

Langue : fr [[équipes](#)]
[cs](#) [fr](#) [en](#) [nl](#) [pl](#)

Autres Documents

[Guide de Mise à Niveau](#)
[Suivre -current](#)
[Suivre -stable](#)
[Guide de Tests des Ports](#)
[Comment Utiliser AnonCVS](#)
[Comment Utiliser CVSup](#)
[Pages de manuel](#)
[Faire un Rapport de Bogues](#)
[Listes de diffusion](#)
[Guide de l'Utilisateur PF](#)
[FAQ OpenSSH](#)

Fichiers PDF

[FAQ OpenBSD](#)
[Guide de l'Utilisateur PF](#)

Fichiers texte

[FAQ OpenBSD](#)
[FAQ PF](#)

Retour à OpenBSD



OpenBSD

Documentation et Questions Fréquemment Posées

[Problèmes Fréquemment Rencontrés](#)

[Mises à jour récentes](#)

Cette FAQ constitue une documentation visant à agréger les pages de manuel, ces dernières étant disponibles aussi bien sur le système installé qu'[en ligne](#). La FAQ couvre la version la plus récente d'OpenBSD, actuellement la [v4.6](#). Il y aura vraisemblablement des nouvelles fonctionnalités ou modifications apportées aux fonctionnalités existantes au niveau de la [version de développement \(-current\)](#) qui ne seront pas couvertes par la présente FAQ.

La FAQ aux formats PDF et texte simple est disponible avec d'autres documents (en anglais) dans le répertoire `pub/OpenBSD/doc` des nombreux [miroirs FTP](#).

1 - Introduction à OpenBSD

- [1.1 - Qu'est-ce que OpenBSD?](#)
- [1.2 - Sur quels systèmes OpenBSD fonctionne-t-il ?](#)
- [1.3 - Pourquoi utiliserais-je OpenBSD ?](#)
- [1.4 - OpenBSD est-il réellement libre ?](#)
- [1.5 - Comment puis-je aider OpenBSD ?](#)
- [1.6 - Qui maintient OpenBSD ?](#)
- [1.7 - Pour quand la prochaine version d'OpenBSD est-elle prévue ?](#)
- [1.8 - Quels logiciels sont inclus avec OpenBSD ?](#)
- [1.9 - Quoi de neuf dans OpenBSD 4.6 ?](#)
- [1.10 - Puis-je utiliser OpenBSD comme station de travail ?](#)

- [1.11 - Pourquoi le Produit "X" est t-il/n'est-il pas inclus ?](#)

2 - Apprendre à connaître OpenBSD

- [2.1 - Pages Web intéressantes](#)
- [2.2 - Listes de discussion](#)
- [2.3 - Pages de manuel](#)
- [2.4 - Rapporter les bugs](#)

3 - Débuter avec OpenBSD

- [3.1 - Acheter un jeu de CD OpenBSD](#)
- [3.2 - Acheter des T-Shirts OpenBSD](#)
- [3.3 - Est-ce qu'OpenBSD fournit une image ISO disponible en téléchargement ?](#)
- [3.4 - Téléchargement via FTP, HTTP ou AFS](#)
- [3.5 - Choisir son matériel](#)
- [3.6 - Quel est le "premier système" approprié pour apprendre OpenBSD dessus ?](#)

4 - Guide d'Installation d'OpenBSD 4.6

- [4.1 - Présentation de la procédure d'installation d'OpenBSD](#)
- [4.2 - Vérifications avant l'installation](#)
- [4.3 - Créer un média d'installation OpenBSD amorçable](#)
- [4.4 - Démarrer le média d'installation OpenBSD](#)
- [4.5 - Installer OpenBSD](#)
- [4.6 - Les détails d'une installation plus complexe](#)
- [4.7 - Quels sont les fichiers nécessaires à l'installation ?](#)
- [4.8 - De combien d'espace disque ai-je besoin pour une installation OpenBSD ?](#)
- [4.9 - Multiboot OpenBSD/i386](#)
- [4.10 - Envoyer votre dmesg à \[dmesg@openbsd.org\]\(mailto:dmesg@openbsd.org\) après l'installation](#)
- [4.11 - Ajouter un paquetage après l'installation](#)
- [4.12 - Qu'est ce que 'bsd.rd' ?](#)
- [4.13 - Problèmes d'installation courants](#)
- [4.14 - Personnaliser la procédure d'installation](#)

- [4.15 - Comment puis-je installer plusieurs systèmes identiques ?](#)
- [4.16 - Comment puis-je obtenir un dmesg\(8\) pour rapporter un problème d'installation ?](#)

5 - Construire le Système à partir des Sources

- [5.1 - Saveurs \('Flavors'\) OpenBSD](#)
- [5.2 - Pourquoi devrais-je compiler mon système depuis les sources?](#)
- [5.3 - Compilation d'OpenBSD depuis les sources](#)
- [5.4 - Compilation d'une Révision](#)
- [5.5 - Compilation de X](#)
- [5.6 - Pourquoi aurais-je besoin d'un noyau sur mesure ?](#)
- [5.7 - Options de configuration du noyau](#)
- [5.8 - Configuration au démarrage](#)
- [5.9 - Utilisation de config\(8\) pour changer le binaire du noyau](#)
- [5.10 - Obtention d'une sortie plus verbeuse lors du démarrage](#)
- [5.11 - Problèmes courants, astuces et questions lors de la Compilation et de la Construction](#)

6 - Mise en place du réseau

- [6.1 - Avant d'aller plus loin](#)
- [6.2 - Configuration initiale du réseau](#)
- [6.3 - Comment filtrer et utiliser un pare-feu sous OpenBSD ?](#)
- [6.4 - Protocole d'attribution dynamique des adresses \(DHCP\)](#)
- [6.5 - Protocole Point à Point \(PPP\)](#)
- [6.6 - Optimisation des paramètres réseau](#)
- [6.7 - Utilisation de NFS](#)
- [6.9 - Mise en place d'un pont \("bridge"\) avec OpenBSD](#)
- [6.10 - Comment démarrer en utilisant PXE ?](#)
- [6.11 - Protocole de redondance d'adresse commune \(CARP\)](#)
- [6.12 - Utiliser OpenNTPD](#)
- [6.13 - Quels sont les types de cartes Sans Fil supportées par OpenBSD ?](#)
- [6.14 - Comment faire du routage "equal-cost multipath" \(répartition de charge entre 2 chemins\) ?](#)

7 - Contrôles du clavier et de l'affichage

- [7.1 - Comment puis-je redéfinir le clavier ? \(wscons\)](#)
- [7.2 - OpenBSD dispose t-il d'un support de la souris en mode console ?](#)
- [7.3 - Accéder au tampon de la console. \(amd64, i386, quelques Alpha\)](#)
- [7.4 - Comment puis-je changer de console ? \(amd64, i386, Zaurus, quelques Alpha\)](#)
- [7.5 - Comment puis-je utiliser une résolution console de 80x50 ? \(amd64, i386, quelques Alpha\)](#)
- [7.6 - Comment puis-je utiliser une console série ?](#)
- [7.7 - Comment effacer la console ? \(wscons\)](#)
- [7.8 - TOUT CE QUE JE TAPE A LA CONNEXION EST EN MAJUSCULES !](#)
- [7.9 - Qu'est ce que tmux\(1\)?](#)

8 - Questions Générales

- [8.1 - J'ai oublié mon mot de passe root... Que dois-je faire !](#)
- [8.2 - X ne veut pas démarrer, j'ai plein de messages d'erreur](#)
- [8.3 - Puis-je utiliser le langage de programmation "L" sous OpenBSD ?](#)
- [8.8 - Existe-il un moyen d'utiliser mon lecteur de disquettes alors qu'il n'était pas connecté durant la phase de démarrage ?](#)
- [8.9 - Le chargeur de démarrage OpenBSD \(*spécifique à i386 et amd64*\)](#)
- [8.10 - Utilisation de S/Key avec votre système OpenBSD](#)
- [8.12 - Est-ce qu'OpenBSD supporte plusieurs processeurs ?](#)
- [8.13 - Parfois, j'ai des erreurs d'entrée/sortie lorsque j'essaie d'utiliser mes périphériques tty](#)
- [8.14 - Quels sont les navigateurs web disponibles sur OpenBSD ?](#)
- [8.15 - Comment s'utilise l'éditeur mg ?](#)
- [8.16 - Apparemment, Ksh ne lit pas mon fichier .profile !](#)
- [8.17 - Pourquoi le contenu de /etc/motd est-il écrasé alors que je l'ai modifié ?](#)
- [8.20 - Les polices anti-aliasées et "TrueType" sous X](#)

- [8.21 - Est-ce qu'OpenBSD supporte des systèmes de fichiers journalisés ?](#)
- [8.22 - Le DNS Inverse ou Pourquoi ça prend autant de temps pour me connecter ?](#)
- [8.23 - Pourquoi les pages web OpenBSD ne sont pas conformes à HTML4/XHTML?](#)
- [8.24 - Mon horloge est décalée d'une vingtaine de secondes. Pourquoi ?](#)
- [8.25 - Pourquoi mon horloge est elle décalée de plusieurs heures ?](#)

9 - Migrer vers OpenBSD

- [9.1 - Astuces pour les utilisateurs d'autres OS Unix-like](#)
- [9.2 - Double démarrage de Linux et d'OpenBSD](#)
- [9.3 - Convertir votre fichier de mots de passe de Linux \(ou de tout autre style Sixth Edition\) au format BSD](#)
- [9.4 - Exécution des binaires Linux sous OpenBSD](#)
- [9.5 - Accéder à vos fichiers Linux depuis OpenBSD](#)

10 - Gestion du Système

- [10.1 - Quand j'essaie de passer root à l'aide de su, on me dit que je suis dans le mauvais groupe](#)
- [10.2 - Comment dupliquer un système de fichiers ?](#)
- [10.3 - Comment démarrer des services en même temps que le système ? \(Vue d'ensemble de rc\(8\)\)](#)
- [10.4 - Pourquoi les utilisateurs sont interdits de relais quand ils envoient des mails à distance à travers mon système OpenBSD ?](#)
- [10.5 - J'ai mis en place POP, mais j'ai des erreurs quand j'accède à ma messagerie via POP. Que puis-je faire ?](#)
- [10.6 - Pourquoi Sendmail ignore /etc/hosts ?](#)
- [10.7 - Configurer HTTP en mode sécurisé à l'aide de SSL\(8\)](#)
- [10.8 - J'ai effectué des changements dans /etc/passwd avec vi \(1\), mais les changements ne semblent pas être pris en compte. Pourquoi ?](#)
- [10.9 - Comment je crée un compte utilisateur ? Ou je supprime](#)

[un compte utilisateur ?](#)

- [10.10 - Comment puis-je créer un compte pour ftp uniquement ?](#)
- [10.11 - Mise en place des quotas](#)
- [10.12 - Mise en place de Clients et de Serveurs KerberosV](#)
- [10.13 - Mise en place d'un serveur FTP Anonyme](#)
- [10.14 - Confiner les utilisateurs à leur répertoire HOME avec ftpd\(8\)](#)
- [10.15 - Appliquer des correctifs sous OpenBSD](#)
- [10.16 - Parlez moi de chroot\(2\) Apache ?](#)
- [10.17 - Puis-je changer le shell de l'utilisateur root ?](#)
- [10.18 - Que puis-je faire d'autre avec ksh ?](#)
- [10.19 - Services d'annuaires, en particulier YP](#)

11 - Le système X Window

- [11.1 - Introduction à X](#)
- [11.2 - Configurer X](#)
- [11.3 - Configurer X sur amd64 et i386](#)
- [11.4 - Démarrer X](#)
- [11.5 - Personnaliser X](#)

12 - Questions Spécifiques Aux Plates-Formes Et Au Matériel

- [12.1 - Remarques Générales sur le Matériel](#)
- [12.2 - DEC Alpha](#)
- [12.3 - AMD 64](#)
- [12.4 - Appliances à base d'ARM](#)
- [12.5 - HP 9000 series 300, 400](#)
- [12.6 - HP Precision Architecture \(PA-RISC\)](#)
- [12.7 - i386](#)
- [12.8 - Landisk](#)
- [12.9 - Luna88k](#)
- [12.10 - Mac68k](#)
- [12.11 - MacPPC](#)
- [12.12 - MVME68k](#)
- [12.13 - MVME88k](#)

- [12.14 - SGI](#)
- [12.15 - SPARC](#)
- [12.16 - UltraSPARC](#)
- [12.17 - DEC VAX](#)
- [12.18 - Sharp Zaurus](#)

13 - Multimédia

- [13.1 - Comment configurer mon périphérique audio ?](#)
- [13.2 - Jouer différents types de formats audio](#)
- [13.3 - Comment jouer des CDs dans OpenBSD ?](#)
- [13.4 - Puis-je utiliser OpenBSD pour enregistrer des séquences audio ?](#)
- [13.5 - Comment configurer un serveur audio](#)
- [13.6 - Que puis-je faire si j'ai des problèmes audio ?](#)
- [13.7 - Comment puis-je utiliser mes instruments MIDI ?](#)
- [13.8 - Parlez moi de l'encodage Ogg Vorbis et MP3 ?](#)
- [13.9 - Comment lire des vidéos DVDs sous OpenBSD ?](#)
- [13.10 - Comment graver des CDs et DVDs ?](#)
- [13.11 - Je voudrais mes fichiers multimédia au format FOO.](#)
- [13.12 - Est-il possible de lire des fichiers en streaming sous OpenBSD ?](#)
- [13.13 - Puis-je utiliser un plugin Java avec mon navigateur ? \(spécifique à i386 & amd64\)](#)
- [13.14 - Puis-je utiliser un plugin Flash plugin avec mon navigateur ? \(spécifique à i386\)](#)

14 - Configuration des Disques

- [14.1 - Utilisation de disklabel\(8\) sous OpenBSD](#)
- [14.2 - Utilisation de fdisk\(8\) sous OpenBSD](#)
- [14.3 - Ajout de nouveaux disques sous OpenBSD](#)
- [14.4 - Comment gérer l'espace de pagination ?](#)
- [14.5 - Soft Updates](#)
- [14.6 - Comment se déroule le processus de démarrage d'OpenBSD/i386 ?](#)
- [14.7 - Quels sont les problèmes liés aux disques de grande capacité sous OpenBSD ?](#)

- [14.8 - Installation des blocs de démarrage \("Bootblocks"\) - spécifique i386/amd64](#)
- [14.9 - Se préparer au désastre : faire une sauvegarde vers une bande et effectuer une restauration.](#)
- [14.10 - Montage des images disque sous OpenBSD](#)
- [14.11 - A l'aide ! J'ai des erreurs avec IDE DMA !](#)
- [14.13 - Options RAID avec OpenBSD](#)
- [14.14 - Pourquoi df \(1 \) me dit que j'ai plus de 100% d'espace disque utilisé ?](#)
- [14.15 - Récupération de partitions après une suppression du disklabel](#)
- [14.16 - Est-il possible d'accéder aux données présentes sur des systèmes de fichiers autres que FFS ?](#)
- [14.17 - Est-il possible d'utiliser un périphérique de masse \('flash memory device'\) sous OpenBSD ?](#)
- [14.18 - Optimiser les performances des disques durs](#)
- [14.19 - Pourquoi nous n'utilisons pas de Montage Asynchrone ?](#)

15 - Le système de paquetages et de ports OpenBSD

- [15.1 - Introduction](#)
- [15.2 - Gestion des paquetages](#)
- [15.3 - Utilisation des ports](#)
- [15.4 - FAQ](#)
- [15.5 - Comment signaler un problème](#)
- [15.6 - Comment nous aider](#)

Guide de l'Utilisateur PF

- Configuration Basique
 - [Principes de Base](#)
 - [Listes et Macros](#)
 - [Tables](#)
 - [Filtrage de Paquets](#)
 - [Traduction d'Adresses Réseau](#)
 - [Redirection de Trafic \("Forwarding" de Ports\)](#)

- [Raccourcis pour la Création de Jeux de Règles](#)
 - Configuration Avancée
 - [Options de Fonctionnement](#)
 - [Ancres](#)
 - [Gestion de La Bande Passante](#)
 - [Ensembles d'Adresses \("Pools"\) et Partage de Charge](#)
 - [Balisage de Paquets \(Politique de Filtrage\)](#)
 - Sujets Additionnels
 - [Journal des Evénements](#)
 - [Performance](#)
 - [Problèmes avec FTP](#)
 - [Authpf : Shell Utilisateur pour les Passerelles d'Authentification](#)
 - [Haute-disponibilité des pare-feu avec CARP et pfsync](#)
 - Exemples de Jeux de Règles
 - [Pare-feu particulier ou pour une petite société](#)
-

Problèmes Fréquemment Rencontrés

- [Problèmes d'Installation Communs](#)
 - [Comment mettre à jour mon système ?](#)
 - [Packet Filter](#)
 - [Dois-je utiliser les Ports ou les Paquetages ?](#)
 - [Comment configurer un système "multi-boot" ?](#)
 - [Réseau sans-fil sous OpenBSD](#)
-

Mises à jour récentes

- [Mettre en place un media USB "live"](#) -- nouveau
- [Configurer un serveur audio](#) -- nouveau
- FAQ mise à jour pour la 4.6
- [Nouvelles instructions d'installation pour la 4.6](#)
- [Qu'est ce que tmux\(1\)?](#) -- nouveau
- [Multiboot](#) -- mis à jour pour Vista, Windows 7
- [Services d'annuaires, en particulier YP](#) -- nouveau

Le mainteneur de la FAQ est : Nick Holland.

Parmi les contributeurs additionnels, on compte Steven Mestdagh, Joel Knight, Eric Jackson, Wim Vandeputte et Chris Cappuccio.

Pour toute information concernant la traduction de cette FAQ et le reste du site web OpenBSD, consultez la [page traduction](#).

Les questions et les commentaires concernant cette FAQ peuvent être envoyés à faq@openbsd.org. Les questions d'ordre général sur OpenBSD devront être envoyées à la [liste de diffusion](#) appropriée.

Retour à OpenBSD



OpenBSD FAQ Copyright © 1998-2009 OpenBSD

\$OpenBSD: index.html,v 1.156 2010/02/16 09:19:53 ajacoutot Exp \$

"Si vous ne le trouvez pas dans l'index, veuillez regarder en détail le catalogue."

Sears, Roebuck, and Co., Consumer's Guide, 1897



[\[Index de La FAQ\]](#) [\[Section 2 - Autres sources d'Information OpenBSD\]](#)

1 - Introduction à OpenBSD

Table des matières

- [1.1 - Qu'est-ce qu'OpenBSD ?](#)
 - [1.2 - Sur quels systèmes OpenBSD fonctionne-t-il ?](#)
 - [1.3 - Pourquoi utiliserais-je OpenBSD ?](#)
 - [1.4 - OpenBSD est-il réellement libre ?](#)
 - [1.5 - Comment puis-je aider OpenBSD ?](#)
 - [1.6 - Qui maintient OpenBSD ?](#)
 - [1.7 - Pour quand la prochaine version d'OpenBSD est-elle prévue ?](#)
 - [1.8 - Quels logiciels sont inclus avec OpenBSD ?](#)
 - [1.9 - Quoi de neuf dans OpenBSD 4.7 ?](#)
 - [1.10 - Puis-je utiliser OpenBSD comme station de travail ?](#)
 - [1.11 - Pourquoi le *ProduitX* est-il/n'est-il pas inclus ?](#)
-

1.1 - Qu'est-ce qu'OpenBSD ?

Le projet [OpenBSD](#) fournit un système d'exploitation de type UNIX, multi plates-formes et basé sur 4.4BSD. Nos [objectifs](#) concernent principalement l'exactitude, la [sécurité](#), la standardisation et la [portabilité](#). OpenBSD supporte l'émulation des binaires SVR4 (Solaris), FreeBSD, Linux, BSD/OS, SunOS et HP-UX.

Cette FAQ couvre spécifiquement la version la plus récente d'OpenBSD, la version 4.7.

1.2 - Sur quels systèmes OpenBSD fonctionne-t-il ?

OpenBSD 4.7 fonctionne sur les plates-formes suivantes :

- [alpha](#) - FTP uniquement
- [amd64](#) - Disponible sur CD
- [armish](#) - FTP uniquement
- [hp300](#) - FTP uniquement
- [hppa](#) - FTP uniquement
- [i386](#) - Disponible sur CD
- [landisk](#) - FTP uniquement
- [loongson](#) - FTP uniquement (nouveau)
- [macppc](#) - Disponible sur CD
- [mvme68k](#) - FTP uniquement
- [mvme88k](#) - FTP uniquement
- [sgi](#) - FTP uniquement
- [socppc](#) - FTP uniquement
- [sparc](#) - FTP uniquement
- [sparc64](#) - Disponible sur CD
- [vax](#) - FTP uniquement
- [zaurus](#) - FTP uniquement

Disponible sur CD indique que le [set de CD officiel](#) inclut cette plate-forme et bon nombre de [paquetages](#) pour celle-ci. Les images ISO du système de base peuvent également être téléchargées pour la plupart des autres plates-formes.

Plus d'informations sur les architectures supportées par OpenBSD sont disponibles sur la page des [plates-formes](#).

De temps à autre, on nous demande pourquoi nous supportons autant de machines exotiques. Tout simplement "parce que nous le voulons". Si un nombre suffisant de personnes compétentes (des fois, c'est une seule personne compétente !) souhaitent maintenir le support d'une plate-forme alors celle-ci sera supportée. Il y a des avantages pratiques au support de multiples plates-formes : lorsque de nouvelles plates-formes sont supportées, l'arbre des sources ne contient pas a priori de bogues relatifs à la portabilité ou à la conception du système. Les plates-formes supportées par OpenBSD incluent des processeurs 32 bits et 64 bits, des machines en "big" et "little" endian, ainsi que d'autres de conception différente. Et bien entendu, le support de plates-formes exotiques nous a aidé à produire un code de meilleure qualité pour les plates-formes les plus communes.

1.3 - Pourquoi utiliserais-je OpenBSD ?

Les nouveaux utilisateurs veulent souvent savoir si OpenBSD est supérieur à d'autres UNIX libres. Il est quasiment impossible de répondre à cette question qui est sujette à de nombreux et inutiles débats "religieux". Ne jamais, en aucune circonstance, poser cette question sur une des listes de discussion

OpenBSD.

Ci-dessous, vous trouverez les raisons qui nous font penser qu'OpenBSD est un système utile. Maintenant, vous seul pouvez répondre à la question qui est de savoir s'il vous sera utile.

- OpenBSD fonctionne sur de [nombreuses plates-formes matérielles](#).
- OpenBSD est considéré par de nombreux professionnels de la sécurité comme étant le système de type UNIX le plus [sûr](#) qui soit. Ceci est dû au fait que le code source fait régulièrement l'objet d'un audit exhaustif.
- OpenBSD est un système UNIX complet disponible gratuitement avec les sources.
- OpenBSD intègre les derniers outils en matière de sécurité pour construire des pare-feu et des [réseaux privés virtuels](#) dans un environnement distribué.
- OpenBSD bénéficie d'un développement fort et continu dans de nombreux domaines, donnant la possibilité de travailler sur des technologies émergentes avec une communauté internationale de programmeurs et d'utilisateurs.
- OpenBSD tente de minimiser le besoin de personnalisation. Pour la majorité des utilisateurs, OpenBSD "fonctionne tout court" sur leur matériel pour leur application. Non seulement la personnalisation est rarement nécessaire, mais elle est activement [déconseillée](#).

1.4 - OpenBSD est-il réellement libre ?

OpenBSD est totalement libre. Les binaires sont libres, le source est libre. Toutes les parties d'OpenBSD ont des copyright raisonnables permettant la libre redistribution. Cela comprend la possibilité de REUTILISER la plupart des sources d'OpenBSD, pour un usage personnel comme pour un usage commercial. OpenBSD ne possède pas de restrictions autres que celles spécifiées dans la licence BSD originelle. Les logiciels qui sont écrits avec une licence trop restrictive ne peuvent pas être inclus dans la distribution standard d'OpenBSD. Ceci dans le but de sauvegarder l'usage libre d'OpenBSD. Par exemple, OpenBSD peut être utilisé librement pour un usage personnel, pour un usage éducatif, par des institutions gouvernementales, par des associations à but non lucratif et par des organisations commerciales. OpenBSD peut être entièrement ou partiellement incorporé dans des [produits commerciaux](#).

Des personnes nous demandent parfois si le fait que notre travail soit utilisé dans des produits commerciaux nous dérange. La réponse est la suivante, nous préférons que notre code de qualité soit *largement utilisé* plutôt que des vendeurs de logiciels commerciaux réécrivent des solutions alternatives incompatibles et mal codées, pour répondre à des problématiques déjà résolues. Par exemple, SSH est largement utilisé à cause de la liberté qui lui attrait, et l'est bien plus que si des restrictions avaient été appliquées à l'utilisation du code d'OpenSSH. Si une solution libre de SSH à utiliser n'avait pas été disponible pour les vendeurs (dans la multitude de leurs logiciels rapidement développés), ils auraient dû développer ou acheter une version minable sortie de nul part à la place.

Cela ne veut pas dire que nous nous opposons à des [donations financières ou à un support matériel](#) en remerciement. En fait, il est aberrant de voir le peu de support de toute sorte que nous recevons des sociétés dépendant d'OpenBSD (ou OpenSSH) pour leurs produits, même si aucune compensation n'est exigée.

Pour de plus amples informations sur les autres licences, veuillez lire : [Politique Copyright d'OpenBSD](#).

Les développeurs du projet OpenBSD le supportent principalement grâce à leurs revenus. Ceci inclut le temps dépensé en programmation pour le projet, le matériel utilisé, les ressources réseaux utilisées pour que vous puissiez obtenir OpenBSD ainsi que le temps dépensé à répondre aux questions et à corriger les bugs trouvés par les utilisateurs. Les développeurs OpenBSD ne sont pas riches et même de petites contributions de temps, de matériel ou autres peuvent apporter de grosses différences.

1.5 - Comment puis-je aider OpenBSD ?

Nous sommes redevables aux personnes et organisations qui ont contribué au projet OpenBSD. Les donateurs sont cités sur la [page des dons](#)

OpenBSD a constamment besoin de plusieurs types de support de la part des utilisateurs. Si vous trouvez OpenBSD utile, nous vous invitons à trouver un moyen de contribuer. Si aucun de ceux proposés ci-dessous ne vous convient, vous pouvez toujours en proposer d'autres en envoyant un courrier électronique à : donations@openbsd.org.

- [Acheter un jeu de CD OpenBSD](#). Il comprend la version actuelle d'OpenBSD et est bootable sur plusieurs plates-formes. Il permet aussi de financer le projet OpenBSD et d'éviter la consommation de bande passante lors d'un téléchargement à travers l'Internet. Ce jeu de trois CD peu onéreux inclut la totalité du code source. Rappelez vous que vos amis doivent avoir leurs propres CD !
- [Financer le projet](#). Le projet a toujours besoin d'argent pour payer les équipements, les connexions réseau et la publication des CD. Presser des CD demande un investissement aux développeurs OpenBSD qu'ils ne sont pas toujours assurés de récupérer. Envoyez un courrier électronique à donations@openbsd.org pour savoir comment contribuer. Même les petits dons font de grandes différences.
- [Donner des équipements](#). Le projet a toujours besoin de matériel, qu'il soit spécifique ou général. Des articles tels que les disques IDE ou SCSI et les barrettes de mémoire RAM sont toujours les bienvenus. Pour d'autres types de matériels tels que les systèmes complets ou les cartes mères, il est préférable de se renseigner au préalable. Écrivez à donations@openbsd.org pour convenir des modalités d'expédition.
- Faites don de votre temps et de vos compétences. Les programmeurs qui aiment écrire des systèmes d'exploitation sont bien entendu les bienvenus mais il existe bien d'autres façons d'être utile. Suivez [les listes de discussion](#) et aidez à répondre aux questions des nouveaux utilisateurs.

- Aidez à maintenir la documentation à jour en soumettant de nouvelles rubriques pour la FAQ (à faq@openbsd.org). Créez un [groupe](#) d'utilisateurs local et faites découvrir OpenBSD à vos amis. Insistez pour utiliser OpenBSD au travail. Si vous êtes étudiant, proposez d'utiliser OpenBSD à vos professeurs en tant qu'outil d'apprentissage pour les cours d'informatique ou de sciences de l'ingénieur. Il est aussi important de mentionner l'une des façons de ne pas "aider" le projet OpenBSD : ne perdez pas votre temps en guerres de religion inter systèmes d'exploitation. Cela n'aide en rien le projet à trouver de nouveaux utilisateurs et nuit aux liens que les développeurs auront liés avec des développeurs d'autres projets.

1.6 - Qui maintient OpenBSD ?

OpenBSD est maintenu par une équipe de développement disséminée à travers de nombreux [pays](#). Le projet est coordonné par Theo de Raadt basé au Canada.

1.7 - Pour quand la prochaine version d'OpenBSD est-elle prévue ?

L'équipe de développement OpenBSD crée une nouvelle version tous les six mois, avec des dates de mise à disposition fixées à Mai et Novembre. Vous pourrez obtenir plus d'informations concernant le cycle de développement [ici](#).

1.8 - Quels logiciels sont inclus avec OpenBSD ?

OpenBSD est distribué avec un certain nombre d'applications tierces telles que :

- [X.org 7.4](#), l'environnement X Window, avec des correctifs locaux. Installé en utilisant les fichiers `x* .tgz` des [ensembles d'installation](#).
- [GCC 2.95.3 et 3.3.5](#). Compilateur GNU C. L'équipe OpenBSD a ajouté la technologie de protection de la pile [Propolice](#), activée par défaut et utilisée par toutes les applications intégrées au système d'exploitation ainsi que sur les applications compilées sur OpenBSD. Gcc fait partie de `comp46 .tgz` présent dans [l'ensemble d'installation](#).
- [Perl 5.10.1](#), avec des correctifs et des améliorations créés par l'équipe OpenBSD
- Notre propre version améliorée et sécurisée d'Apache 1.3.29. L'équipe OpenBSD a ajouté le [confinement chroot par défaut](#), la révocation de privilèges, et d'autres améliorations relatives à la sécurité. `mod_ssl` et DSO sont aussi inclus.
- [OpenSSL 0.9.8k](#), avec des correctifs et des améliorations de la part de l'équipe OpenBSD
- Le processeur de texte [Groff 1.15](#).
- Le serveur de messagerie [Sendmail 8.14.3](#) avec `libmilter`
- Le serveur DNS [BIND 9.4.2-P2](#) (avec correctifs). OpenBSD a implémenté plusieurs

- améliorations au processus de confinement chroot ainsi que d'autres améliorations de sécurité
- Le navigateur web en mode texte [Lynx 2.8.6rel.5](#). Sont inclus le support HTTPS et IPv6 et des correctifs créés par l'équipe OpenBSD
- [Sudo v1.7.2](#), permettant aux utilisateurs d'exécuter des commandes individuelles avec les privilèges root
- [Ncurses 5.7](#)
- IPv6 [KAME](#)
- [Heimdal 0.7.2](#) avec correctifs
- [Arla 0.35.7](#)
- [Binutils 2.15](#) avec correctifs
- [gdb 6.3](#) avec correctifs
- [OpenSSH 5.5](#)
- [OpenNTPD](#), implémentation simple et sécurisée du protocole NTP ("Network Time Protocol")
- [OpenBGPD et OpenOSPFD](#), applications de routage

Comme on peut le constater, l'équipe OpenBSD ajoute souvent des correctifs aux applications tierces (typiquement) pour améliorer la sécurité ou la qualité du code. Dans certains cas, l'utilisateur ne verra aucune différence dans le fonctionnement. Tandis que dans d'autres cas, il existe des différences opérationnelles qui peuvent impacter certains utilisateurs. Gardez ces améliorations en tête avant d'ajouter des versions différentes du même logiciel de manière aveugle. Vous pourrez obtenir un numéro de version plus récent mais un système moins sûr en retour.

Bien entendu, des applications supplémentaires peuvent être ajoutées avec le système des [paquetages et ports](#).

1.9 - Quoi de neuf dans OpenBSD 4.7 ?

La liste complète des changements apportés à OpenBSD 4.6 pour créer OpenBSD 4.7 se trouve [ici](#), et les changements majeurs sur la [page d'information d'OpenBSD 4.7](#). Cependant voici quelques changements que l'équipe de développement OpenBSD juge importants à signaler aux personnes qui vont faire des mises à jour ou des installations d'OpenBSD 4.7 et qui sont familières des versions antérieures :

- [La plateforme OpenBSD/mac68k n'est plus maintenue](#)
Le code reste dans l'arbre, mais l'intérêt d'un maintien actif des ports et de faire des compilations est parti.
- **Multi-adaptateur X est cassé**
Bien que X puisse être utilisé sur plusieurs écrans avec UNE seule carte vidéo, la possibilité d'utiliser plusieurs cartes vidéos sur un système est cassé "upstream" par l'équipe X.org.
- **La configuration des ponts (bridge) est maintenant faite par [ifconfig\(8\)](#)**

brconfig(8) n'est plus utilisé.

- **Changements dans PF**

Les détails dans le [guide de mise à jour](#).

- **Le protocole 1 de SSH est désactivé par défaut**

Cela impacte seulement les nouvelles installations. Vous avez eu bien assez de temps pour migrer vers SSH2, bien plus qu'il n'en faut.

1.10 - Puis-je utiliser OpenBSD comme station de travail ?

Cette question est souvent posée exactement de cette manière, sans aucune précision sur le sens de "station de travail". La seule personne capable de répondre à cette question est vous, étant donné que la réponse dépend de vos besoins et de vos attentes.

Bien qu'OpenBSD possède une bonne réputation en tant que système d'exploitation "serveur", il peut être utilisé sur les stations de travail. Plusieurs applications destinées à des "stations de travail" sont disponibles à travers les [ports et les paquetages](#). Comme pour n'importe quelle autre décision relative à un système d'exploitation, la question est la suivante : est-ce qu'OpenBSD peut remplir telle fonction de telle manière ? Il vous appartient de répondre à cette question.

Il est intéressant de noter qu'une large portion du développement d'OpenBSD est effectué sur des ordinateurs portables.

1.11 - Pourquoi le *ProduitX* est-il/n'est-il pas inclus ?

Souvent, des personnes demandent pourquoi un produit particulier est ou n'est pas inclus dans OpenBSD. La réponse est basée sur deux éléments : les souhaits des développeurs et la compatibilité avec les [objectifs](#) du projet. Un produit ne sera pas inclus simplement parce qu'il est "chic" -- il doit aussi être "libre" d'utilisation, de distribution et de modification selon nos standards. Un produit doit aussi être stable et sûr -- un numéro de version plus récent n'est pas toujours synonyme d'un meilleur produit.

La licence est souvent le plus grand problème : nous voulons qu'OpenBSD reste utilisable par n'importe quelle personne n'importe où dans le monde et dans n'importe quel but.

Une autre considération majeure est le souhait des développeurs. Les développeurs OpenBSD sont les juges ultimes de ce qui est intégré ou non dans le projet. Le fait qu'une application soit "bien" ne signifie pas forcément que le projet OpenBSD veuille y allouer les ressources pour la maintenir, ou qu'ils soient enjoués par sa présence au sein du projet.

Quelques questions communes sur les produits tiers :

- *Pourquoi Sendmail est inclus, il est "connu pour être non sécurisé" ?*
Sendmail a un historique sécurité imparfait, cependant les auteurs de Sendmail et les mainteneurs ont été très réceptifs pour retravailler leur code pour le rendre beaucoup plus sécurisé (et ceci est une réponse malheureusement peu commune). L'historique sécurité récent de Sendmail n'est pas si différent de celui de certaines alternatives supposées "plus sécurisées".
- *Pourquoi Postfix n'est pas inclus ?*
Sa licence n'est pas libre et il ne peut donc pas être inclus.
- *Pourquoi qmail ou djbdns ne sont pas inclus ?*
Ces deux programmes ne constituent pas ce que la plupart des utilisateurs Unix peuvent "attendre" de la part d'une application de messagerie ou DNS.
- *Pourquoi Apache est-il inclus ? Beaucoup de personnes n'en veulent pas !*
Parce que les développeurs le veulent.
- *Pourquoi une version plus récente d'Apache n'est pas incluse ?*
La licence sur les versions plus récentes est inacceptable.
- *Pourquoi bzip2 n'est pas inclus à la place de gzip ?*
Les performances sont horribles, et le bénéfice est minimal. L'impact sur les plates-formes plus lentes, comme hp300 ou VAX, ne serait pas acceptable.
- *Pourquoi aucun logiciel d'installation graphique ou basé sur [curses\(3\)](#) n'est inclus ?*
Pour un certain nombre de raisons, y compris le but de pouvoir utiliser une seule disquette pour initier l'installation du système, sans oublier le fait qu'un seul logiciel d'installation peut être utilisé sur toutes les plates-formes pour tout type de configuration ainsi que le fait qu'après la seconde ou troisième installation d'OpenBSD, la plupart des utilisateurs trouvent que le logiciel d'installation d'OpenBSD est parmi les plus rapides et les plus faciles d'utilisation qui soient. La plupart des développeurs et utilisateurs apprécient la vitesse, la puissance et la facilité d'utilisation du logiciel d'installation à tous les autres logiciels plus "coloriés" ou "jolis" fournis par certaines plates-formes.
- *Est-ce que ZFS sera ajouté à OpenBSD ?*
Non, tant que quelqu'un ne convaincra pas Oracle de modifier sa licence avec quelque chose de compatible avec la [politique](#) d'OpenBSD.

Dans la plupart des cas, ces sujets ont été discutés de manière plus que détaillée sur les [listes de diffusion](#), veuillez consulter les archives si vous avez besoin de plus d'informations.

Bien entendu, si vous souhaitez utiliser un de ces paquetages et dans la mesure où votre utilisation est compatible avec la licence de ces produits, personne ne vous en empêchera (ça ne serait pas vraiment synonyme de "liberté" de le faire, n'est-ce pas ?). Cependant, vos besoins pourraient changer -- vous ne voudriez peut-être pas développer une "Killer Application" que vous ne pouvez pas vendre, distribuer, ou utiliser pour devenir riche parce que vous y avez incorporé des éléments non libres.

[\[Index de La FAQ\]](#) [\[Section 2 - Autres sources d'Information OpenBSD\]](#)



www@openbsd.org

\$OpenBSD: faq1.html,v 1.84 2010/05/27 10:00:57 ajacoutot Exp \$

OpenBSD

[\[Index de la FAQ\]](#) [\[Section 1 - Introduction à OpenBSD\]](#) [\[Section 3 - Démarrer avec OpenBSD\]](#)

2 - Apprendre à connaître OpenBSD

Table des matières

- [2.1 - Pages Web intéressantes](#)
 - [2.2 - Listes de discussion](#)
 - [2.3 - Pages de manuel](#)
 - [2.4 - Rapporter les bugs](#)
-

2.1 - Pages Web intéressantes

Le site officiel pour le projet OpenBSD se trouve à l'adresse : <http://www.OpenBSD.org>.

Beaucoup d'informations utiles s'y trouvent en ce qui concerne tous les aspects du projet OpenBSD.

L'[OpenBSD Journal](#) est un site de nouvelles et d'opinions autour d'OpenBSD.

[OpenBSDsupport.org](#) regroupe de la documentation maintenue par des utilisateurs et de qualité variable, mais qui a le mérite de couvrir des sujets ne faisant pas partie de la présente FAQ ou de toute autre documentation officielle.

Plusieurs utilisateurs ont mis en place des sites et des pages avec des informations spécifiques à OpenBSD. Un bon moteur de recherche vous facilitera la vie, comme le fera aussi une dose raisonnable de scepticisme. Comme d'habitude, n'utilisez pas aveuglément des commandes que vous ne comprenez pas.

2.2 - Listes de discussion

Le projet OpenBSD maintient plusieurs listes de discussion auxquelles les utilisateurs peuvent s'inscrire. Pour s'inscrire à une liste, il suffit d'envoyer un message à majordomo@openbsd.org. Cette adresse correspond à un service d'inscription automatique. Dans le corps du message, sur une seule ligne, vous devez inclure une commande d'inscription pour la liste désirée. Par exemple :

```
subscribe announce
```

Le gestionnaire automatique de la liste vous répondra, pour obtenir une confirmation de votre part afin que d'autres personnes ne vous inscrivent pas à une avalanche de courriel non sollicité. Le message contiendra des instructions sur différentes méthodes de confirmation, y compris un lien vers la page web du [serveur de listes](#), la réponse au message de confirmation ou la réponse à majordomo@openbsd.org. Utilisez la méthode qui vous convient. Il est à noter que les

trois techniques précitées impliquent un identifiant unique et limité dans le temps, tel que A56D-70D4-52C3, encore une fois pour s'assurer que *vous* êtes réellement la personne qui a demandé l'inscription à la liste de diffusion (c'est le *vrai* "opt-in").

Une fois que vous avez confirmé votre intention de vous joindre à la liste, vous serez immédiatement ajouté à celle-ci, et le service vous l'indiquera.

Pour se désabonner d'une liste, il vous faut encore envoyer un message à majordomo@openbsd.org. Cela ressemblera à :

```
unsubscribe announce
```

Si vous avez des difficultés avec le système des listes de discussion, veuillez d'abord lire le fichier d'aide qui peut être obtenu en envoyant un message à majordomo@openbsd.org avec dans le corps de celui-ci : "help".

Votre inscription aux listes de diffusion OpenBSD peut aussi être maintenue à travers l'interface Web disponible à l'adresse <http://lists.openbsd.org>

Parmi les listes de discussions les plus populaires, on peut citer :

- **announce** - Annonces importantes. Il s'agit d'une liste à faible débit.
- **security-announce** - Annonces des problèmes de sécurité. Il s'agit d'une liste à faible débit.
- **misc** - Questions et réponses générales d'utilisateurs. C'est la liste la plus active et elle devrait être la liste par "défaut" pour la plupart des questions.
- **bugs** - Bugs reçus par l'intermédiaire de sendbug(1) et discussions à leur sujet.
- **source-changes** - Liste recevant automatiquement les modifications apportées à l'arbre des sources CVS. Dès qu'un développeur modifie ("commit") l'arbre des sources d'OpenBSD, [CVS](#) envoie une copie (généralement brève) de son message "commit" à cette liste.
- **ports** - Discussions à propos de l'arbre des ports OpenBSD.
- **ports-changes** - Liste recevant automatiquement les modifications apportées à l'arbre des sources CVS qui concernent les ports.
- **advocacy** - Discussions sur la promotion d'OpenBSD et les messages hors-sujet dans **misc**...

Avant d'envoyer une question à **misc** ou toute autre liste de diffusion, parcourez les archives, pour y voir les questions fréquemment posées. Bien que cela puisse être la première fois que vous rencontrez un problème ou que vous avez une question, les autres participants de la liste peuvent avoir vu la même question plusieurs fois en plusieurs semaines et peuvent ne pas apprécier de la voir, une fois encore. Si votre question concerne du matériel, *envoyez toujours un [dmesg](#) (8)!*

Vous pouvez consulter plusieurs archives, d'autres lignes de conduite relatives aux listes de diffusion ainsi que d'autres informations dans la [page des listes de diffusion](#).

Une liste de diffusion non officielle qui pourrait intéresser les nouveaux utilisateurs d'OpenBSD et d'Unix est la liste appelée [OpenBSD Newbies](#).

2.3 - Pages de manuel

OpenBSD est fourni avec une abondante documentation sous la forme de pages de manuel, ainsi que d'autres documents relatifs à des applications spécifiques. Un effort considérable est fourni pour s'assurer que les pages de manuel sont à jour

et correctes. Dans tous les cas, celles-ci sont à considérer comme la source faisant autorité concernant les informations sur OpenBSD.

Pour accéder aux pages du man ainsi qu'au reste de la documentation, assurez-vous d'avoir installé les tarballs `man47.tgz` et `misc47.tgz` de l'[ensembles des fichiers](#).

Voici une liste des pages de manuel les plus utiles pour les nouveaux utilisateurs :

Débutants

- [afterboot\(8\)](#) - les choses à vérifier après le premier redémarrage complet.
- [help\(1\)](#) - aide pour les nouveaux utilisateurs et administrateurs.
- [hier\(7\)](#) - agencement des systèmes de fichiers.
- [man\(1\)](#) - affiche les pages de manuel.
- [intro\(1\)](#) - introduction aux commandes générales, voir aussi les introductions aux autres sections de manuel : [intro\(2\)](#), [intro\(3\)](#), [intro\(4\)](#) (remarque : [intro\(4\)](#) est spécifique à chaque [plate-forme](#)), [intro\(5\)](#), [intro\(6\)](#), [intro\(7\)](#), [intro\(8\)](#) et [intro\(9\)](#).
- [adduser\(8\)](#) - commande pour ajouter de nouveaux utilisateurs.
- [vipw\(8\)](#) - édite le fichier de mots de passe principal.
- [disklabel\(8\)](#) - lecture et écriture du `disklabel`.
- [reboot, halt\(8\)](#) - arrêt et redémarrage du système.
- [shutdown\(8\)](#) - arrêt du système à un temps donné.
- [dmesg\(8\)](#) - réaffichage des messages de démarrage du noyau.
- [sudo\(8\)](#) - exécuter des commandes en root mais sans se loguer en root.
- [mg\(1\)](#) - éditeur de texte fonctionnant comme `emacs`.

Pour des utilisateurs plus avancés

- [boot\(8\)](#) - procédures de bootstrapping système.
- [boot_config\(8\)](#) - comment changer la configuration du noyau au boot.
- [gcc-local\(1\)](#) - modifications spécifiques à OpenBSD pour [gcc\(1\)](#)
- [ifconfig\(8\)](#) - configure les paramètres de l'interface réseau.
- [login.conf\(5\)](#) - format du fichier de configuration des classes de connexion.
- [netstat\(1\)](#) - affiche le statut du réseau.
- [release\(8\)](#) - compile et prépare une distribution OpenBSD.
- [sendbug\(1\)](#) - envoi d'un rapport de bogues (PR) sur OpenBSD à un site central.
- [style\(9\)](#) - guide de style pour le code source du noyau OpenBSD.
- [sysctl\(8\)](#) - obtenir ou paramétrer des états noyau.

Vous pouvez trouver toutes les pages de manuel OpenBSD sur le web à l'adresse <http://www.openbsd.org/cgi-bin/man.cgi> ainsi que sur votre ordinateur si vous installez l'archive de fichiers `man47.tgz`.

En général, si vous connaissez le nom d'une commande ou d'une page de manuel, vous pouvez la lire en tapant "`man commande`". Par exemple : "`man vi`" pour obtenir la page de manuel de l'éditeur `vi`. Si vous ne connaissez pas le nom d'une commande ou si "`man commande`" ne trouve pas la page de manuel vous pouvez chercher dans la base de données des pages de manuel en tapant "``apropos quelque_chose``" ou "`man -k quelque_chose`", où

"quelque_chose" est une expression qui a des chances d'apparaître dans le titre de la page de manuel que vous recherchez. Par exemple :

```
# apropos "time zone"
tzfile (5) - time zone information
zdump (8) - time zone dumper
zic (8) - time zone compiler
```

Le numéro entre parenthèses indique la section de manuel dans laquelle cette page a été trouvée. Dans certains cas, vous pourrez trouver des pages de manuel avec des noms identiques dans différentes sections du manuel. Par exemple, supposons que vous souhaitez connaître le format des fichiers de configuration du service cron. Une fois que vous connaissez la section de manuel pour la page que vous recherchez, tapez "man n commande", où n est le numéro de la section correspondante.

```
# man -k cron
cron (8) - clock daemon
crontab (1) - maintain crontab files for individual users
crontab (5) - tables for driving cron
# man 5 crontab
```

En plus des pages de manuel UNIX, il existe une documentation imprimable (inclue dans le jeu de fichiers misc47.tgz). Celle-ci se trouve dans le répertoire /usr/share/doc. Vous pouvez formater chacune des parties de la documentation avec un "make" dans le répertoire correspondant. Le répertoire psd contient le document "Programmer's Supplementary Documents". Le répertoire smm contient le document "System Manager's Manual". Le répertoire usd contient le document "UNIX User's Supplementary Documents". Vous pouvez effectuer votre "make" dans les trois sous-répertoires ou vous pouvez sélectionner une section spécifique d'un document et faire le `make' dans le répertoire correspondant.

Certaines parties sont vides. Par défaut, les documents seront formatés au format PostScript, utilisable pour l'impression. La taille de la sortie PostScript peut-être importante (attendez-vous à une augmentation de volume de l'ordre de 250-300%). Si vous n'avez pas accès à une imprimante PostScript (ou à un visualiseur PostScript), vous pouvez toujours formater les documents de façon à les lire sur un terminal. Chaque sous-répertoire à documents possède une cible de compilation pour compiler des copies ASCII de ces documents (appelés `paper.txt') qui peuvent être générés avec [make\(1\)](#). Par exemple :

```
# cd /usr/share/doc/usd/06.bc
# make paper.txt
# more paper.txt
```

Les privilèges super-utilisateur peuvent être nécessaires pour compiler les documents dans ces répertoires. Un **make clean** aura pour effet de supprimer tous les documents générés par un make précédent. Veuillez consulter /usr/share/doc/README pour plus de détails concernant les documents présents dans /usr/share/doc/.

Les pages de manuel UNIX sont généralement plus à jour que les documents imprimables. Mais ceux-ci peuvent expliquer des applications complexes avec plus de détails que ne le font les pages de manuel.

Il est utile pour bon nombre de personnes de disposer d'une version papier d'une page de manuel. Voici les indications pour obtenir une version imprimable d'une page de manuel.

Comment est-ce que j'affiche le code source d'une page de manuel (par exemple, un de ces fichiers finissant par un numéro, comme tcpdump.8) ?

On les trouve dans les sources. Les pages de manuel non formatées se trouvent dans les sources et elles seront souvent mises à jour par l'utilisation de [CVS](#). Pour les visualiser, il faut juste taper :

```
# nroff -Tascii -mandoc <file> | more
```

Comment est-ce que j'obtiens une page de manuel sans caractères de contrôle ou de formatage ?

Il est utile d'obtenir une page de manuel sans caractères non imprimables.

Exemple :

```
# man <command> | col -b
```

Comment puis-je obtenir une copie PostScript d'une page de manuel ?

Remarquez que `<file>` doit être le fichier source de la page de manuel (certainement un fichier finissant par un numéro comme `tcpdump.8`). Les versions PostScript des pages de manuel ont une belle apparence. Elles peuvent être imprimées ou visualisées à l'aide d'un programme comme `gv` (GhostView). GhostView est disponible dans notre [collection de Paquetages](#). Utilisez les options suivantes de la commande [nroff\(1\)](#) pour obtenir une version PostScript à partir d'une page de manuel système d'OpenBSD :

```
# nroff -Tps -mandoc <file> > outfile.ps
```

Comment générer des copies compressées des pages de manuel ?

Pour les personnes qui compilent leur système à partir des sources, il existe un certain nombre d'options permettant de contrôler la manière dont les pages de manuel sont construites. Ces options figurent dans `/etc/mk.conf` (il peut être nécessaire de créer ce fichier) et sont incluses durant les compilations système. Une option particulièrement utile permet de générer des pages de manuel compressées pour économiser de l'espace disque. Celles-ci peuvent être consultées de la manière habituelle avec la commande `man`. Pour cela, ajoutez la ligne suivante à `/etc/mk.conf` :

```
MANZ=yes
```

Une autre option utile permet de générer les pages de manuel aux formats PostScript et ASCII. Cette option, `MANPS=yes`, est à ajouter dans `/etc/mk.conf`. Voir [mk.conf\(5\)](#) pour plus de détails.

Que sont les fichiers info ?

Une partie de la documentation d'OpenBSD est sous la forme de fichiers info. Ces fichiers se trouvent typiquement sous `/usr/share/info`. C'est une forme de documentation alternative fournie par GNU. Plusieurs fichiers info sont plus à jour que les pages de manuel fournis par GNU et peuvent être visualisés à l'aide de la commande [info\(1\)](#). Par exemple, pour voir les informations sur le compilateur GNU, [gcc\(1\)](#), tapez :

```
# info gcc
```


Après avoir utilisé info, vous allez vraiment apprécier nos pages de manuel !

Comment afficher les pages de manuel en couleur dans un XTerm ?

Le fichier de configuration par défaut de [xterm\(1\)](#) n'affiche pas les pages de manuel en couleur. Afin d'avoir un affichage couleur, copiez le fichier `/etc/X11/app-defaults/XTerm-color` dans votre répertoire personnel et renommez-le en `.Xdefaults`. Veillez à ne pas écraser un paramétrage particulier dans `.Xdefaults`. Ce fichier contient tous les paramètres dont vous aurez besoin pour activer les couleurs sous XTerm. Cependant, trois lignes doivent être décommentées auparavant :

```
!*VT100*colorULMode: on
!*VT100*underLine: off
!*VT100*colorBDMode: on
```

Le reste du fichier permet de choisir les couleurs pour plusieurs paramètres. Les lignes applicables aux pages de manuel sont :

```
*VT100*colorUL: yellow
*VT100*colorBD: white
```

Ce qui produit des pages de manuel avec une couleur infernale, adaptez-la à votre convenance : pouvons nous nous permettre de suggérer rouge pour "colorUL" et magenta pour "colorBD" ? Il existe aussi un afficheur de pages de manuel sous X11, [xman\(1\)](#), qui fournit une interface alternative (graphique) aux pages de manuel. Consultez les pages de manuel de `xterm` et `xman` pour plus d'informations.

Comment écrire ma propre page de manuel ?

Si vous souhaitez écrire votre propre page de manuel pour une application que vous avez écrite, un tutorial est fourni dans [mdoc.samples\(7\)](#). Il existe aussi un guide de référence bien pratique dans [mdoc\(7\)](#).

2.4 - Rapporter les bugs

Avant de crier "Au bug!", merci de bien vouloir vous assurer que c'est réellement le cas. Par contre, si vous ne comprenez pas comment telle ou telle chose est implémentée par OpenBSD ou comment elle fonctionne, et vous ne trouvez pas comment résoudre le problème à l'aide des [pages de manuel](#) ou du site OpenBSD, utilisez les [listes de discussion](#) (généralement `misc@openbsd.org`) pour demander de l'aide. Si c'est votre première expérience avec OpenBSD, soyez réaliste : vous n'avez probablement pas découvert un bug inconnu. Notez aussi que du matériel défectueux peut mimer un bug logiciel. Alors prenez le temps de vérifier l'état de votre matériel avant de décider que vous avez trouvé un "bug".

Finalement, avant de soumettre un rapport de bug, merci de lire <http://www.openbsd.org/fr/report.html>.

Faire un rapport de bug correct est l'une des plus grandes responsabilités conférée aux utilisateurs. Des informations très détaillées sont nécessaires pour diagnostiquer les bugs les plus sérieux. Les développeurs reçoivent fréquemment des rapports de bugs par mail du style :

```
From: joeuser@example.com
To: bugs@openbsd.org
Subject: HELP!!!
```

I have a PC and it won't boot!!!!!! It's a 486!!!!!!

Heureusement la plupart des gens savent que de tels rapports sont rapidement effacés. Tous les rapports de bugs doivent contenir des informations détaillées. Si Joe User souhaite que son bug soit corrigé, il faudrait que son rapport ressemble à ça :

```
From: smartuser@example.com
To: bugs@openbsd.org
Subject: 3.3-beta panics on a SPARCStation2
```

OpenBSD 3.2 installed from an official CD-ROM installed and ran fine on this machine.

After doing a clean install of 3.3-beta from an FTP mirror, I find the system randomly panics after a period of use, and predictably and quickly when starting X.

This is the dmesg output:

```
OpenBSD 3.3-beta (GENERIC) #9: Mon Mar 17 12:37:18 MST 2003
  deraadt@sparc.openbsd.org:/usr/src/sys/arch/sparc/compile/GENERIC
real mem = 67002368
avail mem = 59125760
using 200 buffers containing 3346432 bytes of memory
bootpath: /sbus@1,f8000000/esp@0,800000/sd@1,0
mainbus0 (root): SUNW,Sun 4/75
cpu0 at mainbus0: CY7C601 @ 40 MHz, TMS390C602A FPU; cache chip bug
- trap page uncached
cpu0: 64K byte write-through, 32 bytes/line, hw flush cache enabled
memreg0 at mainbus0 iaddr 0xf4000000
clock0 at mainbus0 iaddr 0xf2000000: mk48t02 (eeprom)
timer0 at mainbus0 iaddr 0xf3000000 delay constant 17
auxreg0 at mainbus0 iaddr 0xf7400003
zs0 at mainbus0 iaddr 0xf1000000 pri 12, softpri 6
zstty0 at zs0 channel 0 (console i/o)
zstty1 at zs0 channel 1
zs1 at mainbus0 iaddr 0xf0000000 pri 12, softpri 6
zskbd0 at zs1 channel 0: reset timeout
zskbd0: no keyboard
zstty2 at zs1 channel 1: mouse
audioamd0 at mainbus0 iaddr 0xf7201000 pri 13, softpri 4
audio0 at audioamd0
sbus0 at mainbus0 iaddr 0xf8000000: clock = 20 MHz
dma0 at sbus0 slot 0 offset 0x400000: rev 1+
```

```

esp0 at sbus0 slot 0 offset 0x800000 pri 3: ESP100A, 25MHz, SCSI ID 7
scsibus0 at esp0: 8 targets
sd0 at scsibus0 targ 1 lun 0: <SEAGATE, ST1480 SUN0424, 8628> SCSI2 0/direct fixed
sd0: 411MB, 1476 cyl, 9 head, 63 sec, 512 bytes/sec, 843284 sec total
sd1 at scsibus0 targ 3 lun 0: <COMPAQPC, DCAS-32160, S65A> SCSI2 0/direct fixed
sd1: 2006MB, 8188 cyl, 3 head, 167 sec, 512 bytes/sec, 4110000 sec total
le0 at sbus0 slot 0 offset 0xc00000 pri 5: address 08:00:20:13:10:b9
le0: 16 receive buffers, 4 transmit buffers
cgsix0 at sbus0 slot 1 offset 0x0: SUNW,501-2325, 1152x900, rev 11
wsdisplay0 at cgsix0
wsdisplay0: screen 0 added (std, sun emulation)
fdc0 at mainbus0 ioaddr 0xf7200000 pri 11, softpri 4: chip 82072
fd0 at fdc0 drive 0: 1.44MB 80 cyl, 2 head, 18 sec
root on sd0a
rootdev=0x700 rrootdev=0x1100 rawdev=0x1102

```

This is the panic I got when attempting to start X:

```

panic: pool_get(mclpl): free list modified: magic=78746572; page 0xf9a93000;
       item addr 0xf9a93000
Stopped at      Debugger+0x4:      jmp1          [%o7 + 0x8], %g0
RUN AT LEAST 'trace' AND 'ps' AND INCLUDE OUTPUT WHEN REPORTING THIS PANIC!
DO NOT EVEN BOTHER REPORTING THIS WITHOUT INCLUDING THAT INFORMATION!
ddb> trace
pool_get(0xf9a93000, 0x22, 0x0, 0x1000, 0x102, 0x0) at pool_get+0x2c0
sosend(0x16, 0xf828d800, 0x0, 0xf83b0900, 0x0, 0x0) at sosend+0x608
soo_write(0xfac0bf50, 0xfac0bf70, 0xfac9be28, 0xfab93190, 0xf8078f24, 0x0)
at soo_write+0x18
dofilewritev(0x0, 0xc, 0xfac0bf50, 0xf7fff198, 0x1, 0xfac0bf70) at
dofilewritev+0x12c
sys_writev(0xfac87508, 0xfac9bf28, 0xfac9bf20, 0xf80765c8, 0x1000, 0xfac0bf70)
at sys_writev+0x50
syscall(0x79, 0xfac9bfb0, 0x0, 0x154, 0xfcfffffff, 0xf829dea0) at syscall+0x220
slowtrap(0xc, 0xf7fff198, 0x1, 0x154, 0x1, 0xfac87508) at slowtrap+0x1d8
ddb> ps
  PID   PPID   PGRP     UID  S      FLAGS  WAIT      COMMAND
 27765   8819  29550     0   3      0x86   netio     xconsole
  1668   29550  29550     0   3      0x4086  poll      fvwm
15447   29550  29550     0   3      0x44186 poll      xterm
  8819   29550  29550    35   3      0x4186  poll      xconsole
  1238   29550  29550     0   3      0x4086  poll      xclock
29550   25616  29550     0   3      0x4086  pause     sh
  1024   25523  25523     0   3      0x40184 netio     XFree86
*25523  25616  25523    35   2      0x44104          XFree86
25616   30876  30876     0   3      0x4086  wait      xinit
30876   16977  30876     0   3      0x4086  pause     sh
16977     1   16977     0   3      0x4086  ttyin     csh
  5360     1   5360     0   3      0x84   select    cron
14701     1  14701     0   3      0x40184 select    sendmail
12617     1  12617     0   3      0x84   select    sshd

```

```

27515      1  27515      0  3          0x184  select      inetd
 1904      1   1904      0  2           0x84          syslogd
 9125      1   9125      0  3          0x84  poll       dhclient
   7       0     0       0  3      0x100204  crypto_wa  crypto
   6       0     0       0  3      0x100204  aiodoned  aiodoned
   5       0     0       0  3      0x100204  syncer    update
   4       0     0       0  3      0x100204  cleaner   cleaner
   3       0     0       0  3      0x100204  reaper    reaper
   2       0     0       0  3      0x100204  pgdaemon  pagedaemon
   1       0     1       0  3         0x4084  wait      init
   0      -1     0       0  3      0x80204  scheduler swapper

```

Thank you!

Consultez [report.html](#) concernant la manière de créer et d'envoyer des rapports des bogues. Des informations détaillées à propos de votre matériel sont nécessaires si vous pensez que le bogue peut être, *de quelque manière que se soit*, lié à votre matériel ou à sa configuration. Habituellement, un [dmesg\(8\)](#) est suffisant. Une description détaillée de votre problème est nécessaire. Vous remarquerez que le dmesg décrit le matériel, le texte explique pourquoi Smart User pense que son système n'est pas défectueux (3.2 fonctionnait convenablement), comment ce "crash" était causé (en démarrant X), et inclut le détail des commandes "ps" et "trace" afin de permettre aux développeurs de déboguer. Dans ce cas, Smart User a fourni ces détails capturés via une [console série](#), si vous ne savez pas le faire, vous pouvez utiliser une feuille et un stylo afin de retranscrire le "crash". (L'exemple ci-dessus était un problème réel qui affectait les systèmes sun4c, et les informations fournies dans le précédent rapport ont aidé à le résoudre).

Si notre ami Smart User possède un système OpenBSD fonctionnel et qu'il veut soumettre un rapport de bug, il peut le faire en utilisant l'utilitaire [sendbug\(1\)](#) qui enverra le rapport au système de suivi des bugs GNATS. Évidemment, vous ne pouvez utiliser [sendbug\(1\)](#) si votre système ne démarre pas mais utilisez-le dès que c'est possible. Vous aurez toujours à inclure des informations détaillées sur ce qui s'est passé, la configuration de votre système et comment reproduire le problème. La commande [sendbug\(1\)](#) nécessite que votre système puisse envoyer des messages électroniques sur l'Internet. Notez que le serveur de messagerie utilise la fonctionnalité "greylisting" de [spamd\(8\)](#). Il peut alors s'écouler jusqu'à une demi-heure avant que le serveur de messagerie n'accepte votre rapport. Soyez donc patient.

Après avoir envoyé un rapport de bogue via [sendbug\(1\)](#), vous serez averti de son état par e-mail. Vous pourrez être contacté par les développeurs pour fournir de plus amples informations ou tester des correctifs. Vous pouvez également consulter les archives de la liste de diffusion bugs@openbsd.org, de plus amples informations se trouvant sur la [page des listes de discussion](#), ou consulter la base de données des rapports de bogues en ligne sur notre [Système de Consultation des Bogues](#).

Comment fournir plus d'informations utiles aux développeurs

Voici quelques astuces supplémentaires :

Vous avez perdu le "Panic message" ?

Dans certaines conditions, vous pouvez perdre le tout premier message d'une panique système, et il s'agit de celui qui donne la raison de la panique. Ce message est très important, vous devez donc l'inclure dans votre rapport. Vous pouvez le retrouver en utilisant la commande "show panic" dans ddb> comme suit :

```
ddb> show panic
0:      kernel: page fault trap, code=0
ddb>
```

Dans ce cas, le message fût "Kernel: page fault trap, code=0".

Remarque pour les systèmes SMP :

Vous devez obtenir une "trace" pour chaque processeur et les insérer dans votre rapport :

```
ddb{0}> trace
pool_get(d05e7c20,0,dab19ef8,d0169414,80) at pool_get+0x226
fxp_add_rfabuf(d0a62000,d3c12b00,dab19f10,dab19f10) at
fxp_add_rfabuf+0xa5
fxp_intr(d0a62000) at fxp_intr+0x1e7
Xintr_ioapic0() at Xintr_ioapic0+0x6d
--- interrupt ---
idle_loop+0x21:
ddb{0}> machine ddb 1
Stopped at      Debugger+0x4:   leave
ddb{1}> trace
Debugger(d0319e28,d05ff5a0,dab1bee8,d031cc6e,d0a61800) at Debugger+0x4
i386_ipi_db(d0a61800,d05ff5a0,dab1bef8,d01eb997) at i386_ipi_db+0xb
i386_ipi_handler(b0,d05f0058,dab10010,d01d0010,dab10010) at
i386_ipi_handler+0x
4a
Xintripi() at Xintripi+0x47
--- interrupt ---
i386_softintlock(0,58,dab10010,dab10010,d01e0010) at
i386_softintlock+0x37
Xintrltimer() at Xintrltimer+0x47
--- interrupt ---
idle_loop+0x21:
ddb{1}>
```

Répétez la commande "machine ddb x" suivie de "trace" pour chaque processeur de votre machine.

Comment récupérer des informations d'un crash du noyau.

Un crash du noyau typique sur OpenBSD ressemble à cela : (les choses à regarder sont en gras dans le texte)

```
kernel: page fault trap, code=0
Stopped at      _pf_route+0x263:      mov      0x40(%edi),%edx
ddb>
```

La première commande à exécuter du prompt ddb> est "trace" (voir [ddb\(4\)](#) pour plus de détails):

```

ddb> trace
_pf_route(e28cb7e4,e28bc978,2,1fad,d0b8b120) at
_pf_route+0x263
_pf_test(2,1f4ad,e28cb7e4,b4c1) at _pf_test+0x706
_pf_route(e28cbb00,e28bc978,2,d0a65440,d0b8b120) at _pf_route+0x207
_pf_test(2,d0a65440,e28cbb00,d023c282) at _pf_test+0x706
_ip_output(d0b6a200,0,0,0,0) at _ip_output+0xb67
_icmp_send(d0b6a200,0,1,a012) at _icmp_send+0x57
_icmp_reflect(d0b6a200,0,1,0,3) at _icmp_reflect+0x26b
_icmp_input(d0b6a200,14,0,0,d0b6a200) at _icmp_input+0x42c
_ipv4_input(d0b6a200,e289f140,d0a489e0,e289f140) at _ipv4_input+0x6eb
_ipintr(10,10,e289f140,e289f140,e28cbd38) at _ipintr+0x8d
Bad frame pointer: 0xe28cbcac
ddb>

```

Cela nous informe de la fonction qui nous a conduit au crash.

Pour retrouver précisément la ligne de code C qui a causé le crash, vous pouvez faire les choses suivantes :
 Trouvez le fichier source ou la fonction qui a crashé a été définie. Dans cet exemple, cela doit être `pf_route()` dans `sys/net/pf.c`. Recompiliez le fichier source avec les informations de débogage :

```

# cd /usr/src/sys/arch/$(uname -m)/compile/GENERIC/
# rm pf.o
# DEBUG=-g make pf.o

```

Puis utilisez [objdump\(1\)](#) pour avoir le désassemblage :

```

# objdump --line --disassemble --reloc pf.o >pf.dis

```

Sur la sortie, un `grep` sur le nom de la fonction (`pf_route` dans notre exemple) :

```

# grep "<_pf_route>:" pf.dis
00007d88 <_pf_route>:

```

Prenez le premier nombre hexadécimal et ajoutez l'offset de la ligne 'Stopped at' : **`0x7d88 + 0x263 == 0x7feb`**.
 Allez vers cette ligne (l'instruction assembleur doit correspondre à celle de la ligne 'Stopped at'), puis sur le numéro de ligne C le plus proche :

```
# more pf.dis
/usr/src/sys/arch/i386/compile/GENERIC/../../../../net/pf.c:3872
7fe7:      0f b7 43 02      movzwl 0x2(%ebx),%eax
7feb:      8b 57 40      mov    0x40(%edi),%edx
7fee:      39 d0      cmp    %edx,%eax
7ff0:      0f 87 92 00 00 00    ja    8088 <_pf_route+0x300>
```

Dans notre cas, c'est précisément la ligne **3872** de pf.c qui crash :

```
# cat -n pf.c | head -n 3872 | tail -n 1
3872      if ((u_int16_t)ip->ip_len <= ifp->if_mtu) {
```

Il faut remarquer que le noyau qui produit la sortie du crash et le fichier objet pour objdump doivent être compilés avec les mêmes fichiers sources, sinon l'offset ne correspondra pas.

Si vous fournissez aussi bien la sortie de ddb> trace et la section objdump correspondante, c'est très utile.

[\[Index de la FAQ\]](#) [\[Section 1 - Introduction à OpenBSD\]](#) [\[Section 3 - Démarrer avec OpenBSD\]](#)



www@openbsd.org

\$OpenBSD: faq2.html,v 1.65 2010/07/30 16:48:33 ajacoutot Exp \$

[\[Index de La FAQ\]](#) [\[Section 2 - Autres Ressources d'Information OpenBSD\]](#) [\[Section 4 - Guide d'Installation\]](#)

3 - Démarrer avec OpenBSD

Table des matières

- [3.1 - Acheter un jeu de CD OpenBSD](#)
 - [3.2 - Acheter des T-shirts OpenBSD](#)
 - [3.3 - Est-ce qu'OpenBSD fournit une image ISO disponible en téléchargement ?](#)
 - [3.4 - Télécharger par FTP, HTTP ou AFS](#)
 - [3.5 - Choisir son matériel](#)
 - [3.6 - Quel est le "premier système" approprié pour apprendre OpenBSD dessus ?](#)
-

3.1 - Acheter un jeu de CD OpenBSD

Acheter un jeu de CD OpenBSD est sans doute la meilleure manière de commencer. Veuillez consulter la page des commandes pour en obtenir une copie : [Commandes](#).

Il y a plusieurs bonnes raisons d'acheter un jeu de CD OpenBSD :

- La vente des jeux de CDs finance le développement d'OpenBSD.
- Le développement d'un système d'exploitation multi plates-formes nécessite beaucoup d'investissements en matériels.
- Votre contribution sous la forme de l'achat d'un jeu de CD a un véritable impact sur le développement futur.
- Les CDs contiennent les binaires (et les sources) pour toutes les plates-formes supportées.
- Les CDs sont "bootables" sur plusieurs plates-formes et peuvent être utilisés pour démarrer une machine sans système pré-installé.
- Les CD sont utiles pour démarrer un système même si vous choisissez d'installer un "snapshot".
- Installer depuis les CDs est plus rapide ! De plus cela permet de préserver la bande passante.
- Les jeux de CDs OpenBSD sont toujours fournis avec de jolis autocollants. Votre système n'est

pas vraiment complet sans eux. Vous ne pouvez les obtenir que si vous achetez un jeu de CD ou donnez du matériel.

- Les jeux de CDs OpenBSD contiennent un assortiment de [paquetages](#) populaires et utiles. Ceux-ci sont suffisamment complets pour installer un environnement de travail et de développement sans aucune connexion réseau.

Pour installer une Révision (Release) d'OpenBSD, vous devriez utiliser un jeu de CD officiel.

3.2 - Acheter des T-shirts OpenBSD

OpenBSD propose des T-shirts pour votre plaisir vestimentaire. Vous pouvez les visualiser sur la page : [T-shirts OpenBSD](#).

3.3 - Est-ce qu'OpenBSD fournit une image ISO disponible en téléchargement ?

Pour certaines plate-formes, oui !

La plupart des plate-formes peuvent démarrer du CD et peuvent télécharger et installer une image ISO qui peut être utilisée pour créer un CD-ROM bootable et installer tout OpenBSD. Les ISOs "full install" (tous les fichiers nécessaires pour une installation complète de OpenBSD, appele install47.iso) et "boot only" (pour l'installation d'un source de fichiers réseau, appelé cd47.iso) sont fournis pour ces plateformes.

Notez que ces ISOs *ne sont pas* les mêmes images que celle disponible sur la [distribution officielle](#). Ces images sont faites pour une plate-forme précise et n'inclues aucun des [packages](#) précompilés, autocollants ou dessins distribués avec les CD officiels.

Cependant une installation par un fichier ISO n'est PAS la solution optimum pour bien des gens. Il est habituellement toujours plus rapide et plus simple de télécharger les fichiers de base et d'installer par la suite juste les suppléments nécessaires. Bien sûr, pour ceux qui veulent faire un certain nombre d'installations, ou qui ne savent pas comment on grave 10 fichiers sur un CD-ROM ou encore comment configurer un serveur FTP local, les ISOs sont disponibles.

Le projet OpenBSD ne fournit pas en téléchargement les images ISO utilisées pour créer les masters des CDs officiels. La raison est simplement que nous souhaiterions que vous achetiez les CDs, pour aider au financement des développements futurs dans OpenBSD. La disposition du CD-ROM officiel est copyright Theo de Raadt. Theo n'autorise pas les gens à redistribuer des images des CDs OpenBSD officiels. Pour inciter les gens à acheter les CDs, plusieurs extras sont aussi inclus dans le paquetage (Dessins, Autocollants, etc).

Notez que seule la disposition du CD est sous copyright, OpenBSD lui-même est libre. Rien n'empêche quelqu'un d'autre de récupérer OpenBSD et de créer son propre CD.

Pour ceux qui ont besoin d'un CD bootable pour leur système, des images ISO des disquettes de démarrage (appelées `cd47.iso`) sont disponibles pour un certain nombre de plates-formes et permettront d'installer le système par FTP. Ces images ISO ont une taille de quelques mégaoctets seulement, et contiennent uniquement les outils d'installation. Elles ne contiennent pas les jeux de fichiers constituant le système.

3.4 - Télécharger par FTP, HTTP ou AFS

Il y a plusieurs miroirs internationaux offrant un accès FTP et HTTP aux snapshots et versions stables d'OpenBSD. L'accès par AFS est aussi disponible. Il est préférable de toujours utiliser le site le plus proche de vous. Avant de commencer à télécharger vous pouvez utiliser [ping \(8\)](#) et [traceroute\(8\)](#) pour déterminer quel est le site miroir le plus proche et voir si celui-ci fonctionne correctement. Bien sûr votre CD d'OpenBSD est toujours plus proche qu'un site miroir. Les informations sont disponibles ici :

[Page FTP OpenBSD.](#)

3.5 - Choisir son matériel

Sélectionner le matériel approprié pour exécuter un système OpenBSD dessus est important, et cela fait toute la différence entre le succès ou l'échec de ce projet.

Si vous allez acheter un nouveau PC, soit vous allez l'acheter pièce par pièce ou déjà complet, vous voulez être sûr que vous achetez des composants fiables. Dans le monde PC, cela n'est pas simple. **Des composants mauvais ou peu fiables peuvent faire fonctionner OpenBSD lentement et planter souvent.** Le meilleur avis que nous pouvons vous donner et d'être prudent, et d'acheter des marques et composants qui ont été validé par une autorité que vous pouvez croire. Quelquefois, une machine plus chère est une machine de meilleure qualité. D'autres fois c'est simplement plus cher.

Il y a certaines choses qui permettent d'obtenir le maximum de performances de votre système :

- **Laisser l'application choisir le matériel :** Il est habituellement préférable de faire des ajustements sur le matériel que vous prévoyez d'utiliser plutôt que de vous compromettre sur la conception de votre application parce que vous avez quelque chose que vous "voulez absolument utiliser".
- **Identifier les goulets d'étranglement :** Ne payez pas plus pour le dernier modèle de processeur si votre application est limitée par les I/O disque. Ne payez pas pour des disques rapides si votre système est limité par la vitesse du réseau. Ne payez pour rien de plus si votre goulet

d'étranglement est une ligne DSL à 128Kbps.

- **Faire simple** : Un matériel simple a souvent des problèmes simples. Un matériel complexe qui est supposé ne jamais tomber en panne vous prendra beaucoup de temps à être réparé si il tombe en panne quand même.
- **Utilisez du matériel que vous comprenez ou apprenez à utiliser ce nouveau matériel avant de le mettre en production** : Indépendamment de la qualité technique du matériel, en s'engageant à utiliser une type de matériel avant de vous familiariser avec lui, la manière dont il fonctionne et *comment il tombe en panne*, est stupide.
- **Utiliser plusieurs disques** : Au lieu d'acheter un gros disque, achetez en plusieurs petits. Bien que cela coûte plus cher, distribuer la charge sur plusieurs axes va diminuer le temps nécessaire pour accéder aux données sur les disques. Et avec plus d'axes, vous obtiendrez plus de disponibilité et un accès plus rapide aux données avec du RAID.
- **Casser les gros blocs de stockage** : Beaucoup de personnes font l'erreur de concevoir un système qui stocke de grosses quantités de données pour obtenir un Gros Bloc de Stockage. C'est naturellement une mauvaise conception. Vous trouverez finalement que c'est bien mieux de casser votre stockage en blocs administrables. Cela possède plusieurs avantages, deux des plus importants sont que vous pourrez ajouter plus de stockage plus tard quand vous en aurez besoin facilement (vous n'allez pas faire les bonnes estimations des le départ, personne ne le fait jamais), et vous pouvez ajouter une petite quantité de stockage maintenant, et en ajouter plus dans le futur quand le prix aura bien diminué et les capacités augmentées.
- **Evitez les adaptateurs réseaux bon marché** : OpenBSD supporte une grande gamme d'adaptateurs réseaux bon marché. Ces adaptateurs fonctionnent bien sur des systèmes maison, et sur des débits moyens de travail et d'environnements de recherche. Mais si vous avez besoin de plus de débit et un impact faible sur votre serveur, il faut mieux acheter un adaptateur réseau de qualité. Malheureusement, certains adaptateurs de marque ne sont pas meilleurs que des adaptateurs bon marché, et certains adaptateurs potentiellement de meilleure qualité ne possèdent pas la documentation nécessaire pour écrire un bon pilote. Les adaptateurs Gigabit permettent souvent d'obtenir de meilleures performances que des adaptateurs 10Mbps/100Mbps, même utilisés sur des réseaux moins rapides à cause de tampons plus importants.

3.6 - Quel est le "premier système" approprié pour apprendre OpenBSD dessus ?

Bien que OpenBSD fonctionnera sur un ordinateur plus petit, plus vieux et moins puissant comme tous les autres OS modernes, si vous démarrez sous OpenBSD, utiliser une machine trop petite peut-être frustrant. Les lignes de conduite suivantes ne sont QUE des lignes de conduite, OpenBSD fonctionnera très bien sur bien plus d'équipements modestes qu'affiché ici, mais cela sera plus frustrant que nécessaire pour une première utilisation.

- **Plate-forme** : Utilisez une plate-forme dont vous êtes déjà familié. Quand vous apprenez un nouveau système d'exploitation, c'est le mauvais moment pour aussi apprendre une nouvelle plate-forme. Nous considererons que vous allez utiliser la plate-forme [i386](#) ici, qui est probablement la

plus familière pour tout le monde.

- **Processeur:** Pentium 100Mhz ou mieux. Oui, OpenBSD fonctionnera sur un 80486 25Mhz, mais vous ne voudrez pas faire l'expérience, avec l'ennui et les rechargements dont vous aurez besoin pour apprendre le système sur une machine lente. La première irritation que vous rencontrerez avec un processeur aussi lent et le temps que cela prendra pour faire un SSH sur la machine. Si vous voulez exécuter X, vous voudrez probablement aller au minimum vers un 200Mhz. X fonctionne actuellement vraiment bien sur une machine plus lente même chargée, mais il prendra son temps à se charger et démarrer.
- **64Mo de RAM ou plus :** Si vous voulez exécuter X, 128Mo sera mieux pour démarrer.
- **Disque dur :** Un disque dur de 1Go vous permettra une installation facile sur un système simple comme un pare-feu, serveur DNS ou similaire. Si vous voulez recompiler le système des sources, il vous faudra plutôt un disque de 4Go, et vous vous voulez recompiler X aussi, vous aurez besoin de 6Go ou plus. IDE est recommandé pour démarrer. Si vous possédez un disque plus gros, ne vous sentez pas obligé d'allouer tout le disque initialement -- il n'y a rien de mal à laisser 72Go sur un disque de 80Go si votre besoin est uniquement de 8Go.
- **Adaptateur réseau :** Utilisez un adaptateur PCI. Bien que l'urgence d'utiliser cet ancien adaptateur ISA est fort, résistez. Vous ne vous souvenez probablement pas comment le configurer correctement. Si vous prévoyez de mettre plusieurs adaptateurs réseaux dans la machine, écrivez la MAC adresse sur le dos de la carte avant de refermer le boîtier.
- **Multiboot :** Pour votre première installation d'OpenBSD, n'essayez pas de faire un multiboot avec d'autres OS. [Multibooting](#) est un processus difficile à faire fonctionner correctement, et vous devez connaître parfaitement bien les OS à mettre avant de tenter le coup, ce qui est clairement pas le cas avec votre première installation. Il est très possible de supprimer accidentellement toutes les données du système. Au lieu de cela, utilisez un ordinateur dédié, ou au moins, un disque dédié sur la machine.
- **Portables :** Bien que de nombreux portables fonctionnent parfaitement avec OpenBSD, il ne sont quelquefois pas le meilleur système pour le faire fonctionner dessus, donc un portable n'est peut-être pas le meilleur choix pour votre première installation de OpenBSD. Cependant, des que vous êtes confiant avec OpenBSD, un portable peut-être un outil très utile.
- **Nouveau matériel :** Le matériel dernier cri n'est quelquefois pas bien supporté par OpenBSD, donc pour votre premier système OpenBSD, un machine un peu plus ancienne est recommandée.

Evidemment, "plus on en a, mieux c'est". Certaines applications populaires, peuvent apparemment utiliser autant de processeurs et de mémoire que dispose le système.

[\[Index de La FAQ\]](#) [\[Section 2 - Autres Ressources d'Information OpenBSD\]](#) [\[Section 4 - Guide d'installation OpenBSD\]](#)



[www@openbsd.org](http://www.openbsd.org)

\$OpenBSD: faq3.html,v 1.50 2010/05/23 16:20:40 ajacoutot Exp \$

4 - Guide d'Installation d'OpenBSD 4.7

Table des Matières

- [4.1 - Présentation de la procédure d'installation d'OpenBSD](#)
- [4.3 - Vérifications avant l'installation](#)
- [4.3 - Créer un média d'installation OpenBSD amorçable](#)
 - [4.3.1 - Créer un CD-ROM amorçable](#)
 - [4.3.2 - Créer des disquettes de démarrage sur Unix](#)
 - [4.3.3 - Créer des disquettes de démarrage sur Windows](#)
- [4.4 - Démarrer le média d'installation OpenBSD](#)
- [4.5 - Installer OpenBSD simplement](#)
 - [4.5.1 - Commencer l'installation](#)
 - [4.5.2 - Questions lors de l'installation](#)
 - [4.5.3 - Configurer les disques](#)
 - [4.5.4 - Choisir le média d'installation et les sets de fichiers](#)
 - [4.5.5 - Premier démarrage !](#)
 - [4.5.6 - Une dernière chose...](#)
- [4.6 - Détails pour une installation plus complexe](#)
 - [4.6.1 - Configurer le réseau](#)
 - [4.6.2 - Configurer le fuseau horaire](#)
 - [4.6.3 - Personnaliser la configuration via fdisk\(8\)](#)
 - [4.6.4 - Personnaliser la configuration via disklabel](#)
- [4.7 - Quels sont les fichiers nécessaires à l'installation ?](#)
- [4.8 - Comment est-ce que je dois partitionner mon disque ?](#)
- [4.9 - Multiboot OpenBSD/i386](#)
- [4.10 - Envoyer votre dmesg à dmesg@openbsd.org après l'installation](#)
- [4.11 - Ajouter un paquetage après l'installation](#)
- [4.12 - Qu'est ce que 'bsd.rd' ?](#)
- [4.13 - Problèmes d'installation courants](#)
 - [4.13.1 - Mon Compaq ne reconnaît que 16M de RAM](#)
 - [4.13.2 - Mon i386 ne démarre pas après l'installation](#)
 - [4.13.3 - Ma machine a démarré, mais bloque pendant la procédure ssh-keygen](#)
 - [4.13.4 - J'ai le message "Failed to change directory" pendant l'installation](#)
 - [4.13.5 - Ma table de partition fdisk est corrompue ou vide !](#)
 - [4.13.6 - Je ne possède pas de lecteur de disquette ou de CD-ROM sur ma machine](#)
- [4.14 - Personnaliser la procédure d'installation](#)
- [4.15 - Comment puis-je installer plusieurs systèmes identiques ?](#)

- [4.16 - Comment puis-je obtenir un dmesg\(8\) pour rapporter un problème d'installation ?](#)

4.1 - Présentation de la procédure d'installation OpenBSD

OpenBSD a longtemps été respectée pour son processus d'installation simple, et qui est cohérent à travers toutes les plateformes.

La plupart des architectures suivent une procédure d'installation similaire ; cependant quelques détails diffèrent sur quelques plateformes. Dans tous les cas, il vous est vivement conseillé de lire le document `INSTALL` spécifique à votre architecture se trouvant dans le dossier "*platform*" sur le CD-ROM ou les sites FTP (par exemple, `i386/INSTALL.i386`, `macppc/INSTALL.macppc` ou `sparc/INSTALL.sparc`).

L'installation d'OpenBSD utilise un noyau [spécial](#) avec un certain nombre d'utilitaires et de scripts d'installation inclus dans un disque RAM préchargé. Après le démarrage de ce noyau, le système d'exploitation est extrait depuis plusieurs fichiers [tar\(1\)](#) (.tgz) compressés à partir d'une source autre que ce disque RAM préchargé. Il existe de nombreux moyens de démarrer ce noyau d'installation :

- **Disquette** : OpenBSD peut être installé sur beaucoup de plateformes en démarrant l'installation à partir d'une seule disquette. Cependant, à cause de contrainte d'espace, des plateformes plus gourmandes (sparc64, amd64 et alpha) ne possèdent pas certaines utilitaires qui peuvent être important pour vous, comme le client DHCP pour configurer le réseau. Pour ces plateformes, il est préférable d'utiliser l'installation avec le CD. Cependant, pour les plateformes comme sparc et i386, vous trouverez la disquette de démarrage très complète.

Des images de disquettes utilisables pour créer une disquette d'installation depuis un autre système [Compatible Unix](#) ou sur un système [Windows](#) sont fournies. Les noms de fichiers typiques sont `floppy47.fs`, bien que plusieurs architectures possèdent de multiples images de disquettes.

- **CD-ROM** : Sur plusieurs architectures une image CD-ROM (`cd47.iso` pour effectuer uniquement un démarrage, ou `install47.iso` pour l'installation complète) est fournie permettant la création d'un CDROM amorçable.
- **Partition existante** : Le noyau du [disque RAM](#) peut être démarré à partir d'une partition OpenBSD existante (i.e., une mise à jour ou une réinstallation).
- **Réseau**: Quelques architectures supportent le démarrage à travers un réseau (en utilisant par exemple [PXE](#) ou d'autres méthodes de [démarrage réseau](#)).
- **Ecrire un système de fichiers sur disque (miniroot)** : une image de système de fichiers qui peut être écrite sur une partition existante et qui peut être amorcée par la suite.
- **Cassettes amorçables** : Certaines architectures supportent l'amorçage depuis des cassettes. Ces cassettes peuvent être créées en suivant les instructions du fichier `INSTALL.platform`.

Toutes les [architectures](#) ne supportent pas chacune des options de démarrage :

- **alpha** : Disquette, CD-ROM, Réseau, Ecriture d'une image de disquette sur le disque dur.
- **amd64** : Disquette, CD-ROM, [Réseau](#).
- **armish**: Dépend du système hôte.
- **hp300** : CD-ROM, Réseau.
- **hppa** : Réseau.
- **i386** : Disquette, CD-ROM, [Réseau](#).

- **landisk:** miniroot, installation en utilisant un autre système.
- **macppc :** CD-ROM, Réseau.
- **mvme68k :** Réseau, Cassette amorçable.
- **mvme88k :** Réseau, Cassette amorçable.
- **sparc :** Disquette, CD-ROM, Réseau, Ecriture d'une image sur une partition swap existante, Cassette amorçable.
- **sparc64 :** Disquette (U1/U2 uniquement), CD-ROM, Réseau, Ecriture d'une image sur une partition existante.
- **vax :** Disquette, Réseau.
- **zaurus:** Démarrer bsd.rd à partir d'une partition Linux. Veuillez consulter [INSTALL.zaurus](#) pour plus de détails.

Toutes les installations autres que mac68k peuvent aussi utiliser un noyau [bsd.rd](#) lors d'une mise à jour ou d'une réinstallation.

Une fois le noyau démarré, vous avez plusieurs options pour obtenir les [paquetages d'installation](#). Une fois de plus, toutes les architectures ne supportent pas toutes ces options.

- **CD-ROM :** Bien sûr, nous préférons que vous utilisiez les [CD-ROMS Officiels](#), mais vous pouvez utiliser `install47.iso` ou créer les vôtres.
- **FTP :** L'un des différents [sites miroirs FTP](#) OpenBSD ou votre serveur FTP local contenant les paquetages.
- **HTTP :** L'un des différents [sites miroirs HTTP](#) OpenBSD ou votre serveur web local contenant les paquetages.
- **Partition de disque local :** Dans la plupart des cas, vous pouvez installer les paquetages depuis une autre partition de votre disque dur local. Par exemple, sur [i386](#), vous pouvez installer à partir d'une partition FAT ou d'un CD-ROM formaté au format ISO9660, Rock Ridge ou Joliet. Dans certains cas, vous devrez manuellement monter le système de fichiers avant de l'utiliser.
- **NFS :** Quelques architectures supportent les montages NFS de paquetages.
- **Cassette :** Les paquetages peuvent aussi être lus depuis une cassette formatée. Ces cassettes peuvent être créées en suivant les instructions du fichier `INSTALL.platform`.

4.3 - Vérifications avant l'installation

Avant de commencer votre installation, vous devriez avoir une idée de ce que vous allez devoir faire. Vous devriez connaître ces différents éléments au minimum :

- Nom de la machine.
- Matériel installé et disponible.
 - La compatibilité de votre architecture grâce à la page de compatibilité.
 - Dans les cas d'utilisation de matériel ISA, vous devrez aussi connaître les paramètres matériels et confirmer qu'ils sont tels qu'OpenBSD les requiert.
- La méthode d'installation qui sera utilisée (CD-ROM, FTP, etc.)
- Comment le système sera-t-il mis à jour et corrigé ?
 - Si cela est fait localement, vous devrez bénéficier de suffisamment [d'espace libre](#) pour stocker l'arbre des sources et le construire.
 - Autrement, vous aurez besoin d'accéder à une autre machine pour y construire une [release](#) patchée.
- Partitionnement de disque désiré.
 - Y-a-t-il des données à sauvegarder quelque part ?
 - OpenBSD coexistera-t-il sur ce système avec d'autres OS ? Si oui, comment seront-ils démarrés ? Avez-vous besoin d'installer un gestionnaire de démarrage ?
 - La totalité du disque sera-t-elle utilisée par OpenBSD ou voulez-vous conserver une partition ou un OS (ou de la place pour un autre) ?

- Comment souhaitez vous partitionner la partie réservée à OpenBSD de votre disque ?
- Paramètres Réseau, si vous n'utilisez pas DHCP :
 - Nom de domaine.
 - Adresse des Serveurs de Nom de Domaine (DNS).
 - Adresse IP et masque de sous-réseau pour chaque NIC.
 - Adresse de la passerelle.
- Lancerez vous le système de fenêtres X ?

4.3 - Créer un média d'installation OpenBSD amorçable

Comme exemples, nous allons regarder les images d'installation disponibles pour les architectures [i386](#) et [sparc](#).

L'architecture [i386](#) dispose de six images de disques séparées parmi lesquelles choisir :

- **cdemu47.iso** est une image ISO9660 se servant de l'émulation d'une disquette de 2.88M pour démarrer. Il faut espérer que peu de personnes auront à utiliser cette image -- la plupart utiliseront `cd47.iso`, n'utilisez `cdemu47.iso` que si `cd47.iso` ne fonctionne pas pour vous.
- **install47.iso** est une image ISO9660 contenant tous les fichiers d'installation standard. Ce fichier peut être utilisé pour créer un CD permettant de faire une installation OpenBSD de façon autonome.
- **floppy47.fs** (PC de bureau) supporte la plupart des cartes PCI et ISA, des adaptateurs IDE et des adaptateurs SCSI simples ainsi que quelques matériels PCMCIA. La plupart des utilisateurs utiliseront cette image.
- **floppyB47.fs** (Serveurs) supporte plusieurs contrôleurs RAID et quelques adaptateurs SCSI courants. Toutefois le support pour certains adaptateurs SCSI et plusieurs NICs EISA et ISA a été retiré.
- **floppyC47.fs** (Portables) supporte les matériels CardBus et PCMCIA trouvés sur la plupart des portables.

L'architecture [sparc](#) dispose de quatre images disques d'installation parmi lesquelles choisir :

- **floppy47.fs**: Supporte les systèmes équipés d'un lecteur de disquettes.
- **cd47.iso** Une image ISO utilisable pour créer votre propre CD afin d'amorcer les systèmes SPARC à l'aide un CD-ROM.
- **miniroot47.fs** peut être écrite sur une partition swap et démarrée.
- **install47.iso** est une image ISO9660 contenant tous les fichiers d'installation standard. Ce fichier peut être utilisé pour créer un CD permettant de faire une installation OpenBSD de façon autonome.

Sur les plateformes récentes, vous serez plus avisé d'utiliser les images de démarrage CDROM, comme dans certaines plateformes "gourmandes" (comme `sparc64` et `amd64`), les images disquette ont beaucoup de pilotes et d'utilitaires supprimés, ce qui peut compliquer l'installation. Les plateformes plus anciennes, comme le `i386` et `sparc`, sont toujours installable facilement d'une disquette.

4.3.1 - Créer un CD-ROM amorçable

Vous pouvez créer un CDROM en utilisant les fichiers `cd47.iso` ou `install47.iso`. Les détails exactes sont laissés ici au lecteur afin de déterminer quels outils sont à leur disposition.

Avec OpenBSD, vous pouvez créer un CD d'une image ISO en utilisant [cdio\(1\)](#) :

```
# cdio tao cd47.iso
```


La plupart des graveurs de CD vendus avec les systèmes Windows et Macintosh sont livrés avec des logiciels pour graver une image ISO sur un media vierge. Si le votre ne le permet pas, il y a plusieurs applications gratuites qui peuvent le faire pour vous.

D'autres systèmes Unix-like utilisent des applications comme [cdrkit](#).

4.3.2 - Créer des disquettes de démarrage sur Unix

Les détails précis et les noms de périphériques peuvent varier d'une variante Unix à une autre; utilisez ce qui est approprié sur votre système.

Pour créer une disquette formatée, utilisez la commande [fdformat\(1\)](#) pour chacun des formats et recherchez les secteurs défectueux..

```
# fdformat /dev/rfd0c
Format 1440K floppy `/dev/rfd0c'? (y/n): y
Processing VVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVV done.
```

Si votre sortie est identique à celle ci-dessus, votre disque est correct. Cependant, si vous ne voyez pas tous les "V" alors votre disque est probablement endommagé et vous devriez en essayer un autre.

Pour écrire un fichier image sur une disquette, utilisez [dd\(1\)](#). Un exemple d'utilisation de `dd(1)` est fourni ci-dessous :

```
# dd if=floppy47.fs of=/dev/rfd0c bs=32k
```

Une fois l'image écrite, vérifiez qu'elle a été correctement copiée et qu'elle est identique à l'image d'origine avec la commande [cmp\(1\)](#). Si la disquette est identique à l'image, vous ne verrez apparaître qu'une nouvelle invite de commande.

```
# cmp /dev/rfd0c floppy47.fs
```

4.3.3 - Créer des disquettes de démarrage sur Windows

Pour préparer la disquette sous Windows, utilisez tout d'abord les utilitaires natifs pour formater le disque, vérifiez qu'il ne possède pas de secteurs défectueux.

Pour écrire l'image d'installation sur la disquette préparée vous pouvez utiliser `ntrw.exe`, qui peut être téléchargé du répertoire [tools](#) de la plupart des miroirs FTP.

Exemple d'utilisation de `ntrw` :

```
C:\> ntrw floppy47.fs a:
3.5", 1.44MB, 512 bytes/sector
bufsize is 9216
1474560 bytes written
```

4.4 - Démarrer le média d'installation OpenBSD

Démarrer sur i386/amd64

Démarrer une image d'installation sur les architectures PC i386 et amd64 n'est pas nouveau pour la plupart des gens. Votre système devra être informé pour démarrer de n'importe quel média que vous avez choisi d'utiliser, habituellement à travers des options de configuration du BIOS. Si vous souhaitez démarrer depuis un CD-ROM, vous devrez aller dans votre BIOS système et autoriser le démarrage depuis celui-ci. Quelques anciens BIOS ne supportent pas cette option, et vous devrez utiliser une disquette pour démarrer votre image d'installation. Ne vous inquiétez pas ; bien que vous démarriez depuis la disquette, vous pourrez installer depuis le CD si cela est supporté par OpenBSD (par exemple, presque tous les disques IDE).

Vous pouvez aussi démarrer [bsd.rd](#) depuis une partition OpenBSD existante, ou depuis le réseau en utilisant la [procédure de démarrage PXE](#).

Démarrer sur sparc/sparc64

NOTE : Sur l'architecture [sparc64](#), seules les machines SBus (Ultra 1, Ultra 2) sont amorçables depuis une disquette.

Vous aurez besoin que le système soit sur le prompt du moniteur de ROM, qui ressemble habituellement à "ok ". Si vous utilisez un clavier SUN, tapez et pressez en même temps "STOP" et "A". Si vous utilisez une console série, un BREAK devrez vous retourner le prompt du moniteur.

Utilisez ensuite la commande suivante pour démarrer :

```
ok boot floppy
```

Normalement, vous pouvez démarrer du lecteur de CDROM d'un système Sun à partir de l'invite de démarrage en tapant 'boot cdrom' :

```
ok boot cdrom
```

4.5 - Installer OpenBSD simplement

Le nouvel installateur d'OpenBSD a été conçu pour installer et configurer OpenBSD dans une configuration utilisable par défaut avec très peu d'intervention de la part de l'utilisateur. En fait, vous pouvez juste taper *ENTER* un certain nombre de fois pour obtenir une bonne installation d'OpenBSD, pour déplacer vos mains sur le reste du clavier pour uniquement entrer le mot de passe root.

L'installateur créera un plan de partitionnement basé sur la taille de votre disque dur. Même si cela n'est PAS un modèle parfait pour tout le monde, cela fournit un bon point de départ et une bonne stratégie d'ensemble pour définir vos besoins.

Nous allons démarrer avec une installation très simple, avec quelques commentaires sur les options disponibles, et en utilisant la magie des liens hypertextes, cela vous permettra d'en lire plus sur certains sujets qui vous intéressent et d'explorer les options.

Les notes d'installation pour chaque plateforme sont sur les CDs d'installation et les serveurs FTP dans le fichier `INSTALL.<plat>`, ou `<plat>` est votre [plateforme](#), par exemple `i386`.

4.5.1 - Commencer l'installation

Quelle que soit votre méthode de démarrage, il est temps de l'utiliser. Pendant la procédure de démarrage, le noyau et tous les programmes utilisés pour installer OpenBSD sont chargés en mémoire. Après que le noyau d'installation est démarré, le média de démarrage n'est plus nécessaire, tout fonctionne du disque en MEMOIRE. Vous pouvez à ce moment retirer le CD ou la disquette dont vous avez démarré, car vous n'avez plus besoin du CD pour des fichiers d'installation.

A presque tout moment de la procédure d'installation OpenBSD, vous pouvez arrêter la procédure en frappant CTRL-C et pourrez la relancer sans avoir à redémarrer en lançant `install` à l'invite shell. Vous pouvez aussi taper un "!" dans la plupart des cas lors de l'installation pour obtenir l'invite shell, et quitter le shell pour retourner à l'installation.

Si votre démarrage réussi, vous verrez beaucoup de messages défile. Ces textes, dans beaucoup d'architectures en blanc sur bleu, représentent le [dmesg](#), du noyau qui cite les matériels trouvés et comment ils sont connectés aux autres périphériques. Une copie de ce message est conservée dans `/var/run/dmesg.boot`.

Ensuite, vous verrez ce qui suit :

```
...
root on rd0a swap on rd0b dump on rd0b
erase ^?, werase ^W, kill ^U, intr ^C, status ^T

Welcome to the OpenBSD/i386 4.7 installation program.
(I)nstall, (U)pgrade or (S)hell? i
```

Avec cela, nous atteignons notre première question. Vous avez ces trois options affichées :

- **Install** : Charge OpenBSD dans le système, remplaçant un éventuel autre système se trouvant là. Notez qu'il est possible de laisser certaines partitions intactes dans cette procédure, comme `/home`, mais dans les autres cas, considérez que tout le reste sera écrasé.
- **Upgrade** : Installer un nouveau paquetage de [fichiers d'installation](#) sur cette machine, mais ne remplacer aucune information de configuration, donnée utilisateur ou programme additionnel. Aucun formatage de disque n'est réalisé, ni le dossier `/etc` ni `/var` ne sont remplacés. Quelques remarques importantes :
 - Vous n'aurez pas la possibilité d'installer le paquetage `etc47.tgz`. Après l'installation, vous devrez [manuellement greffer](#) `etc47.tgz` sur votre système avant qu'il puisse être complètement fonctionnel. C'est une étape importante qui doit être réalisée, car dans le cas contraire certains services clés (tel que [pf\(4\)](#)) ne démarreront pas.

- o La procédure de mise à jour n'est pas faite pour sauter des "releases" ! Bien que ce saut fonctionnera souvent, il n'est pas supporté. Pour OpenBSD 4.7, mettre à jour de la version 4.6 à la version 4.7 est la seule mise à jour supportée. Si vous mettez à jour depuis une version plus vieille, il est fortement recommandé de mettre à jour en versions intermédiaires d'abord. Cependant, si votre version est très ancienne, une réinstallation complète est recommandée.

Plus d'informations concernant la mise à jour d'une version à une autre se trouvent dans le [Guide de mise à niveau d'OpenBSD 4.7](#).

- **Shell** : Parfois, vous devez réaliser des réparations ou de la maintenance sur un système qui ne pourra pas (ou ne devra pas) démarrer un noyau normal. Cette option vous permettra cette maintenance sur le système. Un certain nombre d'utilitaires importants sont présents sur le media de démarrage.

Nous considérons que vous avez choisi "(I)nstall" ici.

4.5.2 - Questions lors de l'installation

Maintenant nous allons voir les questions qui vont définir comment votre système va être configuré. Vous pouvez remarquer que dans la plupart des cas, toutes les questions sont d'abord posées, puis l'installation se déroule. Si vous possédez un ordinateur lent ou une connexion Internet lente, vous pouvez répondre à ces questions, sortir, revenir plus tard et seulement redémarrer votre ordinateur pour finaliser l'installation.

```
At any prompt except password prompts you can escape to a shell by
typing '!'. Default answers are shown in []'s and are selected by
pressing RETURN. You can exit this program at any time by pressing
Control-C, but this can leave your system in an inconsistent state.
```

```
Terminal type: [vt220] Enter
```

Dans la plupart des cas, le type de terminal par défaut est approprié ; cependant si vous utilisez une [console série](#) pour l'installation, ne prenez pas l'option par défaut, répondez de manière appropriée.

```
System hostname? (short form, e.g. 'foo') puffy
```

Cette valeur, de même que le nom de domaine DNS (spécifié [plus loin](#)), sera sauvegardé dans le fichier `/etc/myname`, qui sera utilisé durant le démarrage normal pour configurer le nom d'hôte du système. Si vous ne configurez pas le nom de domaine du système, la valeur par défaut 'my.domain' sera utilisée.

```
Available network interfaces are: fxp0 vlan0.
Which one do you wish to configure? (or 'done') [fxp0] Enter
```

`vlan0` est l'interface virtuelle VLAN. Pour nos besoins ici, nous allons ignorer cette option et rester avec les interfaces physiques. Si vous avez plusieurs interfaces physiques, elles seront affichées ici. Il faut remarquer qu'elles sont identifiées par le nom de leur pilote, pas par un périphérique générique Ethernet. Dans notre cas, "fxp0" fait référence au premier périphérique utilisant le pilote [fxp\(4\)](#), `fxp1` sera le second périphérique, etc... Plus sur le nommage des périphériques dans la [FAQ 6](#).

Après avoir sélectionné le périphérique que vous voulez configurer, vous allez maintenant le configurer. Dans la plupart des cas, vous voudrez le configurer en utilisant DHCP :

```
IPv4 address for fxp0? (or 'dhcp' or 'none') [dhcp] Enter
Issuing hostname-associated DHCP request for fxp0.
DHCPDISCOVER on fxp0 to 255.255.255.255 port 67 interval 1
DHCPOFFER from 192.168.1.250 (08:00:20:94:0b:c8)
DHCPREQUEST on fxp0 to 255.255.255.255 port 67
DHCPACK from 192.168.1.250 (08:00:20:94:0b:c8)
bound to 192.168.1.199 -- renewal in 43200 seconds.
```

DHCP va configurer l'adresse IP, le masque du sous-réseau, la passerelle par défaut, le nom de domaine DNS et les serveurs DNS. Si vous n'utilisez pas DHCP, vous devrez spécifier toutes ces choses manuellement; voir le sujet plus détaillé [plus loin](#).

Si vous avez une configuration IPv6 à faire ou si il y a une autre interface à configurer (ou si vous n'aimez pas comment vous venez de configurer précédemment), vous pouvez le faire maintenant, mais dans notre cas, nous avons terminé :

```
IPv6 address for fxp0? (or 'rtsol' or 'none') [none] Enter
Available network interfaces are: fxp0 vlan0.
Which one do you wish to configure? (or 'done') [done] Enter
Using DNS domainname in.nickh.org
Using DNS nameservers at 192.168.1.252
Do you want to do any manual network configuration? [no] Enter
```

Si vous avez répondu "yes" à la question "manual network configuration", vous vous retrouverez devant une invite shell, ou vous pourrez configurer tout ce que nécessite votre configuration, puis tapez "exit" pour retourner au programme d'installation.

```
Password for root account? (will not echo) PaSsWoRd
Password for root account? (again) PaSsWoRd
```

Utilisez un mot de passe fort pour le compte root, rappelez-vous : sur Internet, ils SONT là pour pénétrer dans votre ordinateur, ils essayeront des mots de passe que la plupart des gens trouvent intelligents.

Nous vous laisserons plus tard une chance de créer un compte administrateur et de désactiver l'accès distant (SSH) sur le compte root, mais vous avez quand même besoin d'un bon mot de passe pour le compte root.

```
Start sshd(8) by default? [yes] Enter
```

Habituellement, vous voudrez que sshd(8) fonctionne. Si votre application n'a pas besoin de sshd(8), il y a un petit avantage de sécurité de ne pas le faire fonctionner.

```
Start ntpd(8) by default? [no] y
NTP server? (hostname or 'default') [default] Enter
```

Vous avez ici la possibilité de faire fonctionner [OpenNTPD](#), l'implémentation OpenBSD de NTP. OpenNTPD a très peu d'impact et permet d'avoir votre horloge d'ordinateur synchronisée avec précision. La configuration par défaut, utilisant [pool.ntp.org](#), utilise un grand nombre de serveurs de temps ouverts dans le monde.

Une raison où vous NE voudrez pas faire fonctionner ntpd(8) et si vous utilisez un système [dual-boot](#) avec un autre OS qui n'utilise pas une horloge matérielle configurée en GMT, et vous ne voulez pas qu'OpenBSD modifie le temps pour votre autre OS.

```
Do you expect to run the X Window System? [yes] Enter
```

Toutes les plateformes ne vous poseront pas la question si vous voulez utiliser X, celles qui ne demandent pas que sysctl configure l'utilisation de X. Répondez "y" ici modifiera `/etc/sysctl.conf` pour ajouter la ligne `machdep.allowaperture=1` ou `machdep.allowaperture=2`, cela dépend de votre plateforme.

Si vous n'avez pas l'intention d'utiliser X sur votre système ou vous n'êtes pas sûr, répondez 'N' ici, vous pourrez facilement le modifier en éditant `/etc/sysctl.conf` et redémarrant, en fonction de vos besoins futurs. Il y a un avantage de sécurité possible en laissant le pilote d'ouverture [xf86\(4\)](#) désactivé, car le moteur graphique sur les cartes vidéos modernes peut être utilisé potentiellement pour modifier la mémoire au delà du contrôle du processeur. Il faut remarquer que les applications non-graphiques qui nécessitent les bibliothèques et utilitaires X pour fonctionner n'ont PAS besoin de ce sysctl configuré.

```
Do you want the X Window System to be started by xdm(1)? [no] y
```

[xdm\(1\)](#) démarre l'environnement X au démarrage du système. Nous vous recommandons de faire cela à l'installation seulement si vous êtes très confiant que X fonctionnera sur votre système par défaut. Sinon, [configurez X](#) avant d'activer xdm(1).

```
Change the default console to com0? [no] Enter
```

Si vous désirez configurer une console série plutôt que votre système par défaut (habituellement un clavier et écran), c'est votre chance. Si vous modifiez le défaut par "y", on vous demandera de configurer la vitesse. Il faut remarquer que pour les consoles séries, plus vite n'est pas toujours le meilleur choix, utiliser le paramètre par défaut de votre plateforme et souvent hautement recommandé.

```
Setup a user? (enter a lower-case loginname, or 'no') [no] Enter
```

Vous avez ici l'opportunité de créer un utilisateur AUTRE que root pour la maintenance du système. Cette utilisateur sera membre du groupe "wheel" donc vous pourrez utiliser [su\(1\)](#) et on vous demandera un mot de passe.

Il faut remarquer que si vous voulez créer l'utilisateur, entrez le nom de l'utilisateur, pas "y" ou "yes".

```
What timezone are you in? ('?' for list) [Canada/Mountain] US/Michigan
```

OpenBSD considère que l'horloge temps réel de votre ordinateur est configurée en GMT, mais vous devez aussi spécifier dans quel fuseau horaire vous êtes. Il peut y avoir plusieurs réponses valides pour votre localisation physique. En tapant "?" à l'invite vous serez guidé pour trouver un nom de fuseau horaire valide.

Il faut remarquer que l'installateur devinera souvent correctement votre fuseau horaire, et vous n'aurez juste qu'à taper "Enter".

Plus d'informations sur la configuration du fuseau horaire [ici](#).

4.5.3 - Configurer les disques

Remarque importante : Les utilisateurs avec de gros disques durs (plus gros que ce qui était disponible normalement quand votre ordinateur a été fabriqué) voudront voir [cette section](#) avant d'aller plus loin.

Configurer votre disque de manière appropriée et probablement la partie la plus compliquée de l'installation de OpenBSD.

Configurer les disques sous OpenBSD varie un peu en fonction des plateformes. Pour [i386](#), [amd64](#), [macppc](#), [zaurus](#) et [armish](#), la configuration du disque est réalisée en deux étapes. En premier, la partition OpenBSD sur le disque dur est définie en utilisant fdisk(8), puis cette partition est subdivisée en partitions OpenBSD en utilisant disklabel(8).

Certains utilisateurs seront un peu confus avec la terminologie utilisée ici. Il apparaît que nous utilisons le mot "partition" de deux façons différentes. Cette observation est correcte. Il y a deux niveaux de partitionnement dans les plateformes OpenBSD suivantes, le premier, celui qui peut-être considéré comme le partitionnement du système d'exploitation, qui est la façon dont plusieurs OSs sur un seul ordinateur marquent leur propre espace sur le disque, et le second qui est celui dont la partition OpenBSD sous-partitionne dans plusieurs systèmes de fichiers individuels. La premier niveau est visible comme une partition disque au DOS, Windows et tous les autres OS qui utilisent ce schéma de disque système, le second niveau de partitionnement est visible uniquement par OpenBSD et d'autres OSs qui peuvent directement lire un système de fichiers OpenBSD.

Le nouvel installateur d'OpenBSD essaye de vous rendre la tâche plus simple pour configurer votre disque en ayant une configuration par défaut saine pour une utilisation "standard". Il faut remarquer que beaucoup de personnes voudront continuer à personnaliser la configuration par défaut, ou utiliser leur propre configuration disque, mais les nouveaux utilisateurs commenceront sûrement avec cette configuration tant qu'ils ne verront pas qu'ils ont des besoins différents. Il faut remarquer que la configuration par défaut variera en fonction de la taille du disque système.

Pour maintenant, nous allons prendre la configuration par défaut de notre disque 40G.

```

Available disks are: wd0.
Which one is the root disk? (or 'done') [wd0]
Disk: wd0          geometry: 4863/255/63 [78125000 Sectors]
Offset: 0          Signature: 0xAA55

#  id      Starting          Ending          LBA Info:
#  id      C  H  S -          C  H  S [          start:          size ]
-----
0: 06      0  1  1 -          521 254 63 [          63:          8385867 ] DOS > 32MB
1: 00      0  0  0 -          0  0  0 [          0:           0 ] unused
2: 00      0  0  0 -          0  0  0 [          0:           0 ] unused
3: 00      0  0  0 -          0  0  0 [          0:           0 ] unused
Use (W)hole disk or (E)dit the MBR? [whole] Enter
Setting OpenBSD MBR partition to whole wd0...done.

```

Il faut remarquer que ce disque possède une partition pré-existante -- *en utilisant "whole" cela la supprimera du disque !*.

Configurer le disque "whole" pour OpenBSD fait plusieurs choses importantes :

- Supprimer toutes les partitions existantes sur le disque
- Créer un MBR et une signature disque pour rendre le disque démarrable
- Créer une partition OpenBSD en utilisant tout le disque
- Configurer la partition comme "active".

Il y a plusieurs raisons pour lesquelles vous ne voudrez pas faire cela, incluant :

- Vous désirez garder d'autres partitions d'OS
- Vous désirez conserver "setup", "suspend to disk", ou d'autres partitions systèmes
- Vous désirez avoir un système [multi-boot](#)

Il faut remarquer qu'il est critique qu'un nouveau (ou jamais utilisé pour démarrer) disque possède un MBR valide, une signature valide, une partition OpenBSD et une partition marquée comme "active". Si vous ne faites pas ces choses en utilisant l'option "Use whole disk", vous devrez être sûr que c'est réalisé manuellement.

Plus d'informations sur le partitionnement avec fdisk de votre disque [plus loin](#).

Maintenant nous allons casser notre partition fdisk OpenBSD en partitions disque OpenBSD en utilisant [disklabel\(8\)](#) :

```

Setting OpenBSD MBR partition to whole wd0...done.
The auto-allocated layout for wd0 is:
#          size          offset  fstype [fsize bsize  cpg]
a:         1024.0M          63  4.2BSD  2048 16384   1 # /
b:          507.1M       2097215   swap
c:        38147.0M           0  unused
d:         2620.4M       3135791  4.2BSD  2048 16384   1 # /tmp
e:         4143.1M       8502287  4.2BSD  2048 16384   1 # /var
f:         2048.0M       16987323  4.2BSD  2048 16384   1 # /usr
g:         1024.0M       21181627  4.2BSD  2048 16384   1 # /usr/X11R6
h:         3610.7M       23278779  4.2BSD  2048 16384   1 # /usr/local
i:         1961.6M       30673543  4.2BSD  2048 16384   1 # /usr/src

```



```

j:          1961.6M          34690971  4.2BSD   2048 16384    1 # /usr/obj
k:          19245.9M         38708399  4.2BSD   2048 16384    1 # /home
Use (A)uto layout, (E)dit auto layout, or create (C)ustom layout? [a] Enter

```

L'installateur nous présente ce qu'il propose comme "Auto layout" pour les partitions OpenBSD sur notre disque, ce que nous allons accepter.

Si la configuration proposée n'est pas appropriée à vos besoins, vous pouvez, bien sur, l'éditer ou la personnaliser complètement, plus de détails sur le partitionnement disklabel [plus loin](#).

REMARQUE pour les re-installations : Le nouvel installateur ne supprimera pas votre ancien disklabel si vous choisissez "(C)ustom Layout", mais vous devrez re-spécifier chaque point de montage en utilisant l'option 'm' de disklabel(8).

Le nouvel installateur crée maintenant ces partitions et crée les systèmes de fichiers dessus en utilisant [newfs\(8\)](#), et les montent pour installation :

```

/dev/rwd0a: 1024.0MB in 2097152 sectors of 512 bytes
6 cylinder groups of 202.47MB, 12958 blocks, 25984 inodes each
/dev/rwd0k: 19245.9MB in 39415696 sectors of 512 bytes
96 cylinder groups of 202.47MB, 12958 blocks, 25984 inodes each
/dev/rwd0d: 2620.4MB in 5366496 sectors of 512 bytes
13 cylinder groups of 202.47MB, 12958 blocks, 25984 inodes each
/dev/rwd0f: 2048.0MB in 4194304 sectors of 512 bytes
11 cylinder groups of 202.47MB, 12958 blocks, 25984 inodes each
/dev/rwd0g: 1024.0MB in 2097152 sectors of 512 bytes
6 cylinder groups of 202.47MB, 12958 blocks, 25984 inodes each
/dev/rwd0h: 3610.7MB in 7394764 sectors of 512 bytes
18 cylinder groups of 202.47MB, 12958 blocks, 25984 inodes each
/dev/rwd0j: 1961.6MB in 4017428 sectors of 512 bytes
10 cylinder groups of 202.47MB, 12958 blocks, 25984 inodes each
/dev/rwd0i: 1961.6MB in 4017428 sectors of 512 bytes
10 cylinder groups of 202.47MB, 12958 blocks, 25984 inodes each
/dev/rwd0e: 4143.1MB in 8485036 sectors of 512 bytes
21 cylinder groups of 202.47MB, 12958 blocks, 25984 inodes each
/dev/wd0a on /mnt type ffs (rw, asynchronous, local)
/dev/wd0k on /mnt/home type ffs (rw, asynchronous, local, nodev, nosuid)
/dev/wd0d on /mnt/tmp type ffs (rw, asynchronous, local, nodev, nosuid)
/dev/wd0f on /mnt/usr type ffs (rw, asynchronous, local, nodev)
/dev/wd0g on /mnt/usr/X11R6 type ffs (rw, asynchronous, local, nodev)
/dev/wd0h on /mnt/usr/local type ffs (rw, asynchronous, local, nodev)
/dev/wd0j on /mnt/usr/obj type ffs (rw, asynchronous, local, nodev, nosuid)
/dev/wd0i on /mnt/usr/src type ffs (rw, asynchronous, local, nodev, nosuid)
/dev/wd0e on /mnt/var type ffs (rw, asynchronous, local, nodev, nosuid)

```

Vous remarquerez qu'il y a une partition *c* qui semble avoir été ignoré. Cette partition est l'ensemble de votre disque dur; ne tentez pas de la modifier.

4.5.4 - Choisir le média d'installation et les sets de fichiers

Ensuite, vous allez avoir une chance de choisir votre média d'installation. Dans notre cas, nous allons installer d'un serveur FTP.

```
Location of sets? (cd disk ftp http or 'done') [cd] ftp
HTTP/FTP proxy URL? (e.g. 'http://proxy:8080', or 'none') [none]
Enter
Server? (hostname, list#, 'done' or '?') [mirror.example.org] obsd.cec.mtu.edu
```

Si vous ne vous souvenez pas de la localisation de votre miroir préféré (ou plus !), l'installateur sera souvent capable de vous fournir un miroir par défaut qui sera adapté à votre situation. Sinon, tapez "?" pour l'affichage d'une liste des miroirs, et sélectionner le numéro du miroir qui vous convient.

```
Server directory? [pub/OpenBSD/4.7/i386] Enter
Login? [anonymous] Enter
```

Les miroirs FTP publiques supportent tous le téléchargement anonyme, bien évidemment, mais vous pouvez avoir un serveur local qui nécessite un login et un mot de passe.

Vous pouvez maintenant ajuster la liste des [sets de fichiers](#).

```
Select sets by entering a set name, a file name pattern or 'all'. De-select
sets by prepending a '-' to the set name, file name pattern or 'all'. Selected
sets are labelled '[X]'.
  [X] bsd             [X] etc47.tgz       [X] game47.tgz       [X] xfont47.tgz
  [X] bsd.rd         [X] misc47.tgz      [X] xbase47.tgz      [X] xserv47.tgz
  [ ] bsd.mp         [X] comp47.tgz      [X] xetc47.tgz
  [X] base47.tgz     [X] man47.tgz       [X] xshare47.tgz
Set name(s)? (or 'abort' or 'done') [done] Enter
```

Au minimum, vous avez besoin d'un noyau (bsd), le set de fichiers base47.tgz et etc47.tgz. A moins que vous ne sachiez ce que vous faites, gardez les sets par défaut. Vous pouvez ajouter ou supprimer les sets de fichiers en utilisant les caractères "+" et "-" devant le nom du set, et aussi utiliser les wildcards :

- `-comp47.tgz` supprime `comp47.tgz`
- `+bsd.mp` ajoute `bsd.mp`
- `-x*` supprime tous les composants `X`

Mais encore un fois, on va prendre par défaut. Cette machine est un système mono-processeur, donc `bsd.mp` n'est pas installé, mais tout le reste l'est. Si il devait être mise à jour plus tard pour un système multi-processeurs, vous préférerez installer `bsd.mp` bien évidemment.

Et maintenant nous démarrons notre installation ! C'est à ce moment la ou vous pouvez revenir plus tard si vous avez un ordinateur ou une connexion Internet lente, pensez qu'avoir un ordinateur rapide et les fichiers en local, ce processus ne prend que quelques minutes ou moins !

```

bsd          100% | ***** | 7404 KB 00:04
bsd.rd       100% | ***** | 6147 KB 00:03
base47.tgz   100% | ***** | 49365 KB 01:46
etc47.tgz    100% | ***** | 510 KB 00:01
misc47.tgz   100% | ***** | 356 KB 00:06
comp47.tgz   100% | ***** | 90468 KB 03:08
man47.tgz    100% | ***** | 9240 KB 00:21
game47.tgz   100% | ***** | 2556 KB 00:04
xbase47.tgz  100% | ***** | 15185 KB 00:21
xetc47.tgz   100% | ***** | 70053   00:00
xshare47.tgz 100% | ***** | 2887 KB 00:12
xfont47.tgz  100% | ***** | 38754 KB 00:57
xserv47.tgz  100% | ***** | 19413 KB 00:39
Location of sets? (cd disk ftp http or 'done') [done] Enter

```

Une fois de plus, nous avons choisi par défaut, nous en avons fini avec l'installation des fichiers.

```

Saving configuration files...done.
Generating initial host.random file...done.
Making all device nodes...done.

```

```

CONGRATULATIONS! Your OpenBSD install has been successfully completed!
To boot the new system, enter 'reboot' at the command prompt.
When you login to your new system the first time, please read your mail
using the 'mail' command.

```

```
#
```

4.5.5 - Premier démarrage !

OpenBSD est maintenant installé sur votre système et prêt pour son premier démarrage, mais avant il faut...

Avant de redémarrer

A cet instant, votre système est installé et prêt à être redémarré et configuré pour le service. Avant cela, cependant, il serait prudent de regarder la [page Errata](#) pour voir si il y a des bogues qui vous impacterez dans l'immédiat.

Une astuce que vous pouvez utiliser pour certaines configurations "avant le premier démarrage" est d'exécuter :

```
# /mnt/usr/sbin/chroot /mnt
```

à l'invite du shell. Cela configurera vos points de montage à ce qu'ils seraient sur un redémarrage normal de votre nouveau système installé. Vous pouvez maintenant faire quelques configurations basiques du système, comme ajouter des utilisateurs, changer les points de montage, etc...

Après le redémarrage

Au premier démarrage, les clés [SSH](#) seront générées. Sur un ordinateur moderne, cela ne prendra que quelques secondes, vous ne le remarquerez même pas. Sur des systèmes plus anciens, cela peut prendre quelques minutes, même potentiellement une heure ou plus sur des systèmes très lents.

Une des premières chose à lire après avoir installé votre système est [afterboot\(8\)](#).

Vous trouvez aussi utile les liens suivants :

- [Ajoutez des utilisateurs dans OpenBSD](#)
- [Configuration réseau initiale](#)
- [Pages Man des commandes utiles/populaires](#)
- [Pages man OpenBSD sur le Web](#)
- [Le système de paquetages et ports OpenBSD pour installer un logiciel](#)

4.5.6 - Une dernière chose...

Les développeurs OpenBSD vous demande [d'envoyer une copie de votre dmesg](#). C'est très apprécié par les développeurs, et finalement, tous les utilisateurs.

4.6 - Détails pour une installation plus complexe

Parfois vous ne pouvez pas juste utiliser la configuration par défaut. Vous trouverez ici plus de détails sur certaines parties du processus d'installation.

4.6.1 - Configurer le réseau

Si vous n'avez pas de serveur DHCP disponible, vous devrez configurer votre adaptateur(s) réseau manuellement. Voici un exemple :

```
Which one do you wish to configure? (or 'done') [xl0] Enter
IPv4 address for xl0? (or 'dhcp' or 'none') [dhcp] 192.168.1.37
Netmask? [255.255.255.0] 255.255.254.0
IPv6 address for xl0? (or 'rtsol' or 'none') [none] Enter
```

Après cet ensemble de questions, vous aurez la possibilité de configurer d'autres adaptateurs réseaux si cette machine en possède. Si vous spécifiez un autre adaptateur réseau ici, les questions précédentes se répéteront.

```
Available network interfaces are: xl0 vlan0.
Which one do you wish to configure? (or 'done') [done]
```

Maintenant vous allez configurer la passerelle par défaut et les serveurs DNS, choses qui impactent tous les adaptateurs réseaux :

```

Default IPv4 route? (IPv4 address, 'dhcp' or 'none') 192.168.1.1
add net default: gateway 192.168.1.1
DNS domain name? (e.g. 'bar.com') [my.domain] example.org
DNS nameservers? (IP address list or 'none') [none] 192.168.1.250 192.168.1.251

```

Il faut remarquer que plusieurs serveurs DNS peuvent être indiqués, séparés par des espaces.

Quelquefois, vous devez faire des choses en plus, par exemple configurer une clé d'accès sans-fil ou configurer en dur une vitesse ou un duplex (ne faite cela que si c'est ABSOLUMENT nécessaire, corriger la configuration de votre commutateur est une bien meilleure idée !).

```

Do you want to do any manual network configuration? [no] y
Type 'exit' to return to install.
# ifconfig xl0 media
xl0: flags=8843<UP,BROADCAST,RUNNING,SIMPLEX,MULTICAST> mtu 1500
    lladdr 00:08:74:2c:df:9c
    groups: egress
    media: Ethernet autoselect (100baseTX full-duplex)
    status: active
    supported media:
        media 10baseT
        media 10baseT mediaopt full-duplex
        media 100baseTX
        media 100baseTX mediaopt full-duplex
        media autoselect
    inet 192.168.1.37 netmask 0xfffffe00 broadcast 192.168.1.255
# ifconfig xl0 media 100baseTX mediaopt full-duplex
# ifconfig xl0
xl0: flags=8843<UP,BROADCAST,RUNNING,SIMPLEX,MULTICAST> mtu 1500
    lladdr 00:08:74:2c:df:9c
    groups: egress
    media: Ethernet 100baseTX full-duplex
    status: active
    inet6 fe80::208:74ff:fe2c:df9c%xl0 prefixlen 64 scopeid 0x1
    inet 192.168.1.37 netmask 0xfffffe00 broadcast 192.168.1.255
# exit
...setup resumes...

```

[\(revenir là où on aurait pu\)](#)

4.6.2 - Configurer le fuseau horaire

Le temps dans Unix n'est pas une chose simple (ou d'une autre façon, le temps dans Unix est une chose très simple, le temps humain est un désordre politiquement manipulé). Les fichiers de fuseau horaire aident le système à convertir le temps Unix (le nombre de secondes après minuit GMT, 1 Janvier 1970) en temps humain, prenant en compte des choses comme les fuseaux horaires, l'heure avancée (DST), les changements de règles DST, etc... Ils incluent aussi l'historique

des changements.

Plusieurs fichiers de définitions de fuseaux horaires peuvent parfois fournir le même temps courant, mais peuvent avoir des histoires historiques différentes. Par exemple, EST5EDT et US/Michigan possèdent le même temps MAINTENANT, mais en retournant en 1975, les règles étaient différentes, donc si vous faites des calculs avec des dates et le temps en 1975, vous devrez faire attention aux différences. Vous devez utiliser le fichier de fuseau horaire le plus spécifique et se rapprochant le plus de votre région, plutôt que celui qui vous donnera un temps correct à l'instant donné.

L'installateur d'OpenBSD vous aidera à trouver de manière appropriée le fichier de fuseau horaire pour vous si vous n'êtes pas sûr. Tapez simplement "?" à chaque invite, et l'installateur vous affichera les options. Si le premier niveau de réponses ne vous convient pas, sélectionnez un continent ou un pays, et regardez vos options :

```
What timezone are you in? ('?' for list) [right/EST5EDT] ?
Africa/          Chile/          GB-Eire         Israel          NZ-CHAT        UCT
America/         Cuba           GMT             Jamaica        Navajo         US/
Antarctica/     EET           GMT+0          Japan          PRC            UTC
Arctic/         EST           GMT-0          Kwajalein     PST8PDT       Universal
Asia/           EST5EDT       GMT0           Libya          Pacific/       W-SU
Atlantic/       Egypt         Greenwich      MET           Poland         WET
Australia/     Eire          HST            MST           Portugal       Zulu
Brazil/         Etc/          Hongkong       MST7MDT       ROC            posix/
CET             Europe/       Iceland        Mexico/        ROK            posixrules
CST6CDT        Factory       Indian/        Mideast/      Singapore     right/
Canada/         GB            Iran           NZ            Turkey
What timezone are you in? ('?' for list) [right/EST5EDT] US
What sub-timezone of 'US' are you in? ('?' for list) ?
Alaska          Central         Hawaii          Mountain       Samoa
Aleutian        East-Indiana   Indiana-Starke Pacific
Arizona         Eastern        Michigan        Pacific-New
What sub-timezone of 'US' are you in? ('?' for list) Michigan
```

Nous avons maintenant configuré le temps sur "US/Michigan". Cela créera un lien symbolique dans /etc pointant sur le fichier de zoneinfo dans /usr/share/zoneinfo, quelque chose comme cela :

```
/etc/localtime -> /usr/share/zoneinfo/US/Michigan
```

Remarquez le répertoire "right/", ce répertoire inclut des ajustements de bond en seconde, mais duplique aussi le choix de zoneinfo standard. Plus [ici](#).

[\(revenir là où on aurait pu\)](#)

4.6.3 - Personnaliser la configuration via fdisk(8)

Remarque : seulement certaines plateformes utilisent uniquement fdisk, et d'habitude, seulement les utilisateurs de [i386](#) et [amd64](#) devront craindre faire des fantaisies avec fdisk. Les utilisateurs de la plupart des plateformes utilisant fdisk(8) ne seront généralement pas ennuyés avec le multiboot ou la configuration/diagnostic de partitions. Pour cette raison, cette partie est dédié au i386 et amd64.

`fdisk(8)` est utilisé pour délimiter la partie OpenBSD sur votre disque dur. Il aide à délimiter la partie du disque utilisée par OpenBSD des parties utilisées par d'autres OSs ou fonctions systèmes.

Si vous avez une partition sur votre disque que vous voulez garder ou garder de l'espace libre pour une autre partition, vous ne devez pas choisir "(W)hole disk", mais vous devrez éditer la table de partition avec `fdisk(8)`. Plus d'informations sur l'utilisation manuelle de `fdisk(8)` peuvent être trouvées dans [ici](#). **Avant de travailler sur un système qui possède des données que vous ne voulez pas perdre, soyez sur d'avoir une bonne sauvegarde.** Il est très facile dans ce processus de perdre des données importantes, donc soyez sur que vous êtes prêt pour revenir en arrière si c'est nécessaire.

Si vous ajoutez OpenBSD à un système déjà existant, vous devrez surement libérer de l'espace avant d'installer OpenBSD. Cela conduit souvent à supprimer ou réduire la taille de partitions existantes. Le programme `gparted` nous semble utile pour réduire la taille de partitions de la plupart des OSs populaires, permettant d'installer OpenBSD sur l'espace libéré.

Dans cet exemple, nous considérerons que l'on commence avec un disque de 40G vierge et que l'on désire installer un système multi-boot, réservant au début du disque 5G pour Windows et le reste pour OpenBSD. Il faut remarquer qu'un disque vierge doit avoir un MBR valide dans le code de démarrage et une signature avant qu'il ne soit démarré.

Le processus est très similaire avec une partition déjà existante, vous devez juste sauter les parties où l'on crée la partition pour Windows et l'inquiétude sur l'installation du code de démarrage MBR.

```
Available disks are: wd0.
Which one is the root disk? (or 'done') [wd0] Enter
MBR has invalid signature; not showing it.
```

SI le disque possède déjà un MBR valide, il vous montrera la table de partition existante, ce qui est une bonne façon de savoir si le disque possède déjà des données

```
Use (W)hole disk or (E)dit the MBR? [whole] e
```

You will now create a single MBR partition to contain your OpenBSD data. This partition must have an id of 'A6'; must *NOT* overlap other partitions; and must be marked as the only active partition. Inside the `fdisk` command, the 'manual' command describes all the `fdisk` commands in detail.

```
Disk: wd0          geometry: 4998/255/63 [80293248 Sectors]
Offset: 0         Signature: 0x0
```

#	id	Starting			Ending			LBA Info:	start:	size]]]
		C	H	S	-	C	H						
0:	00	0	0	0	-	0	0	0	[0:	0]	unused
1:	00	0	0	0	-	0	0	0	[0:	0]	unused
2:	00	0	0	0	-	0	0	0	[0:	0]	unused
3:	00	0	0	0	-	0	0	0	[0:	0]	unused

```
Enter 'help' for information
fdisk: 1>
```

En premier lieu notez l'invite `fdisk`. Le chiffre "1" indique le premier niveau des tables de partition -- si vous éditez une partition étendue, cela sera "2" (ou plus). Les partitions étendues sont des partitions qui possèdent leur propre table de sous-partition, permettant de contourner la limite de quatre partitions sur IBM AT. Les partitions étendues ne seront pas traitées ici.

En premier, nous allons faire de la partition "0" une partition de 5G Windows (utilisant NTFS), et la partition "1" sera notre partition OpenBSD utilisant le reste du disque.

```
fdisk: 1> e 0
Starting           Ending           LBA Info:
#: id            C   H   S -       C   H   S [      start:           size ]
-----
0: 00            0   0   0 -       0   0   0 [           0:           0 ] unused
Partition id ('0' to disable) [0 - FF]: [0] (? for help)
```

Comme nous ne savons pas de mémoire l'ID pour une partition NTFS, nous tapons "?" ici pour obtenir la liste.

```
Partition id ('0' to disable) [0 - FF]: [0] (? for help) ?
Choose from the following Partition id values:
00 unused           20 Willowsoft       66 NetWare 386      A9 NetBSD
01 DOS FAT-12       24 NEC DOS           67 Novell           AB MacOS X boot
02 XENIX /           27 Win Recovery     68 Novell           AF MacOS X HFS+
03 XENIX /usr       38 Theos            69 Novell           B7 BSDI filesy*
04 DOS FAT-16       39 Plan 9            70 DiskSecure      B8 BSDI swap
05 Extended DOS     40 VENIX 286        75 PCIX            BF Solaris
06 DOS > 32MB       41 Lin/Minux DR     80 Minix (old)     C0 CTOS
07 NTFS             42 LinuxSwap DR    81 Minix (new)    C1 DRDOSs FAT12
08 AIX fs           43 Linux DR         82 Linux swap      C4 DRDOSs < 32M
09 AIX/Coherent     4D QNX 4.2 Pri     83 Linux files*   C6 DRDOSs >=32M
0A OS/2 Bootmgr     4E QNX 4.2 Sec     84 OS/2 hidden    C7 HPFS Disabled
0B Win95 FAT-32     4F QNX 4.2 Ter     85 Linux ext.     DB CPM/C.DOS/C*
0C Win95 FAT32L     50 DM              86 NT FAT VS      DE Dell Maint
0E DOS FAT-16       51 DM              87 NTFS VS        E1 SpeedStor
0F Extended LBA     52 CP/M or SysV   8E Linux LVM      E3 SpeedStor
10 OPUS             53 DM              93 Amoeba FS      E4 SpeedStor
11 OS/2 hidden      54 Ontrack         94 Amoeba BBT     EB BeOS/i386
12 Compaq Diag.     55 EZ-Drive        99 Mylex          EE EFI GPT
14 OS/2 hidden      56 Golden Bow     9F BSDI           EF EFI Sys
16 OS/2 hidden      5C Priam          A0 NotebookSave  F1 SpeedStor
17 OS/2 hidden      61 SpeedStor      A5 FreeBSD       F2 DOS 3.3+ Sec
18 AST swap         63 ISC, HURD, *   A6 OpenBSD       F4 SpeedStor
19 Willowtech       64 NetWare 2.xx   A7 NEXTSTEP      FF Xenix BBT
1C ThinkPad Rec     65 NetWare 3.xx   A8 MacOS X
Partition id ('0' to disable) [0 - FF]: [0] (? for help) 07
```

Maintenant on définit ces points de début et fin :


```
Do you wish to edit in CHS mode? [n]
```

Le mode CHS vous permet de spécifier la taille du disque en cylindres, têtes et secteurs. Gardez en mémoire que sur les disques modernes, les numéros CHS sont complètement faux, juste trois nombres qui traduisent en secteurs sur le disque, qui est traduit sur la géométrie physique de votre disque (qui varie sûrement à travers le disque de toute façon).

Si vous répondez "y" ici, on vous demandera le début et la fin du cylindre, tête secteur. Si vous répondez "no" ici (comme nous allons le faire), on vous demandera le secteur de démarrage et la taille. Editer par CHS est quelquefois plus simple pour travailler sur une partition existante, le secteur de démarrage et la taille sont souvent plus simple quand vous voulez créer rapidement une partition d'une certaine taille.

```
offset: [0] 63
```

Les plateformes fdisk ont besoin que la première partition commence après la première piste du disque. Une piste est un nombre de secteurs "secteur par piste", le dernier des nombres de "geometry:". Il est facile de prendre l'habitude de taper "63" ici, comme c'est courant sur les disques IDE et SATA modernes, mais tous les disques n'utilisent pas cette taille de piste, les exceptions courantes modernes sont les disques SCSI et Flash.

```
size: [0] 5g
Rounding to nearest cylinder: 10490382
```

La valeur de "Size" peut être le nombre de secteurs (512 octets chacuns), ou la capacité désirée suivie par un "k", "m" ou "g". Quand on édite en utilisant la taille et l'offset, fdisk arrondira votre partition pour que cela se termine sur la limite d'un cylindre (OpenBSD ne se soucie pas de cela, et il est possible qu'aucun système d'exploitation moderne ne se soucie à ce sujet, mais certains peuvent en avoir à un moment).

Maintenant, regardons notre nouvelle partition :

```
fdisk:*1> p
Disk: wd0      geometry: 4998/255/63 [80293248 Sectors]
Offset: 0     Signature: 0x0

#  id      Starting      Ending      LBA Info:
#  id      C   H   S -         C   H   S [ start:      size ]
-----
0: 07      0   1   1 -         652 254 63 [ 63:      10490382 ] NTFS
1: 00      0   0   0 -         0   0   0 [ 0:        0 ] unused
2: 00      0   0   0 -         0   0   0 [ 0:        0 ] unused
3: 00      0   0   0 -         0   0   0 [ 0:        0 ] unused
fdisk:*1>
```

Remarquez que l'invite inclut maintenant un "", cela signifie qu'il y a des modifications non sauvegardées.*

Nous avons maintenant créer notre partition Windows. Remarquez que cette partition réserve juste de l'espace sur le

disque, elle n'est pas formatée; aucun système de fichiers n'existe ici. Vous vous inquiétez de cela quand vous installerez Windows; nous avons accompli notre but en réservant de l'espace pour qu'une partition Windows soit créée plus tard.

Maintenant nous créons notre partition OpenBSD. Dans notre cas, l'ID de la partition sera "A6".

```
fdisk:*1> e 1
Starting           Ending           LBA Info:
#: id             C  H  S -           C  H  S [         start:           size ]
-----
1: 00             0  0  0 -           0  0  0 [           0:           0 ] unused
Partition id ('0' to disable) [0 - FF]: [0] (? for help) a6
Do you wish to edit in CHS mode? [n] Enter
offset: [0]
```

uh-oh ! Quel est notre offset ? Simple -- l'offset de la partition précédente plus la taille de la partition, dans notre cas, $63 + 10490382 = 10490445$.

```
offset: [0] 10490445
size: [0] *
fdisk:*1>
```

Remarquez qu'ici nous avons tapé "*" comme taille, signifiant "ce qui reste du disque". Encore un fois, nous aurions pu entrer la taille en secteurs, "m" ou "g" si on voulait laisser de l'espace quelque part.

Maintenant nous allons voir notre table de partition :

```
fdisk:*1> p
Disk: wd0           geometry: 4998/255/63 [80293248 Sectors]
Offset: 0           Signature: 0x0
Starting           Ending           LBA Info:
#: id             C  H  S -           C  H  S [         start:           size ]
-----
0: 07             0  1  1 -           652 254 63 [           63:           10490382 ] NTFS
1: A6            653  0  1 -           4998  5 63 [          10490445:           69802803 ] OpenBSD
2: 00             0  0  0 -           0  0  0 [           0:           0 ] unused
3: 00             0  0  0 -           0  0  0 [           0:           0 ] unused
fdisk:*1>
```

CE N'EST PAS ENCORE TERMINE !

Ce disque n'est toujours pas démarrable ! Comme c'est un nouveau disque, la [MBR](#) du disque est complètement vide. Le message "Signature: 0x0" affiché ici montre qu'il n'y a pas de signature valide (0xAA55), ce qui indique définitivement un code de démarrage non valide. Bien sûr, vous pouvez avoir une signature valide sans code de démarrage valide, à cause d'un manque de chance aléatoire ou de dommage sur le code de démarrage existant, mais une signature invalide indique bien le plus souvent un manque de code de démarrage, nous allons donc l'installer maintenant en utilisant la commande "update" :

```
fdisk:*1> update
Machine code updated.
fdisk:*1>
```

Nous devons aussi "marquer" la partition comme "active" pour que la ROM de démarrage sache de quelle partition démarrer :

```
fdisk:*1> f 1
Partition 1 marked active.
```

Maintenant, regardons à quoi cela ressemble :

```
fdisk:*1> p
Disk: wd0          geometry: 4998/255/63 [80293248 Sectors]
Offset: 0         Signature: 0xAA55
```

#:	id	Starting			Ending			LBA Info:		size	
		C	H	S	C	H	S	start:			
0:	07	0	1	1	652	254	63	63:	10490382] NTFS	
*1:	A6	653	0	1	4998	5	63	10490445:	69802803] OpenBSD	
2:	00	0	0	0	0	0	0	0:	0] unused	
3:	00	0	0	0	0	0	0	0:	0] unused	

```
fdisk:*1>
```

Une checklist des choses que vous voudrez surement faire avant de sortir de `fdisk(8)` :

- Signature valide ?
- Partitions qui ne se recouvrent pas ?
- Partition OpenBSD avec un id "A6" ?
- Bonne partition (probablement OpenBSD) marquée active?

[\(Revenir là où on aurait pu\)](#)

4.6.4 - Personnaliser la configuration via `disklabel`

Dans une partition OpenBSD `fdisk(8)`, nous utilisons [`disklabel\(8\)`](#) pour créer les partitions de système de fichiers OpenBSD. OpenBSD marque ces partitions de système de fichiers en utilisant seize lettres, "a" à "p". La partition "a" sur le disque de démarrage est définie comme partition racine, "b" est la partition de pagination par défaut. "c" sur tous les disques est la partition "whole disk", elle est utilisée par les programmes qui ont un accès direct au disque physique, comme `fdisk(8)` et `disklabel(8)`. La partition "c" est créée automatiquement pour vous, et ne doit pas être supprimée ou modifiée. Les lettres restantes sont disponibles pour vous pour définir les points de montage. Vous pouvez sauter des lettres, vous pouvez les définir dans n'importe quel ordre, et peuvent être dans n'importe quel ordre sur le disque (bien que pour certaines plateformes la partition "a" doit être à un endroit précis). Vous pouvez aussi laisser des espaces sur le disque qui ne sont pas alloués, et les allouer plus tard, ou potentiellement agrandir une partition existante plus tard dans

cette espace non alloué en utilisant [growfs\(8\)](#).

Toutes les partitions qui possèdent des partitions FFS natives doivent les avoir dans la partition OpenBSD fdisk(8), cependant des partitions [non-OpenBSD](#) peuvent (et normalement doivent) être en dehors de la partition OpenBSD fdisk (8).

Plus d'informations sur l'utilisation de disklabel peuvent être trouvées [ici](#).

Plus d'informations sur pourquoi le partitionnement est une bonne chose et la stratégie pour un bon plan de partitionnement sont [plus loin](#).

La procédure d'installation d'OpenBSD essaiera d'auto-partitionner votre disque de façon utile, en configuration "d'utilisation générale", basé sur la taille de votre disque. Si votre disque est assez gros, l'espace inutilisé sera alloué à la partition /home. Bien que cela soit souvent utile, cela ne satisfait pas tous les besoins des utilisateurs.

Pour notre exemple, nous allons considérer que nous allons construire un serveur web statique utilisé par certains de nos amis. Nous avons une machine connectée à un connexion Internet modeste, un disque de 40G, qui est utilisé en grande partie par OpenBSD (avec ma même partition de 5G utilisée par Windows de l'exemple précédent. Pourquoi ? Peut-être que ce système possède un contrôleur RAID qui est supporté par OpenBSD, mais uniquement configurable sous Windows. Plus vraisemblablement, parce que l'éditeur de la FAQ n'a pas envie de maintenir un trop grand nombre d'exemples de systèmes différents).

Les pages web distribuées par un serveur web sous OpenBSD seront dans /var/www, et très peu seront stockées dans /home, donc cela indique qu'un changement définitif de la configuration par défaut doit être effectué. Pour les besoins du débat, nous considérerons que nous n'avons pas à reconstruire le système des sources sur cette machine (nous le ferons ailleurs). Le système n'utilisera pas X, cependant comme certaines applications web [ont besoin de X d'être installé](#), nous aurons X d'installé. La machine ne sera pas trop puissante, il n'y aura pas plus de 1G de RAM dessus, et il est peu probable que notre application en demande plus.

Donc, après un peu de réflexion, notre plan est de partitionner le système comme cela :

- / - root: 100m. Cela sera 'a'.
- swap: 1G (donc nous avons toujours assez d'espace pour un core dump), cela sera la partition 'b'
- /usr: 2g, partition d
- /tmp: 100m (nous ne l'utiliserons pas beaucoup), partition e
- /usr/local: 2g, partition f
- /usr/X11R6: 1g, partition g
- /home: 1g, partition h
- /var: 1g (beaucoup de fichiers de log système), partition j
- /var/www: ce qui reste du disque, partition k

The auto-allocated layout for wd0 is:

```
#          size          offset  fstype [fsize bsize  cpg]
a:         1024.0M        10490445  4.2BSD  2048 16384   1 # /
b:          252.1M        12587597   swap
c:         39205.7M           0  unused
d:         2319.3M        13103933  4.2BSD  2048 16384   1 # /tmp
e:         3653.9M        17853877  4.2BSD  2048 16384   1 # /var
f:         1149.8M        25337016  4.2BSD  2048 16384   1 # /usr
g:         1024.0M        27691862  4.2BSD  2048 16384   1 # /usr/X11R6
h:         3422.6M        29789014  4.2BSD  2048 16384   1 # /usr/local
i:         5122.3M           63   NTFS
j:         1848.7M        36798433  4.2BSD  2048 16384   1 # /usr/src
k:         1848.7M        40584654  4.2BSD  2048 16384   1 # /usr/obj
l:         17540.2M       44370875  4.2BSD  2048 16384   1 # /home
```

Use (A)uto layout, (E)dit auto layout, or create (C)ustom layout? [a] **c**

Si nous avons uniquement des révisions mineures, nous opterons probablement pour "Edit" la configuration personnalisée plutôt que de démarrer d'un état vierge, mais nous allons utiliser la méthode difficile dans notre cas.

You will now create an OpenBSD disklabel inside the OpenBSD MBR partition. The disklabel defines how OpenBSD splits up the MBR partition into OpenBSD partitions in which filesystems and swap space are created. You must provide each filesystem's mountpoint in this program.

The offsets used in the disklabel are ABSOLUTE, i.e. relative to the start of the disk, NOT the start of the OpenBSD MBR partition.

Label editor (enter '?' for help at any prompt)

```
> p
OpenBSD area: 10490445-80293248; size: 69802803; free: 69802803
#          size          offset  fstype [fsize bsize  cpg]
c:         80293248           0  unused
i:         10490382           63   NTFS
>
```

Remarquez qu'il y a déjà deux partitions ici -- la partition "c" qui est toujours présente et crée pour vous, mais disklabel (8) a aussi remarqué la partition NTFS existante et l'a assignée à une partition disklabel qui peut être potentiellement accédée par OpenBSD (remarque, actuellement, le support de NTFS est expérimental et nécessite un noyau personnalisé mais le support FAT/FAT32 est aussi bon).

Nous allons maintenant créer nos partitions. Nous allons démarrer avec la partition "a", notre partition racine :

```
> a a
offset: [10490445] Enter
size: [69802803] 100m
Rounding to cylinder: 208845
FS type: [4.2BSD] Enter
mount point: [none] /
>
```

Remarquez que `disklabel` est par défaut sur le premier secteur disponible pour OpenBSD sur le disque, ce qui est ce que l'on veut. Il est aussi par défaut avec la taille maximale du disque, ce qui n'est PAS ce que l'on veut. Ici nous allons le modifier avec nos tailles préférées, ce qui peut-être spécifié en secteurs, "M" ou "G".

Vous voudrez surement utiliser le FS par défaut de type "4.2BSD" pour une partition FFS (Fast File System) ou FFS2, bien qu'il y ai d'autres types utiles incluant "swap" et "RAID".

Finalement le point de montage. Notre partition "a" est la partition racine, par définition.

Maintenant , nous créons le swap, qui est notre partition "b" (encore, c'est une nécessité -- 'b' sur le disque de démarrage est le swap) :

```
> a b
offset: [10699290] Enter
size: [69593958] 1g
Rounding to cylinder: 2104515
FS type: [swap] Enter
>
```

Encore un fois, `disklabel` a correctement calculé notre secteur de démarrage, et nous l'a présenté en suggérant une taille de "entire remaining space", ce qui encore une fois sera surchargé avec notre choix de taille. Comme c'est la partition "b", `disklabel` considère qu'il va être utilisé comme espace de pagination, et quand nous le confirmerons, il ne nous demandera pas de point de montage.

Nous sommes maintenant prés pour créer le reste des partitions.

```
> a d
offset: [12803805] Enter
size: [67489443] 2g
Rounding to cylinder: 4209030
FS type: [4.2BSD] Enter
mount point: [none] /usr
> a e
offset: [17012835] Enter
size: [63280413] 100m
Rounding to cylinder: 208845
FS type: [4.2BSD] Enter
mount point: [none] /tmp
```

```

> a f
offset: [17221680] Enter
size: [63071568] 2g
Rounding to cylinder: 4209030
FS type: [4.2BSD] Enter
mount point: [none] /usr/local
> a g
offset: [21430710] Enter
size: [58862538] 1g
Rounding to cylinder: 2104515
FS type: [4.2BSD] Enter
mount point: [none] /usr/X11R6
> a h
offset: [23535225] Enter
size: [56758023] 1g
Rounding to cylinder: 2104515
FS type: [4.2BSD] Enter
mount point: [none] /home
> a j
offset: [25639740] Enter
size: [54653508] 1g
Rounding to cylinder: 2104515
FS type: [4.2BSD] Enter
mount point: [none] /var
> a k
offset: [27744255] Enter
size: [52548993] Enter
FS type: [4.2BSD] Enter
mount point: [none] /var/www
>

```

Remarquez que sur la partition /var/www ("k"), nous avons juste utilisé le défaut pour utiliser tout l'espace disque restant. Avec les disques modernes monstrueusement gros, c'est normalement une mauvaise idée. Si vous savez que vous ne l'utiliserez jamais, ne l'allouez pas, et gardez le pour des utilisations futures.

Maintenant, regardons le résultat, utilisant les commandes "p" et "p m" :

```

> p
OpenBSD area: 10490445-80293248; size: 69802803; free: 0
#          size          offset  fstype  [fsize bsize  cpg]
a:         208845         10490445 4.2BSD  2048 16384   1 # /
b:         2104515        10699290  swap
c:         80293248         0  unused
d:         4209030        12803805 4.2BSD  2048 16384   1 # /usr
e:         208845        17012835 4.2BSD  2048 16384   1 # /tmp
f:         4209030        17221680 4.2BSD  2048 16384   1 # /usr/local
g:         2104515        21430710 4.2BSD  2048 16384   1 # /usr/X11R6
h:         2104515        23535225 4.2BSD  2048 16384   1 # /home
i:         10490382         63      NTFS

```

```

j:          2104515          25639740  4.2BSD   2048 16384    1 # /var
k:          52548993          27744255  4.2BSD   2048 16384    1 # /var/www
> p m
OpenBSD area: 10490445-80293248; size: 34083.4M; free: 0.0M
#           size           offset  fstype [fsize bsize  cpg]
a:          102.0M          10490445  4.2BSD   2048 16384    1 # /
b:          1027.6M          10699290  swap
c:          39205.7M          0         unused
d:          2055.2M          12803805  4.2BSD   2048 16384    1 # /usr
e:          102.0M          17012835  4.2BSD   2048 16384    1 # /tmp
f:          2055.2M          17221680  4.2BSD   2048 16384    1 # /usr/local
g:          1027.6M          21430710  4.2BSD   2048 16384    1 # /usr/X11R6
h:          1027.6M          23535225  4.2BSD   2048 16384    1 # /home
i:          5122.3M           63        NTFS
j:          1027.6M          25639740  4.2BSD   2048 16384    1 # /var
k:          25658.7M          27744255  4.2BSD   2048 16384    1 # /var/www
>

```

Comme sur fdisk, vous ne voulez pas que votre partition disklablel OpenBSD ne se chevauche (autre que la partition 'c', qui chevauche tout, bien évidemment).

Ecrivez vos changements et quittez disklablel :

```

> w
> q
No label changes.
newfs: reduced number of fragments per cylinder group from 13048 to 12992 to
enlarge last cylinder group
/dev/rwd0a: 102.0MB in 208844 sectors of 512 bytes
5 cylinder groups of 25.38MB, 1624 blocks, 3328 inodes each
/dev/rwd0h: 1027.6MB in 2104512 sectors of 512 bytes
6 cylinder groups of 202.47MB, 12958 blocks, 25984 inodes each
newfs: reduced number of fragments per cylinder group from 13048 to 12992 to
enlarge last cylinder group
/dev/rwd0e: 102.0MB in 208844 sectors of 512 bytes
5 cylinder groups of 25.38MB, 1624 blocks, 3328 inodes each
/dev/rwd0d: 2055.2MB in 4209028 sectors of 512 bytes
11 cylinder groups of 202.47MB, 12958 blocks, 25984 inodes each
/dev/rwd0g: 1027.6MB in 2104512 sectors of 512 bytes
6 cylinder groups of 202.47MB, 12958 blocks, 25984 inodes each
/dev/rwd0f: 2055.2MB in 4209028 sectors of 512 bytes
11 cylinder groups of 202.47MB, 12958 blocks, 25984 inodes each
/dev/rwd0j: 1027.6MB in 2104512 sectors of 512 bytes
6 cylinder groups of 202.47MB, 12958 blocks, 25984 inodes each
/dev/rwd0k: 25658.7MB in 52548992 sectors of 512 bytes
127 cylinder groups of 202.47MB, 12958 blocks, 25984 inodes each
/dev/wd0a on /mnt type ffs (rw, asynchronous, local)
/dev/wd0h on /mnt/home type ffs (rw, asynchronous, local, nodev, nosuid)
/dev/wd0e on /mnt/tmp type ffs (rw, asynchronous, local, nodev, nosuid)

```



```

/dev/wd0d on /mnt/usr type ffs (rw, asynchronous, local, nodev)
/dev/wd0g on /mnt/usr/X11R6 type ffs (rw, asynchronous, local, nodev)
/dev/wd0f on /mnt/usr/local type ffs (rw, asynchronous, local, nodev)
/dev/wd0j on /mnt/var type ffs (rw, asynchronous, local, nodev, nosuid)
/dev/wd0k on /mnt/var/www type ffs (rw, asynchronous, local, nodev, nosuid)

```

Let's install the sets!

...

[\(Revenir là où on aurait pu\)](#)

4.7 - Quels sont les fichiers nécessaires à l'installation ?

L'installation complète d'OpenBSD est divisée en plusieurs paquetages de fichiers séparés. Chaque application ne demande pas tous les paquetages de fichiers, cependant on recommande aux nouveaux utilisateurs de TOUS les installer. Voici une vue d'ensemble de chacun :

- **bsd** - Ceci est le noyau. **Requis**
- **bsd.mp** - Noyau pour les systèmes multiprocesseurs (SMP) (seulement sur certaines plates-formes)
- **bsd.rd** - [Kernel de disque RAM](#)
- **base47.tgz** - Contient le système de base OpenBSD **Requis**
- **etc47.tgz** - Contient tous les fichiers de /etc **Requis**
- **comp47.tgz** - Contient le compilateur et ses outils, en-têtes et bibliothèques **Recommandé**
- **man47.tgz** - Contient les pages de manuels **Recommandé**
- **misc47.tgz** - Contient différentes informations et documentations de configuration
- **game47.tgz** - Contient les jeux pour OpenBSD
- **xbase47.tgz** - Contient les librairies de base et utilitaires pour X11
- **xetc47.tgz** - Contient les fichiers de configuration /etc/X11 et /etc/fonts
- **xfont47.tgz** - Contient le serveur de fontes X11 et les polices
- **xserv47.tgz** - Contient les serveurs X de X11
- **xshare47.tgz** - Contient les pages de manuels, les options de localisations, les inclusions, etc. pour X

Les paquetages **etc47.tgz** et **xetc47.tgz** ne sont pas installés lors d'une mise à niveau mais uniquement lors d'une installation complète, de ce fait, toute configuration que vous ferez sera conservée. Vous devrez mettre à jour vos dossiers /etc, /dev et /var manuellement.

Pourquoi dois-je installer X pour mes applications non-graphiques ?

Même si vous n'avez pas l'intention de faire tourner X, certains [paquetages](#) nécessitent que les bibliothèques ou d'autres utilitaires présents dans X soient installées sur votre système. Cette dépendance peut facilement être résolue en installant **xbase47.tgz**, le reste de X n'étant pas nécessaire. Beaucoup de personnes résistent dans l'installation de X sur leur système sans raison valable :

- De lui-même, l'installation de X n'engendre pas l'exécution de programmes sur le système.
- De lui-même, l'installation de X sur un système ne change pas le risque de problèmes de sécurité externe.
- Si quelqu'un est déjà SUR votre système, il peut déjà installer ce qu'il veut, donc la présence ou l'absence de X ne change pas sensiblement la situation.

- *Les seules parties de X qui s'exécutent sont celles nécessaires à votre application.*
- *L'espace requis pour X est relativement modeste sur du matériel moderne.*

Les gens perdent quelquefois beaucoup de temps et d'efforts en essayant de chercher et récupérer dans xbase47.tgz juste les fichiers dont ils ont besoin pour installer leur application. Ce n'est pas seulement inutile, mais un effort qui devra être répétée pour chaque cycle de mise à niveau, ce qui signifie probablement que vous ne mettrez pas à jour votre système correctement, créant de VÉRITABLES problèmes de sécurité.

SI vous avez besoin de X, installez le. Cela ne vous fera pas plus mal que l'application qui en a besoin.

Je ne désire pas installer les compilateurs

Ok, ne le faite pas, mais s'il vous plait ne vous dites pas que c'est pour des "raisons de sécurité". Avec le temps quelqu'un qui est sur votre système n'a que faire de la présence ou l'absence de compilateur, il peut facilement l'installer lui-même. Cependant, le paquetage de fichiers compXX.tgz est relativement gros et contient de nombreux fichiers, donc cela peut prendre du temps à l'installer et le mettre à jour, et sur des petits ou lents systèmes, cela peut compter.

Si vous décidez de ne pas installer le compilateur, vous aurez certainement besoin d'un autre système pour maintenir et contruire les mises à jour des logiciels. Il y a eu beaucoup plus de système qui ont été compromis à cause d'une mauvaise maintenance plutôt qu'a cause d'un compilateur installé.

4.8 - Comment est-ce que je dois partitionner mon disque ?

Evidemment, la réponse à cette question dépend de l'utilisation que vous voulez faire du système. OpenBSD peut-être installé sur un espace aussi petit de 512M, mais l'installer sur un périphérique aussi petit nécessite d'être un utilisateur avancé. Tant que vous n'avez pas d'expérience, un HD de 4G ou 8G est recommandé pour démarrer.

A l'inverse de beaucoup d'autres OSs, OpenBSD encourage ces utilisateurs à partitionner leur disque en un nombre de partitions, plutôt que d'avoir une ou deux grosses partitions Il y a un nombre important de raisons pour partitionner son disque :

- **Sécurité** : Vous pouvez marquer certains systèmes de fichiers d'un 'nosuid', 'nodev', 'noexec', 'readonly', etc. Cela est fait par la procédure d'installation, si vous utilisez les partitions recommandées.
- **Stabilité** : Un utilisateur, ou un programme se conduisant mal, peut remplir votre système de fichiers s'il a les droits d'écriture pour le faire. Vos programmes critiques, qui bien sûr sont lancés sur un système de fichier à part, ne seront pas interrompus.
- **Vitesse** : Un système de fichiers sur lequel on écrit souvent peut devenir quelque peu fragmenté. (Par chance, le système de fichiers ffs utilisé par OpenBSD n'est pas enclin à être très fragmenté.)
- **Intégrité** : Si un système de fichiers est corrompu pour quelque raison que ce soit, les autres restent en bon état.
- **Taille** : Beaucoup de machines ont une limite sur la taille de la zone d'un disque d'où la ROM de démarrage peut charger le noyau. Dans certains cas, cette limite peut être vraiment basse (504Mo pour un ancien 486), dans d'autre cas, elle peut être plus grande (par exemple 2Go, 8Go ou 128Go sur les systèmes i386). Comme le système peut être installé n'importe où sur la partition racine, la totalité de la partition racine devrait être comprise dans cette zone. Pour plus de détails, regardez [cette section](#). Une bonne ligne de conduite peut être de conserver votre partition / inférieure à 2Go, dans le cas où vous ne savez pas si votre architecture (et votre machine) peuvent en supporter plus (ou moins).
- **Lecture seule** : Vous pouvez monter des partitions que vous utilisez rarement ou jamais en écriture en "Lecture seule" la plupart du temps, ce qui éliminera le besoin de fsck(8) après un crash ou un arrêt électrique, et previendra l'altération inattendue de données.

- **fsck(8)** : Les très grandes partitions requièrent plus de RAM pour fsck(8), et sur un système avec peu de mémoire, vous pouvez finir par utiliser l'espace de pagination, donnant des temps très long pour fsck.

Donner un espace disque suffisant, la procédure d'installation d'OpenBSD créera par défaut les partitions suivantes :

- **/ - root** : En plus d'être l'endroit où les autres systèmes de fichiers seront montés, le système de fichiers racine contient tous les fichiers nécessaires pour démarrer OpenBSD. Cela inclut le noyau, les utilitaires de base dans /sbin et /bin, les fichiers de configuration dans /etc, le répertoire périphérique, /dev qui sont tous utilisés pour mettre en place le système. Le système de fichiers racine peut-être aussi petit que 60M, de 100M à 200M cela sera plus simple pour une machine de faire des cycles de mise à jour. La partition 'a' de votre disque de démarrage devient automatiquement votre partition racine. CERTAINES plateformes donnent des restrictions sur l'emplacement physique du disque (par exemple le premier disque) pour pouvoir démarrer.
- **Swap** : En plus d'être l'espace de pagination, cette partition est aussi utilisée pour stocker les "core dumps" après un crash du système, il est donc suggéré que l'espace de pagination (si il est mis en place) soit plus grand que la plus grande quantité de RAM jamais disponible sur votre machine. Vous pouvez en savoir plus dans la [FAQ 14](#), [Swap](#).
- **/tmp** : C'est le répertoire accessible en écriture par tous utilisé comme (comme son nom l'indique) espace de stockage temporaire. La plupart des systèmes pourront utiliser une quantité de stockage modeste ici, 50M est courant et supérieur à ce que vous n'aurez jamais besoin, mais il y a quelques applications qui peuvent en utiliser beaucoup, beaucoup plus. Bien que ce répertoire soit accessible en écriture par tous, quand il est dans une partition séparée, OpenBSD le monte par défaut avec nodev et nosuid, ce qui minimise son action pour abuser le système. Remarquez que ce répertoire est vidé au redémarrage, et les fichiers non touchés depuis plus de 24 heures sont supprimés.
- **/var** : Ce répertoire et point de montage est utilisé pour BEAUCOUP de choses, et en fonction de votre utilisation, peut-être un candidat de choix pour être subdivisé en plus de partitions. Certaines des choses qui se retrouvent ici (et potentiellement des points de montage supplémentaires) :
 - /var/log : Journaux système.
 - /var/mail : Boîtes de messagerie en entrée.
 - /var/spool : Envoi des messages (et d'autres choses)
 - /var/www : Le [serveur web](#) d'OpenBSD vit ici.
 - /var/tmp : C'est un répertoire de fichiers temporaires persistents, les fichiers placés ici NE sont PAS supprimés au redémarrage. Par exemple, vi(1) utilise ce répertoire pour le stockage temporaire, donc si votre ordinateur plante ou redémarre pendant que vous éditez un fichier, les fichiers trouvés ici peuvent être utilisés pour récupérer votre session d'édition. Les fichiers laissés ici plus de 24 heures seront supprimés par un script de nettoyage chaque nuit, [daily\(8\)](#).
 - /var/crash : Si le système panique, il essaiera de sauvegarder un core dump dans la partition d'espace de pagination avant de redémarrer. Ce core dump sera ensuite sauvegardé dans /var/crash au redémarrage, donc /var a besoin d'au moins autant d'espace disque que le système de RAM pour que cela fonctionne automatiquement.
- **/usr** : C'est ici que réside la plus grande partie d'OpenBSD. Les programmes binaires, bibliothèques, documentation, pages de manuel, etc... sont tous localisés dans le répertoire /usr. Les fichiers dans ce point de montage sont rarement modifiés -- dans la plupart des cas, vous pouvez facilement monter la partition /usr en lecture seule sans aucun changement système jusqu'à la prochaine mise à jour ou changement de version.
- **/usr/X11R6** : C'est ici que le système X Window réside. Les binaires X, fichiers de fontes, bibliothèques, etc... sont tous ici. La seule chose qui n'est pas ici ce sont les fichiers de configuration.
- **/usr/local** : Sur une installation d'OpenBSD par défaut, ce point de montage/répertoire est complètement vide. Il est utilisé pour installer localement des binaires et bibliothèques d'applications locales.
- **/usr/src** : Ce répertoire comprend les fichiers des sources basiques du système, excluant X et les ports. Ce répertoire est vide par défaut, vous devez le remplir comme c'est expliqué dans la [FAQ 5](#).
- **/usr/obj** : Ce répertoire est utilisé lors du processus de construction avec des fichiers objets et binaires. Avoir ce

répertoire comme point de montage permet de le formater plutôt que d'effacer fichier par fichier, ce qui est beaucoup plus rapide.

- **/home** : C'est ici que vont les fichiers des utilisateurs. L'avoir comme une partition distincte permet de réinstaller plus facilement votre système; il faut simplement ne pas formater cette partition au chargement.

Quelques remarques sur le partitionnement :

- Pour votre première tentative sur un système d'expérimentation, une grosse partition / et une zone "swap" devraient être plus simples jusqu'à ce que vous sachiez combien de place vous est nécessaire. En faisant cela vous sacrifierez certaines fonctions de sécurité d'OpenBSD qui requièrent des systèmes de fichiers séparés pour /, /tmp, /var, /usr et /home. Cependant, vous ne devriez probablement pas mettre votre première installation OpenBSD en production.
- Un système exposé à l'Internet ou à d'autres forces hostiles devrait avoir une partition /var séparée (et peut être même une partition /var/log séparée) pour la journalisation.
- Une partition /home peut être intéressante. Nouvelle version de l'OS ? Supprimez et rechargez tout le reste, conservez votre partition /home intacte. Rappelez vous de garder une copie de vos fichiers de configuration tout de même.
- Une partition séparée pour tout ce qui risque d'accumuler une grosse quantité de fichiers qui devront être supprimés peut être plus rapide à reformater puis re-créeer que de supprimer les fichiers. Regardez la [FAQ de la compilation du code source](#) pour un exemple (/usr/obj).
- Si vous souhaitez reconstruire votre système depuis les sources pour quelque raison que ce soit, les sources seront dans /usr/src. Si vous ne faites pas une partition séparée pour /usr/src, soyez sûr que /usr est suffisamment grand.
- Un fait souvent oublié : vous n'avez **pas** besoin d'allouer toute la place sur un disque quand vous configurez un système ! Etant donné qu'il est maintenant compliqué de trouver un disque dur plus petit que 100Go, il devient sensé de laisser une certaine portion du disque non allouée. Si vous dépassez la taille d'une partition, vous pouvez allouer une nouvelle partition dans l'espace non utilisé, [dupliquer](#) votre partition existante sur la nouvelle, changer /etc/fstab pour pointer vers votre nouvelle partition, la remonter, vous disposez maintenant de plus d'espace.
- Si vous créez vos partitions avec une taille trop proche de la taille minimum requise, vous allez probablement le regretter plus tard lorsqu'il sera temps de mettre à jour votre système.
- Si vous créez des partitions très larges, il est à signaler que la vérification des systèmes de fichiers à l'aide de [fsck\(8\)](#) nécessite approximativement 1Mo de RAM par gigaoctet, et peut être très consommatrice en temps voire infaisable sur des systèmes anciens et lents (merci de consulter [cette section](#)).
- Si vous autorisez les utilisateurs à écrire dans /var/www (ex. pages web personnelles), vous devriez le placer dans une partition séparée, vous pouvez par exemple utiliser les [quotas](#) pour restreindre la place qu'ils utiliseront, de manière à ce que s'ils remplissent la partition, les autres parties du système ne seront pas affectées.
- Vous pourrez éventuellement créer une partition /altroot tel que décrit dans [daily\(8\)](#). Ceci fera une copie quotidienne de votre partition /, vous fournissant ainsi une copie supplémentaire de votre noyau et de vos fichiers de configuration /etc si jamais quelque chose devait arriver à votre partition racine. Bien évidemment, la partition /altroot devrait être aussi grande que /. Si vous disposez d'un second disque et vous avez quelque chose qui duplique le reste de votre disque, soit du [raid\(4\)](#) logiciel ou une copie périodique utilisant [dump\(8\)/restore\(8\)](#), ce disque peut être amorcé après le retrait du disque principal.
- Compiler certains [ports](#) des sources peut utiliser beaucoup d'espace disque sur vos partitions /usr et /tmp. C'est une autre raison de vous suggérer d'utiliser les [paquetages pré-compilés](#) à la place.
- Certains éditeurs utilisent /var/tmp pour de l'espace en cas de plantage, et nécessite donc qu'elle soit aussi grande que le plus grand fichier que vous éditez. Si vous pensez éditer des fichiers de 500M, votre partition /var ou /var/tmp doit être plus grande que les fichiers que vous pensez éditer.

4.9 - Multiboot OpenBSD

Le "Multibooting" est le fait d'avoir plusieurs systèmes d'exploitation sur le même ordinateur, et de pouvoir choisir depuis lequel vous souhaitez démarrer. Ce n'est pas une tâche triviale ! Si vous ne comprenez pas ce que vous êtes en train de faire, vous finirez par perdre une somme conséquente de données sur votre ordinateur. Les nouveaux utilisateurs OpenBSD sont vivement encouragés à démarrer avec un disque dur vierge et sur une machine dédiée, afin d'essayer la configuration désirée sur un système qui n'est pas en production avant d'installer une configuration "multiboot" sur une machine de production. [La FAQ 14](#) donne plus d'informations sur la procédure d'amorçage d'OpenBSD.

Utilisez de préférence une partition parmi les quatre partitions MBR primaires pour démarrer OpenBSD (i.e., les partitions étendues ne fonctionneront pas).

Il faut remarquer que Windows 7 et Vista peuvent modifier la taille de leur partition système : allez dans le panneau de configuration, recherchez "partition" et entrez dans l'outil système correspondant. Cliquez droit sur la partition et vous remarquerez que vous pouvez la réduire. La principale limitation est que le système de fichier d'échange Windows ne peut pas être déplacé, si vous avez besoin de plus d'espace, vous devrez le bouger/supprimer.

Voici plusieurs options pour le "multiboot" :

Configurer la partition active

C'est probablement la solution la plus négligée, et parfois la plus intéressante pour le "multiboot". Configurez simplement comme partition active, la partition d'OS depuis laquelle vous souhaitez démarrer par défaut au prochain démarrage. Chaque OS offre un programme pour faire ceci ; celui d'OpenBSD est [fdisk\(8\)](#), des programmes portant des noms similaires sont disponibles sous Windows 9x et DOS, et la plupart des autres systèmes d'exploitation. Ceci peut être très utile pour les OS ou systèmes long à s'arrêter et redémarrer -- vous pouvez le configurer et lancer la procédure de redémarrage, ensuite aller faire un tour, prendre une tasse de café, et revenir devant le système démarré comme vous le souhaitez -- pas d'attente du Moment Magique pour choisir le système d'exploitation désiré.

Disquette d'amorçage

Si vous avez un système qui utilise OpenBSD peu fréquemment (ou que vous ne voulez pas que les autres utilisateurs de l'ordinateur notent que quoi que ce soit ait changé), vous pouvez utiliser une disquette d'amorçage. Utilisez simplement l'une des [disquettes d'installation standard d'OpenBSD](#), et créez un fichier `/etc/boot.conf` (oui, vous devrez aussi créer un dossier `/etc` sur la disquette) ayant le contenu suivant :

```
boot hd0a:/bsd
```

pour que le système démarre sur le disque dur 0, la partition OpenBSD 'a' et le fichier de noyau `/bsd`. Notez que vous pouvez aussi démarrer d'autres disques avec une ligne comme : `"boot hd2a:/bsd"` pour lancer le troisième disque dur de votre système. Pour lancer OpenBSD, insérez la disquette dans le lecteur et redémarrez. Pour lancer un autre système d'exploitation, éjectez la disquette et redémarrez. (Vous pouvez, bien évidemment, utiliser cette disquette pour créer un CD démarrable aussi)

Le programme [boot\(8\)](#) chargé depuis la disquette, cherche et lit `/etc/boot.conf`. L'instruction `"boot hd0a:/bsd"` indique à `boot(8)` depuis quel endroit charger le noyau -- dans ce cas, le premier disque dur que le BIOS voit. Gardez à l'esprit que seulement un petit fichier (`/boot`) est chargé depuis la disquette -- le système charge le noyau entier depuis le disque dur, cela ne ralentit que de quelques secondes la procédure de démarrage.

Windows NT/2000/XP NTLDR

Pour un "multiboot" entre OpenBSD et Windows NT/2000/XP, vous pouvez utiliser NTLDR, le chargeur de démarrage que NT utilise. Pour "multi-booter" avec NT, vous aurez besoin d'une copie de votre "Partition Boot Record" (PBR) OpenBSD. Après avoir lancé "installboot" vous pouvez en obtenir une copie dans un fichier en utilisant [dd\(1\)](#), en suivant une procédure similaire à la procédure suivante :

```
# dd if=/dev/rsd0a of=openbsd.pbr bs=512 count=1
```

Remarque : ceci est une excellente occasion pour vous rappeler que saisir bêtement des commandes auxquelles vous ne comprenez rien est vraiment une mauvaise idée. La commande ci-dessus ne fonctionnera pas directement sur la plupart des machines. Au lecteur de l'adapter à son équipement.

Maintenant démarrez sous NT et mettez `openbsd.pbr` dans C:. Ajoutez une ligne comme celle-ci à la fin du fichier C:\BOOT.INI :

```
c:\openbsd.pbr="OpenBSD"
```

Quand vous redémarrerez, vous devriez être en mesure de choisir OpenBSD dans le menu de chargement NT. D'autres informations sur le NTLDR sont disponibles dans le ["NTLDR Hacking Guide"](#).

Sur Windows XP vous pouvez aussi éditer les informations en utilisant la "GUI" ; consultez le [XP Boot.ini HOWTO](#).

Des programmes faisant la plupart de ce travail sont à votre disposition, par exemple [BootPart](#). Ce programme peut être lancé depuis Windows NT/2000/XP, et ira chercher le PBR OpenBSD, le mettra dans votre partition NT/2000/XP partition et le rajoutera dans C:\BOOT.INI.

Note : Le chargeur de démarrage Windows NT/2000/XP est seulement capable de démarrer des systèmes depuis le premier disque dur. Vous ne pouvez pas l'utiliser pour charger OpenBSD depuis le second disque sur un système.

Windows Vista

Avec Vista, Microsoft a supprimé le support de NTLDR en faveur de leur nouveau stockage des données de configuration de démarrage (BCD) utilisé pour contrôler l'environnement de démarrage. Depuis que BOOT.INI n'est plus disponible pour la configuration, un utilitaire en ligne de commande, `bcdedit`, le remplace.

Des que la PBR OpenBSD est copiée dans la partition système Vista, les trois commandes suivantes sont nécessaires pour sélectionner et démarrer OpenBSD quand le système redémarre :

```
C:\Windows\system32> bcdedit /create /d "OpenBSD/i386 4.7" /application
bootsector
The entry {05a763ce-d81b-11db-b3ec-000000000000} was successfully
created.
```

```
C:\Windows\System32>
```

Le GUID retourné ici, `05a763ce-d81b-11db-b3ec-000000000000`, est affiché pour l'exemple. Il faut noter que

le GUID affiché quand vous utiliserez cet commande devra être copié dans les commandes suivantes. En copiant simplement le GUID affiché ici ne fonctionnera pas.

Les deux commandes suivantes sont aussi nécessaires :

```
C:\Windows\system32> bcdedit /set {05a763ce-d81b-11db-b3ec-000000000000}
device boot
The operation completed successfully.
```

```
C:\Windows\system32> bcdedit /set {05a763ce-d81b-11db-b3ec-000000000000}
path \openbsd.pbr
The operation completed successfully.
```

```
C:\Windows\system32>
```

Cela doit être fait dans un shell avec les droits administrateurs. Des que vous avez localisé `cmd.exe`, click droit pour être capable de sélectionner "run as administrator".

Il faut remarquer que le chemin absolu du fichier PBR importé. N'ajoutez pas de lettre de lecteur car il est acqui que le fichier est placé dans la partition système. `bcdedit` ne se plaindra pas de la fourniture d'un lecteur spécifique, mais le gestionnaire de démarrage se plaindra plus tard qu'il ne peut pas résoudre le chemin fourni.

Après redémarrage, Vista sera le premier affiché dans le gestionnaire de démarrage suivi de OpenBSD. En sélectionnant l'une des deux entrée vous démarrerez le système d'exploitation sélectionné.

Si rien ne se passe, regardez du coté du panneau de contrôle pour les informations de démarrage. La plupart du temps, votre Windows démarre sans délai, donc vous ne pouvez pas voir le menu de démarrage. Vous pouvez aussi l'utiliser pour démarrer OpenBSD par défaut.

Pour plus d'informations, consultez l'aide de `bcdedit` en tapant :

```
C:\Windows\system32> bcdedit /?
```

ou en cherchant dans la documentation de Microsoft ou sur le site Web. Une bonne introduction peut-être trouvée sur cet article TechNet [Frequently Asked Questions](#).

Pour ceux qui trouvent cette configuration manuelle compliquée, [EasyBCD](#) fournit une alternative GUI.

Windows 7

Microsoft a amélioré BCD depuis Vista en permettant plusieurs versions de Windows de démarrer à travers `bcdedit`. A cause de ce plus grand contrôle, cinq commandes sont nécessaires pour configurer un environnement multiboot avec OpenBSD.

Après la copie de la PBR OpenBSD dans la partition système de Windows 7, il faut taper les commandes suivantes pour initialiser les registres nécessaires :

```
C:\Windows\system32> bcdedit /create /d "OpenBSD/i386 4.7" /application
```

```
bootsector
```

The entry {0154a872-3d41-11de-bd67-a7060316bbb1} was successfully created.

```
C:\Windows\system32>
```

Comme signalé précédemment, le GUID {0154a872-3d41-11de-bd67-a7060316bbb1} est dépendant du système. Conservez la valeur que vous avez lors de l'exécution et copiez la dans les commandes suivantes :

```
C:\Windows\system32> bcdedit /set {0154a872-3d41-11de-bd67-a7060316bbb1}
device boot
The operation completed successfully.
```

```
C:\Windows\system32> bcdedit /set {0154a872-3d41-11de-bd67-a7060316bbb1}
path \openbsd.pbr
The operation completed successfully.
```

```
C:\Windows\system32> bcdedit /set {0154a872-3d41-11de-bd67-a7060316bbb1}
device partition=c:
The operation completed successfully.
```

```
C:\Windows\system32> bcdedit /displayorder {0154a872-3d41-11de-bd67-
7060316bbb1} /addlast
The operation completed successfully.
```

```
C:\Windows\system32>
```

Autres chargeurs de démarrage

D'autres utilisateurs de chargeurs de démarrage OpenBSD ont inclus avec succès [GAG](#), [The Ranish Partition Manager](#), [rEFIt](#) et [GRUB](#).

OpenBSD et Linux (i386)

Veuillez vous référer au [INSTALL.linux](#), qui donne les instructions en profondeur pour faire fonctionner OpenBSD avec Linux.

Problèmes de fuseau horaire

OpenBSD s'attend à ce que l'horloge de la machine soit réglée sur UTC ("Universal Coordinated Time" - Temps universel coordonné). D'autres systèmes d'exploitation s'attendent au contraire que celle-ci soit réglée sur le fuseau local. Bien évidemment, cela peut engendrer un certain problème si vous utilisez ces deux types de systèmes sur une seule et même machine. L'un ou l'autre aura besoin d'être configuré en conséquence. Plus d'informations sur cette manipulation sont disponibles dans la [FAQ 8 - Pourquoi mon horloge avance/retarde-t-elle de plusieurs](#)

4.10 - Envoyer votre dmesg à dmesg@openbsd.org après l'installation

Rappelez vous, il est important pour les développeurs OpenBSD de garder une trace de quels matériels fonctionnent, et de

quels matériels ne fonctionnent pas parfaitement, incluant les capteurs matériels qui sont trouvés dans ces machines.

Un commentaire de /usr/src/etc/root/root.mail

If you wish to ensure that OpenBSD runs better on your machines, please do us a favor (after you have your mail system configured!) and type something like:

```
# (dmesg; sysctl hw.sensors) | \
    mail -s "Sony VAI0 505R laptop, apm works OK" dmesg@openbsd.org
```

so that we can see what kinds of configurations people are running. As shown, including a bit of information about your machine in the subject or the body can help us even further. We will use this information to improve device driver support in future releases. (Please do this using the supplied GENERIC kernel, not for a custom compiled kernel, unless you're unable to boot the GENERIC kernel. If you have a multi-processor machine, dmesg results of both GENERIC.MP and GENERIC kernels are appreciated.) The device driver information we get from this helps us fix existing drivers. Thank you!

Soyez sûr d'envoyer le mail depuis un compte sous lequel vous serez habilité à recevoir pour que les développeurs puissent vous contacter s'il ont quelque chose qu'il voudraient que vous testiez ou changiez afin que votre configuration fonctionne. Il n'est pas important d'envoyer le mail depuis la même machine que celle sur laquelle tourne OpenBSD, donc si cette dernière n'est pas en mesure de recevoir des mails, faites simplement :

```
$ (dmesg; sysctl hw.sensors) | mail your-account@yourmail.dom
```

et transférez le message à

dmesg@openbsd.org

Où your-account@yourmail.dom est votre compte de messagerie régulier.

NOTE

- *Merci de n'envoyer que des dmesg concernant les noyaux GENERIC. Les noyaux personnalisés qui ont des drivers de matériels en moins ne sont pas utiles.*
- *Si vous avez un système multi-processeurs et utilisez généralement le noyau GENERIC.MP, il est utile pour les développeurs d'avoir le dmesg des deux noyaux GENERIC et GENERIC.MP ; merci de les envoyer dans des emails différents.*
- *Les dmesgs sont reçus sur une machine utilisant le système de filtrage de spam [spamd](#). Cela peut causer le rejet temporaire de votre dmesg par les serveurs de mails. Soyez patient, après une demi-heure voire une heure, il sera reçu.*

La méthode précédente est très simple mais si vous avez choisi de ne pas configurer le système de messagerie de votre machine OpenBSD, vous devriez toutefois envoyer votre dmesg aux développeurs. Sauvegardez simplement la sortie de votre dmesg dans un fichier texte.

```
$ (dmesg; sysctl hw.sensors) > ~/dmesg.txt
```

Puis transférez ce fichier (en utilisant FTP/scp/floppydisk/pigeon voyageur/...) vers le système depuis lequel vous utilisez votre messagerie. Puisque la réception des messages dmesg est gérée automatiquement, merci de vous référer aux recommandations suivantes lors de l'utilisation de systèmes/clients de messagerie alternatifs :

- *Configurez votre client de messagerie afin que celui-ci envoie les messages en tant que texte simple ("plain text") ; n'utilisez pas le format HTML*
- *Désactivez le retour à la ligne automatique. De nombreux clients de messagerie sont configurés pour effectuer un renvoi à la ligne automatiquement à la 72^e colonne (la norme pour les listes de diffusion).*
- *Soyez certains que votre client reformate pas les messages en "text-flow"*
- *N'envoyez pas la sortie du dmesg en tant que fichier joint. Mettez le dmesg directement dans le corps du message.*

4.11 - Ajouter un paquetage après l'installation

"Oh non ! J'ai oublié de rajouter un paquetage quand j'ai fait l'installation !"

Parfois, vous réalisez que vous AURIEZ vraiment eu besoin de `comp47.tgz` (ou de n'importe quel composant système) après tout, mais vous ne l'avez pas réalisé quand vous avez installé votre système. Bonne nouvelle : Il y a deux voies relativement simples pour rajouter un paquetage après l'installation initiale :

En utilisant la procédure de mise à jour

Démarrez simplement votre média d'installation (CD-ROM ou disquette), et choisissez "Upgrade" (plutôt que "Install"). Quand vous aurez la liste des paquetages, sélectionnez simplement celui que vous avez oublié d'installer la première fois, choisissez la source, et laissez-le l'installer pour vous.

En utilisant tar(1)

Les paquetages d'installation sont de simples fichiers compressés tar, et vous pouvez les décompresser vous même manuellement depuis la racine du système de fichiers.

```
# cd /
# tar xzvpf comp47.tgz
```

N'oubliez PAS l'option 'p' ci-dessus qui restaurera correctement les permissions sur les fichiers !

Une méprise courante est de croire qu'il est possible d'utiliser [pkg_add\(1\)](#) pour rajouter des sets d'installation manquants. Cela ne fonctionne pas. `pkg_add(1)` est l'[outil de gestion des paquetages](#) pour installer des applications tierces. Cet outil prend en compte les fichiers paquetages et non des archives tar génériques telles que les ensembles d'installation.

Si vous installez l'ensemble de fichier `xbase` sur votre système pour la première fois en utilisant `tar(1)` et sans

redémarrer, le cache des bibliothèques partagées devra être mis à jour après installation en utilisant [ldconfig\(8\)](#). Pour ajouter toutes les bibliothèques X au cache :

```
# ldconfig -m /usr/X11R6/lib
```

Alternativement, vous pouvez juste redémarrer votre système, et ceci sera fait automatiquement par le script de démarrage [rc\(8\)](#).

4.12 - Qu'est ce que 'bsd.rd' ?

`bsd.rd` est un noyau "RAM Disk". Ce fichier peut être vraiment intéressant ; beaucoup de développeurs prudents en conservent un tout le temps à la racine de leur système.

Le noyau "RAM Disk" définit la racine du système de fichiers du noyau -- plutôt qu'être stockés sur un disque physique, les utilitaires disponibles après l'amorçage de `bsd.rd` sont enregistrés dans le kernel, et lancés depuis un système de fichiers basé en mémoire RAM. `bsd.rd` comporte aussi une floppée d'utilitaires vous permettant de faire de la maintenance système et de lancer une installation.

Sur certaines architectures, `bsd.rd` est actuellement la méthode d'installation privilégiée -- vous placez ce noyau sur un système de fichiers, vous le démarrez, et lancez l'installation depuis ce dernier. Sur la plupart des architectures, si vous utilisez une ancienne version d'OpenBSD, vous pouvez obtenir une nouvelle version de `bsd.rd` par FTP, redémarrer à partir de lui, et installer la nouvelle version d'OpenBSD sans avoir besoin de quelque média amovible que ce soit.

Voici un exemple d'amorçage de `bsd.rd` sur un système i386 :

```
Using Drive: 0 Partition: 3
reading boot.....
probing: pc0 com0 com1 apm mem[639k 255M a20=on]
disk: fd0 hd0+
>> OpenBSD/i386 BOOT 3.02
boot> boot hd0a:/bsd.rd
. . . normal boot to install . . .
```

Comme indiqué, vous allez être amené au programme d'installation, mais vous pouvez aussi aller à une invite de shell pour faire de la maintenance sur votre système.

La règle générale en lançant `bsd.rd` est de changer votre noyau d'amorçage de `/bsd` à `bsd.rd` quelle que soit la méthode pour votre architecture.

4.13 - Problèmes d'installation courants

4.13.1 - Mon Compaq ne reconnaît que 16Mo de RAM

Certains systèmes Compaq rencontrent un problème où la mémoire RAM n'est pas complètement détectée par le [Chargeur d'amorçage de second niveau OpenBSD](#) et seulement 16Mo seront détectés et utilisés par OpenBSD. Ceci peut être corrigé en créant/éditant le fichier [/etc/boot.conf](#), ou en entrant des commandes à l'invite "boot>" avant qu'OpenBSD ne se charge. Si vous avez une machine avec 64Mo de RAM, mais qu'OpenBSD n'en a détecté que 16Mo, la commande

devrait être :

```
machine mem +0x3000000@0x1000000
```

pour ajouter 48Mo (0x3000000) après les premiers 16Mo (0x1000000). Typiquement, si vous avez une machine avec ce problème, vous devriez entrer la commande précédente d'abord dans l'invite `boot>` du CD-ROM ou de la Disquette, charger la disquette, redémarrer et créer un fichier `/etc/boot.conf` avec la ligne précédente pour que dans les démarrages suivants OpenBSD reconnaisse toute la mémoire disponible.

Une mise à jour ROM règlera ce problème sur certains systèmes.

4.13.2 - Mon i386 ne démarre pas après l'installation

Votre installation a eu l'air de bien se dérouler, mais lors de votre premier démarrage, vous ne voyez aucun signe montrant qu'OpenBSD essaye de démarrer. Plusieurs problèmes courants peuvent expliquer ce phénomène :

- **Aucune partition n'a été définie active dans `fdisk(8)`.** Pour corriger cela, relancez la machine en utilisant une disquette de démarrage ou tout autre média, et marquez une partition comme active. Regardez [ici](#) et [ici](#).
- **Aucun chargeur de démarrage valide n'a été installé sur le disque.** Si vous avez tapé "Enter" ou répondu "Y" à la question "Use (W)hole disk or (E)dit the MBR?" pendant l'installation, ou utilisé l'option "reinit" de `fdisk(8)`, l'amorce OpenBSD a été installée sur le "Master Boot Record" du disque ; autrement, le programme d'amorce est conservé intact. Ce sera un problème si aucun autre programme d'amorce n'existe. Une solution est de démarrer le média d'installation une nouvelle fois, basculer dans le shell et invoquer la commande `fdisk(8)` pour mettre à jour le MBR depuis la ligne de commande :

```
# fdisk -u wd0
```

Note : l'option "update" du mode ("-e") interactif de `fdisk` n'écrira pas les bits de signature requis pour rendre le disque amorçable.

- **Dans quelques rares cas, quelque chose s'est mal déroulé dans l'installation du chargeur de démarrage de stage 2.** La réinstallation du chargeur de démarrage de stage 2 est vue [ici](#).

4.13.3 - Ma (vieille et lente) machine a démarré, mais bloque pendant la procédure `ssh-keygen`

Il semble que votre machine fonctionne correctement, mais prend juste beaucoup de temps pendant la procédure de génération de clés `ssh`. Une SPARCStation2 ou un Macintosh Quadra peut prendre plusieurs heures ou plus pour terminer les trois étapes [ssh-keygen\(1\)](#), Laissez le simplement terminer ; cela n'est réalisé qu'une fois par installation.

La taille de la clé par défaut a été augmentée pour OpenBSD 3.8, en conséquence de quoi les temps de génération de clés sont beaucoup plus élevés que ce qu'ils furent. Les utilisateurs possédant des machines très lentes souhaiteront peut-être générer leurs clés sur une autre machine, les placer dans l'archive [site47.tgz](#), et les installer avec le reste des "file sets".

4.13.4 - J'ai le message "Failed to change directory" pendant l'installation

Quand vous faites l'installation d'un [snapshot](#) durant la phase -beta du cycle de développement OpenBSD, vous devriez voir ceci :

```
Display the list of known ftp servers? [no] yes
Getting the list from 192.128.5.191 (ftp.openbsd.org)... FAILED
Failed to change directory.
Server IP address or hostname?
```

Cela est normal et souhaité pendant la version précédant la sortie officielle dans le cycle. Le programme d'installation cherche la liste FTP sur le premier serveur FTP dans un dossier qui ne sera pas disponible avant [la date de "release"](#), vous obtiendrez donc les messages précédents.

Utilisez simplement la [liste de miroirs FTP](#) pour trouver votre site miroir FTP favori, et entrez manuellement son nom lorsque cela vous est demandé.

Note : *Vous ne devriez pas voir cela si vous installez une version "-release" ou depuis un CD-ROM.*

4.13.5 - Ma table de partition fdisk est corrompue ou vide !

Occasionnellement, un utilisateur trouvera un système fonctionnant, mais en faisant un `fdisk wd0`, il trouvera une table de partitions vide (ou polluée). Cela est usuellement dû à la création d'une partition dans [fdisk\(8\)](#) ayant un offset de zero secteur, au lieu de [l'offset d'une piste](#) qu'elle est sensée avoir. (note : cela ne concerne que les plates-formes [i386](#) et [amd64](#). Les autres plates-formes requièrent des offsets différents, certaines n'en requièrent pas). Le système [démarr](#)e ensuite en utilisant le PBR, pas le MBR.

Bien que cette configuration peut fonctionner, cela peut causer des problèmes de maintenance et devrait être corrigé. Pour corriger ce problème, le système de fichiers doit généralement être recréé depuis le début (si vous savez VRAIMENT ce que vous faites, vous devriez être en mesure de recréer juste le disklabel et le MBR et ne perdre que la première partition OpenBSD du disque).

4.13.6 - Je ne possède pas de lecteur de disquette ou de CD-ROM sur ma machine

Certaines personnes possédant un ordinateur pourraient vouloir utiliser OpenBSD avec une absence évidente de moyen pour installer OpenBSD, n'ayant aucun lecteur de disquette ou de lecteur de CD-ROM. Soit la machine a été conçue sans eux (par exemple plusieurs portables ou machines basées sur des "flash", comme les systèmes Soekris ou l'ALIX), ou le périphérique de démarrage est défectueux ou absent, et est difficile à remplacer. Voici quelques conseils et techniques que vous pouvez utiliser pour installer OpenBSD sur ces systèmes.

- Démarrage Réseau utilisant [PXE](#) (i386 ou amd64) ou [diskless\(8\)](#) (autres plateformes).
- Lecteur de CD-ROM USB ou lecteur de disquette USB, si votre machine peut démarrer de l'un d'eux.
- Disque Flash USB ou disque dur, encore si votre ordinateur peut démarrer d'un périphérique USB. Préparez le périphérique sur un autre ordinateur comme décrit dans [FAQ 14](#). Démarrez le mais choisissez le kernel `bsd.rd`, puis installez le normalement. Vous pouvez aussi avoir les groupes de fichiers pré-chargés sur un média flash bien évidemment.
- Dans le pire de cas, si rien de ce qui a été décrit précédemment n'est valable, vous pouvez éventuellement sortir le disque du système cible, utiliser un adaptateur adapté pour l'installer dans un ordinateur "normal", installer OpenBSD et replacer le disque dans le système cible. OpenBSD démarrera bien maintenant dans la machine cible, vous devrez peut-être très probablement adapter la configuration réseau. Vous devrez aussi modifier `/etc/fstab` si vous avez (par exemple) utilisé un adaptateur USB->IDE/SATA ou votre cible ou la machine utilise [ahci\(4\)](#) et l'autre pas. Les disques IDE et certains SATA sont reconnus normalement comme des périphériques [wd\(4\)](#),

mais si ils sont connectés à un adaptateur USB, ils deviendront [sd\(4\)](#). Un disque SATA connecté à une interface [pciide\(4\)](#) sera vu comme `wd(4)`, mais connecté à une interface [ahci\(4\)](#), deviendra un périphérique `sd(4)`. Cependant, des que vous aurez corrigé le fichier `/etc/fstab`, le système démarrera normalement.

Dans tous les cas, rappelez vous que la machine a eu un OS installé avant cela et donc que cet OS pouvait être réinstallé. Comment cela a été réalisé initialement vous donnera surement une bonne idée de la façon d'installer OpenBSD maintenant.

4.14 - Personnaliser la procédure d'installation

Fichier `siteXX.tgz`

Les scripts d'installation/mise à jour d'OpenBSD autorisent la création d'un set utilisateur nommé "`siteXX.tgz`", où `XX` représente la version (ex. 47). Le fichier `siteXX.tgz` est, comme les autres [paquetages](#), une compression [gzip\(1\)](#) d'archive [tar\(1\)](#) dont la racine est `/` et est décompressé comme les autres avec les options `xzphf`. Ce paquetage sera installé en dernier, après tous les autres paquetages.

Ce paquetage vous permet d'ajouter et/ou écraser des fichiers installés dans les paquetages 'normaux' et donc de personnaliser l'installation ou la mise à jour.

Vous pouvez également créer des sets d'installation spécifiques à un hôte, nommés `siteXX-<hostname>.tgz`, par exemple, "`site47-puffy.tgz`". Cela facilite les installations customisées, mises à jour ou récupération après désastre.

Quelques exemples d'utilisation de fichier `siteXX.tgz` :

- Créer un fichier `siteXX.tgz` qui contient tous les changements que vous avez fait depuis la première installation de OpenBSD. Ensuite, si vous avez à récréer le système, sélectionnez simplement `siteXX.tgz` pendant la procédure de réinstallation et toutes les modifications que vous avez faites seront répliquées sur le nouveau système.
- Créer une série de dossiers spécifiques machine qui contiennent chacun un fichier `siteXX.tgz` dans lequel se trouvent les fichiers spécifiques à la machine. L'installation de machines (ex. machines avec des cartes graphiques différentes) d'une catégorie particulière peut être faite en choisissant le fichier `siteXX.tgz` approprié.
- Mettez les fichiers que vous paramétrez constamment dans un fichier `siteXX.tgz` similaire -- fichiers [/etc/skel](#), [/etc/pf.conf](#), [/var/www/conf/httpd.conf](#), [/etc/rc.conf.local](#), etc.

Scripts `install.site/upgrade.site`

A la dernière étape de la procédure d'installation/mise à jour, le script cherche dans la racine un `install.site` ou un `upgrade.site` d'un nouveau système ou d'une nouvelle mise à jour, selon la procédure en cours, et lance le script dans un environnement [chrooté](#) de la racine système de l'installation/de la mise à jour. Rappelez-vous, la mise à jour est faite depuis un système de fichiers démarré donc votre système de fichiers cible est actuellement monté dans `/mnt`. Cependant, votre script peut être écrit tel quel à cause du `chroot`, comme s'il était écrit dans la racine normale de votre système de fichiers. Comme ce script est lancé après que tous les fichiers aient été installés, vous avez la plupart des fonctionnalités d'un système complet quand votre script est exécuté. Gardez en mémoire que vous exécutez un noyau minimal, toutes les fonctionnalités ne sont pas disponibles, et à cause des contraintes d'espaces, les choses qui fonctionnent aujourd'hui peuvent ne pas fonctionner dans les prochaines versions.

Notez que le script `install.site` devra être placé dans un fichier `siteXX.tgz`, tandis que le script `upgrade.site`.

`site` pourra être placé à la racine du système de fichiers avant la mise à jour ou bien être placé lui aussi dans fichier `siteXX.tgz`.

Ce script peut être utilisé pour faire de nombreuses choses.

- Supprimer des fichiers qui sont installés/mis à jour et que vous ne souhaitez pas présents dans votre système.
- Supprimer/mettre à jour/installer les [paquetages](#) que vous souhaitez sur le système installé (peut ne pas fonctionner pour tous les paquetages !).
- Faire une [sauvegarde/archive immédiate](#) de votre nouveau système avant de l'exposer au reste du monde.
- Utilisez [rdate\(8\)](#) pour paramétrer l'horloge du système.

La combinaison de `siteXX.tgz` et de `install.site/upgrade.site` a pour but de donner de larges capacités de personnalisation sans avoir à créer ses propres paquetages d'installation.

Remarque : Si vous comptez effectuer votre installation à partir d'un serveur HTTP, vous devez ajouter votre ou vos fichier(s) `site*.tgz` au fichier `index.txt` se trouvant dans le répertoire source afin que ce ou ces fichier(s) soi(en)t proposé(s) en option durant l'installation. Cette opération n'est pas nécessaire pour les installations à partir de FTP ou toute autre type d'installation.

4.15 - Comment puis-je installer plusieurs systèmes identiques ?

Voici quelques utilitaires que vous pouvez utiliser lorsque vous avez plusieurs systèmes OpenBSD identiques à déployer.

Les fichiers `siteXX.tgz` **et** `install/upgrade.site`

Voir l'article [précédent](#).

Restauration depuis `dump(8)`

Sur la plupart des architectures, le média de démarrage inclut le programme [restore\(8\)](#) qui peut être utilisé pour restaurer une sauvegarde faite par [dump\(8\)](#). Ainsi, vous pouvez démarrer depuis [disquettes](#), [CD](#), ou fichier [bsd.rd](#), ensuite [fdisk](#), [disklabel](#), et [restore](#) pour restaurer la configuration désirée depuis une bande ou autre média, et installer les [blocs de d'amorce](#). Plus de détails [ici](#).

Image de disque

Malheureusement, il n'existe pas de paquetage d'image de disque reconnaissant le FFS, et qui pourrait faire une image contenant simplement l'espace de disque utilisé. La plupart des solutions d'image de disque traiteront la partition OpenBSD comme une partition "générique", et pourront simplement faire l'image de l'intégralité du disque. Cela rejoint souvent notre but, mais souvent avec d'énormes quantités d'espace perdu -- une partition `/home` de 10Go vide demandera 10Go d'espace dans l'image, même s'il n'y a aucun fichier à l'intérieur. Tandis que vous pouvez typiquement installer une image de disque sur un disque plus grand, vous ne pourrez pas l'installer sur un disque de plus petite taille.

Si cela est pour vous acceptable, vous devriez trouver dans la commande [dd](#) tout ce dont vous avez besoin, autorisant la copie d'un disque vers un autre, secteur-par-secteur. Celui-ci vous fournira souvent les mêmes fonctionnalités que les programmes commerciaux, sans le prix.

4.16 - Comment puis-je obtenir un dmesg(8) pour rapporter un problème d'installation ?

Lorsque vous [rapportez un problème](#), il est important d'inclure le [dmesg\(8\)](#) complet du système. Souvent cependant, vous en avez besoin parce que le système ne fonctionne pas correctement ou ne s'installe pas, donc vous n'aurez pas de disque, pas de réseau ou manquez d'une autre ressource vous permettant d'envoyer votre dmesg à la [mailing liste](#) appropriée. Il y a d'autres façons de faire cela, cependant :

- **Disquette** : Les disques d'amorçage et les CD-ROM ont assez d'outils pour permettre d'enregistrer votre dmesg sur une disquette MSDOS afin de le lire sur une autre machine. Mettez une disquette MSDOS formatée dans votre lecteur de disquette et tapez les commandes suivantes :

```
mount -t msdos /dev/fd0a /mnt
dmesg >/mnt/dmesg.txt
umount /mnt
```

Si vous avez un autre système OpenBSD, vous pouvez aussi écrire sur une disquette compatible OpenBSD -- souvent, la disquette d'amorce a encore assez d'espace libre pour contenir le dmesg. Dans ce cas, retirez le "-t msdos" ci-dessus.

- **Console série** : Utiliser une console série et capturer la sortie sur un autre ordinateur est souvent la meilleure solution pour obtenir des informations de diagnostic - particulièrement si l'ordinateur "panic" immédiatement après le démarrage. Aussi bien qu'un second ordinateur, vous aurez besoin d'un câble série (souvent un câble null-modem), et d'un émulateur de terminal pouvant capturer la sortie de l'écran dans un fichier.

Des informations sur la configuration d'une console série sont données [ailleurs dans la FAQ](#); dans le but de capturer le log de l'installation, les commandes suivantes sont souvent suffisantes.

i386

A l'invite de démarrage tapez :

```
boot> set tty com0
```

Cela indiquera à OpenBSD d'utiliser le premier port série (souvent appelé COM1 ou COMA dans la documentation du PC) en tant que console série. La bande passante par défaut est 9600 bauds.

Sparc/Sparc64

Ces machines utiliseront automatiquement une console série si elles sont lancées sans clavier. Si vous avez un clavier et un écran attachés, vous pouvez toujours forcer le système à utiliser une console série avec l'invocation suivante à l'invite ok.

```
ok setenv input-device ttya
ok setenv out-device ttya
ok reset
```

[\[Retour à l'Index principal\]](#) [\[Section 3 - Démarrer avec OpenBSD\]](#) [\[Section 5 - Construire le Système à partir des](#)

[Sources\]](#)



www@openbsd.org

\$OpenBSD: faq4.html,v 1.99 2010/07/04 17:26:24 ajacoutot Exp \$

[\[Index de La FAQ\]](#) [\[Section 4 - Guide d'Installation\]](#) [\[Section 6 - Mise en place du réseau\]](#)

5 - Construire le Système à partir des Sources

Table des matières

- [5.1 - Les Saveurs \("Flavors"\) OpenBSD](#)
- [5.2 - Pourquoi devrais-je compiler mon système depuis les sources ?](#)
- [5.3 - Compilation d'OpenBSD depuis les sources](#)
 - [5.3.1 - Aperçu du processus de compilation](#)
 - [5.3.2 - Installer ou mettre à niveau depuis les binaires les plus récents](#)
 - [5.3.3 - Téléchargement des sources appropriées](#)
 - [5.3.4 - Compilation du noyau](#)
 - [5.3.5 - Compilation du "userland"](#)
- [5.4 - Compilation d'une Révision](#)
- [5.5 - Compilation de X](#)
- [5.6 - Pourquoi aurais-je besoin d'un noyau sur mesure ?](#)
- [5.7 - Options de configuration du noyau](#)
- [5.8 - Configuration au démarrage](#)
- [5.9 - Utilisation de config\(8\) pour changer le binaire du noyau](#)
- [5.10 - Obtention d'une sortie plus verbeuse lors du démarrage](#)
- [5.11 - Problèmes courants, astuces et questions lors de la compilation et de la construction](#)
 - [5.11.1 - La compilation s'est arrêtée avec une erreur "Signal 11"](#)
 - [5.11.2 - "make build" échoue en générant un message "cannot open output file snake: is a directory"](#)
 - [5.11.3 - Mon système sans IPv6 ne fonctionne pas !](#)
 - [5.11.4 - Oops ! J'ai oublié de créer le répertoire `/usr/obj` avant de commencer !](#)
 - [5.11.5 - Placer `/usr/obj` sur sa propre partition](#)
 - [5.11.6 - Comment ne pas compiler certaines parties de l'arbre ?](#)
 - [5.11.7 - Où puis-je avoir plus d'informations sur le processus de compilation ?](#)
 - [5.11.8 - Je ne vois aucun snapshot sur le site FTP. Où sont t'ils passés ?](#)
 - [5.11.9 - Comment puis-je amorcer sur une nouvelle version du compilateur \(`gcc`\)?](#)
 - [5.11.10 - Quel est le meilleur moyen de mettre à jour `/etc`, `/var`, et `/dev` ?](#)

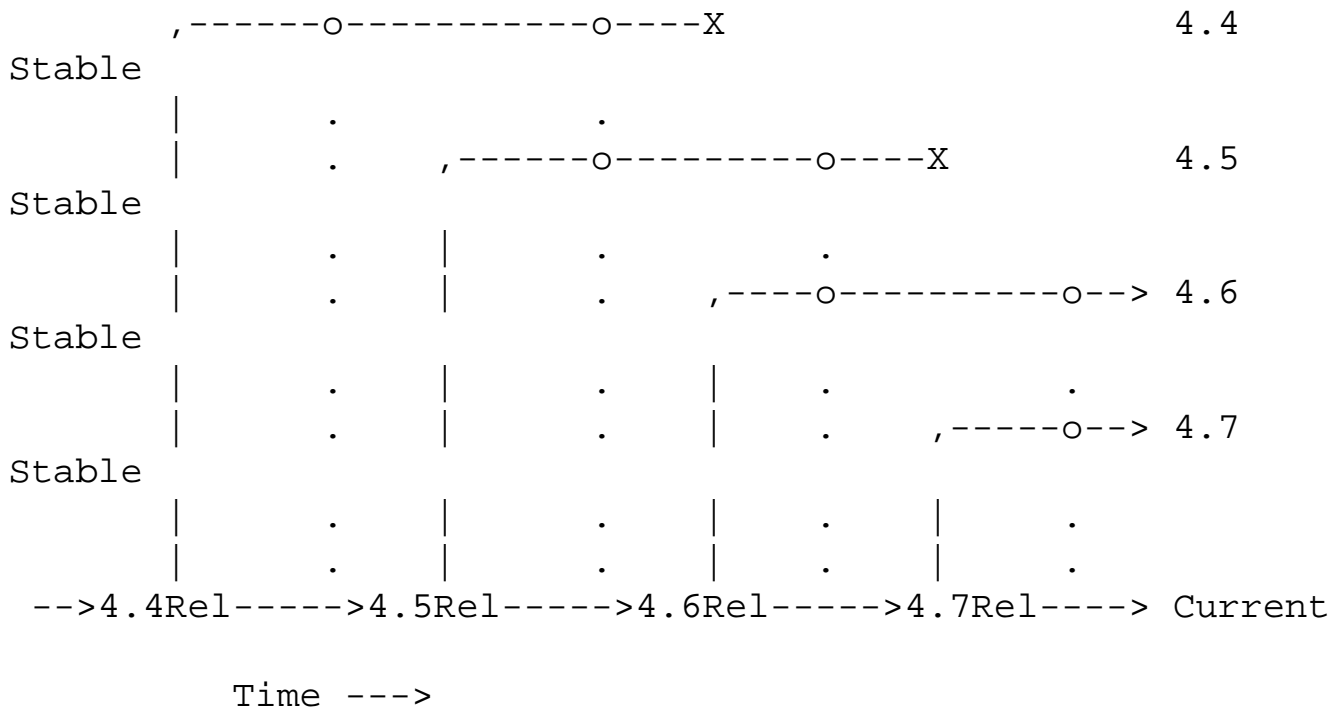
- [5.11.11 - Y a-t'il un moyen facile de faire des changements sur tous les fichiers de la hiérarchie ?](#)
- [5.11.12 - Puis-je cross-compiler ? Pourquoi pas ?](#)

5.1 - Les Saveurs ("Flavors") OpenBSD

Il existe trois "saveurs" OpenBSD :

- **-release:** La version d'OpenBSD livrée tous les six mois sur CD.
- **-stable:** "release", plus les correctifs considérés comme critiques pour la sécurité et la fiabilité.
- **-current:** La version courante pour laquelle le développement est réalisé, et qui deviendra la prochaine "release".

Schématiquement, le développement de ces versions ressemble à ceci :



-*Current*: La version courante pour laquelle le développement est réalisé, et qui deviendra la prochaine *-release* d'OpenBSD. Tous les six mois, lorsqu'une version d'OpenBSD est révisée, *-current* est taguée, et devient *-release* : un point gelé dans l'histoire de l'arbre des sources. Chaque *-release* demeure inchangée; c'est pour cela qu'on la trouve sur les [CDs](#) et [serveurs FTP](#).

-*Stable* est basée sur *-release*, et est une branche du chemin de développement principal d'OpenBSD. Quand des corrections très importantes sont faites sur *-current*, elles sont "backportées" (intégrées) aux

branches *-stable*; à cause de cela, *-stable* est aussi connue sous le nom de *branche patch*. Dans l'illustration ci-dessus, la ligne verticale en pointillés symbolise les corrections de bogues incorporées aux branches *-stable*. Vous remarquerez aussi dans l'exemple ci-dessus que la branche *4.4-stable* a été supprimée à la sortie de *4.6-release*, et que la branche *4.5-stable* a été supprimée à la sortie de *4.7-release* -- les anciennes versions sont typiquement supportées durant deux "releases" au maximum. Le support d'anciennes versions nécessite des ressources et du temps, et alors que nous pourrions vouloir plutôt fournir un support continu pour les anciennes versions, nous préférons nous concentrer sur les nouvelles fonctionnalités. La branche *-stable* est, par conception, très facile à construire à partir de *-release* de la même version (c'est à dire en allant de *4.7-release* vers *4.7-stable*).

La branche *-stable* est *-release* à laquelle on a ajouté les correctifs listés dans la [page des errata](#). Habituellement, le fonctionnement de *-stable* est le même que celui de *-release* sur laquelle elle est basée. Si les [pages de manuel](#) doivent être modifiées, il est très probable que ces modifications ne fassent pas partie de *-stable*. En d'autres termes, le support de nouveaux périphériques NE sera pas ajouté à *-stable*.

Il est utile de préciser que le nom "*-stable*" ne signifie pas pour autant que *-current* est instable. *-current* change et évolue, alors que *-stable* ne change que très peu (tant au niveau des APIs qu'au niveau opérationnel), et vous n'aurez pas à apprendre à nouveau votre système, à changer des fichiers de configuration ou à ajouter de nouvelles applications à votre système.

En fait, notre préoccupation étant de faire continuellement évoluer OpenBSD, le but avoué est de rendre *-current* plus fiable, plus sécurisée, et bien sûr offrant de meilleurs outils que *-stable*. Tout simplement, la "meilleure" version d'OpenBSD est *-current*.

La plupart des utilisateurs devraient utiliser *-stable* ou *-release*. Cela dit, plusieurs personnes utilisent *-current* sur des systèmes en production, et il est important que les personnes fassent cela pour identifier des bogues et tester les nouvelles fonctionnalités. Cependant, si vous ne savez pas comment décrire, diagnostiquer et gérer proprement un problème, ne vous dites pas (ou à quelqu'un d'autre) que vous êtes entrain d'"aider le projet" en utilisant *-current*. "Ça ne marche pas !" n'est pas un [rapport de bogue utile](#). "Les changements récents dans le pilote pciide ont brisé la compatibilité avec mon interface IDE basée sur Slugchip, ci-joint le dmesg d'un système fonctionnel et un système ne fonctionnant pas..." peut être un rapport utile.

Des fois, les utilisateurs "normaux" veulent disposer des derniers développements et utiliser *-current*. La raison la plus commune de faire cela est que l'utilisateur possède un périphérique qui n'est pas supporté par *-release* (et donc, par *-stable* non plus), ou qu'il souhaite utiliser une nouvelle fonctionnalité de *-current*. Dans ce cas, soit l'utilisateur utilise *-current* soit il n'utilise pas le périphérique, et utiliser *-current* est peut-être l'option la plus logique. Cependant, il ne faut pas espérer que les développeurs vous tiennent la main.

Snapshots

Entre les versions formelles d'OpenBSD, des *snapshots* sont mis à disposition sur les [sites FTP](#). Comme le nom l'indique, ce sont des images du code dans l'arborescence à l'instant où le créateur de l'image a pris une copie du code pour une plate-forme donnée. Il est à noter que, pour certaines plates-formes, il peut s'écouler des jours entiers avant que l'image ne soit complètement construite et mise à disposition. Aucune garantie n'est donnée quant au bon fonctionnement ou à la possibilité d'installer des snapshots. Souvent, une modification qui a besoin d'être testée peut enclencher le processus de création des snapshots. Quelques plates-formes ont des snapshots construits pratiquement tous les jours, d'autres en ont beaucoup moins fréquemment. Si vous souhaitez utiliser *-current*, un snapshot récent est tout ce dont vous aurez besoin, et mettre à jour un snapshot est un point de départ nécessaire avant de tenter de compiler *-current* depuis les sources.

Parfois, on demande s'il y a un moyen d'obtenir une copie exacte du code qui a servi à construire un snapshot. La réponse est non. Primo, il n'y a aucun intérêt. Secundo, les snapshots sont construits selon le souhait des développeurs, lorsque le planning le permet, et lorsque des ressources sont disponibles. Sur les plates-formes rapides, il est possible de créer plusieurs snapshots en un jour. Sur les plates-formes lentes, la création d'un snapshot peut durer une semaine ou plus. Fournir des balises ou des marques dans l'arborescence des sources pour chaque snapshot peut s'avérer peu pratique. Tertio, les snapshots contiennent souvent du code expérimental non encore soumis dans l'arborescence

Mise à jour majeure vs. mise à jour mineure

Il existe deux types de mises à jour sous OpenBSD : des mises à jour majeures et des mises à jour mineures. Il existe des différences notables entre ces deux types de mises à jour que nous allons tâcher d'expliquer ci-après.

Une **mise à jour majeure** consiste à installer une nouvelle version d'OpenBSD, qui inclut généralement de nouvelles fonctionnalités. Par exemple, le passage d'une version 4.6 à une version 4.7 ou la mise à jour d'un snapshot du 12 juin avec un snapshot du 24 juin sont considérés comme mises à jour majeures. Lors d'une mise à jour majeure, vous devez typiquement consulter [Suivre la version de développement "-current"](#) ou le [Guide de Mise à niveau d'OpenBSD](#) (lors du passage d'une version à une plus récente) pour effectuer les modifications nécessaires au bon fonctionnement de la nouvelle version d'OpenBSD. Une **mise à jour mineure** consiste à appliquer des correctifs à un système afin d'améliorer son fonctionnement SANS modification des fonctionnalités de base ou de la compatibilité binaire. Ceci est généralement effectué en suivant le processus d'[Application des correctifs sous OpenBSD](#) ou en suivant la procédure décrite dans le document intitulé [Maintenir son système à niveau par rapport à -stable](#). Quand vous effectuez une mise à jour mineure, votre système passe d'un état *-release* à un état *-stable* (ou *-release* corrigé) de la même version. Par exemple, de 4.7-release à 4.7-stable. Vous pouvez ensuite effectuer une nouvelle mise à jour mineure vers un *-stable* plus récent de la même version. Le processus de mise à jour est généralement sans conséquence étant donné que les fichiers `/etc` ou d'autres configurations système n'ont pas besoin d'être modifiés.

Ainsi, vous pouvez installer un système (tel que 4.5-release) à partir du CD, puis vous pourrez le mettre à jour un certain nombre de fois en 4.5-stable et par la suite effectuer une mise à jour majeure vers 4.6-release et ainsi de suite.

Garder les Composants Synchronisés

Il est important de comprendre qu'OpenBSD est un Système d'Exploitation, et il faut le prendre en tant que tel et non pas comme un noyau entouré d'un ensemble d'outils. Vous devez vous assurer que votre noyau, le "userland" (les utilitaires et fichiers complétant le noyau) et l'arborescence des [ports](#) sont synchronisés, autrement des choses désagréables peuvent arriver. Dit autrement (vu que les gens continuent à commettre les mêmes erreurs), vous ne pouvez pas utiliser des `ports` tout neufs sur un système datant d'il y a un mois, ou reconstruire un noyau à partir de `-current` et espérer qu'il fonctionne avec un "userland" `-release`. Oui, cela veut dire que vous aurez besoin d'effectuer une mise à jour majeure de votre système si vous voulez utiliser un nouveau programme qui a été rajouté aujourd'hui à l'arborescence des ports. OpenBSD n'a malheureusement que des ressources limitées.

Il faut aussi comprendre que le processus de mise à jour majeure est uniquement supporté **dans une seule direction uniquement : de l'ancien au nouveau**, et de `-stable` vers `-current`. Vous ne pouvez pas utiliser `4.7-current` (ou un snapshot), puis décider que c'est trop dangereux, et revenir vers `4.7-stable`. Vous ne pouvez compter que sur vous-même si vous choisissez un autre chemin que celui, supporté, consistant à réinstaller votre système proprement. Vous ne devez espérer aucune aide de la part de l'équipe de développement OpenBSD.

Oui, cela veut dire que vous devez prendre le temps de réfléchir avant d'utiliser `-current`.

5.2 - Pourquoi devrais-je compiler mon système depuis les sources ?

Actuellement, il y a de fortes chances pour que vous n'en ayez pas besoin.

Voici quelques raisons pour NE PAS compiler votre systèmes depuis les sources :

- La compilation de votre propre système à des fins de mise à niveau n'est pas supportée.
- Vous n'obtiendrez pas de meilleurs performances en compilant votre système.
- Le changement des options de compilation est plus enclin à briser votre système qu'à l'améliorer.

Voici quelques raisons d'avoir besoin de compiler depuis les sources :

- Tester ou développer de nouvelles fonctionnalités.
- La compilation du système est éprouvante pour l'ordinateur, ceci peut être un bon moyen de s'assurer du bon fonctionnement de ce dernier.

- Vous souhaitez suivre la branche [stable](#).
- Vous souhaitez construire un système OpenBSD hautement personnalisé pour des utilisations spécifiques.

L'équipe OpenBSD réalise fréquemment des instantanés basés sur le code de *-current* pour toutes les plates-formes. Dans la plupart des cas, cela vous suffira pour utiliser *-current*.

La raison la plus courante de compiler depuis les sources est de suivre la branche *-stable*, pour laquelle la compilation depuis les sources est la seule issue supportée.

5.3 - Compilation d'OpenBSD depuis les sources

5.3.1 - Aperçu du processus de compilation

La compilation d'un système OpenBSD depuis les sources implique un certain nombre d'étapes :

- [5.3.2 - Installer ou mettre à niveau depuis les binaires les plus récents](#)
- [5.3.3 - Téléchargement des sources appropriées](#)
- [5.3.4 - Compilation du noyau](#)
- [Compilation du "userland" \("make build"\)](#).

Il y a deux étapes supplémentaires que certains utilisateurs pourraient vouloir réaliser, ceci dépend de la raison de la compilation et de la présence ou non de X :

- [Compilation d'une Release](#).
- [Compilation de X](#).

5.3.2 - Installer ou mettre à niveau depuis les binaires les plus récents

La première étape dans la compilation depuis les sources est de s'assurer d'avoir le binaire le plus récent installé. Utilisez ce tableau afin de savoir où vous en êtes, où vous voulez aller, et avec quel binaire commencer :

Vous êtes au	But	Mise à niveau binaire vers	ensuite ...
Ancienne -release	Nouvelle release	Dernière release	C'est fait !
-release	-stable	Dernière release	Récupérer & compiler <i>-stable</i>
Ancienne -stable	-stable	Nouvelle release	Récupérer & compiler <i>-stable</i>

-release	-current	Dernier instantané	Récupérer (optionnel) & compiler - <i>current</i>
Ancienne - current	-current	Dernier instantané	Récupérer (optionnel) & compiler - <i>current</i>

Il est recommandé d'installer le binaire via l'option "Upgrade" du média d'installation. Si ce n'est pas possible, vous pouvez aussi décompacter les binaires comme décrits [ici](#). Vous pouvez faire le processus complet de mise à niveau sans soucis, incluant la création d'utilisateurs ou d'autres changements de répertoires de /etc.

5.3.3 - Téléchargement des sources appropriées

Le source d'OpenBSD est géré en utilisant le système de contrôle de versions [CVS](#), et [cvs\(1\)](#) est utilisé pour récupérer les sources désirées sur votre machine locale, à des fins de compilation. Ceci peut être réalisé en utilisant le serveur [AnonCVS](#) (une machine proposant une copie du dépôt CVS utilisé par le projet OpenBSD) et ce publiquement, ou depuis une archive locale que vous maintenez en utilisant les programmes [CVSup](#), ou [CVSync](#), disponibles dans les [paquetages](#). CVSup peut aussi être utilisé en mode "checkout", mais cela n'est pas couvert par le présent document. Si vous avez plusieurs machines sur lesquelles vous désirez maintenir le code source, vous devriez trouver utile d'avoir un dépôt CVS local, créé et maintenu en utilisant CVSup ou CVSync.

Après avoir décidé quel [Serveur AnonCVS](#) vous allez utiliser, vous devez effectuer un "checkout" (récupération) de l'arbre des sources, que vous maintiendrez par la suite en lançant des "updates" (mises à jour), afin de récupérer les fichiers les plus récents dans votre arbre local.

La commande [CVS\(1\)](#) dispose de nombreuses options, certaines d'entre elles sont *requis* afin de récupérer et de mettre à jour un arbre. Les autres commandes peuvent résulter en un arbre cassé. Le fait de comprendre et de suivre les directions est ici important.

Suivis de *-current*

Dans ce cas, nous supposons que vous utilisez un serveur AnonCVS publique, anoncvs@anoncvs.example.org:/cvs. Nous supposons également que vous utilisez le shell [sh\(1\)](#), vous devrez dans le cas contraire ajuster certaines commandes.

Pour récupérer un arbre des sources CVS de *-current*, vous pouvez utiliser la commande suivante :

```
# cd /usr
# export CVSROOT=anoncvs@anoncvs.example.org:/cvs
```



```
# cvs -d$CVSROOT checkout -P src
```

Une fois l'arbre récupéré, vous pourrez plus tard le mettre à jour avec :

```
# cd /usr
# export CVSROOT=anoncvs@anoncvs.example.org:/cvs
# cvs -d$CVSROOT up -Pd
```

Suivis de *-stable*

Si vous souhaitez récupérer une "branche" alternative de l'arbre, comme la branche *-stable*, vous devez utiliser le modificateur "-r" :

```
# cd /usr
# export CVSROOT=anoncvs@anoncvs.example.org:/cvs
# cvs -d$CVSROOT checkout -rOPENBSD_4_7 -P src
```

Ceci aura pour effet de récupérer les fichiers sources de la branche OPENBSD_4_7, aussi connue sous le nom de "Branche patchée" ou "[-stable](#)". Vous pourrez mettre à jour le code de façon similaire :

```
# cd /usr/src
# export CVSROOT=anoncvs@anoncvs.example.org:/cvs
# cvs -d$CVSROOT up -rOPENBSD_4_7 -Pd
```

CVS est vraiment agréable car il permet de suivre une balise dans les fichiers récupérés, vous n'avez ainsi pas à vous rappeler de la partie "-rOPENBSD_4_7" de la ligne de commande, ceci sera mémorisé jusqu'à ce que vous l'effaciez ou en spécifiez un autre via l'option "-A" de "update". Cependant, il est probablement plus correct de fournir trop d'infos que pas assez sur vos lignes de commande CVS.

L'arbre "src" ayant le seul à avoir été montré jusqu'à présent, vous pouvez faire les mêmes étapes pour "xenocara" et "ports". Toutes les parties d'OpenBSD devant être en synchronisation, tous les arbres utilisés devraient être mis à jour en même temps. Vous pouvez allier plusieurs récupérations sur une seule ligne (*-stable* est montré) avec :

```
# export CVSROOT=anoncvs@anoncvs.example.org:/cvs
# cd /usr
# cvs -d$CVSROOT checkout -rOPENBSD_4_7 -P src ports xenocara
```

Cependant, les mises à jour doivent être faites répertoire par répertoire.

A ce niveau, que vous ayez suivi *-stable* ou *-current* vous devriez avoir un arbre des sources correct. Soyez attentif à ce que vous récupérez -- vous pourriez compiler *-current* en pensant compiler *-stable*.

Pré-chargement de l'arbre des sources : `src.tar.gz`, `sys.tar.gz`

Au lieu de télécharger l'arbre des sources dans sa totalité à partir d'un serveur AnonCVS, vous pouvez le "pré-charger" à partir des fichiers des sources se trouvant sur le CD d'OpenBSD ou sur les serveurs FTP. Vous économiseriez ainsi beaucoup de temps et de bande passante. Ceci est particulièrement vrai si vous utilisez [-stable](#), étant donné le peu de différences entre cette version et *-release*.

Pour extraire l'arbre des sources à partir du CD vers `/usr/src` (en supposant que le CD est monté dans `/mnt`) :

```
# cd /usr/src; tar xzf /mnt/src.tar.gz
# cd /usr; tar xzf /mnt/xenocara.tar.gz
# cd /usr; tar xzf /mnt/ports.tar.gz
```

Les fichiers sources téléchargeables depuis les serveurs FTP sont séparés en deux fichiers pour minimiser le temps nécessaire à leur téléchargement pour les personnes souhaitant travailler avec telle ou telle partie de l'arbre uniquement. Ces deux fichiers sont `sys.tar.gz`, qui contient les fichiers utilisés pour créer le noyau, et `src.tar.gz` qui contient toutes les autres applications "userland", hormis l'arbre des ports et les sources X11. Cependant, et de manière générale, vous aurez besoin des deux. En supposant que vous les ayez téléchargé `src.tar.gz` et `sys.tar.gz` dans `/usr` :

```
# cd /usr/src
# tar xzf ../sys.tar.gz
# tar xzf ../src.tar.gz
# cd /usr
# tar xzf xenocara.tar.gz
# tar xzf ports.tar.gz
```

L'extraction de toutes les parties de l'arbre source n'est pas obligatoire mais si on souhaite garder un système cohérent, il est conseillé de les extraire toutes.

Astuces CVS courantes

Comme indiqué précédemment, certaines options sont obligatoires afin d'obtenir un arbre `src` d'OpenBSD valide. L'option `-P` ci-dessus est l'une d'entre elle : elle "prune" (efface) les répertoires vides. Avec les années, des répertoires ont été créés puis supprimés, et les noms de ces répertoires sont parfois utilisés actuellement pour des fichiers. Sans l'option `-P`, votre arbre fraîchement récupéré NE compilera PAS.

Même chose pour l'option `-d` sur une commande 'update' -- elle crée les nouveaux répertoires ayant pu être ajoutés au dépôt depuis votre dernier "checkout". Pour réussir une mise à jour, il faut absolument utiliser les options `-Pd`

Les utilisateurs de CVS expérimentés auront pu se demander pourquoi le CVSROOT est précisé et utilisé dans cet exemple, alors que cvs(1) enregistre la situation du serveur dans l'arbre ainsi obtenu. Ceci est correct, cependant, un utilisateur devra souvent outrepasser le serveur enregistré, et de nombreuses personnes recommandent de *toujours* spécifier le dépôt à utiliser. Il est aussi à noter que si la variable d'environnement CVSROOT peut être utilisée directement par cvs(1), elle est utilisée uniquement si rien ne viens la remplacer, et la spécification de la ligne de commande est prépondérante.

Il est souvent utile d'utiliser un `.cvsrc` dans votre répertoire home afin de spécifier les options par défaut. Un fichier `.cvsrc` d'exemple :

```
$ more ~/.cvsrc
cvs -q -danoncvns@anoncvns.example.org:/cvs
diff -up
update -Pd
checkout -P
```

Ce fichier ordonnera à cvs(1) d'utiliser le serveur `anoncvns@anoncvns.example.org:/cvs`, supprimant les sorties souvent inutiles ("`-q`" pour "quiet", tranquille) pour toutes les opérations, la commande "cvs up" utilisant par défaut `-Pd`, la commande "cvs diff" réalisant par défaut des "diffs unifiés" suite à l'option "`-u`", et un "cvs checkout" utilisera l'option "`-P`". Tandis que cela est confortable, si vous oubliez que ce fichier existe, ou si vous essayez de lancer ces commandes auxquelles vous êtes habitué sans ce fichier, vous rencontrerez des problèmes.

L'arbre des sources est constitué d'un grand nombre de petits fichiers. Il est donc conseillé d'activer [soft updates](#) sur la partition où se trouve cet arbre afin d'améliorer significativement les performances.

5.3.4 - Compilation du noyau

Nous supposons que vous désirez construire un noyau standard (GENERIC ou GENERIC.MP). Normalement, c'est ce que vous devriez faire. *N'essayez pas de construire un noyau personnalisé si vous ne maîtrisez pas le processus de compilation standard.*

Evidemment, le noyau est un composant TRES dépendant du matériel du système. Les sources du noyau sont dans le répertoire `/usr/src/sys`. Certaines parties du code du noyau d'OpenBSD sont utilisées sans distinction de plate-forme, alors que d'autres sont très spécifiques à un processeur ou une architecture. Si vous regardez dans le répertoire `/usr/src/sys/arch/`, vous verrez certaines choses quelque peu intrigantes -- par exemple, il y a des répertoires `mac68k`, `m68k` et `mvme68k`. Dans ce cas,

les systèmes mvme68k et mac68k utiliseront tous deux le même processeur, mais les machines sur lesquelles ils sont basés sont très différentes, et elles demandent ainsi un noyau très différent (il y a bien plus de choses qu'un processeur dans un ordinateur !). Cependant, certaines parties du noyau sont communes, elles demeurent dans le répertoire m68k. Si vous compilez simplement un noyau, les répertoires de l'architecture de base comme m68k ne sont pas un soucis, vous devriez travailler uniquement avec les répertoires de "l'architecture composant", comme mvme68k.

Les noyaux compilés sont basés sur les [fichiers de configuration du noyau](#), qui se trouvent dans le répertoire `/usr/src/sys/arch/<votre plate-forme>/conf`. La compilation du noyau consiste à l'utilisation du programme [config\(8\)](#) pour créer et peupler un répertoire compile, qui se terminera dans `/usr/src/sys/arch/<votre plate-forme>/compile/<nom du noyau>`. Pour cet exemple, nous supposons que vous utilisez la plate-forme i386 :

```
# cd /usr/src/sys/arch/i386/conf
# config GENERIC
# cd ../compile/GENERIC
# make clean && make depend && make
  [...beaucoup de sortie...]
# make install
```

Remplacez "i386" sur la première ligne par le nom de votre plate-forme. La commande [machine\(1\)](#) peut vous donner le nom de la plate-forme sur laquelle vous êtes afin de pouvoir utiliser "`cd /usr/src/sys/arch/`machine`/conf`" à la place de la première ligne.

A ce stade, redémarrez votre machine afin d'activer le nouveau noyau. Notez que le nouveau noyau devrait être lancé avant l'étape suivante, ce que vous savez si vous avez suivis les explications [ci-dessus](#) sur la mise à niveau vers les instantanés les plus récents. Parfois, les APIs changent cependant, et l'ancien noyau sera dans l'incapacité de lancer de nouvelles applications, mais les nouveaux noyaux supportent en général les applications plus anciennes.

Variation du processus ci-dessus : arbre des sources en lecture seule

Parfois, vous voudrez peut-être vous assurer que `/usr/src/sys` reste intacte. Ceci peut être fait en utilisant les procédés suivants :

```
$ cd /somewhere
$ cp /usr/src/sys/arch/i386/conf/GENERIC .
$ config -s /usr/src/sys -b . GENERIC
$ make clean && make depend && make
  ... beaucoup de messages affichés ...
```

Remarquez que vous pouvez compiler un noyau sans accès root, mais vous devez être root pour pouvoir

l'installer.

5.3.5 - Compilation du "userland"

Il y a une marche à suivre spécifique afin que cela fonctionne, sans quoi vous vous amusez à trouver d'où les problèmes viennent.

- Nettoyez votre répertoire `/usr/obj` et reconstruisez les liens symboliques :

```
# rm -rf /usr/obj/*
# cd /usr/src
# make obj
```

Remarquez que l'utilisation du répertoire `/usr/obj` est obligatoire. Echouer à cette étape avant la compilation laissera votre arbre `src` dans un mauvais état.

- Soyez sûr que tous les répertoires nécessaires sont présents.

```
# cd /usr/src/etc && env DESTDIR=/ make distrib-dirs
```

- Compilez le système :

```
# cd /usr/src
# make build
```

Ceci compile et installe tous les outils du "userland" dans un ordre approprié. Cette étape est très coûteuse en temps -- une machine très rapide devrait la réaliser sous une heure, une machine très lente pourrait mettre plusieurs jours. Lorsque cette étape est terminée, vous avez donc de nouveaux binaires à leurs places sur votre système.

- **Si vous compilez *-current*** : Mettez à jour `/dev` et `/etc`, avec les changements listés dans [current.html](#). Si vous suivez *-stable* d'après le [processus de mise à niveau](#) ou via une installation du [binaire correspondant](#), cette étape n'est ni désirée ni nécessaire.

5.4 - Compilation d'une Révision

Qu'est ce qu'une "release" (révision), et pourquoi voudrais-je en créer une ?

Une "release" est le jeu de fichiers complet pouvant être utilisé pour installer OpenBSD sur un ordinateur. Si vous n'avez qu'une seule machine sous OpenBSD, vous n'avez pas de réelle motivation à

construire une release, le processus [ci-dessus](#) vous apportant tout ce dont vous avez besoin. Un exemple d'utilisation du processus de release pourrait être de compiler une *-stable* sur une machine puissante, et de faire une release installable sur tous vos ordinateurs.

Le processus de release utilise les binaires créés dans le répertoire `/usr/obj` lors du processus `ci-dessus`, vous devez donc accomplir celui-ci au préalable, et rien ne doit perturber le répertoire `/usr/obj`. Ceci peut être un problème si vous utilisez un [disque mémoire](#) pour `/usr/obj` avec de petites performances lors du processus de compilation, vous ne voudrez pas redémarrer l'ordinateur entre les étapes de compilations et de "release" !

Le processus de release requiert deux répertoires de travail, appelés `DESTDIR` et `RELEASEDIR`. Tous les fichiers faisant partie d'une installation OpenBSD "propre" seront copiés à l'endroit correct au sein de `DESTDIR`. Ils seront `tar(1)`és puis placés dans `RELEASEDIR`. A la fin du processus, `RELEASEDIR` arborera la release OpenBSD intégralement. Le processus de release utilisera aussi `/mnt`, ce point ne doit donc pas être utilisé par autre chose pendant le processus de release. A titre d'exemple, nous utiliserons `DESTDIR` à `/usr/dest` et `RELEASEDIR` à `/usr/rel`.

Le processus de release utilisera un utilitaire, [crunchgen\(8\)](#), qui est utilisé pour créer un exécutable unique composé de plusieurs binaires. Le nom qui l'invoque détermine quel composant binaire est utilisé. C'est un peu comme si plusieurs fichiers de programmes individuels étaient concentrés sur un ramdisk noyau qui existe sur les disquettes et autres médias d'installation.

Vous devez avoir les privilèges root pour créer une release.

Faire une release

Définissons nos variables d'environnement `DESTDIR` et `RELEASEDIR` :

```
# export DESTDIR=/usr/dest
# export RELEASEDIR=/usr/rel
```

Nettoyons `DESTDIR` et créons le répertoire :

```
# test -d ${DESTDIR} && mv ${DESTDIR} ${DESTDIR}.old && rm -rf
${DESTDIR}.old &
# mkdir -p ${DESTDIR} ${RELEASEDIR}
```

`RELEASEDIR` ne doit pas forcément être vide lors du lancement du processus mais, cependant, s'il y a des changements dans les fichiers de la release ou dans leurs noms, les anciens fichiers seront conservés. Vous voudrez ainsi certainement effacer ce répertoire.

Passons à présent à la release elle-même :

```
# cd /usr/src/etc
# make release
```

Une fois la release construite, il peut être bon de la vérifier afin d'être sûr que les fichiers tar reflètent bien le contenu de DESTDIR. La sortie de cette étape devrait être très courte :

```
# cd /usr/src/distrib/sets
# sh checkflist
```

Vous avez à présent un jeu de fichiers complet et fonctionnel dans RELEASDIR. Ces fichiers peuvent être utilisés afin d'installer ou de mettre à jour OpenBSD sur d'autres machines.

Les instructions faisant foi sur la création d'une release sont dans [release\(8\)](#).

Remarque : Si vous souhaitez distribuer les fichiers résultants par HTTP pour être utilisables par les scripts de mise à jour ou d'installation, vous aurez besoin d'ajouter un fichier "index.txt". Ce fichier contient la liste de tous les fichiers constituant votre release fraîchement créée.

```
# /bin/ls -1 >index.txt
```

Dés que vous avez une release complète créée, vous pouvez utiliser ces fichiers pour une [installation](#) standard ou une [mise à jour](#) sur une autre machine, ou si vous mettez à jour une autre machine pour une nouvelle *-stable*, [décompressez](#) simplement les fichiers tar dans le répertoire racine de la machine cible.

5.5 - Compilation de X (Xenocara)

Commençons par [X.org v7](#), X est devenu un système à "compilation modulaire", découpant l'arborescence des sources de x.org en plus de trois cents packages plus ou moins indépendants.

Pour simplifier la vie des utilisateurs d'OpenBSD, un procédé de "meta-compilation" nommée [Xenocara](#) a été développé. Ce système "reconvertie" X en une seule grosse arborescence pour permettre la compilation en une seule étape. Comme point positif supplémentaire, ce processus de compilation se rapproche bien plus du processus utilisé pour le reste d'OpenBSD que la version précédente.

Les instructions officielles pour compiler X se trouvent dans le fichier `/usr/xenocara/README` ainsi que dans [release\(8\)](#).

Obtenir le code source

L'arborescence des sources de xenocara se trouve "habituellement" dans `/usr/src/xenocara`. Les sources sont stockées dans le module `xenocara` du CVS. Le checkout se fera donc ainsi :

```
$ cd /usr
$ cvs -qdanoncv@anoncv.example.org:/cvs checkout -P
xenocara
```

Compiler Xenocara

Aucun outil externe n'est nécessaire pour compiler l'arbre standard de xenocara telle que prévue par OpenBSD.

```
# cd /usr/xenocara
# rm -rf /usr/xobj/*
# make bootstrap
# make obj
# make build
```

Si vous voulez faire des modifications dans le code source, vous aurez probablement besoin d'ajouter plusieurs [packages](#). Les détails sont dans le fichier `/usr/xenocara/README`.

Création d'une release X

Ceci est similaire au processus de release principal. Après avoir compilé X avec succès, vous définirez `DESTDIR` et `RELEASEDIR`, de la même manière qu'évoqué précédemment. `RELEASEDIR` peut être le même répertoire que la principale `RELEASEDIR` système, mais `DESTDIR` sera effacé et reconstruit lors de ce processus. Si cela est fait avec attention, ce n'est pas un problème, mais l'utilisation d'une `DESTDIR` séparée est plus sûr.

Pour cet exemple, nous utiliserons les `DESTDIR` et `RELEASEDIR` comme `/usr/dest` et `/usr/rel`, respectivement. Ceci doit être fait après le processus de compilation ci-dessous.

```
# export DESTDIR=/usr/dest
# export RELEASEDIR=/usr/rel
# test -d ${DESTDIR} && mv ${DESTDIR} ${DESTDIR}- && \
  rm -rf ${DESTDIR}- &
# mkdir -p ${DESTDIR} ${RELEASEDIR}
# make release
```

Une fois le processus complété, vous aurez un ensemble de fichiers release dans le `$RELEASEDIR`

5.6 - Pourquoi aurais-je besoin d'un noyau sur mesure ?

En réalité, vous n'en avez très probablement pas besoin.

Un noyau sur mesure est un noyau construit à partir d'un fichier de configuration autre que `GENERIC`, le fichier de configuration fourni de base. Un noyau sur mesure peut se baser sur du code source [-release](#), [-stable](#) ou [-current](#) comme c'est le cas pour le noyau `GENERIC`. Alors que la compilation de votre propre noyau `GENERIC` est supportée par l'équipe OpenBSD, la compilation de votre propre noyau sur mesure *ne* l'est pas.

Le fichier de configuration noyau OpenBSD standard (`GENERIC`) est conçu pour convenir à la plupart des utilisateurs. Bon nombre de personnes ont rendu leur système inopérant en essayant d'optimiser le noyau au lieu d'améliorer son fonctionnement. Il existe certaines personnes qui pensent qu'un noyau et un système d'exploitation doivent être taillés sur mesure pour obtenir des performances optimales. Ceci n'est pas vrai dans le cas d'OpenBSD. Seules les personnes très compétentes avec des applications très particulières doivent penser à faire un noyau et un système sur mesure.

Voici quelques raisons pour lesquelles vous devriez créer un noyau sur mesure :

- Vous savez vraiment ce que vous faites, et vous souhaitez utiliser OpenBSD sur une machine disposant de peu de ressources mémoire en supprimant tous les pilotes de périphériques dont vous n'avez pas besoin.
- Vous savez vraiment ce que vous faites, et vous souhaitez supprimer des options par défaut ou ajouter des options qui ne sont pas activées par défaut (et vous avez vraiment une bonne raison pour le faire).
- Vous savez vraiment ce que vous faites, et vous souhaitez activer des options expérimentales.
- Vous savez vraiment ce que vous faites, et vous avez un besoin spécifique auquel le noyau `GENERIC` ne répond pas. Si quelque chose ne marche pas comme prévu, vous n'allez pas demander à autrui le pourquoi du comment.

Voici quelques raisons pour lesquelles vous ne devez pas compiler un noyau sur mesure :

- Vous n'en avez pas besoin en temps normal.
- Votre système n'en sera pas plus rapide.
- Vous rendrez probablement votre machine moins fiable.
- Vous n'obtiendrez aucune aide de la part des développeurs.
- Tout problème rencontré devra être obligatoirement reproduit avec un noyau `GENERIC` avant que les développeurs ne le prennent au sérieux.
- Les autres utilisateurs et les développeurs vous riront au nez si vous cassez votre système.
- D'habitude, des options de compilation sur mesure exposent les problèmes de compilateur au lieu d'améliorer les performances du système.

La suppression de pilotes pourrait rendre plus rapide la phase de démarrage système. Cependant, elle peut compliquer la récupération suite à problème matériel. La suppression de pilotes est une tâche très souvent mal réalisée. La suppression de pilotes *ne rendra pas* votre système plus rapide de manière perceptible même si elle peut produire un noyau plus petit. La suppression des parties liées au débogage et à la vérification d'erreurs peut améliorer les performances, mais rendra impossible l'analyse du système si quelque chose ne fonctionne plus ou pas.

Encore une fois, les développeurs ignorent d'habitude les rapports de bogues relatifs à des noyaux personnalisés, sauf si le problème peut être reproduit avec un noyau `GENERIC`. Vous aurez été prévenu.

5.7 - Options de configuration du noyau

Nous partons du fait que vous avez lu [ci-dessus](#), et que vous aimez la douleur. Il est aussi supposé que vous avez un but ne pouvant être atteint ni avec [Configuration au démarrage \(UKC<\)](#), ni avec [config\(8\) urer un noyau GENERIC](#). Si toutes les possibilités sont fausses, vous devriez vous en tenir à utiliser `GENERIC`. Vraiment.

La création d'un noyau OpenBSD est contrôlée par le biais de fichiers de configuration, se trouvant dans le répertoire `/usr/src/sys/arch/<arch>/conf/` par défaut. Toutes les architectures possèdent un fichier, `GENERIC`, qui peut être utilisé pour générer un noyau OpenBSD standard pour une plateforme donnée. Il peut aussi y avoir d'autres fichiers de configuration qui peuvent être utilisés pour créer des noyaux avec des objectifs différents tels que la minimisation de l'utilisation de la RAM, les stations de travail "diskless", etc.

Le fichier de configuration est traité par [config\(8\)](#), qui crée et peuple un répertoire de compilation situé sous `../compile`. Pour une installation typique, le chemin absolu du répertoire serait situé sous `/usr/src/sys/arch/<arch>/compile/`. `config(8)` peut aussi créer un fichier [Makefile](#), et d'autres fichiers requis pour créer avec succès un noyau.

Les options de configuration du noyau sont des options que vous ajoutez à la configuration de votre noyau pour activer certaines caractéristiques dans celui-ci. Ceci vous permet d'avoir exactement le support que vous voulez sans vous encombrer des pilotes inutiles. Il y a une multitude d'options qui vous permettront de personnaliser votre noyau. Veuillez consulter la page de manuel [options\(4\)](#) pour une liste complète des options. Vous pouvez aussi consulter les fichiers d'exemples de configurations qui sont disponibles pour votre architecture.

L'ajout, la suppression, ou la modification d'options dans votre noyau ne doivent être effectués que si vous avez une bonne raison pour le faire ! N'éditez pas le fichier de configuration `GENERIC` !! La seule configuration du noyau supportée par l'équipe OpenBSD est le noyau `GENERIC`, la combinaison d'options figurant dans les fichiers `/usr/src/sys/arch/<arch>/conf/GENERIC` et `/usr/src/sys/conf/GENERIC` tels que livrés par l'équipe OpenBSD (i.e. non

édités). Emettre un rapport de bogue concernant un noyau personnalisé va dans la plupart des cas se résumer à une réponse vous demandant d'essayer de reproduire le problème avec un noyau GENERIC. Les options ne sont pas toutes compatibles entre elles, et plusieurs options sont nécessaires au bon fonctionnement du système. Il n'y a aucune garantie quant au fonctionnement d'un noyau personnalisé que vous avez réussi à compiler. Il n'y a aucune garantie sur le fait qu'un noyau pouvant être "configuré" puisse être compilé.

Vous pouvez voir les fichiers de configuration spécifiques à une plate-forme donnée ici :

- [Fichiers de Configuration du Noyau alpha](#)
- [Fichiers de Configuration du Noyau i386](#)
- [Fichiers de Configuration du Noyau macppc](#)
- [Fichiers de Configuration du Noyau sparc](#)
- [Fichiers de Configuration du Noyau sparc64](#)
- [Fichiers de Configuration du Noyau vax](#)
- [Fichiers de Configuration du Noyau hppa](#)
- [Autres architectures](#)

Si vous lisez attentivement ces fichiers, vous verrez une ligne comme :

```
include "../.../.../conf/GENERIC"
```

Cela signifie que l'on fait référence à un autre fichier de configuration. Ce fichier comprend toutes les options qui ne sont pas dépendantes de l'architecture. Donc quand vous créez votre fichier de configuration, soyez sûr de regarder [/sys/conf/GENERIC](#) pour voir ce que vous voulez.

Toutes les options ci-dessous doivent être placées dans le fichier de configuration du noyau avec le format :

```
option      nom
```

ou

```
option      nom=valeur
```

Par exemple, pour utiliser l'option "DEBUG" dans le noyau, il faut mettre la ligne suivante :

```
option      DEBUG
```

Les options dans le noyau OpenBSD sont traduites en tant qu'options du préprocesseur, donc une option

telle que `DEBUG` compilerait les sources avec l'option `-DDEBUG`. Ce qui est équivalent à placer un `#define DEBUG` à travers les sources du noyau.

Vous aurez peut-être parfois besoin de désactiver une option précédemment définie dans le fichier `"src/sys/conf/GENERIC"` typiquement. Bien entendu, vous pouvez modifier une copie de ce fichier, mais une meilleure méthode consiste à utiliser la clause `rmoption`. Par exemple, si vous souhaitez vraiment désactiver le débogueur intégré au noyau (*non recommandé !*), vous ajouteriez la ligne suivante :

```
rmoption DDB
```

à votre fichier de configuration du noyau. `option DDB` est définie dans `src/sys/conf/GENERIC`, mais la ligne `rmoption` ci-dessus la désactive.

Encore une fois, veuillez consulter [options\(4\)](#) pour plus d'informations concernant les spécificités de ces options. Il est à noter que plusieurs options possèdent leurs propres pages de manuel -- il faut toujours lire toutes les informations disponibles au sujet d'une option avant de l'ajouter ou la supprimer de votre noyau.

Compiler un noyau sur mesure

Dans ce cas, nous compilerons un noyau supportant les cartes série ISA multi-port [boca\(4\)](#). Cette carte n'est pas dans le noyau `GENERIC`, du fait qu'elle entre en conflit avec d'autres drivers. Il y a deux moyens courants de faire un noyau personnalisé : copier le fichier de configuration `GENERIC` vers un autre nom et l'éditer, ou créer un fichier "wrapper" qui inclut le noyau standard `GENERIC` ainsi que toutes les options dont vous avez besoin, et qui ne sont pas dans `GENERIC`. Dans ce cas, votre fichier "wrapper" pourrait ressembler à :

```
include "arch/i386/conf/GENERIC"

boca0 at          isa? port 0x100 irq 10      # BOCA 8-port
serial cards
com* at          boca? slave ?
```

Les deux lignes au regard des cartes [boca\(4\)](#) sont copiées depuis les lignes commentées dans `GENERIC`, avec l'IRQ ajusté comme requis. L'avantage d'utiliser ce fichier "wrapper" est que les changements imprévus dans `GENERIC` sont automatiquement mis à jour comme lors de toute mise à jour du code. L'inconvénient est qu'on ne peut supprimer un périphérique (cependant, c'est en général une mauvaise idée).

L'autre moyen de générer un noyau personnalisé est de faire une copie du `GENERIC` standard, en lui donnant un autre nom, et en l'éditant selon les besoins. L'inconvénient est que les futures mises à jour du

fichier de configuration GENERIC devront être fusionnés dans votre copie, ou vous devrez refaire votre fichier de configuration.

Dans un autre événement, après avoir fait votre fichier de configuration personnalisé, utilisez `config(8)` et construisez le noyau comme documenté [ci-dessus](#).

Les instructions complètes pour la création de votre propre noyau sont dans la page de manuel [config\(8\)](#).

5.8 - Configuration au démarrage

Lorsque vous démarrez votre système il est possible que vous remarquiez que votre noyau trouve vos périphériques mais à la mauvaise IRQ. Et que vous aillez immédiatement besoin de ce périphérique. Sans recompiler votre noyau vous pouvez utiliser la configuration au démarrage de celui-ci. Cela vous permettra de corriger votre problème pour cette fois et uniquement pour cette fois. Si vous redémarrez le système il faudra recommencer la procédure. Il s'agit donc d'une méthode temporaire. Le problème devra être corrigé en utilisant [config\(8\)](#). De plus votre noyau à besoin de l'**option BOOT_CONFIG** dans la configuration du noyau. Le noyau GENERIC possède cette option.

Une grande partie de ce document peut-être trouvée dans la page de manuel [boot_config\(8\)](#).

Pour démarrer avec UKC (User Kernel Config), il faut spécifier l'option `-c` au démarrage.

```
boot> boot hd0a:/bsd -c
```

ou n'importe quel autre noyau que vous voulez démarrer. L'invite de commandes UKC apparaîtra, et vous pourrez spécifier au noyau les périphériques que vous désirez modifier, ceux que vous voulez activer ou désactiver.

Voici un liste des commandes les plus utilisées dans UKC.

- `add device` - Ajoute un périphérique en en copiant un autre
- `change devno | device` - Modifie un ou plusieurs périphériques
- `disable devno | device` - Désactive un ou plusieurs périphériques
- `enable devno | device` - Active un ou plusieurs périphériques
- `find devno | device` - Trouver un ou plusieurs périphériques
- `help` - Rapide sommaire de ces commandes
- `list` - Liste TOUS les périphériques connus
- `exit/quit` - Continue le démarrage
- `show [attr [val]]` - Montre les périphériques avec un attribut et une valeur spécifiée optionnelle

Une fois votre noyau configuré, utilisez `quit` ou `exit` et continuez le démarrage. Une fois votre système démarré, vous devriez rendre la modification permanente au niveau de votre binaire du noyau, tel que c'est décrit dans [utilisation de config\(8\) pour changer le binaire du noyau](#)

5.9 - Utilisation de config(8) pour changer le binaire du noyau

Les options `-e` et `-u` de [config\(8\)](#) peuvent être très utiles et vous évitent de perdre du temps à recompiler votre noyau. Le drapeau `-e` vous permet de rentrer en configuration UKC alors que le système fonctionne. Les changements prendront effets au prochain redémarrage. Le drapeau `-u` permet de voir si des changements ont été effectués au noyau pendant le démarrage, signifiant que vous avez utilisé `boot -c` pour entrer en configuration UKC.

Les exemples suivants montrent la désactivation des périphériques `ep*` dans le noyau. Pour plus de sécurité, il est préférable d'utiliser l'option `-o` qui écrira les changements dans un fichier spécifié. Par exemple : `config -e -o bsd.new /bsd` écrira les changements dans `bsd.new`. Les exemples n'utilisent pas l'option `-o`, mais les changements sont ignorés et ne sont pas écrits dans le binaire du noyau. Pour plus d'informations sur les messages d'erreur et autres avertissements, lisez la page de manuel [config\(8\)](#).

```
$ sudo config -e /bsd
OpenBSD 4.7 (GENERIC) #558: Thu Mar 17 20:46:15 MDT 2010
  deraadt@i386.openbsd.org:/usr/src/sys/arch/i386/compile/
GENERIC
warning: no output file specified
Enter 'help' for information
ukc> ?

      help                                Command help
list
      add                dev                Add a device
      base                8|10|16          Base on large
numbers
      change            devno|dev          Change device
      disable           attr val|devno|dev  Disable device
      enable            attr val|devno|dev  Enable device
      find              devno|dev          Find device
      list              List
configuration
      lines            count                # of lines per
page
      show             [attr [val]]        Show attribute
      exit              Exit, without
saving changes
      quit              Quit, saving
```

```

current changes
    timezone          [mins [dst]]          Show/change
timezone
    bufcachepersent  [number]              Show/change
BUFCACHEPERCENT
    nkmempg          [number]              Show/change
NKMEMPAGES
    shmseg           [number]              Show/change
SHMSEG
    shmmaxpgs        [number]              Show/change
SHMMAXPGS
ukc> list
    0 video* at uvideo* flags 0x0
    1 audio* at uaudio*|btsc0*|sb0|sb*|gus0|pas0|sp0|ess*|
wss0|wss*|ym*|eap*|envy*
|eso*|sv*|neo*|cmpci*|clcs*|clct*|auacer*|auglx*|auich*|
auixp*|autri*|auvia*|aza
lia*|fms*|maestro*|esa*|yds*|emu* flags 0x0
    2 midi* at umidi*|sb0|sb*|opl*|opl*|opl*|opl*|opl*|ym*|
mpu*|mpu*|autri*|eap* f
lags 0x0
    3 midi* at pcppi0 flags 0x0
    4 drm* at inteldrm*|radeondrm* flags 0x0
    5 inteldrm* at vga0|vga* flags 0x0
    6 radeondrm* at vga0|vga* flags 0x0
    7 radio* at udsbr*|bktr0|fms* flags 0x0
    8 softraid0 at root flags 0x0
    9 nsphy* at url*|udav*|mos*|axe*|aue*|xe*|ef*|hme*|lii*|
bce*|ale*|age*|jme*|et
*|nfe*|stge*|vge*|bnx*|bge*|lge*|nge*|msk*|sk*|ste*|sis*|
wb*|tl*|vr*|pcn*|sf*|ge
m*|ne0|ne1|ne2|ne*|ne*|ne*|epic*|sm0|sm*|dc*|dc*|re*|re*|
rl*|rl*|mtd*|fxp*|fxp*|
xl*|xl*|ep0|ep0|ep0|ep*|ep*|ep*|ep*|ep* phy -1 flags 0x0
    10 nsphyter* at url*|udav*|mos*|axe*|aue*|xe*|ef*|hme*|
lii*|bce*|ale*|age*|jme*
|et*|nfe*|stge*|vge*|bnx*|bge*|lge*|nge*|msk*|sk*|ste*|sis*|
wb*|tl*|vr*|pcn*|sf*
|gem*|ne0|ne1|ne2|ne*|ne*|ne*|epic*|sm0|sm*|dc*|dc*|re*|re*|
rl*|rl*|mtd*|fxp*|fx
p*|xl*|xl*|ep0|ep0|ep0|ep*|ep*|ep*|ep*|ep* phy -1 flags 0x0
[...snip...]
ukc> disable ep

```

```

98 ep0 disabled
99 ep* disabled
100 ep* disabled
278 ep0 disabled
279 ep0 disabled
280 ep* disabled
281 ep* disabled
344 ep* disabled
ukc> quit
not forced

```

Dans l'exemple ci-dessus, tous les périphériques `ep*` sont désactivés du noyau et ne seront donc pas testés. Dans certaines situations où vous aurez effectué ces changements au démarrage avec UKC et **boot -c**, il vous faudra les rendre définitifs. Pour ce faire, il faudra utiliser l'option **-u**. Dans l'exemple suivant, l'ordinateur a été démarré avec UKC et les périphériques [wi\(4\)](#) sont désactivés. Étant donné que les changements fait par `boot -c` ne sont pas permanents, ceux-ci doivent être écrits sur le disque. Cet exemple montre comment écrire les changements effectués par `boot -c` dans un nouveau noyau binaire `bsd.new`.

```

$ sudo config -e -u -o bsd.new /bsd
OpenBSD 4.7 (GENERIC) #58: Thu Mar 17 20:46:15 MDT 2010
  deraadt@i386.openbsd.org:/usr/src/sys/arch/i386/compile/
GENERIC
Processing history...
162 wi* disabled
163 wi* disabled
416 wi* disabled
Enter 'help' for information
ukc> quit

```

5.10 - Obtention d'une sortie plus verbeuse lors du démarrage

L'obtention d'une sortie plus verbeuse peut s'avérer être très utile lors des tentatives de résolution de problème au boot. Si vous avez un problème inhérent à votre disquette de boot et désirez obtenir plus d'informations, redémarrez. A l'arrivée au prompt "`boot>`", bootez avec l'option `-c`. Ceci vous permettra d'être dans UKC>, puis de faire :

```

UKC> verbose
autoconf verbose enabled
UKC> quit

```

Vous aurez à présent une sortie très verbeuse au démarrage.

5.11 - Problèmes courants, astuces et questions lors de la compilation et de la construction

La plupart du temps, les problèmes lors de la compilation sont causés par le fait de ne pas suivre précisément les indications précédentes. Cependant, il peut exister occasionnellement de véritables problèmes lors de la compilation de *-current* à partir du snapshot le plus récent mais les erreurs lors de la compilation de *-release* ou *-stable* sont presque toujours dues à une erreur de l'utilisateur.

Les problèmes les plus courants sont :

- Ne pas démarrer à partir du [binaire approprié](#), essayer de mettre à jour à partir des sources ou penser qu'un snapshot vieux d'une semaine est "suffisamment proche".
- [Récupérer](#) la mauvaise branche de l'arbre CVS.
- Ne pas suivre la [procédure](#).
- Tenter d'[adapter](#) ou d'"optimiser" votre système.

Voici tout de même certains problèmes auxquels vous pourriez être confronté :

5.11.1 - La compilation s'est arrêtée avec une erreur "Signal 11"

La compilation d'OpenBSD et d'autres programmes à partir des sources est une tâche qui sollicite le matériel plus que la plupart des autres opérations, faisant un usage intensif du CPU, du disque et de la mémoire. Par conséquent, si votre matériel a des problèmes, ces derniers apparaîtront plus facilement durant une compilation. Les défaillances "Signal 11" sont *typiquement* causées par des problèmes matériel, et la plupart du temps à cause de problèmes de mémoire. Mais ces défaillances peuvent aussi causer par le CPU, la carte mère, ou des problèmes de surchauffe. Votre système peut d'ailleurs être stable la plupart du temps et connaître des défaillances lors de la compilation de programmes.

Il est recommandé de réparer ou remplacer les composants à l'origine des défaillances; les problèmes pouvant se manifester sous d'autres formes plus tard. Si vous avez du matériel que vous souhaitez utiliser et qui ne vous pose aucun problème, installez simplement un snapshot ou une "révision".

Pour plus d'informations, consultez la [Faq Sig11](#).

5.11.2 - "make build" échoue en générant un message "cannot open output file snake: is a directory"

Ceci est le résultat de deux erreurs séparées :

- **Vous n'avez pas proprement récupéré ou mis à jour votre arborescence CVS.** Lorsque vous effectuez une opération "CVS checkout", vous devez utiliser l'option "-P", et lorsque vous mettez à jour votre arborescence des sources à partir de CVS, vous devez utiliser les options "-Pd" de [cvs\(1\)](#), comme documenté [ci-dessus](#). Ces options s'assurent que les nouveaux répertoires sont ajoutés ou supprimés au fur et à mesure qu'OpenBSD évolue.
- **Vous n'avez pas correctement créé le répertoire `obj` avant de commencer la compilation.** La compilation de l'arborescence sans répertoire `/usr/obj` n'est pas supportée.

Il est important de suivre soigneusement les instructions lors de la [récupération](#) et de la [compilation](#) de votre arbre.

5.11.3 Mon système sans IPv6 ne fonctionne pas !

Oui. Nous vous prions de ne pas faire de modifications au système de base pour lesquelles vous ne comprenez pas toutes les conséquences. Un "petit" changement dans le noyau peut avoir un impact très large sur le reste entier du système. Relisez s'il vous plaît [ceci](#).

5.11.4 - Oops ! J'ai oublié de créer le répertoire `/usr/obj` avant de commencer !

En faisant un "make build" avant de faire un "make obj", vous devriez en terminer avec les fichiers objets accumulés dans votre répertoire `/usr/src`. C'est une mauvaise chose. Si vous désirez essayer d'avoir à télécharger l'arbre des sources une nouvelle fois, vous pouvez essayer les manipulations suivantes afin de nettoyer les fichiers objets :

```
# cd /usr/src
# find . -type l -name obj | xargs rm
# make cleandir
# rm -rf /usr/obj/*
# make obj
```

5.11.5 - Conseil : Placer `/usr/obj` sur sa propre partition

Si vous compilez souvent, vous trouverez plus rapide de mettre `/usr/obj` sur sa propre partition. Le bénéfice est simple, il est typiquement plus rapide de :

```
# umount /usr/obj
# newfs VotrePartitionObj
# mount /usr/obj
```

que de faire "rm -rf /usr/obj".

5.11.6 - Comment ne pas compiler certaines parties de l'arbre ?

Vous souhaitez peut-être de pas compiler certaines parties de l'arbre, typiquement parce que vous l'avez remplacé par une application incluse dans un paquetage, ou parce que vous voulez créer une installation "plus petite" pour diverses raisons. La solution est d'utiliser l'option SKIPDIR de [/etc/mk.conf](#).

Note : il est de ce fait possible de compiler des systèmes cassés. Les résultats de cette option ne sont pas supportés par le projet OpenBSD.

5.11.7 - Ou puis-je avoir plus d'informations sur le processus de compilation ?

- [release\(8\)](#)
- [afterboot\(8\)](#)
- [mk.conf\(5\)](#)
- [/usr/src/Makefile](#)
- [Branches Patchées](#) (-stable)
- (pour X) [/usr/X11R6/README](#) sur votre système

5.11.8 - Je ne vois aucun snapshot sur le site FTP. Ou sont-ils passés ?

Les snapshots peuvent être retirés lorsqu'ils deviennent anciens (ou obsolètes) ou lorsque la sortie d'une *release* approche.

5.11.9 - Comment puis-je amorcer sur une nouvelle version du compilateur (gcc)?

Vous devriez vraiment [installer le dernier snapshot](#).

OpenBSD supporte désormais deux compilateurs dans l'arbre, gcc v3.3.5 par la plupart des plates-formes, et également gcc v2.95.3 utilisé par des plates-formes qui n'ont pas encore été converties, ou qui ne seront pas converties à cause des manques du support de gcc3 ou de ses performances pauvres.

Les deux compilateurs sont dans des parties différentes de l'arbre :

- gcc3: `/usr/src/gnu/usr.bin/gcc`
- gcc2: `/usr/src/gnu/egcs/gcc`

Parce que la mise à niveau du compilateur est un peu comme le problème du poulet et de l'oeuf, les changements sur le compilateur intégré à l'arbre demandent une attention toute particulière. Vous devez construire le compilateur deux fois -- la première construction produit un compilateur qui génère un nouveau code mais utilise l'ancien code pour fonctionner, la deuxième construction construit un nouveau compilateur intégralement. En général, vous voudrez réaliser la procédure suivante :

Si votre plate-forme utilise gcc 2.95.3 :

```
# rm -r /usr/obj/gnu/egcs/gcc/*
# cd /usr/src/gnu/egcs/gcc
- ou -
```

Si votre plate-forme utilise gcc 3.3.5 :

```
# rm -r /usr/obj/gnu/usr.bin/gcc/*
# cd /usr/src/gnu/usr.bin/gcc
```

Procédure commune de compilation pour v3.3.5 ou v2.95.3

```
# make -f Makefile.bsd-wrapper clean
# make -f Makefile.bsd-wrapper obj
# make -f Makefile.bsd-wrapper depend
# make -f Makefile.bsd-wrapper
# make -f Makefile.bsd-wrapper install
# make -f Makefile.bsd-wrapper clean
# make -f Makefile.bsd-wrapper depend
# make -f Makefile.bsd-wrapper
# make -f Makefile.bsd-wrapper install
```

Puis lancer un [make build](#) classique.

5.11.10 - Quel est le meilleur moyen de mettre à jour /etc, /var, et /dev ?

En tant que politique, les logiciels dans l'arbre OpenBSD ne modifient pas de fichiers dans /etc automatiquement. Cela signifie que les modifications à réaliser incombent *toujours* à l'administrateur. Les mises à niveau n'y font pas exception. Pour mettre à jour les fichiers dans ces répertoires, déterminez tout d'abord les changements qui ont été opérés sur les fichiers de la base (distribution) et répétez manuellement ces changements.

Par exemple, pour voir les fichiers dans l'arbre qui ont changé récemment, faites un :

```
# cd /usr/src/etc
# ls -lt |more
```

Pour voir tous les changements dans le répertoire /etc entre des versions arbitraires d'OpenBSD, vous pouvez utiliser [CVS](#). Par exemple, pour voir les changements entre 4.6 et 4.7, faites un :

```
# cd /usr/src/etc
# cvs diff -u -rOPENBSD_4_6 -rOPENBSD_4_7
```

Pour voir les changements entre 4.7 et *-current* ("HEAD"), utilisez :

```
# cd /usr/src/etc
# cvs diff -u -rOPENBSD_4_7 -rHEAD
```

Le script [/dev/MAKEDEV](#) n'est pas mis à jour automatiquement lors du processus de compilation, mais il est cependant installé comme partie de la [mise à jour binaire](#). En règle générale, c'est une bonne idée que de copier (si besoin est) et de lancer ce script depuis votre arbre des sources lors de la mise à niveau :

```
# cd /dev
# cp /usr/src/etc/etc.`machine`/MAKEDEV ./
# ./MAKEDEV all
```

Une fois que vous avez identifié les changements, appliquez à nouveau ceci à votre arbre local, en préservant toute configuration que vous auriez pu faire.

Les changements typiques de `/etc` à considérer, entre les release concernent :

- Additions à `/etc/protocols` et `/etc/services`
- Nouveaux `sysctls` (regardez `/etc/sysctl.conf`)
- Changements aux travaux de cron par défaut. Regardez `/etc/daily`, `/etc/weekly`, `/etc/monthly`, et `/etc/security`
- Tous les scripts `rc`, incluant `netstart`
- Les changements de périphériques, regardez ci-dessus
- Les changements de la hiérarchie des fichiers dans `/etc/mtree`, voyez [ci-dessus](#)
- Les nouveaux utilisateurs (`/etc/passwd`) et groupes (`/etc/group`)

Ces changements sont résumés dans [upgrade47.html](#) (pour aller vers 4.7-release) ou [current.html](#) (pour aller vers *-current*).

5.11.11 - Y a-t'il un moyen facile de faire des changements sur tous les fichiers de la hiérarchie ?

De temps en temps, des fichiers et répertoires sont ajoutés, ou supprimés dans le fichier [hierarchy](#). Aussi, le possesseur de l'information pour les portions du système de fichiers peut changer. Un moyen facile de s'assurer que votre hiérarchie est à jour est d'utiliser l'outil [mtree\(8\)](#).

Tout d'abord, procurez vous les dernières sources, en faisant :

```
# cd /usr/src/etc/mtree
# install -c -o root -g wheel -m 600 special /etc/mtree
# install -c -o root -g wheel -m 444 4.4BSD.dist /etc/mtree
# mtree -qdef /etc/mtree/4.4BSD.dist -p / -u
```

Votre hiérarchie de fichiers devrait à présent être à jour.

5.11.12 - Puis-je cross-compiler ? Pourquoi pas ?

Les outils de cross-compilation sont présent dans le système, et sont utilisés par les développeurs lors de la création d'un nouveau port. Cependant, ils ne sont pas maintenus pour une utilisation généralisée.

Lorsque les développeurs créent le support d'une nouvelle plate-forme, l'un des premiers gros tests est la compilation native. Compiler le système depuis les sources augmente très fortement la charge de l'OS et de la machine, et permet de voir à quel point le système fonctionne correctement. Pour cette raison, OpenBSD réalise les compilations sur les plates-formes que la compilation concerne, ceci est appelé le "native building". Sans compilation native, il serait bien plus difficile de s'assurer que les différentes plates-formes sont stables et qu'elles ne font pas que booter.

[\[Index de La FAQ\]](#) [\[Section 4 - Guide d'Installation\]](#) [\[Section 6 - Le réseau\]](#)



www@openbsd.org

\$OpenBSD: faq5.html,v 1.94 2010/05/23 16:20:41 ajacoutot Exp \$

OpenBSD

[\[Index de la FAQ\]](#) [\[Section 5 - Construire le Système à partir des Sources\]](#) [\[Section 7 - Contrôles du clavier et de l'affichage\]](#)

6 - Le réseau

Table des matières

- [6.1 - Avant d'aller plus loin](#)
 - [6.2 - Configuration du réseau](#)
 - [6.2.1 - Identifier et configurer vos interfaces réseau](#)
 - [6.2.2 - Passerelle par défaut](#)
 - [6.2.3 - Résolution DNS](#)
 - [6.2.4 - Nom d'hôte](#)
 - [6.2.5 - Activer les changements](#)
 - [6.2.6 - Vérifier les routes](#)
 - [6.2.7 - Mettre en place une passerelle OpenBSD](#)
 - [6.2.8 - Configurer les alias sur une interface](#)
 - [6.3 - Comment filtrer et utiliser un pare-feu sous OpenBSD ?](#)
 - [6.4 - Protocole d'attribution dynamique des adresses \(DHCP\)](#)
 - [6.4.1 - Client DHCP](#)
 - [6.4.2 - Server DHCP](#)
 - [6.5 - Protocole Point à Point \(PPP\)](#)
 - [6.6 - Optimisation des paramètres réseau](#)
 - [6.7 - Utilisation de NFS](#)
 - [6.9 - Mise en place d'un pont \("bridge"\) avec OpenBSD](#)
 - [6.10 - Comment démarrer en utilisant PXE ?](#)
 - [6.11 - Protocole de redondance d'adresse commune \(CARP\)](#)
 - [6.12 - Utiliser OpenNTPD](#)
 - [6.13 - Quels sont les types de cartes Sans Fil supportées par OpenBSD ?](#)
 - [6.14 - Comment puis-je mettre en oeuvre un routage multi-chemin à coût égal ?](#)
-

6.1 - Avant d'aller plus loin

Afin de mieux comprendre ce document, vous devriez lire et assimiler au moins partiellement la section [Construire le Système à partir des Sources](#) de la FAQ ainsi que le manuel [ifconfig\(8\)](#) et [netstat\(1\)](#).

Si vous êtes administrateur de réseau et que vous mettez en place des protocoles de routage, si vous utilisez OpenBSD en tant que routeur, ou si vous souhaitez en savoir plus sur les réseaux IP, vous devriez lire "[Understanding IP Addressing](#)" (Comprendre l'adressage IP). Il s'agit d'un excellent document. Vous pouvez vous appuyer sur celui-ci afin de vous aider à travailler sur les réseaux IP, surtout si vous en avez un ou plusieurs sous votre responsabilité.

Si vous travaillez sur des applications telles que des serveurs web, des serveurs ftp et des serveurs de messagerie, vous pourriez bénéficier de la [lecture des RFCs](#). A priori, vous ne pourrez pas toutes les lire. Choisissez les sujets qui vous intéressent ou les technologies que vous utilisez sur votre réseau. Lisez-les et voyez comment ces technologies fonctionnent. Les RFCs standardisent beaucoup (plusieurs milliers) de protocoles Internet et la façon dont ils sont censés fonctionner.

6.2 - Configuration du réseau

Normalement, OpenBSD est configuré initialement durant la procédure d'[installation](#). Cependant il est intéressant de comprendre ce qui se passe pendant cette procédure et la façon dont cela fonctionne. Toute la configuration réseau est effectuée en utilisant de simples fichiers texte sous le répertoire */etc*.

6.2.1 - Identifier et configurer vos interfaces réseau

Sous OpenBSD, les interfaces sont nommées en fonction du type de la carte réseau, et non en fonction du type de la connexion. Vous pouvez voir l'initialisation de votre carte pendant la procédure de démarrage, ou après celle-ci en utilisant la commande [dmesg\(8\)](#). Vous avez aussi la possibilité de voir votre carte réseau grâce à l'utilisation de la commande [ifconfig\(8\)](#). Voici par exemple la sortie de la commande `dmesg` pour une carte Intel Fast Ethernet qui utilise le nom de périphérique `fxp`.

```
fxp0 at pci0 dev 10 function 0 "Intel 82557" rev 0x0c: irq
5, address 00:02:b3:2b:10:f7
inphy0 at fxp0 phy 1: i82555 10/100 media interface, rev. 4
```

Si vous ne connaissez pas le nom du périphérique associé à votre carte, regardez dans la liste des [plates-formes actuellement supportées](#) par votre architecture. Vous pourrez y trouver le nom des cartes les plus courantes et leur équivalent sous OpenBSD. Ajoutez au nom alphabétique du périphérique (par exemple `fxp`) le numéro assigné par le noyau et vous aurez le nom de votre interface (par exemple `fxp0`). Le chiffre est attribué selon plusieurs critères dépendant du type de carte et d'autres détails du système.

Certaines cartes se voient assigner un chiffre selon l'ordre dans lequel elles ont été détectées au démarrage et d'autres selon la configuration des ressources matérielles ou de l'adresse MAC.

Vous pouvez connaître quelles interfaces réseaux ont été identifiées à l'aide de l'utilitaire [ifconfig\(8\)](#). La commande suivante montrera toutes les interfaces réseau d'un système. Cet exemple montre qu'il n'y a qu'une seule interface Ethernet physique : [fxp\(4\)](#).

```
$ ifconfig
lo0: flags=8049<UP,LOOPBACK,RUNNING,MULTICAST> mtu 33224
    inet 127.0.0.1 netmask 0xff000000
    inet6 ::1 prefixlen 128
    inet6 fe80::1%lo0 prefixlen 64 scopeid 0x5
lo1: flags=8008<LOOPBACK,MULTICAST> mtu 33224
fxp0: flags=8843<UP,BROADCAST,RUNNING,SIMPLEX,MULTICAST>
mtu 1500
    address: 00:04:ac:dd:39:6a
    media: Ethernet autoselect (100baseTX full-duplex)
    status: active
    inet 10.0.0.38 netmask 0xfffff00 broadcast
10.0.0.255
    inet6 fe80::204:acff:fedd:396a%fxp0 prefixlen 64
scopeid 0x1
pflog0: flags=0<> mtu 33224
pfsync0: flags=0<> mtu 2020
sl0: flags=c010<POINTOPOINT,LINK2,MULTICAST> mtu 296
sl1: flags=c010<POINTOPOINT,LINK2,MULTICAST> mtu 296
ppp0: flags=8010<POINTOPOINT,MULTICAST> mtu 1500
ppp1: flags=8010<POINTOPOINT,MULTICAST> mtu 1500
tun0: flags=10<POINTOPOINT> mtu 3000
tun1: flags=10<POINTOPOINT> mtu 3000
enc0: flags=0<> mtu 1536
bridge0: flags=0<> mtu 1500
bridge1: flags=0<> mtu 1500
vlan0: flags=0<> mtu 1500
    address: 00:00:00:00:00:00
vlan1: flags=0<> mtu 1500
    address: 00:00:00:00:00:00
gre0: flags=9010<POINTOPOINT,LINK0,MULTICAST> mtu 1450
carp0: flags=0<> mtu 1500
carp1: flags=0<> mtu 1500
gif0: flags=8010<POINTOPOINT,MULTICAST> mtu 1280
gif1: flags=8010<POINTOPOINT,MULTICAST> mtu 1280
gif2: flags=8010<POINTOPOINT,MULTICAST> mtu 1280
```

```
gif3: flags=8010<POINTOPOINT,MULTICAST> mtu 1280
```

Comme vous pouvez le voir, [ifconfig\(8\)](#) nous fournit ici beaucoup plus d'informations que nécessaire. Mais nous pouvons tout de même voir notre interface. Dans l'exemple précédent, la carte est déjà configurée. Ceci est évident de part la présence d'une configuration réseau IP pour fxp0, à savoir "inet 10.0.0.38 netmask 0xfffff00 broadcast 10.0.0.255". Les indicateurs **UP** et **RUNNING** sont également présents.

Finalement, vous noterez que d'autres interfaces sont activées par défaut. Il s'agit d'interfaces virtuelles servant différentes fonctions. Les manuels suivants les décrivent :

- [lo](#) - Interface "Loopback"
- [pflog](#) - Interface d'enregistrement du filtre de paquet (PF)
- [sl](#) - Interface SLIP (protocole Internet sur ligne série)
- [ppp](#) - Protocole point à point
- [tun](#) - Interface de tunnel
- [enc](#) - Interface d'encapsulation
- [bridge](#) - Interface de pont Ethernet
- [vlan](#) - Interface d'encapsulation IEEE 802.1Q
- [gre](#) - Interface d'encapsulation GRE/MobileIP
- [gif](#) - Interface générique de tunnel IPv4/IPv6
- [carp](#) - Interface du protocole de redondance d'adresse (CARP)
- [vether](#) - Interface Ethernet Virtuelle

L'interface est configurée au démarrage en utilisant les fichiers [/etc/hostname.if\(5\)](#) où *if* est remplacé par le nom complet de votre interface, c'est à dire dans notre exemple: */etc/hostname.fxp0*.

Le format de ce fichier est simple :

```
address_family address netmask broadcast [autres options]
```

Beaucoup plus d'informations sur la syntaxe de ce fichier sont disponibles dans le manuel [hostname.if\(5\)](#). Vous devrez le lire si vous avez besoin de configurations moins triviales.

Un fichier de configuration typique pour une interface IPv4 ressemblera à :

```
$ cat /etc/hostname.fxp0
inet 10.0.0.38 255.255.255.0 NONE
```

Dans ce cas, nous avons défini une adresse IPv4 (inet), avec une adresse IP de 10.0.0.38, un masque de

sous-réseau de 255.255.255.0 et pas d'adresse de broadcast spécifique (qui sera mise par défaut à 10.0.0.255 dans notre cas).

Vous pouvez également spécifier le type de media d'une connexion Ethernet pour, par exemple, forcer le mode 100baseTX full-duplex.

```
inet 10.0.0.38 255.255.255.0 NONE media 100baseTX mediaopt
full-duplex
```

(Bien entendu, vous ne devriez jamais forcer le mode full duplex à moins que les deux extrémités de la connexion ne soient configurées ainsi ! Si ce n'est pour des besoins spécifiques, la configuration du type de media n'est pas nécessaire. Un cas plus réaliste serait de forcer le mode 10base-T ou half duplex lorsque cela est rendu nécessaire par votre infrastructure.)

Vous pouvez aussi vouloir utiliser des indicateurs spécifiques à une certaine interface. Le format du fichier hostname ne change que très peu.

```
$ cat /etc/hostname.vlan0
inet 172.21.0.31 255.255.255.0 NONE vlan 2 vlandev fxp1
```

6.2.2 - Passerelle par défaut

Mettez l'adresse IP de votre passerelle par défaut dans le fichier [/etc/mygate](#). Cela permettra à votre route par défaut d'être configurée au démarrage. Le fichier n'est constitué que d'une seule ligne comportant l'adresse IP de la passerelle pour cette machine :

```
10.0.0.1
```

Il est possible d'utiliser un nom symbolique ici mais soyez prudents : vous ne pouvez jamais être certain que le résolveur soit configuré ni même accessible AVANT que la passerelle par défaut n'ait été elle-même configurée. En d'autres termes, il est souhaitable que cela soit une adresse IP ou un objet défini dans le fichier [/etc/hosts](#).

6.2.3 - Résolution DNS

La résolution de noms DNS est configurée par le fichier [/etc/resolv.conf](#). Voici un exemple de fichier [/etc/resolv.conf](#) :

```
search example.com
nameserver 125.2.3.4
nameserver 125.2.3.5
```

```
lookup file bind
```

Dans notre cas, le domaine par défaut sera `example.com`, il y a deux serveurs DNS, `125.2.3.4` et `125.2.3.5`, spécifiés et le fichier `/etc/hosts` sera consulté avant que les DNS ne le soient.

Comme sous la plupart des Unix (ainsi que de nombreux systèmes non-Unix), il existe un fichier `/etc/hosts` permettant de définir des systèmes qui ne sont pas (ou, si utilisés avec la directive de priorité "lookup", ne sont pas désirés) dans le système DNS formel.

Si vous utilisez DHCP, lisez [6.4 - DHCP](#) en faisant attention à la partie sur [resolv.conf.tail\(5\)](#).

6.2.4 - Nom d'hôte

Chaque machine sous Unix possède un nom. Sous OpenBSD, ce nom est défini comme FQDN ("Fully Qualified Domain Name" - Nom d'Hôte Pleinement Qualifié) dans le fichier `/etc/myname`. Si cette machine se nomme "puffy" et que son domaine est "example.com", le fichier contiendra la ligne :

```
puffy.example.com
```

6.2.5 - Activer les changements

A présent, vous pouvez soit redémarrer, soit lancer le script `/etc/netstart`. Vous pouvez l'exécuter simplement en tapant (en tant que root) :

```
# sh /etc/netstart
writing to routing socket: File exists
add net 127: gateway 127.0.0.1: File exists
writing to routing socket: File exists
add net 224.0.0.0: gateway 127.0.0.1: File exists
```

Notez que plusieurs erreurs sont apparues. En exécutant ce script, vous reconfigurez certaines choses qui le sont déjà. De fait, certaines routes sont déjà présentes dans la table de routage du noyau. A présent, votre système devrait être en état de fonctionner. Une nouvelle fois, vous pouvez vérifier que votre interface a été correctement configurée avec [ifconfig\(8\)](#).

Bien que vous puissiez complètement reconfigurer le réseau sous OpenBSD sans redémarrer, un reboot est FORTEMENT conseillé après tout changement significatif dans la configuration. La raison en est que l'environnement de démarrage est légèrement différent de celui d'un système en production. Ainsi, si vous avez défini un nom symbolique à résoudre par DNS dans un de vos fichiers de configuration, vous vous apercevrez probablement que cela fonctionnait après une reconfiguration mais que lors du démarrage, votre résolveur pourrait ne pas être disponible et la configuration échouer.

6.2.6 - Vérification des routes

Vous pouvez vérifier vos routes via [netstat\(1\)](#) ou [route\(8\)](#). Si vous avez des problèmes de routage, vous pouvez utiliser l'option `-n` de la commande `route(8)` qui affichera l'adresse IP plutôt que d'effectuer une requête DNS et d'afficher le nom d'hôte. Voici un exemple qui vous permettra de voir vos tables de routage en utilisant ces deux programmes.

```
$ netstat -rn
Routing tables
```

```
Internet:
```

Destination	Gateway	Flags	Refs
Use Mtu Interface			
default	10.0.0.1	UGS	0
86 - fxp0			
127/8	127.0.0.1	UGRS	0
0 - lo0			
127.0.0.1	127.0.0.1	UH	0
0 - lo0			
10.0.0/24	link#1	UC	0
0 - fxp0			
10.0.0.1	aa:0:4:0:81:d	UHL	1
0 - fxp0			
10.0.0.38	127.0.0.1	UGHS	0
0 - lo0			
224/4	127.0.0.1	URS	0
0 - lo0			

```
Encap:
```

Source	Port	Destination	Port	Proto	SA
(Address/SPI/Proto)					

```
$ route show
```

```
Routing tables
```

```
Internet:
```

Destination	Gateway	Flags
default	10.0.0.1	UG
127.0.0.0	LOCALHOST	UG
localhost	LOCALHOST	UH
10.0.0.0	link#1	U
10.0.0.1	aa:0:4:0:81:d	UH

```
10.0.0.38          LOCALHOST          UGH
BASE-ADDRESS.MCA LOCALHOST          U
```

6.2.7 - Mettre en place une passerelle OpenBSD

Voici les informations nécessaires à la mise en place d'une passerelle OpenBSD (appelé aussi routeur). Si vous devez installer OpenBSD pour en faire un routeur Internet, nous vous suggérons de lire les instructions sur la mise en place du filtre de paquets (plus loin) afin de bloquer le trafic non-autorisé. Avec le peu de disponibilité d'adresses [IPv4](#) de la part des fournisseurs d'accès à Internet ainsi que des registres Internet régionaux, vous pourriez vouloir vous renseigner sur la translation d'adresses (NAT) afin d'économiser votre adressage IP.

Le noyau GENERIC est déjà configuré pour permettre le routage IP, mais celui-ci doit être explicitement activé. Vous pouvez l'activer avec l'utilitaire [sysctl\(8\)](#). Afin d'autoriser le routage de façon permanente, éditez le fichier [/etc/sysctl.conf](#) et ajoutez-y la ligne suivante.

```
net.inet.ip.forwarding=1
```

Pour prendre en compte ce changement sans redémarrer, vous utiliserez directement l'utilitaire [sysctl\(8\)](#). Souvenez-vous que ce changement ne sera pas sauvegardé au redémarrage et que vous devrez être root pour utiliser cette commande.

```
# sysctl net.inet.ip.forwarding=1
net.inet.ip.forwarding: 0 -> 1
```

A présent, modifiez les routes sur les hôtes aux deux extrémités. Il existe beaucoup d'autres usages d'OpenBSD en tant que routeur avec l'aide de programmes comme [OpenBGPD](#) (qui fait partie du projet OpenBSD), [routed\(8\)](#), [mrtd](#), [zebra](#) et [quagga](#). Sous OpenBSD, zebra et mrtd sont disponibles en tant que ports. OpenBGPD et routed sont inclus dans le système de base. OpenBSD supporte différentes interfaces T1, HSSI, ATM, FDDI, Ethernet et série (PPP/SLIP).

6.2.8 - Configurer les alias sur une interface

OpenBSD possède un mécanisme simple pour la mise en place d'alias IP sur une interface. Pour ce faire, il suffit d'éditer le fichier [/etc/hostname.<if>](#). Ce fichier est lu au boot par le script [/etc/netstart\(8\)](#) faisant partie de la séquence de démarrage [rc](#). Admettons par exemple qu'un utilisateur utilise l'interface **dc0** et se trouve sur le réseau 192.168.0.0. Autres informations importantes :

- l'adresse IP pour dc0 est 192.168.0.2
- le masque de sous-réseau est 255.255.255.0

Notes sur les alias. Sous OpenBSD, vous utilisez le nom de l'interface uniquement. Il n'y a pas de différence entre le premier et le second alias. A la différence d'autres systèmes d'exploitation, OpenBSD ne s'y réfère pas en tant que `dc0:0`, `dc0:1`. Si vous vous référez à une adresse IP d'alias avec `ifconfig` ou si vous ajoutez un alias, soyez sûr d'utiliser la commande "`ifconfig int alias`" au lieu de "`ifconfig int`". Vous pouvez supprimer les alias en utilisant "`ifconfig int delete`".

En admettant que vous utilisiez plusieurs adresses avec alias sur le même sous-réseau IP, le masque correspondant à chaque alias devient `255.255.255.255`. Il n'est pas nécessaire d'utiliser le masque correspondant à l'adresse IP primaire de l'interface. Dans cet exemple, `/etc/hostname.dc0`, deux alias sont configurés pour le périphérique `dc0`, qui lui-même possède l'adresse `192.168.0.2` avec un masque de `255.255.255.0`.

```
# cat /etc/hostname.dc0
inet 192.168.0.2 255.255.255.0 NONE media 100baseTX
inet alias 192.168.0.3 255.255.255.255
inet alias 192.168.0.4 255.255.255.255
```

Après avoir édité ce fichier, il suffit de redémarrer pour que les changements prennent effet. Mais vous pouvez aussi créer les alias à la main en utilisant l'utilitaire [ifconfig\(8\)](#). Pour créer le premier alias, lancez la commande suivante :

```
# ifconfig dc0 inet alias 192.168.0.3 netmask
255.255.255.255
```

(mais encore une fois, un redémarrage est conseillé afin d'être certain que vous ayez rentré la configuration que vous pensiez !)

Pour voir les alias, utilisez la commande suivante :

```
$ ifconfig -A
dc0: flags=8863<UP,BROADCAST,NOTRAILERS,RUNNING,SIMPLEX,
MULTICAST>
    media: Ethernet manual
    inet 192.168.0.2 netmask 0xffffffff00 broadcast
192.168.0.255
    inet 192.168.0.3 netmask 0xffffffff broadcast
192.168.0.3
```

6.3 - Comment filtrer et utiliser un pare-feu sous OpenBSD ?

Sous OpenBSD, le filtrage du trafic IP ainsi que la translation d'adresses (NAT) est pris en charge par

"Packet Filter" (filtre de paquet ; auquel nous nous référerons à présent sous le nom de PF) . PF est aussi capable de normaliser et de traiter le trafic IP, d'offrir une gestion de la bande passante et une prioritarisation des paquets ainsi que d'être utilisé dans la création de puissants et flexibles pare-feu. Tout ceci est décrit dans le [Guide de l'Utilisateur PF](#).

6.4 - DHCP

Le protocole d'attribution dynamique des adresses (DHCP) permet de configurer les interfaces réseau automatiquement. OpenBSD peut être serveur DHCP (afin de configurer les autres machines), client DHCP (afin de recevoir sa configuration à partir d'une autre machine) et, dans certains cas, les deux.

6.4.1 Client DHCP

Pour utiliser le client DHCP [dhclient\(8\)](#) inclus avec OpenBSD, éditez le fichier `/etc/hostname.xl0` (sous-entendu que l'interface Ethernet soit xl0. La vôtre pouvant être aussi bien ep0 que fxp0 ou encore autre chose). Tout ce que vous avez besoin d'écrire dans ce fichier est 'dhcp' :

```
# echo dhcp > /etc/hostname.xl0
```

Ceci aura pour effet de démarrer automatiquement le client DHCP au démarrage. OpenBSD se verra attribuer son adresse IP, sa passerelle par défaut et ses serveurs de noms (DNS) à partir du serveur DHCP.

Si vous souhaitez démarrer le client DHCP à partir de la ligne de commande, vérifiez bien que `/etc/dhclient.conf` existe, puis lancez la commande :

```
# dhclient fxp0
```

Où `fxp0` représente l'interface que vous voulez configurer par DHCP.

Peu importe la façon dont vous démarrez le client DHCP, vous pouvez éditer `/etc/dhclient.conf` afin de ne **pas** mettre à jour vos DNS à l'aide du serveur dhcp en décommentant les lignes 'request' (il y a des exemples de configurations, mais vous devez les décommenter afin de changer le comportement par défaut de dhclient.)

```
request subnet-mask, broadcast-address, time-offset,
routers,
        domain-name, domain-name-servers, host-name, lpr-
servers, ntp-servers;
```

et supprimez `domain-name-servers`. Bien sûr, vous pouvez également supprimer `host-name`

ainsi que d'autres options.

En changeant les options dans le fichier [dhclient.conf\(5\)](#), vous indiquez au client DHCP la façon de créer votre fichier [resolv.conf\(5\)](#). Le client DHCP écrase toutes les données déjà présentes dans `resolv.conf(5)` en utilisant les informations envoyées par le serveur DHCP. De fait, tous les changements que vous aurez apportés manuellement seront perdus.

Il existe deux mécanismes pour empêcher cela :

- [OPTION MODIFIERS](#) (**default**, **supersede**, **prepend**, et **append**) vous permet de modifier n'importe quelles options du fichier `dhclient.conf(5)`.
- [resolv.conf.tail\(5\)](#) vous permet d'ajouter ce que vous souhaitez au fichier `resolv.conf(5)` créé par `dhclient(8)`.

Imaginons que vous souhaitiez utiliser le client DHCP mais que vous vouliez ajouter `lookup file bind` au fichier `resolv.conf(5)` créé par `dhclient(8)`. Il n'existe pas de telle option dans le fichier `dhclient.conf` et vous devrez donc utiliser `resolv.conf.tail` pour préserver ce changement.

```
# echo "lookup file bind" > /etc/resolv.conf.tail
```

A présent, votre fichier `resolv.conf(5)` devrait inclure "lookup file bind" à la fin.

```
nameserver 192.168.1.1
nameserver 192.168.1.2
lookup file bind
```

6.4.2 Serveur DHCP

Pour utiliser OpenBSD en tant que serveur DHCP [dhcpd\(8\)](#), éditez le fichier `/etc/rc.conf.local` afin qu'il contienne la ligne `dhcpd_flags="interface"`, remplacez `interface` avec la liste des interfaces que le serveur écoute.

```
# echo 'dhcpd_flags="x11 x12 x13"' >>/etc/rc.conf.local
```

Ensuite, éditez `/etc/dhcpd.conf`. Les options sont plutôt explicites.

```
option domain-name "example.com";
option domain-name-servers 192.168.1.3, 192.168.1.5;

subnet 192.168.1.0 netmask 255.255.255.0 {
    option routers 192.168.1.1;
```

```

        range 192.168.1.32 192.168.1.127;
    }

```

Ceci indique à vos clients DHCP que le domaine à ajouter aux requêtes DNS est `example.com` (ainsi, si un utilisateur lance la commande `'telnet joe'`, il sera renvoyé vers `joe.example.com`). Les clients auront comme serveurs DNS 192.168.1.3 et 192.168.1.5. Pour les hôtes présents sur le sous-réseau correspondant à l'interface du serveur OpenBSD, à savoir dans la plage 192.168.1.0/24, ils se verront attribuer une adresse IP entre 192.168.1.32 et 192.168.1.127. Leur passerelle par défaut sera 192.168.1.1.

Si vous souhaitez démarrer `dhcpcd(8)` à partir de la ligne de commandes, une fois `/etc/dhcpcd.conf` configuré, tapez :

```

# touch /var/db/dhcpcd.leases
# dhcpcd fxp0

```

La ligne invoquant `touch` est nécessaire afin de créer un fichier `dhcpcd.leases` vide avant que `dhcpcd(8)` ne démarre. Les [scripts de démarrage](#) d'OpenBSD se chargeront de créer ce fichier au boot, mais si vous lancez `dhcpcd(8)` manuellement, vous devez au préalable créer ce fichier. `fxp0` représente l'interface sur laquelle vous voulez répondre aux requêtes DHCP.

Si vous prévoyez de servir DHCP à des machines Windows, pour pourriez souhaiter donner à ces clients une adresse de serveur 'WINS'. Pour ce faire, ajoutez simplement la ligne suivante dans votre fichier `/etc/dhcpcd.conf` :

```

option netbios-name-servers 192.168.92.55;

```

(où 192.168.92.55 est l'adresse IP de votre serveur Windows ou Samba.) Reportez vous au manuel [dhcpc-options\(5\)](#) si vos clients DHCP ont besoin de plus d'options.

6.5 - PPP

Le protocole point à point est généralement utilisé afin de créer une connexion modem vers votre FAI (fournisseur d'accès à Internet). Sous OpenBSD, deux solutions existent.

- [pppd\(8\)](#) - implémentation noyau du service ppp.
- [ppp\(8\)](#) - implémentation ppp en espace utilisateur ("userland").

Chacune des implémentations (`ppp` et `pppd`) offrent des fonctionnalités similaires de différentes façons. `pppd` utilise le pilote [ppp\(4\)](#) inclus dans le noyau alors que `ppp` utilise [tun\(4\)](#) dans l'espace utilisateur. Ce

document ne couvrira que l'implémentation en espace utilisateur du service PPP car il est plus simple à utiliser et diagnostiquer. Pour commencer, vous aurez besoin de certaines informations concernant votre FAI. En voici une liste non-exhaustive.

- Le numéro de téléphone de votre FAI
- Votre serveur de noms
- Votre nom d'utilisateur et votre mot de passe
- Votre passerelle

Certaines de ces options ne sont pas obligatoires mais vous aideront à la mise en place de ppp. Le fichier de configuration du service PPP est [/etc/ppp/ppp.conf](#). Selon votre situation, les nombreux fichiers présents dans */etc/ppp* peuvent vous aider à la mise en place de votre configuration. Vous devriez jeter un oeil à ce répertoire.

Configuration initiale - pour PPP(8)

La configuration initiale de PPP nécessite l'édition du fichier */etc/ppp/ppp.conf*. Ce fichier n'existe pas par défaut, mais vous pouvez vous inspirer du fichier */etc/ppp/ppp.conf.sample* afin de créer votre propre *ppp.conf*. Je commencerai par une configuration simple et généralement très employée. Voici rapidement un petit fichier *ppp.conf* définissant quelques valeurs par défaut :

```
default:
  set log Phase Chat LCP IPCP CCP tun command
  set device /dev/cua01
  set speed 115200
  set dial "ABORT BUSY ABORT NO\\sCARRIER TIMEOUT 5 \"\" AT
  OK-AT-OK ATE1Q0 OK \\dATDT\\T TIMEOUT 40 CONNECT"
```

La section définie par le marqueur `default:` sera exécutée à chaque fois. Cette section recense des informations importantes. "set log" définit le niveau de verbosité des logs. Il peut être changé : référez-vous au manuel de [ppp\(8\)](#) pour plus d'informations concernant les niveaux d'enregistrement des différents événements. Le périphérique utilisé est indiqué par "set device". Il s'agit du port sur lequel est présent notre modem. Dans cet exemple, le modem est branché sur le port com 2. Ainsi, le port com 1 sera indiqué par */dev/cua00*. La vitesse de la connexion est précisée par "set speed" et "set dial" renseigne nos paramètres de numérotation. Avec ceci, nous pouvons changer l'expiration ("timeout") de la connexion, etc. Ceci étant, cette ligne ne devrait pas tellement varier.

Nous pouvons maintenant continuer et configurer les informations spécifiques à notre FAI. Pour ce faire, nous allons rajouter une autre section en dessous de `default:`. Le marqueur définissant cette nouvelle section pourra être ce que vous désirez - le plus simple étant d'utiliser le nom de votre FAI. Ici, nous utiliserons `myisp:` pour indiquer le commencement de la section correspondant à notre FAI. Voici une configuration simple contenant le nécessaire afin de nous connecter :

```

myisp:
  set phone 1234567
  set login "ABORT NO\\sCARRIER TIMEOUT 5 ogin:--ogin: ppp
word: ppp"
  set timeout 120
  set ifaddr 10.0.0.1/0 10.0.0.2/0 255.255.255.0 0.0.0.0
  add default HISADDR
  enable dns

```

Ici nous avons fourni les éléments essentiels concernant ce FAI. La première option, "set phone", donne le numéro de téléphone du FAI. Nos options d'ouverture de session sont renseignées par "set login". Notre "timeout" est égal à 5 ; ce qui signifie que notre tentative de connexion expirera après 5 secondes si aucune porteuse n'est trouvée. Dans le cas contraire, "login:" sera envoyé avec notre nom d'utilisateur et notre mot de passe.

Dans cet exemple, notre nom d'utilisateur est : ppp ; notre mot de passe est : ppp. Ces valeurs doivent être changées. La ligne "set timeout" permet de couper la connexion après 120 secondes de non-utilisation. L'option "set ifaddr" est un peu plus compliquée. En voici une explication plus complète.

```
set ifaddr 10.0.0.1/0 10.0.0.2/0 255.255.255.0 0.0.0.0
```

La ligne précédente se définit avec le format suivant : "**set ifaddr [myaddr[/nn] [hisaddr[/nn] [netmask [triggeraddr]]]]**". La première IP désigne l'adresse IP que nous souhaitons nous voir attribuer. Si vous avez une adresse IP statique, vous devez l'indiquer ici. Dans notre exemple, nous utilisons la notation /0 qui dit qu'aucun "bit" de cette adresse ne doit forcément correspondre et que celle-ci peut être changée. La deuxième adresse IP est celle du FAI. La troisième option est notre masque de sous-réseau, ici défini à 255.255.255.0. Si triggeraddr est renseigné, il remplacera myaddr comme adresse IP utilisée pour la négociation IPCP initiale. Cependant, seule une adresse incluse dans la gamme d'adressage correspondant à myaddr sera acceptée. Ceci peut être utile dans la négociation avec certaines implémentations PPP qui n'attribuent pas d'adresse IP à moins que l'initiateur de la connexion ne demande explicitement l'adresse "0.0.0.0".

L'option suivante "add default HISADDR" attribue comme route par défaut l'adresse IP de votre FAI. Cette entrée permet d'ajuster automatiquement notre route par défaut en cas de changement d'adresse IP de celui-ci. "enable dns" est utilisé afin de récupérer la liste des serveurs DNS du FAI. N'utilisez PAS cette option si vous avez votre propre serveur DNS local car ppp empêchera les requêtes vers celui-ci en remplaçant les lignes "nameserver" de votre fichier */etc/resolv.conf*.

A la place des méthodes traditionnelles de connexion, certains FAI utilisent à présent l'authentification CHAP ou PAP. Si c'est le cas, notre configuration sera un peu différente:

```

myisp:
  set phone 1234567
  set authname ppp
  set authkey ppp
  set login
  set timeout 120
  set ifaddr 10.0.0.1/0 10.0.0.2/0 255.255.255.0 0.0.0.0
  add default HISADDR
  enable dns

```

Dans l'exemple précédent, nous spécifions notre nom d'utilisateur (ppp) et notre mot de passe à l'aide des options `authname` et `authkey`, respectivement. Il n'est pas nécessaire de spécifier le type d'authentification CHAP ou PAP utilisé, car il sera négocié automatiquement. "set login" signifie simplement de tenter de se connecter en utilisant le nom d'utilisateur et le mot de passe précisés précédemment.

Utiliser PPP(8)

Maintenant que `ppp.conf` est configuré, nous pouvons essayer d'initier une connexion vers notre FAI. Je détaillerais certaines des options les plus utilisées :

- `ppp -auto myisp` - Cette commande lancera ppp, configurera vos interfaces, vous connectera à votre FAI et se placera en arrière-plan.
- `ppp -ddial myisp` - Cette commande est similaire à `-auto` mais si votre connexion est rompue, ppp essaiera de se reconnecter automatiquement.

Si les commandes précédentes ne fonctionnent pas, lancez `/usr/sbin/ppp` sans option, ce qui vous placera en mode interactif. Les options peuvent être spécifiées une par une afin de diagnostiquer les erreurs et autres problèmes. En utilisant la configuration précédente, ppp enregistrera les événements dans `/var/log/ppp.log`. Ce fichier de log ainsi que la page de manuel contiennent des informations utiles.

Autres options de ppp(8)

Dans certaines situations, vous pourriez avoir besoin de lancer certaines commandes au lancement ou à la coupure de votre connexion. Si vous vous trouvez dans cette situation, il existe deux fichiers que vous pouvez créer : `/etc/ppp/ppp.linkup` et `/etc/ppp/ppp.linkdown`. Des exemples de configurations sont disponibles ici :

- [ppp.linkup](#)
- [ppp.linkdown](#)

variations sur ppp(8)

De nos jours, beaucoup de FAI offrent des services xDSL plus rapides que les méthodes traditionnelles de connexion. Ceux-ci incluent des variantes de type ADSL et SDSL. Bien qu'aucun appel à distance n'est lieu, la connexion reste basée sur le protocole point-à-point. En voici des exemples:

- PPPoE
- PPPoA
- PPTP

PPPoE/PPPoA

Le protocole d'encapsulation de PPP dans Ethernet (PPPoE - "Point to Point Protocol over Ethernet") permet d'envoyer des paquets PPP à l'intérieur de trames Ethernet. Le protocole d'encapsulation de PPP dans ATM (PPPoA - "Point to Point Protocol over ATM") est généralement utilisé dans les réseaux ATM tels qu'au Royaume-Uni ou en Belgique.

En résumé cela signifie que vous pouvez établir une connexion vers votre FAI à l'aide d'une carte Ethernet standard et d'un modem DSL-Ethernet (à l'inverse des modems USB).

Si vous possédez un modem compatible PPPoE/PPPoA, il est possible de le configurer afin qu'il initie lui-même la connexion. Dans le cas contraire, si le modem est capable de fonctionner en mode "bridge", il est possible d'activer ce mode afin que celui-ci fasse "transiter" les paquets vers une machine exécutant un logiciel PPPoE (voir plus loin).

Sous OpenBSD, l'interface logicielle PPPoE/PPPoA principale est [pppoe\(8\)](#), qui est l'implémentation en espace utilisateur (de la même manière que [ppp\(8\)](#), précédemment décrit). Une implémentation noyau de PPPoE, [pppoe\(4\)](#), a été ajoutée dans OpenBSD.

PPTP

Le "Point to Point Tunneling Protocol" (PPTP) est un protocole propriétaire Microsoft. Un client `pptp` s'interfaçant avec [pppd\(8\)](#) est disponible et est capable de se connecter à un réseau privé virtuel PPTP (VPN) utilisé par certains fournisseurs d'accès câble ou xDSL. Le logiciel `pptp` quant à lui doit être installé en utilisant les [paquets](#) ou les [ports](#). De plus amples instructions sur la façon de mettre en place et d'utiliser `pptp` sont disponibles dans la page de manuel installée avec le paquet `pptp`.

6.6 - Optimisation des paramètres réseau

L'un des objectifs de OpenBSD est d'avoir le système *Just Work* pour la grande majorité de nos

utilisateurs. Les optimisations que vous ne comprenez pas sont beaucoup plus susceptibles de briser le système que d'en améliorer ses performances. Toujours commencer par les paramètres par défaut et ajuster les choses *uniquement* si vous avez un problème.

TRES PEU DE personne ont besoin de cela !

6.6.1 - Comment configurer le noyau pour avoir un plus grand nombre d'essais et augmenter le délai d'expiration des sessions TCP ?

Généralement, vous utiliserez ceci en cas de problèmes de routage ou de connexion. Bien sûr, pour que cette configuration soit la plus efficace, les deux extrémités de la connexion doivent utiliser des valeurs similaires.

Pour optimiser ceci, utilisez `sysctl` et augmentez les valeurs de :

```
net.inet.tcp.keepintime
net.inet.tcp.keepidle
net.inet.tcp.keepintvl
```

Avec `sysctl -a`, vous pouvez voir les valeurs courantes de ces paramètres (ainsi que beaucoup d'autres). Pour en changer un, tapez par exemple `sysctl net.inet.tcp.keepidle=28800`.

6.6.2 - Comment activer les émissions ("broadcasts") dirigées ?

Normalement, vous ne devriez pas utiliser ceci. Cette option permet à quelqu'un de diriger le trafic vers la ou les adresses "broadcast" des réseaux sur lesquels vous êtes connecté si vous utilisez OpenBSD en tant que routeur.

Dans certaines situations, sur des réseaux fermés, cette option peut-être utile, surtout lors de l'utilisation de vieilles implémentations du protocole NetBIOS. Il s'agit d'un autre paramètre `sysctl`. `sysctl net.inet.ip.directed-broadcast=1` active cette option. Lisez la section sur les ["attaques de type smurf"](#) si vous voulez savoir pourquoi cette option est désactivée par défaut.

6.6.3 - Je ne veux pas que le noyau alloue dynamiquement un port donné

Il existe également un paramètre `sysctl` pour cela. D'après [sysctl\(8\)](#) :

```
Définie la liste des ports TCP ne devant pas être alloués
dynamiquement
par le noyau :
```

```
# sysctl net.inet.tcp.  
baddynamic=749,750,751,760,761,871
```

Ceci peut être utile afin d'éviter l'appropriation d'un port spécifique dont un autre programme a besoin pour fonctionner. Les éléments listés peuvent être séparés par une virgule et/ou un espace.

Il est aussi possible d'ajouter ou de retirer des ports de la liste courante.

```
# sysctl net.inet.tcp.baddynamic=+748  
# sysctl net.inet.tcp.baddynamic=-871
```

6.6.4 - Comment augmenter les performances sur les liens à grande vitesse et à fort trafic ?

Si vous avez des soucis de performances sur des liens WAN à grande vitesse lorsque vous transférez beaucoup de données, vous devriez pouvoir optimiser ces performances en éditant les valeurs sysctls suivantes :

```
net.inet.tcp.recvspace  
net.inet.tcp.sendspace
```

Essayez une valeur comme 65536 au lieu de 16384. Notez que très peu de personnes verront un gain réel. N'ajustez ces valeurs que si vous obtenez des performances en deçà de vos attentes.

6.7 - Utilisation simple de NFS

NFS ("Network File System") est utilisé afin de partager un système de fichiers à travers un réseau. Voici un certain nombre de manuels à lire avant la mise en place d'un serveur NFS :

- [nfsd\(8\)](#)
- [mountd\(8\)](#)
- [exports\(5\)](#)

Cette section détaillera les étapes nécessaires à la mise en oeuvre d'une configuration NFS simple. Cet exemple détaille un serveur et ses clients NFS sur un LAN. Il ne s'agit pas d'étudier la sécurisation de NFS. Nous assumerons que le filtre de paquets (PF) ou qu'un autre type de pare-feu est configuré afin d'interdire les accès extérieurs. Si vous autorisez l'accès à votre serveur NFS depuis l'extérieur et que

vous hébergez des données sensibles, nous vous conseillons fortement d'utiliser IPsec. Autrement certaines personnes pourraient voir ce qui transite dans votre trafic NFS. Quelqu'un pourrait également usurper l'adresse IP que vous autorisez à se connecter à votre serveur NFS. Plusieurs types d'attaques peuvent en découler. Mais lorsqu'IPsec est correctement configuré, il offre une protection contre ces attaques.

Mise en place d'un serveur NFS

Les services suivants doivent être activés et mis en route sur serveur :

- [portmap\(8\)](#)
- [mountd\(8\)](#)
- [nfsd\(8\)](#)

Ces services sont désactivés par défaut sous OpenBSD. Afin de les activer, veuillez ajouter les lignes suivantes au fichier [rc.conf.local\(8\)](#) :

```
portmap=YES
nfs_server=YES
```

L'étape suivante consiste à configurer la liste des systèmes de fichiers qui doivent être mis à disposition des clients.

Dans cet exemple, nous disposons d'un serveur dont l'adresse IP est 10.0.0.1. Ce serveur est destiné à fournir le service FS uniquement aux clients de son propre sous-réseau. Tout ceci est configurable dans le fichier `/etc/exports`. Ce fichier liste les systèmes de fichiers destinés à être mis à disposition via NFS et définit les droits d'accès associés. Plusieurs options sont disponibles. Il convient de lire la page du manuel [exports\(5\)](#). Pour notre serveur, nous utiliserons le fichier exports suivant :

```
#
# NFS exports Database
# See exports(5) for more information.  Be very careful,
# misconfiguration
# of this file can result in your filesystems being
# readable by the world.
/work -alldirs -ro -network=10.0.0 -mask=255.255.255.0
```

Ceci signifie que le système de fichiers local `/work` sera accessible par NFS. L'option `-alldirs` spécifie que les clients seront capables de monter n'importe quel répertoire situé sous `/work` ainsi que `/work` lui-même. Par exemple, s'il existait un répertoire appelé `/work/monday`, les clients pourraient monter `/work` (et avoir ainsi accès à tous les fichiers/répertoires situés sous ce répertoire) ou ils

pourraient tout simplement monter `/work/monday` et avoir ainsi juste accès aux fichiers/répertoires qu'il contient. L'option `-ro` spécifie que les clients sont autorisés à faire de la lecture uniquement. Les deux derniers arguments spécifient que seuls les clients présents dans le réseau 10.0.0.0 et utilisant un masque de 255.255.255.0 seront autorisés à monter ce système de fichiers. Ceci est important pour certains serveurs accessibles dans différents réseaux.

Une autre remarque sécurité importante : ne vous contentez pas d'ajouter tout simplement un système de fichiers à `/etc/exports` sans l'associer à une liste d'hôte(s) autorisé(s). Sans cela, n'importe qui pourrait monter vos répertoires exportés par NFS.

Vous pouvez maintenant démarrer les services du serveur. Vous pouvez soit redémarrer (après activation telle qu'indiquée plus haut) soit exécuter ces services manuellement.

```
# /usr/sbin/portmap
# echo -n >/var/db/mountdtab
# /sbin/mountd
# /sbin/nfsd -tun 4
```

Les arguments passés à `nfsd` autorisent les connexions TCP (`-t`) et UDP (`-u`) et autorisent aussi 4 instances (`-n`) de `nfsd` en exécution. Vous devriez paramétrer un nombre approprié d'instances de serveur NFS pour gérer le maximum de connexions simultanées que vous souhaitez être en mesure de traiter.

Vous êtes maintenant prêt à monter les systèmes de fichiers exportés depuis le(s) client(s).

Rappel : Si vous effectuez des changements au niveau de `/etc/exports` alors que NFS est en cours d'exécution, vous avez juste besoin d'en informer `mountd` ! Il suffit d'envoyer un signal HUP à ce service et les modifications seront effectives.

```
# kill -HUP `cat /var/run/mountd.pid`
```

Montage de Systèmes de Fichiers NFS

Les systèmes de fichiers NFS peuvent être montés à partir d'un client sans avoir besoin d'activer aucun service. Ils peuvent être montés de la même manière que n'importe quel autre système de fichiers.

Les systèmes de fichiers NFS devraient être montés via `mount(8)`, ou plus précisément, [mount_nfs\(8\)](#). Pour monter un système de fichiers `/work` partagé par l'hôte 10.0.0.1 au niveau du système de fichiers local `/mnt`, effectuez l'opération suivante (veuillez noter que vous n'avez pas besoin d'utiliser obligatoirement une adresse IP. `mountd` peut utiliser le DNS) :

```
# mount -t nfs 10.0.0.1:/work /mnt
```

Pour monter le système de fichiers au démarrage, ajoutez la ligne suivante à `/etc/fstab` :

```
10.0.0.1:/work /mnt nfs rw 0 0
```

Il est important d'utiliser `0 0` à la fin de cette ligne pour éviter que le système essaie de faire un `fsck` des systèmes de fichiers NFS au démarrage. Les autres options de sécurité standard telles que `noexec`, `nodev`, et `nosuid`, devraient aussi être utilisées si applicable. Par exemple :

```
10.0.0.1:/work /mnt nfs rw,nodev,nosuid 0 0
```

De cette manière, aucun périphérique ou programme `setuid` sur le serveur NFS ne pourra mettre en défaut les mesures de sécurité prises sur le client NFS. Si vous ne comptez pas utiliser de programmes mis à disposition par le serveur NFS, ajoutez `noexec` à la liste.

Lorsqu'on accède à un montage NFS avec l'identité de l'utilisateur `root`, le serveur réalisera une correspondance automatique de cet accès `root` vers l'utilisateur et le groupe "`nobody`". C'est une information importante à connaître lorsqu'on prend en compte les permissions sur les fichiers. Par exemple, prenez un fichier avec les permissions suivantes :

```
-rw----- 1 root wheel 0 Dec 31 03:00
_daily.B20143
```

Si le fichier était sur un partage NFS et l'utilisateur `root` essaie d'accéder à ce fichier depuis le client NFS, l'accès sera interdit. Ceci est dû au fait que le serveur utilise les identifiants de l'utilisateur "`nobody`" lorsque `root` essaie d'accéder au fichier. Étant donné que l'utilisateur `nobody` n'a pas de permissions d'accès au fichier, l'accès est interdit.

L'utilisateur et le groupe auxquels `root` est "traduit" sont configurables à l'aide du fichier [exports\(5\)](#) sur le serveur NFS.

Statistiques NFS

Pour s'assurer que NFS fonctionne correctement il faut vérifier que tous ces services sont lancés et enregistrés avec RPC. Pour ce faire, utilisez `rpcinfo(8)`.

```
$ rpcinfo -p 10.0.0.1
  program vers proto  port
    100000   2   tcp    111  portmapper
    100000   2   udp    111  portmapper
    100005   1   udp    633  mountd
    100005   3   udp    633  mountd
```

```

100005      1      tcp      916      mountd
100005      3      tcp      916      mountd
100003      2      udp      2049     nfs
100003      3      udp      2049     nfs
100003      2      tcp      2049     nfs
100003      3      tcp      2049     nfs

```

Dans le cadre d'une utilisation normale, d'autres utilitaires peuvent vous permettre de voir ce qui se passe au niveau NFS. Un de ces utilitaires est [showmount\(8\)](#), qui permet de voir ce qui est monté et par qui. Il existe aussi [nfsstat\(1\)](#) qui affiche beaucoup plus de statistiques. Pour utiliser showmount(8), essayez la commande `/usr/bin/showmount -a hôte`. Par exemple :

```

$ /usr/bin/showmount -a 10.0.0.1
All mount points on 10.0.0.1:
10.0.0.37:/work

```

Cette sortie montre que le client 10.0.0.37 a monté le système de fichiers exporté `/work` mis à disposition par le serveur 10.0.0.1.

6.9 - Mise en place d'un pont ("bridge") avec OpenBSD

Un pont ("[bridge](#)") est un lien entre deux (ou plusieurs) réseaux séparés. A l'inverse d'un routeur, les paquets sont transférés à travers le pont de manière "invisible" -- au niveau logique, les deux segments de réseau semblent n'en faire qu'un de chaque côté du pont. Le pont ne transfèrera que les paquets passant d'un segment à l'autre, ce qui entre autres, permet de réduire le trafic sur un réseau complexe tout en permettant à n'importe quel noeud d'accéder à un autre en cas de besoin.

Notez qu'à cause de sa nature "invisible", une interface faisant partie d'un pont peut, ou non, posséder une adresse IP. Si c'est le cas, l'interface aura deux modes d'opération, l'un faisant parti du pont et l'autre se comportant comme une interface normale. Si aucune interface ne possède d'adresse IP, le pont fera transiter le trafic mais ne sera pas administrable par le réseau (ce qui peut être une fonctionnalité voulue).

Exemple d'application d'un pont

Un de mes racks possède un certain nombre d'anciens systèmes, lesquels ne sont pas équipés de carte réseau 10BASE-TX. Bien qu'ils possèdent des ports AUI ou AAUI, ma réserve de transmetteurs est limitée à des câbles coaxiaux. Une des machines de cette rangée est un serveur d'accès terminal sous OpenBSD, toujours allumée et connectée au réseau à haut débit. L'ajout d'une seconde carte équipée d'un port coaxial permettra d'utiliser cette machine comme un pont vers le réseau coaxial.

Ce système a, pour le moment, deux cartes réseau, une Intel EtherExpress/100 ([fxp0](#)) et une carte 3c590-Combo ([ep0](#)) pour le port coaxial. `fxp0` fait le lien avec le reste du réseau et possède donc une adresse IP, `ep0`, ne faisant quant à elle que du "bridging", n'en possède pas. Les machines connectées sur le segment coaxial communiqueront comme les autres présentes sur le reste du réseau. A présent, voyons comment arriver à ce résultat.

Le fichier `hostname.fxp0` contient les informations concernant la carte `fxp0`. Cette machine est configurée pour DHCP, donc le fichier ressemble à celui-ci :

```
$ cat /etc/hostname.fxp0
dhcp NONE NONE NONE
```

Ici, aucune surprise.

Comme vous pouvez le deviner, la configuration de la carte `ep0` est un peu différente :

```
$ cat /etc/hostname.ep0
up media 10base2
```

Ici, nous demandons au système d'activer cette interface en utilisant [ifconfig\(8\)](#) et de la configurer pour du 10BASE-2 (coaxial). Aucune adresse ou information similaire n'est nécessaire pour cette interface. Les options possibles pour la carte `ep` sont disponibles en détail dans le [manuel](#).

A présent, il nous faut paramétrer le pont. Un pont est initialisé par l'existence d'un fichier du type [bridgename.bridge0](#). Dans ma situation, voici un exemple possible :

```
$ cat /etc/bridgename.bridge0
add fxp0
add ep0
up
```

Cela indique de mettre en place un pont constitué de deux interfaces, `fxp0` et `ep0` et de l'activer. Est-ce que l'ordre dans lequel les cartes sont énoncées est important ? Non, souvenez-vous qu'un pont est très symétrique -- les paquets entrent et sortent dans les deux directions.

C'est tout ! Redémarrez et vous aurez un pont fonctionnel.

Le filtrage sur un pont ("filtering bridge")

Bien qu'il existe certainement des usages pour de simples ponts du genre évoqué, il est probable que vous souhaitiez FAIRE quelque chose avec les paquets qui le traversent. Comme vous pouvez vous en

douter, [Packet Filter](#) peut être utilisé pour restreindre le trafic traversant votre pont (pont filtrant).

Gardez à l'esprit que de part la nature d'un pont, les mêmes données traversent les deux interfaces, ce qui signifie que vous n'avez besoin de filtrer que sur l'une d'entre elles. Vos déclarations par défaut "Pass all" ressembleront à l'exemple suivant :

```
pass in on ep0 all
pass out on ep0 all
pass in on fxp0 all
pass out on fxp0 all
```

Maintenant, admettons que je souhaite filtrer le trafic dirigé vers les vieux systèmes évoqués précédemment, ne permettant qu'aux protocoles Web et SSH de les atteindre. Dans ce cas, nous allons autoriser tout le trafic entrant et sortant sur l'interface ep0, mais nous filtrerons sur l'interface fxp0 en utilisant `keep state` pour prendre en charge les données retournées :

```
# Autoriser le trafic entrant et sortant sur ep0
pass in quick on ep0 all
pass out quick on ep0 all

# Bloquer tout le trafic sur fxp0
block in on fxp0 all
block out on fxp0 all

pass in quick on fxp0 proto tcp from any to any port {22,
80} \
    flags S/SA keep state
```

Notez que cette règle bloquera tout, à l'exception des requêtes entrantes HTTP et SSH, vers la machine qui fait le pont ainsi qu'en direction des autres noeuds "derrière" elle. Il est possible d'obtenir d'autres résultats en filtrant sur l'autre interface.

Pour surveiller et contrôler le pont que vous venez de créer, servez-vous de la commande [brconfig\(8\)](#) qui peut aussi être utilisée pour créer un pont après le démarrage.

Astuces sur les ponts

- Il est HAUTEMENT recommandé de ne filtrer que sur une seule interface. Bien qu'il soit possible de filtrer sur les deux, vous devez vraiment en connaître toutes les implications avant de pouvoir le faire de la bonne manière.
- En utilisant l'option `blocknonip` de [brconfig\(8\)](#) ou dans [bridgenam.bridge0](#), vous pouvez empêcher le trafic non-IP (comme IPX ou NETBEUI) de passer oûtre votre filtre. Dans certaines

situations, cela peut-être important et vous devez savoir qu'un pont fonctionne pour tous les types de trafic, pas seulement IP.

- Un pont requiert que les interfaces réseau soient en mode "Promiscuous" -- elles écoutent TOUT le trafic réseau, pas seulement celui leur étant dirigé. Ceci augmentera la charge du processeur et du bus. Certaines cartes ne fonctionnent pas correctement dans ce mode, le circuit TI ThunderLAN ([tl\(4\)](#)) en est un exemple.

6.10 - Comment démarrer en utilisant PXE ? (i386, amd64)

PXE ("Preboot Execution Environment", environnement d'exécution avant démarrage) permet de démarrer un ordinateur à partir du réseau plutôt que d'un disque dur, une disquette ou un CD-ROM. A l'origine, cette technologie a été développée par Intel mais est maintenant supportée par la plupart des contrôleurs réseau et des constructeurs. Sachez qu'il existe plusieurs protocoles de démarrage par le réseau, PXE étant relativement récent. Traditionnellement, un démarrage PXE est effectué en utilisant des ROMs présentes sur la carte réseau ou la carte mère elle-même, mais plusieurs disquettes permettant de démarrer en PXE sont disponibles sur différentes sources. Beaucoup de ROMs présentes sur des anciens contrôleurs supportent le démarrage en réseau mais sont incompatibles avec PXE ; s'il est équipé de tels contrôleurs, un système OpenBSD/i386 ou amd64 ne pourra pas être démarré par le réseau.

Comment fonctionne un démarrage PXE ?

Tout d'abord, il serait sage de savoir [comment se déroule le processus de démarrage d'OpenBSD/i386 ?](#) sur les plates-formes i386 et amd64. Au démarrage, chaque interface compatible PXE émet une requête DHCP en broadcast sur le réseau. Le serveur DHCP lui attribue alors une adresse IP en lui indiquant l'emplacement du fichier à exécuter sur le serveur [tftp\(1\)](#). Ce fichier se charge ensuite de gérer le reste du démarrage. Sous OpenBSD [pxeboot](#) remplace le fichier [boot\(8\)](#) standard. Il est alors capable de charger et d'exécuter un noyau (comme `bsd` ou [bsd.rd](#)) à partir du serveur `tftp(1)`.

Comment le mettre en place ?

Le point évident est que vous avez besoin d'une machine ou d'un contrôleur compatible avec un démarrage PXE. Certaines documentations précisent que toutes les cartes modernes sont compatibles PXE, mais c'est tout simplement faux -- de nombreux systèmes à bas prix n'incluent pas de ROMs PXE ou utilisent un ancien protocole de démarrage sur réseau. Vous avez également besoin d'un serveur [DHCP](#) configuré ainsi qu'un serveur TFTP.

En admettant qu'une machine sous OpenBSD serve les fichiers de démarrage (cela n'étant PAS obligatoire), le fichier [dhcpcd.conf](#) de votre serveur DHCP devra contenir la ligne suivante :

```
filename "pxeboot";
```

afin de pouvoir offrir ce fichier de démarrage à une station de travail. Par exemple :

```
shared-network LOCAL-NET {
    option domain-name "example.com";
    option domain-name-servers 192.168.1.3, 192.168.1.5;

    subnet 192.168.1.0 netmask 255.255.255.0 {
        option routers 192.168.1.1;
        filename "pxeboot";
        range 192.168.1.32 192.168.1.127;
        default-lease-time 86400;
        max-lease-time 90000;
    }
}
```

Vous devrez aussi activer le service [tftpd\(8\)](#). Pour ce faire, il suffit de configurer [inetd\(8\)](#). L'installation standard d'OpenBSD fournit une ligne d'exemple dans `inetd.conf` qui devrait vous satisfaire :

```
#tftp dgram udp wait root /usr/libexec/tftpd tftpd -s /
tftpboot
```

Retirez simplement le caractère '#' et envoyez à `inetd(8)` un signal `-HUP` afin que celui-ci relise son fichier `/etc/inetd.conf`. Les fichiers accessibles par `tftpd(8)` sont contenus dans un répertoire particulier, ici `/tftpboot`, que nous utiliserons pour cet exemple. Bien évidemment, ce répertoire doit être créé et contenir les fichiers nécessaires. Pour un démarrage PXE, vous n'aurez typiquement besoin que de quelques fichiers :

- [pxeboot](#), le chargeur de démarrage (offre la même fonction que [boot](#) pour un démarrage sur disque).
- [bsd.rd](#), le noyau d'installation ou `bsd`, un noyau adapté.
- [/etc/boot.conf](#), le fichier de configuration de boot.

Notez que `/etc/boot.conf` n'est nécessaire qu'au cas où vous souhaiteriez démarrer un noyau ne se nommant pas `bsd`, ou que les options par défaut de `pxeboot` ne vous conviennent pas (par exemple si vous utilisez une console série). Vous pouvez tester votre serveur `tftpd(8)` en utilisant un client [tftp\(1\)](#) afin de vérifier que vous pouvez bien récupérer les fichiers nécessaires.

Une fois vos serveurs DHCP et TFTP démarrés, vous êtes prêt pour un essai. Vous devez activer PXE sur votre système ou votre carte réseau ; consultez la documentation fournie avec votre matériel. Une

fois que PXE est activé, vous devriez voir apparaître des lignes similaires à celles-ci :

```
Intel UNDI, PXE-2.0 (build 067)
Copyright (C) 1997,1998 Intel Corporation
```

```
For Realtek RTL 8139(X) PCI Fast Ethernet Controller v1.00
(990420)
```

```
DHCP MAC ADDR: 00 E0 C5 C8 CF E1
CLIENT IP: 192.168.1.76 MASK: 255.255.255.0 DHCP IP:
192.168.1.252
GATEWAY IP: 192.168.1.1
probing: pc0 com0 com1 apm pxe![2.1] mem[540k 28m a20=on]
disk: hd0*
net: mac 00:e0:c5:c8:cf:e1, ip 192.168.1.76, server 192.168.1.252
>> OpenBSD/i386 PXEBOOT 2.03
boot>
```

A présent, vous obtenez l'invite de commandes standard d'OpenBSD. Si vous tapez simplement "bsd.rd" pour récupérer le fichier bsd.rd à partir du serveur TFTP.

```
>> OpenBSD/i386 PXEBOOT 2.03
boot> bsd.rd
booting tftp:bsd.rd: 4375152+733120 [58+122112+105468]=0x516d04
entry point at 0x100120
```

```
Copyright (c) 1982, 1986, 1989, 1991, 1993
```

```
The Regents of the University of California. All rights
reserved.
```

```
Copyright (c) 1995-2010 OpenBSD. All rights reserved. http://
www.OpenBSD.org
```

```
OpenBSD 4.7 (RAMDISK_CD) #353: Wed Mar 17 21:02:53 MST 2010
```

```
...
```

Le noyau d'installation [bsd.rd](#) va maintenant se lancer.

Est-il possible de démarrer d'autres noyaux que bsd.rd en utilisant PXE ?

Oui, bien qu'avec les outils actuellement présents dans OpenBSD, le démarrage PXE ne soit essentiellement prévu que pour installer le système d'exploitation.

6.11 - Protocole de redondance d'adresse commune (CARP)

6.11.1 - Qu'est-ce que CARP et comment fonctionne-t-il ?

CARP ("Common Address Redundancy Protocol") est un outil aidant à la redondance du système en ayant plusieurs ordinateurs créant une interface réseau unique entre eux, afin que si l'un d'eux ne fonctionne plus, un autre puisse répondre à sa place ; cet outil permet aussi de mettre en place un certain partage de charge entre les différents systèmes. CARP est une avancée par rapport au VRRP ("Virtual Router Redundancy Protocol" - protocole de redondance de routeur virtuel) standard. Il a été développé une fois que VRRP fut considéré comme non suffisamment libre à cause d'un brevet Cisco pouvant le couvrir. Pour plus d'informations sur les origines de CARP et les problèmes légaux entourant VRRP, rendez-vous sur [cette page](#).

Afin d'éviter tout problème légal, Ryan McBride (avec l'aide de Michael Shalayeff, Marco Pfatschbacher et Markus Friedl) a conçu CARP de manière à être fondamentalement différent. Si l'inclusion de la cryptographie reste le changement le plus visible, il n'en reste pas moins un seul parmi beaucoup d' autres.

Comment fonctionne-t-il ? CARP est un protocole multicast. Il regroupe plusieurs systèmes physiques en une ou plusieurs adresses virtuelles. L'un d'eux est le maître et répond aux paquets destinés au groupe, alors que les autres se comportent comme des "hot spares" (remplacement à chaud et automatique du maître). Peu importe les adresses IP et MAC de l'interface physique locale, les paquets envoyés vers l'adresse CARP reviennent avec les informations CARP.

A intervalles paramétrables, le maître annonce son état sur le port IP 112. Si le maître est déconnecté, les autres systèmes du groupe CARP commencent à annoncer leur présence. L'hôte qui parvient à s'annoncer le plus fréquemment devient le nouveau maître. Lorsque le système principal est reconnecté, il se transforme par défaut en hôte de secours, bien qu'il soit possible de configurer un hôte spécifique en tant que maître par défaut lorsque cela est faisable (ex. un hôte est un Sun Fire V120 rapide et les autres sont des SPARCstation IPCs, comparativement plus lentes).

Bien que les équipements à haute redondance et tolérance de pannes minimisent le besoin de CARP, ils ne le suppriment pas. Il n'existe pas de matériel à tolérance de pannes capable de gérer le fait que quelqu'un débranche le cordon d'alimentation ou que l'administrateur système tape `reboot` dans la mauvaise fenêtre. L'utilisation de CARP aide à rendre transparent une mise à jour ou un redémarrage du point de vue des utilisateurs, ainsi que le test d'un programme ou une mise à jour matérielle : si cela ne fonctionne pas, vous pouvez basculer sur le reste du groupe jusqu'à ce que le problème soit résolu.

Il existe cependant certaines situations où CARP ne pourra pas vous venir en aide. La conception de CARP est telle que les membres d'un groupe doivent appartenir au même réseau physique avec une adresse IP fixe, bien que depuis l'introduction de la directive `carpdev` il n'y ait plus besoin d'adresse IP

pour les interfaces physiques. De même, les services nécessitant une connexion constante au serveur (tels que SSH ou IRC) ne seront pas transférés de manière transparente aux autres systèmes, bien que dans ce cas, CARP puisse aider à minimiser le temps d'arrêt. CARP, lui-même, ne synchronise pas les données entre applications, ceci doit être fait au travers de "procédés alternatifs" tels que [pfsync\(4\)](#) (pour du filtrage redondant), [rsync](#), pour dupliquer manuellement les données entre les machines, ou tout ce qui semble approprié à votre application.

6.11.2 - Configuration

Le contrôle de CARP s'effectue en deux endroits : [sysctl\(8\)](#) et [ifconfig\(8\)](#). Regardons tout d'abord les paramètres sysctls.

La première variable sysctl, `net.inet.carp.allow`, définit si l'hôte gère ou non les paquets CARP. Bien évidemment, ceci est nécessaire à l'utilisation de CARP. Cette variable sysctl est activée par défaut.

La seconde, `net.inet.carp.log`, enregistre les changements d'états de CARP, les mauvais paquets et d'autres erreurs. Activée pour enregistrer les changements d'états par défaut.

La troisième, `net.inet.carp.preempt` active la sélection naturelle parmi les hôtes CARP. Le plus à même d'effectuer le travail (à savoir, celui qui est capable de s'annoncer le plus fréquemment) deviendra le maître. Celle-ci est désactivée par défaut, ce qui signifie qu'un système qui n'est pas maître ne tentera pas de (re)gagner ce status.

Toutes ces variables sysctl sont documentées dans [sysctl\(3\)](#).

Pour ce qui concerne le reste de la configuration de CARP, nous dépendrons d' [ifconfig\(8\)](#). Les commandes `advbase` et `advskew`, spécifiques à CARP, définissent l'intervalle entre les annonces CARP. La formule (en secondes) est `advskew` divisée par 256 puis ajoutée à `advbase`. `advbase` peut être utilisée pour diminuer le trafic réseau ou autoriser une plus grande latence avant qu'un hôte de sauvegarde ne prenne le relais ; `advskew` vous permet de contrôler quel hôte sera le maître sans trop de délais de basculement (si cela est nécessaire).

Ensuite, `pass` crée un mot de passe et `vhid` définit le numéro d'identification d'hôte virtuel du groupe CARP. Vous devez assigner un numéro unique pour chaque groupe CARP, même s'ils partagent (pour des raisons de répartition de charge) la même adresse IP. CARP est limité à 255 groupes.

Enfin, `carpdev` spécifie l'interface appartenant à ce groupe CARP particulier. Par défaut, n'importe quelle interface possédant une adresse IP dans le même sous-réseau que celui assigné à l'interface CARP sera utilisée.

Voyons l'emploi de ces différents paramètres dans une configuration simple. Admettons que vous

déployez deux serveurs Web identiques, *rachael* (192.168.0.5) et *pris* (192.168.0.6), afin de remplacer un ancien système à l'adresse 192.168.0.7. Les commandes :

```
rachael# ifconfig carp0 create
rachael# ifconfig carp0 vhid 1 pass tyrell carpdev fxp0 \
192.168.0.7 netmask 255.255.255.0
```

créent l'interface `carp0` et lui donnent un `vhid` de 1, le mot de passe *tyrell*, l'adresse IP 192.168.0.7 de masque 255.255.255.0 et assignent `fxp0` en tant qu'interface membre. Afin de rendre ces changements permanents après un redémarrage, vous pouvez créer un fichier `/etc/hostname.carp0` ressemblant à :

```
inet 192.168.0.7 255.255.255.0 192.168.0.255 vhid 1 pass
tyrell carpdev
fxp0
```

Notez qu'il faut ajouter l'adresse de diffusion ("broadcast") en plus du `vhid` et du mot de passe. Ce champ est nécessaire et est souvent cause d'erreurs en cas d'oubli.

Faites la même chose sur *pris*. Le système démarrant en premier son interface CARP deviendra le maître (en admettant que "preempt" est désactivé ; sinon, c'est l'inverse qui se produit).

Mais admettons que vous ne déployez pas cette solution à partir de zéro. *Rachael* était déjà en place à l'adresse 192.168.0.7. Comment allez-vous gérer cette situation ? Heureusement, CARP sait gérer ce genre de situation. Vous assignez simplement l'adresse à l'interface CARP et laissez vide l'adresse de l'interface physique spécifiée par le mot-clef ``carpdev'`. Cependant, il est plus propre d'avoir une adresse IP différente pour chaque système, cela rend les accès et la surveillance individuels bien plus simples.

Ajoutons un nouveau niveau de complexité ; nous souhaitons que *rachael* reste maître lorsque cela est possible. Plusieurs raisons peuvent nous pousser à cela : différences matérielles, simple préjudice, "si ce système n'est pas maître, c'est qu'il y a un problème", ou simplement connaître le maître par défaut sans avoir à recourir à des scripts envoyant par courriel l'analyse de la sortie d'`ifconfig`.

Sur *rachael*, nous utiliserons la variable `sysctl` créée précédemment puis nous éditerons `/etc/sysctl.conf` afin de rendre ce paramètre permanent.

```
rachael# sysctl net.inet.carp.preempt=1
```

Configurons également *pris* :

```
pris# ifconfig carp0 advskew 100
```

Ceci décale les annonces de *pris*, ce qui signifie que *rachael* sera maître s'il est fonctionnel.

Notez que si vous utilisez PF sur une machine CARP, vous devez utiliser "proto carp" sur toutes les interfaces concernées, avec une ligne similaire à :

```
pass on fxp0 proto carp keep state
```

6.11.3 - Répartition de charge ("load balancing")

Effectuons une avance rapide de quelques mois. Notre entreprise de l'exemple précédent a grossi au point qu'un seul serveur Web interne arrive tout juste à gérer la charge. Que faire ? CARP à la rescousse. Il est temps d'essayer la répartition de charge. Créons une nouvelle interface et un nouveau groupe CARP sur *rachael* :

```
rachael# ifconfig carp1 create
rachael# ifconfig carp1 vhid 2 advskew 100 pass bryant
carpdev fxp0 \
    192.168.0.7 netmask 255.255.255.0
```

Sur *pris*, nous allons également créer un nouveau groupe et une nouvelle interface, puis activer la variable `sysctl "preempt"` :

```
pris# ifconfig carp1 create
pris# ifconfig carp1 vhid 2 pass bryant carpdev fxp0 \
    192.168.0.7 netmask 255.255.255.0
pris# sysctl net.inet.carp.preempt=1
```

A présent nous avons deux groupes CARP avec la même adresse IP. Chaque groupe est dirigé vers un hôte différent, ce qui signifie que *rachael* restera maître du premier groupe, mais que *pris* lui succédera comme maître dans le second.

Bien que ces exemples ne soient que pour un cluster de deux machines, le même principe reste valable avec plus de systèmes. Notez en revanche qu'il n'est pas assuré que vous parveniez à une distribution de 50/50 sur les deux machines : CARP utilise une empreinte de l'adresse IP d'origine afin de savoir quel système répondra à la requête, indépendamment de la charge.

6.11.4 - Plus d'information sur CARP

- [carp\(4\)](#)
- [ifconfig\(8\)](#)
- [sysctl\(8\)](#)

- [sysctl\(3\)](#)
- ["Firewall Failover with pfsync and CARP"](#) (Redondance de pare-feu avec pfsync et CARP) par Ryan McBride

6.12 - Utiliser OpenNTPD

Une horloge précise est importante pour plusieurs applications informatiques. Cependant, plusieurs personnes ont remarqué que leur montre à 5 euros est plus précise que leur ordinateur à 2000 euros. Certes, connaître la date et l'heure est une chose importante. Mais souvent, il est plus important de synchroniser plusieurs machines afin qu'elles s'accordent sur la date et l'heure actuelles. Durant un temps, [ntp.org](#) a produit une application "Network Time Protocol" ([RFC1305](#), [RFC2030](#)), disponible dans le système des [ports](#), et qui peut être utilisée pour synchroniser les horloges des machines à travers Internet. Cependant, c'est un programme difficile à configurer, dont le code est difficile à auditer, et qui nécessite une quantité de mémoire vive conséquente. En somme, c'est un programme qui rend bien des services à certaines personnes mais qui est loin d'être une solution pour tous les utilisateurs.

[OpenNTPD](#) est une tentative pour résoudre certains de ces problèmes en créant un programme compatible NTP qui soit facile à administrer, sûr, simple et qui vous permette d'avoir une horloge exacte sur votre ordinateur. Le service [ntpd\(8\)](#) d'OpenBSD est contrôlé travers le fichier de configuration [/etc/ntpd.conf](#) facile à comprendre.

En activant tout simplement [ntpd\(8\)](#) dans [rc.conf.local](#), l'horloge de votre ordinateur avancera légèrement et restera synchronisée avec les serveurs de [pool.ntp.org](#), une collection de serveurs de temps publics. Une fois votre horloge réglée de manière précise, [ntpd](#) la gardera à un haut degré d'exactitude. Cependant, si votre horloge est décalée de plus de quelques minutes, il est *vivement* recommandé de la régler plus précisément au démarrage car la synchronisation d'une horloge très décalée pourrait prendre des jours, voire des semaines. Vous pouvez la régler manuellement en utilisant l'option "-s" de [ntpd](#) ou tout autre moyen permettant de mettre l'horloge de votre système à l'heure.

6.12.1 - "Mais OpenNTPD n'est pas aussi exact que l'application de ntp.org !"

C'est probablement vrai. Ce n'est pas un des [objectifs de la conception](#) d'OpenNTPD. Ce programme a été d'abord conçu pour être libre, simple, fiable et sécurisé. Si vous avez réellement besoin d'une précision de l'ordre de la micro- seconde, alors OpenNTPD n'est pas pour vous. Utilisez plutôt le [ntpd](#) de [ntp.org](#). Ce dernier restera disponible dans les [ports](#) et comme paquetage. Nous n'avons aucun plan ou souhait de rendre OpenNTPD bouffi en y intégrant toutes les fonctionnalités possibles et imaginables.

6.12.2 - "Quelqu'un a prétendu qu'OpenNTPD était 'nuisible' !"

Certaines personnes n'ont pas compris le but d'OpenNTPD, à savoir un moyen de synchroniser l'horloge

de votre ordinateur qui soit simple, sécurisé et facile à maintenir. Si la maintenance d'un temps aussi exact que possible est une chose importante, un certain nombre d'utilisateurs ont rapporté de meilleurs résultats avec OpenNTPD par rapport au programme ntpd de ntp.org. Si la sécurité est importante, le code d'OpenNTPD est beaucoup plus lisible (et donc auditable) et a été écrit en utilisant les appels de fonction natifs d'OpenBSD tels que [strncpy](#), au lieu d'utiliser des fonctions plus portables telles que [strcpy](#). Il a été écrit avec des principes sécurité dès le début. La sécurité n'a pas été ajoutée par la suite. Si le fait d'amener autant de personnes que possible à utiliser la synchronisation du temps est une chose précieuse, OpenNTPD rend une telle entreprise très facile pour beaucoup de personnes. Si cela est "nuisible", alors nous souhaitons ce genre de nuisance.

Il y a des applications pour lesquelles le programme ntpd de ntp.org est plus approprié, cependant nous croyons qu'OpenNTPD est plus que suffisant pour une large majorité d'utilisateurs.

Une réponse plus complète donnée par l'un des mainteneurs d'OpenNTPD et concernant ce fait est disponible [ici](#).

6.12.3 - Pourquoi mes autres machines ne peuvent pas se synchroniser avec OpenNTPD?

Par défaut, ntpd(8) n'écoute sur aucune adresse. Ainsi, pour l'utiliser en tant que serveur, vous devez décommenter la ligne "#listen on *" dans [/etc/ntp.conf](#) et redémarrer le service ntpd(8). Bien entendu, si vous préférez écouter sur une adresse particulière plutôt que sur toutes celles disponibles, remplacez "*" par l'adresse correspondante.

Lorsque ntpd(8) écoute il se peut que les autres machines ne puissent toujours pas se synchroniser ! Lorsque que le service ntpd(8) vient juste de démarrer (par exemple si vous venez de le redémarrer suite à un changement dans ntpd.conf) il se peut qu'il refuse la synchronisation des clients en attendant que lui-même ajuste sa propre horloge a un certain degré de stabilité. Lorsque ntpd(8) considère que son information temporelle est stable, il l'annonce par le message "clock now synced" dans `/var/log/daemon`. Même si l'horloge système est assez correcte au démarrage, cela peut prendre jusqu'à 10 minutes avant d'être synchronisé et des heures voire des jours si celle-ci est décalée.

6.13 - Quels sont les types de cartes Sans Fil supportées par OpenBSD ?

OpenBSD supporte un certain nombre de chipsets sans fil :

- [acx\(4\)](#) TI ACX100/ACX111. (NFF) (AP)
- [an\(4\)](#) Aironet Communications 4500/4800.
- [ath\(4\)](#) pilote pour Atheros 802.11a/b/g. (AP)

- [athn\(4\)](#) pilote pour Atheros 80211/a/g/n.
- [atu\(4\)](#) Atmel AT76C50x USB 802.11b
- [atw\(4\)](#) ADMtek ADM8211.
- [awi\(4\)](#) AMD 802.11 PCnet Mobile.
- [bwi\(4\)](#) Broadcom AirForce 802.11b/g
- [cnwi\(4\)](#) Xircom CreditCard Netwave
- [ipw\(4\)](#) Intel PRO/Wireless 2100 802.11b. (NFF)
- [iwi\(4\)](#) Intel PRO/Wireless 2200BG/2225BG/2915ABG 802.11a/b/g. (NFF)
- [iwn\(4\)](#) Intel WiFi Link 4965/5100/5300 802.11a/b/g/Draft-N wireless.
- [malo\(4\)](#) Marvell Libertas 802.11b/g
- [pgt\(4\)](#) Conexant/Intersil Prism GT Full-MAC 802.11a/b/g
- [ral\(4\)](#) et [ural\(4\)](#) [USB] Ralink Technology RT25x0 802.11a/b/g. (AP)
- [ray\(4\)](#) Raytheon Raylink/WebGear Aviator 802.11FH
- [rtw\(4\)](#) Realtek 8180 802.11b. (AP)
- [rum\(4\)](#) Ralink Technology RT2501USB. (AP)
- [run\(4\)](#) Ralink Technology USB 802.11a/b/g/Draft-N
- [uath\(4\)](#) Atheros USB 802.11a/b/g
- [upgt\(4\)](#) Conexant/Intersil PrismGT SoftMAC USB 802.11b/g
- [urtw\(4\)](#) Realtek RTL8187L USB 802.11b/g
- [wi\(4\)](#) Prism2/2.5/3. (AP)
- [wpi\(4\)](#) Intel PRO/Wireless 3945ABG. (NFF)
- [zyd\(4\)](#) ZyDAS ZD1211/ZD1211B USB 802.11b/g

(AP) signifie que la carte peut être utilisée comme point d'accès.

(NFF) signifie que la puce nécessite un firmware non-libre qui ne peut pas être inclus dans OpenBSD.

Les adaptateurs basés sur ces composants peuvent être utilisés de la même façon que n'importe quel autre adaptateur réseau pour connecter un système OpenBSD à un réseau sans fil existant, et sont configurés de façon standard à l'aide d' [ifconfig\(8\)](#) (prière de consulter les pages du manuel pour les détails précis). Certaines cartes peuvent aussi être utilisées en mode "Host-Based Access Point", ce qui leur permet d'être utilisées comme un point d'accès sans fil pour votre réseau, intégré à votre pare-feu.

Notez que pour utiliser les cartes à base de chipset Intel, vous devrez récupérer les firmwares dont Intel refuse la [libre](#) distribution, ce qui empêche qu'ils soient inclus dans OpenBSD. Lorsque cela est possible, les pages de manuel correspondantes incluent les informations de contact afin que vous puissiez contacter les bonnes personnes chez les constructeurs pour leur faire savoir ce que vous pensez de cette situation ou de les informer sur le produit que vous avez acheté à la place.

Une autre méthode pour permettre à un pare-feu OpenBSD de fournir un accès sans fil est d'utiliser une

carte conventionnelle et un point d'accès externe en mode pont. Cette méthode a l'avantage de vous permettre de positionner l'antenne là où elle est la plus efficace.

6.14 - Comment puis-je mettre en oeuvre un routage multi-chemin à coût égal ?

Le routage multi-chemin à coût égal permet d'avoir de multiples routes vers le même réseau, tel que la route par défaut (0.0.0.0/0), dans la table de routage. Lorsque le noyau recherche une route pour déterminer où acheminer des paquets destinés à un réseau donné, il pourra alors choisir parmi les routes à coût égal. Dans la majorité des cas, un routage multi-chemin est utilisé pour redonder des connexions "montantes" telles que les connexions vers Internet.

La commande [route\(8\)](#) est employée pour ajouter, modifier et supprimer des routes dans la table de routage. L'argument `-mpath` est utilisé lors de l'ajout de routes s'inscrivant dans un politique de routage multi-chemin.

```
# route add -mpath default 10.130.128.1
# route add -mpath default 10.132.0.1
```

Vérifiez les routes :

```
# netstat -rnf inet | grep default
default      10.130.128.1      UGS          2          134
-           fxp1
default      10.132.0.1        UGS          0          172
-           fxp2
```

A travers cet exemple, nous pouvons voir que la route par défaut pointe vers 10.130.128.1, accessible par le biais de l'interface fxp1 tandis que l'autre route par défaut pointe vers 10.132.0.1 qui est accessible par le biais de l'interface fxp2.

Etant donné que le fichier [mygate\(5\)](#) ne supporte pas encore les routes par défaut multi-chemin, les commandes précitées devraient être ajoutées à la fin des fichiers [hostname.if\(5\)](#) correspondant aux interfaces fxp1 et fxp2. Le fichier `/etc/mygate` devrait être supprimé.

/etc/hostname.fxp1

```
!route add -mpath default 10.130.128.1
```

/etc/hostname.fxp2

```
!route add -mpath default 10.132.0.1
```

Enfin, n'oubliez pas d'activer le support de routes multi-chemin à l'aide de la variable `sysctl(3)` adéquate.

```
# sysctl net.inet.ip.multipath=1
# sysctl net.inet6.ip6.multipath=1
```

Assurez-vous d'éditer [sysctl.conf\(5\)](#) pour rendre les modifications permanentes.

Maintenant essayez d'effectuer un traceroute vers des destinations différentes. Le noyau partagera le trafic entre les différentes routes multi-chemin.

```
# traceroute -n 154.11.0.4
traceroute to 154.11.0.4 (154.11.0.4), 64 hops max, 60 byte
packets
 1  10.130.128.1  19.337 ms  18.194 ms  18.849 ms
 2  154.11.95.170 17.642 ms  18.176 ms  17.731 ms
 3  154.11.5.33   110.486 ms 19.478 ms 100.949 ms
 4  154.11.0.4   32.772 ms  33.534 ms  32.835 ms

# traceroute -n 154.11.0.5
traceroute to 154.11.0.5 (154.11.0.5), 64 hops max, 60 byte
packets
 1  10.132.0.1  14.175 ms  14.503 ms  14.58 ms
 2  154.11.95.38 13.664 ms  13.962 ms  13.445 ms
 3  208.38.16.151 13.964 ms  13.347 ms  13.788 ms
 4  154.11.0.5   30.177 ms  30.95 ms  30.593 ms
```

Pour de plus amples informations concernant la manière de choisir une route, veuillez vous référer à la [RFC2992](#), "Analysis of an Equal-Cost Multi-Path Algorithm".

Il est utile de noter que si une interface utilisée par une route multi-chemin tombe (perte de signal par exemple), le noyau acheminera tout de même des paquets à travers cette interface. Ces paquets seront bien entendus perdus. Il est hautement recommandé d'utiliser [ifstated\(8\)](#) pour vérifier l'état des interfaces et ajuster la table de routage si besoin.

[\[Index de la FAQ\]](#) [\[Section 5 - Construire le Système à partir des Sources\]](#) [\[Section 7 - Contrôles du clavier et de l'affichage\]](#)



www@openbsd.org

\$OpenBSD: faq6.html,v 1.102 2010/05/23 16:20:41 ajacoutot Exp \$

7 - Contrôles du clavier et de l'affichage

Table des matières

- [7.1 - Comment puis-je redéfinir le clavier ? \(wscons\)](#)
 - [7.2 - OpenBSD dispose t-il d'un support de la souris en mode console ?](#)
 - [7.3 - Accéder au tampon de la console. \(amd64, i386, quelques Alpha\)](#)
 - [7.4 - Comment puis-je changer de console ? \(amd64, i386, Zaurus, quelques Alpha\)](#)
 - [7.5 - Comment puis-je utiliser une résolution console de 80x50 ? \(amd64, i386, quelques Alpha\)](#)
 - [7.6 - Comment puis-je utiliser une console série ?](#)
 - [7.7 - Comment effacer la console ? \(wscons\)](#)
 - [7.8 - TOUT CE QUE JE TAPE A LA CONNEXION EST EN MAJUSCULES !](#)
 - [7.9 - Qu'est ce que tmux\(1\) ?](#)
-

7.1 - Comment puis-je redéfinir le clavier ? (wscons)

En utilisant le pilote console [wscons\(4\)](#) : [alpha](#), [amd64](#), [hppa](#), [i386](#), [mac68k](#), [macppc](#), [sparc](#), [sparc64](#), [vax](#), et [zaurus](#).

Avec [wscons\(4\)](#), plusieurs options peuvent être configurées en utilisant l'utilitaire [wsconsctl\(8\)](#). Par exemple, pour changer la configuration des touches du clavier avec [wsconsctl\(8\)](#) vous devez exécuter ceci :

```
# wsconsctl keyboard.encoding=fr
```

Dans l'exemple suivant, on redéfinit la touche "Caps Lock" en "Control L" :

```
# wsconsctl keyboard.map+="keysym Caps_Lock = Control_L"
```

7.2 - OpenBSD dispose t-il d'un support de la souris en mode console ?

Pour les plates-formes [alpha](#), [amd64](#) et [i386](#), OpenBSD fournit [wsmoused\(8\)](#), un port du service moused (8) de FreeBSD. Il peut être lancé au démarrage en copiant et éditant la ligne appropriée de `rc.conf` dans [rc.conf.local\(8\)](#).

7.3 - Accéder au tampon de la console (*amd64, i386, quelques Alpha*)

Sur quelques plates-formes, OpenBSD offre la possibilité de revenir en arrière dans la console. Ceci permet de voir les informations inscrites précédemment sur votre écran. Pour monter et descendre dans le tampon, utilisez simplement les combinaisons de touches `[SHIFT] + [PGUP]` et `[SHIFT] + [PGDN]`.

La valeur par défaut de pages que vous pouvez consulter en revenant en arrière dans la console est de 8. Ceci est une fonctionnalité du driver [vga\(4\)](#), donc cela ne fonctionnera pas sans carte vga quelque soit la plate-forme (beaucoup de systèmes Alpha ont une carte vidéo de type TGA).

Par manque d'espace, les noyaux d'installation ne fournissent pas la possibilité de revenir en arrière dans la console. [Changer de console](#) effacera le tampon de retour en arrière.

7.4 - Comment puis-je changer de console ? (*amd64, i386, Zaurus, quelques Alpha*)

Sur `amd64`, `i386` et quelques systèmes Alpha, avec les cartes [vga\(4\)](#), OpenBSD fournit six terminaux virtuels par défaut, allant de `/dev/ttyC0` à `/dev/ttyC5`. `ttyC4` est réservée pour l'utilisation de X, laissant donc cinq consoles texte. Vous pouvez passer de l'une à l'autre en utilisant `[CTRL] + [ALT] + [F1]`, `[CTRL] + [ALT] + [F2]`, `[CTRL] + [ALT] + [F3]`, `[CTRL] + [ALT] + [F4]` et `[CTRL] + [ALT] + [F6]`.

L'environnement X utilise `ttyC4`, `[CTRL] + [ALT] + [F5]`. Lorsque vous utilisez X, les combinaisons `[CTRL] + [ALT] + [Fn]` vous basculeront sur les consoles textes ; `[CTRL] + [ALT] + [F5]` vous ramènera à l'environnement graphique.

Si vous souhaitez avoir plus de console virtuelles que le nombre par défaut, utilisez la commande [wsconscfg\(8\)](#) pour créer des consoles pour `ttyC6`, `ttyC7` et au delà. Par exemple :

```
wsconscfg -t 80x25 6
```

Créera un terminal virtuel pour `ttyC6`, accessible par `[CTRL] + [ALT] + [F7]`. N'oubliez pas d'ajouter cette commande à votre fichier [rc.local\(8\)](#) si vous souhaitez cette console supplémentaire au prochain redémarrage de votre ordinateur.

Notez que vous n'obtiendrez pas une invite `login:` sur la nouvelle console virtuelle tant que vous ne l'aurez pas mis à "on" dans [/etc/ttys\(5\)](#), puis au choix redémarré ou envoyé un signal HUP à [init\(8\)](#) par le biais de [kill\(1\)](#).

Sur le Zaurus, deux terminaux virtuels (`/dev/ttyC0` and `/dev/ttyC1`) sont disponibles par défaut. Ils sont accessibles à l'aide des combinaisons de touches `[ALT] + [CALENDAR]` et `[ALT] + [ADDRESS]` (La touche `[ALT]` se situe à gauche de la touche `[CTRL] key`).

7.5 - Comment puis-je utiliser une résolution console de 80x50 ? (*amd64, i386, quelques Alpha*)

Les utilisateurs de systèmes `i386`, `amd64` et `Alpha` (VGA) ont normalement une console de 25 lignes de 80 caractères. Cependant, beaucoup de cartes vidéo VGA sont capables d'afficher une résolution supérieure à 50 lignes de 80 caractères.

Premièrement, une police supportant la résolution désirée doit être chargée en utilisant la commande [wsfontload](#). L'écran standard 80x25 utilise des polices de 8x16 pixels, pour doubler la résolution verticale, nous allons devoir utiliser des polices de 8x8 pixels.

Ensuite, nous allons devoir supprimer et recréer une [console virtuelle](#) de la résolution désirée en utilisant la commande [wsconscfg](#).

Ceci peut être fait automatiquement en ajoutant les lignes suivantes à la fin de votre fichier [rc.local](#) :

```
wsfontload -h 8 -e ibm /usr/share/misc/pcvtfonts/vt2201.808
wsconscfg -dF 5
wsconscfg -t 80x50 5
```

Comme pour toutes les configurations de votre système, il est recommandé de prendre le temps de lire les pages de manuels (`man`) afin de comprendre ce que font les commandes ci-dessus.

La première commande ci-dessus charge une police 8x8. La seconde supprime la console 5 (qui aurait été accessible via `[CTRL] + [ALT] + [F6]`). La troisième crée une nouvelle console 5 avec une résolution de 50 lignes de 80 caractères chacune. Si vous procédez de la sorte, vous verrez votre console principale, ainsi que les trois autres consoles par défaut, en mode standard 80x25, mais une nouvelle

console 5 en mode 80x50 accessible via [CTRL] + [ALT] + [F6].

Rappelez vous que [CTRL] + [ALT] + [F1] est la console 0 (ttyC0). Si vous désirez modifier les autres consoles, réalisez simplement les étapes de suppression et d'ajout des consoles pour lesquelles vous désirez une résolution de 80x50.

Vous devriez éviter de modifier la console 4 (ttyC4, [CTRL] + [ALT] + [F5]), qui est utilisée par X en tant que console graphique. Il n'est pas non plus possible de changer la résolution de la première console (exemple ttyC0).

Comme certains ont pu le remarquer, toutes ces commandes peuvent être tapées sur la ligne de commande, en tant que root, ou (mieux), en utilisant [sudo\(8\)](#).

Note : ceci ne fonctionnera pas avec toutes les cartes vidéo. Malheureusement, toutes les cartes vidéo ne supportent pas les polices fournies qui sont nécessaires à [wscons](#) pour parvenir à configurer le mode texte 80x50. Dans ces cas là vous devriez reconsidérer le fait d'utiliser X.

7.6 - Comment puis-je utiliser une console série ?

Il y a plusieurs raisons pour que vous souhaitiez utiliser une console série sur votre système OpenBSD :

- Enregistrer la sortie de la console (pour de la documentation).
- Administration distante.
- Maintenance simplifiée d'un grand nombre de machines.
- Obtenir un très utile dmesg de machines depuis lesquelles cette obtention aurait été difficile.
- Obtenir une sortie "trace" et de "ps" si votre système se crash, pour que les développeurs aient une chance de régler le problème.

OpenBSD supporte les consoles séries sur la plupart des plates-formes, cependant certains détails varient considérablement entre plates-formes.

Notez que l'interfaçage série n'est PAS une tâche triviale -- vous serez souvent obligé d'utiliser des câbles peu communs, et les ports ne sont pas standardisés entre machines, dans certains cas, parfois même incompatibles sur une machine. Nous considérons que vous savez comment utiliser le câble adéquat entre votre ordinateur et le matériel faisant office de terminal série. Un tutorial complet sur l'interfaçage série sortirait du cadre de ce document, cependant, nous vous donnons un petit conseil : ce n'est pas parce que les prises se connectent que cela fonctionnera.

Modification de */etc/ttys*

Deux conditions doivent être réunies pour obtenir une console série fonctionnelle sur un système

OpenBSD. Premièrement, vous devez indiquer à OpenBSD d'utiliser le port série comme console pour les modes "status" et "single user". Cette étape dépend de la plate-forme que vous utilisez. Ensuite, vous devez activer votre port série comme terminal interactif, de façon à ce qu'un utilisateur puisse s'y connecter en mode multi-utilisateurs. Cette partie est relativement similaire sur les différentes plates-formes, et est détaillée ici.

Les sessions de terminaux sont contrôlées par le fichier [/etc/ttys](#). Avant qu'OpenBSD ne vous présente une invite "login:" sur un périphérique, celui-ci doit être activé dans [/etc/ttys](#), après tout, il y a d'autres utilisations possibles que celle d'un terminal pour un port série. Sur les plates-formes qui ont un clavier et un écran attaché comme console, le terminal série est désactivé par défaut. Nous utiliserons la plate-forme i386 comme exemple. Dans ce cas vous devez éditer la ligne suivante :

```
tty00    "/usr/libexec/getty std.9600"    unknown off
```

ainsi (par exemple) :

```
tty00    "/usr/libexec/getty std.9600"    vt220    on secure
```

Ici, `tty00` est le port série que nous utilisons comme console. `vt220` représente l'entrée [termcap\(5\)](#) équivalente à votre terminal (d'autres options similaires pourraient inclure `vt100`, `xterm`, etc.). Le "on" active le [getty](#) pour le port série afin que l'invite "login:" soit présentée, le "secure" permet une connexion root (uid 0) sur cette console (ce qui peut être, ou pas, ce que vous souhaitez), et le "9600" est le débit en baud du terminal. Essayez de ne pas augmenter le débit en baud du terminal jusqu'au maximum de ce que votre matériel peut supporter car cela risque de poser plus de problèmes que de bénéfices. La plupart des systèmes ont une vitesse "par défaut" (supportée par défaut par la ROM de démarrage et/ou le chargeur de démarrage, généralement 9600), utilisez celle-ci à moins que vous n'ayez une raison valable de faire autrement.

Notez que vous pouvez utiliser une console série pour l'installation sans devoir réaliser tout cela étant donné que le système est lancé en mode "single user" et n'utilise pas *getty* pour la connexion.

Sur certaines plates-formes ou configurations, vous devrez démarrer le système en mode "single user" pour faire cette modification si vous n'avez rien d'autre qu'une console série.

amd64 et i386

Pour que la procédure de démarrage utilise le port série comme console, créez ou éditez votre fichier [/etc/boot.conf](#) pour y inclure la ligne :

```
set tty com0
```

afin d'utiliser votre premier port série comme console. La bande passante en baud par défaut est 9600bps, ceci peut être changé par une ligne du fichier `/etc/boot.conf` en utilisant l'option `stty`. Ce fichier est stocké dans votre média de démarrage, qui peut aussi être votre disquette d'installation, la commande peut être entrée à l'invite `boot>` depuis le [chargeur de démarrage de second stage OpenBSD](#) pour une première utilisation de la commande série.

notes pour amd64 et i386 :

- OpenBSD numérote les ports séries en commençant à `tty00`, les labels DOS/Windows commencent eux à `COM1`. Gardez donc à l'esprit que `tty02` est `COM3`, et non `COM2`.
- Quelques systèmes pourront fonctionner sans carte vidéo présente dans la machine, mais sûrement pas tous -- la plupart des systèmes considèrent cela comme une erreur. Certaines machines refuseront même de fonctionner sans clavier attaché.
- Certains systèmes sont capables de rediriger les activités BIOS du clavier et de l'écran sur un port série au travers d'une option de configuration, pour que la machine puisse être complètement maintenue par le port série. Les résultats que vous obtiendrez varieront, quelques BIOSs peuvent empêcher le chargeur de démarrage de voir le port série, et donc le noyau ne saura pas qu'il doit l'utiliser. Certains BIOS ont une option "Continuer la redirection de la console après le POST" (Power On Self Test), cela devra être désactivé, pour que le chargeur de démarrage et le noyau puissent disposer de leur propre console. Malheureusement cette option n'est pas universelle.
- Les ordinateurs compatibles PC, ne sont pas pensés dans le but d'être utilisés depuis une console série, à la différence d'autres plates-formes. Même sur les systèmes qui supportent une console série, celle-ci doit être activée dans la configuration BIOS -- et si les informations de configuration de celui-ci sont corrompues, vous serez face à un système qui recherche de nouveau un moniteur et un clavier. Vous devrez généralement, disposer d'un moyen d'obtenir un clavier et un moniteur pour vos systèmes amd64 et i386 en cas d'urgence.
- Vous devrez éditer `/etc/ttys` comme décrit [ci-dessus](#).

SPARC et UltraSPARC

Ces machines sont construites dans le but d'être totalement configurables par console série. Retirez simplement le clavier de la machine et le système lancera une console série.

Notes SPARC et UltraSPARC

- Les ports sur un SPARC sont appelés `ttya`, `ttyb`, etc.
- A la différence des autres plates-formes, il n'est pas nécessaire de faire des changements à `/etc/ttys` pour utiliser une console série.
- Les systèmes SPARC/UltraSPARC interprètent le signal BREAK sur le port série de la même manière que la commande STOP-A, et renvoient le système à l'invite "Forth", stoppant toute application et le système lui-même à ce point. Cela est intéressant lorsque souhaité, mais

malheureusement, quelques terminaux série lors de l'arrêt et quelques matériels de switch RS-232 envoient quelques codes à l'ordinateur que celui-ci interprète comme un signal BREAK, arrêtant la machine. Testez cela avant de passer la machine en production.

- Si vous avez un clavier et une souris connectés, vous pouvez toujours forcer l'utilisation de la console série en utilisant les commandes suivantes à l'invite `ok` :

```
ok setenv input-device ttya
ok setenv output-device ttya
ok reset
```

Si le clavier et l'écran (`ttyC0`) sont activés dans `/etc/ttys` ([ci-dessus](#)), vous pouvez utiliser le clavier et le moniteur dans X.

MacPPC

Les machines MacPPC, sont configurées pour utiliser une console série au travers de "OpenFirmware". L'utilisation des commandes

```
ok setenv output-device scca
ok setenv input-device scca
ok reset-all
```

Vous configurera une console série à 57600bps, 8N1.

Notes MacPPC

- Malheureusement, l'utilisation des consoles séries n'est pas directement possible sur la plupart des MacPPCs. Alors que la plupart des machines auront le matériel série, il n'est pas accessible à l'extérieur de la machine. Heureusement, quelques sociétés offrent des matériel tiers pour la plupart des modèles Macintosh qui rendront ce port accessible pour une utilisation de console série (ou autre). Utilisez votre moteur de recherche préféré et recherchez "Macintosh internal serial port".
- Vous devrez positionner `tty00` dans `/etc/ttys` à `on` et mettre la vitesse à 57600 plutôt que la valeur par défaut 9600 comme décrit [ci-dessus](#) dans le mode "single user" avant de démarrer en mode multi-utilisateurs et obtenir une console série fonctionnelle.

Mac68k

La console série est sélectionnée dans le programme *Booter*, sous le menu "Options", puis "Serial Ports". Cochez le bouton "Serial Console", puis choisissez le port du modem ou de l'imprimante. Vous

aurez besoin d'un modem Macintosh ou d'un câble d'imprimante à attacher à un port série Mac. Si vous souhaitez que ces paramètres soient utilisés par défaut, demandez au programme "Booter" de sauvegarder vos options.

Notes Mac68k

- Le port du modem est *tty00*, celui de l'imprimante *tty01*.
- Le Mac68k n'active pas son port série tant que cela n'est pas demandé, aussi, votre machine n'enverra aucun signal sur la console série avant que la procédure de démarrage OpenBSD ne commence.
- Vous devrez activer le port (*tty00* ou *tty01*) comme indiqué [ci-dessus](#).

7.7 - Comment effacer la console ? (wscons)

Si vous souhaitez effacer votre console après une certaine période d'inactivité sans utiliser X, vous pouvez modifier les variables [wscons\(4\)](#) suivantes :

- **display.vblank** réglé à `on` désactivera les rafraîchissements verticaux, ce qui mettra la plupart des moniteurs dans le mode "économie d'énergie". Il faudra plus de temps pour retrouver l'affichage, mais cela réduira la consommation d'énergie. Quand il est réglé à `off`, l'affichage sera désactivé, mais le moniteur continuera de recevoir les signaux de synchronisation verticaux et horizontaux, donc l'affichage sera très vite réactivé.
- **display.screen_off** règle le nombre de nettoyages en millisecondes, par exemple, 60000 serait le temps pour une minute.
- **display.kbdact** détermine si l'activité du clavier provoquera le rafraîchissement de l'écran. En général, ceci est souhaité.
- **display.outact** détermine si une sortie texte sur l'écran provoquera le retour de l'affichage.

Vous pouvez positionner ces variables en ligne de commande en utilisant l'utilitaire [wsconsctl\(8\)](#) :

```
# wsconsctl display.screen_off=60000
display.screen_off -> 60000
```

ou les positionner de manière permanente en éditant le fichier [/etc/wsconsctl.conf](#) de manière à ce que les paramètres soient repris au prochain redémarrage :

```
display.vblank=on           # enable vertical sync blank
display.screen_off=600000   # set screen blank timeout to 10
minutes                    #
display.kbdact=on          # Restore screen on keyboard
input
```

```
display.outact=off # Restore screen on display
output
```

Le vidage est activé à partir du moment où `display.kbdact` ou `display.outact` sont positionnés à "on".

7.8 - TOUT CE QUE JE TAPE A LA CONNEXION EST EN MAJUSCULES !

En réalité, ceci est une fonctionnalité, non un bogue.

Virtuellement toutes les commandes et les noms d'utilisateurs sont entrés en minuscules. Cependant, quelques très vieux terminaux sont simplement incapables d'afficher des caractères en minuscules, ce qui rend leur utilisation difficile, voir impossible à utiliser avec UNIX. Pour pallier cela, si vous utilisez un nom d'utilisateur entièrement en majuscules, [getty\(8\)](#) assumera que votre terminal ne supporte pas les minuscules, et interprétera simplement tout ce que vous tapez comme des minuscules alors qu'il affichera tout en majuscules. Si votre mot de passe est composé de minuscules et de majuscules, la connexion sera impossible.

L'appui sur CTRL-D à la connexion provoquera l'arrêt de `getty(0)` et [init\(8\)](#) en relancera un nouveau, qui acceptera correctement les minuscules et les majuscules.

7.9 - Qu'est ce que tmux ?

Pour ceux qui sont familiés avec le programme "screen", disponible comme paquetage, ou `window(1)` qui a été dans la base, il est plus simple de répondre en disant que [tmux\(1\)](#) fournit plusieurs fonctions de screen et window, avec beaucoup de fonctionnalités en plus.

Pour ceux qui ne sont pas familier avec ces programmes, tmux est un *multiplexeur de terminal*. C'est un programme qui permet qu'un nombre de plusieurs processus partagent le même écran pour les entrées et sorties. Dans tmux, cet ensemble de programmes d'appelle une *session*, dont chaque programme est contenu dans une *windows* tmux.

En plus de partager le terminal, tmux vous permet de détacher une session et ces fenêtres de l'écran, les laissant continuer à fonctionner en arrière-plan, pour les réattacher plus tard dans le même écran ou dans un autre. Une session peut-être détachée manuellement ou à cause d'un événement inattendu comme une déconnexion réseau, dans tous les cas les programmes survivent et continuent de fonctionner normalement.

tmux possède aussi beaucoup d'autres fonctionnalités comme scinder une fenêtre simple en plusieurs

sections (appelés *panes*), un historique du texte affiché dans chaque fenêtre, copier et coller du texte entre fenêtres, combinaisons de touches configurable et verrouillage du terminal. Jetez un oeil sur la page de man [tmux\(1\)](#) pour plus d'information.

Comment est-ce que j'utilise tmux ?

La première étape est de lancer tmux :

```
$ tmux
```

Ce démarre un nouveau tmux avec une nouvelle session unique (appelée "0") et lui crée un affichage client sur l'écran. La plupart de l'écran montrera une fenêtre contenant une invite de commande, et vous remarquerez que la dernière ligne est occupée par une ligne d'état. Cela affiche le nom de la session entre crochets à gauche, le nom de la fenêtre (normalement vide pour les shells) et le temps sur la droite, et un résumé de la fenêtre courante ouverte au milieu. Dans notre nouvelle session, les fenêtres courantes ouvertes contiendront une entrée, par exemple :

```
0:ksh*
```

Un rappel bref sur les terminaux : sur OpenBSD les applications sont faites pour être indépendante des capacités du terminal grâce à la variable d'environnement TERM. C'est configuré par le nom d'une entrée dans la base de données [terminfo\(5\)](#) et dit aux programmes que le terminal, par exemple, supporte la couleur, ou possède la possibilité d'insérer des lignes, ou bien d'autres choses. Une chose importante à remarquer est que l'entrée "xterm" dans la base de données n'inclue pas la couleur, donc tmux n'utilise pas la couleur par défaut dans xterm, TERM doit être configuré à "xterm-xfree86" à la place si la couleur est désirée (la ressource X "XTerm*termName" doit être configurée dans .Xdefaults pour être utilisée dans tous les xterms). Il est aussi important que le TERM dans les shells démarrés dans tmux soient configurés à "screen" ou les programmes executés pour eux ne s'afficheront pas correctement - tmux le configurera lui-même, mais il faut faire attention à ne pas le surcharger dans des fichiers de démarrage shell.

Retournons à la ligne d'état, le numéro "0" est l'index de la fenêtre, "ksh" le nom de la fenêtre, et le "*" que c'est la fenêtre courante qui est affichée au dessus de la ligne d'état. Tous ce que l'on tape est envoyé au shell et toutes sorties affichées. Par exemple, si vous lancez "[top](#)" :

```
$ top
```

Il fonctionnera normalement, occupant l'espace de l'écran au dessus de la ligne d'état. Vous remarquerez aussi que le nom de la fenêtre dans la ligne d'état a été modifié de "ksh" à "top" - tmux renomme les fenêtres pour refléter le programme qui fonctionne actuellement en elles-mêmes.

Maintenant, disons que vous voulez détacher tmux de l'écran et retourner au shell d'origine d'ou vous l'avez lancé. Une session tmux peut-être détachée en pressant en premier les touches *Ctrl* et *b* ensembles puis la touche *d*. La combinaison de touches *Ctrl-b* (raccourcie dans tmux et dans sa page de man à "*C-b*") est appelée *touche de préfixe* et est utilisée pour dire à tmux que la prochaine touche pressée est une instruction qui doit générer une action, plutôt que de passer la touche à travers un programme le programme dans la fenêtre.

Après avoir pressé *Ctrl-b d* et retourné à l'invite de commande, reattachez la session tmux en utilisant la commande "attach" :

```
$ tmux attach
```

La session tmux réapparaîtra, avec la ligne d'état et "top" fonctionnera joyeusement. (Si à la place vous exécutez tmux sans argument, une deuxième sessions sera créée, nommée "1").

Ensuite, créons une deuxième fenêtre. C'est réalisé en utilisant la touche "c" : pressez la touche préfixe, *Ctrl-b*, puis la touche "c". Une nouvelle fenêtre sera créée et une invite de commande sera encore affichée sur l'écran. La ligne d'état sera mise à jour pour afficher la nouvelle fenêtre :

```
0:top- 1:ksh*
```

Le "-" après "top" affiche la fenêtre courante précédente (la dernière fenêtre). En Pressant encore *Ctrl-b c* créera une nouvelle invite de commande :

```
0:top 1:ksh- 2:ksh*
```

Il y a plusieurs commandes pour se déplacer entre les fenêtres. De la fenêtre 2, vous pouvez vous déplacer vers la fenêtre précédente, numéro 1, en tapant *Ctrl-b p*. *Ctrl-b n* déplacera vers la fenêtre suivante : dans ce cas, il n'y a pas de fenêtre 3 donc la fenêtre courante deviendra la fenêtre 0. Vous pouvez aussi taper *Ctrl-b w* pour obtenir un menu interactif des fenêtres ouvertes, *Ctrl-b l* pour se déplacer à la dernière fenêtre (celle marquée avec un "-"), ou *Ctrl-b 0* pour se déplacer à la fenêtre 0, *Ctrl-b 1* à la fenêtre 1 et jusqu'à *Ctrl-b 9* pour la fenêtre 9. Donc, pour retourner au "top" de la fenêtre 0, vous pouvez presser *Ctrl-b 0* pour aller directement à la fenêtre 0, *Ctrl-b p* deux fois pour aller à la fenêtre 1, *Ctrl-b n* pour sauter de la fenêtre 2 à la fenêtre 0, ou presser *Ctrl-b w* et sélectionner la fenêtre 0 de la liste.

Parfois vous voulez créer une fenêtre exécutant directement un programme, sans passer par l'invite de commande. Cela peut-être fait de tmux via *l'invite de commande*. En pressant la séquence de touches *Ctrl-b* : la ligne d'état change pour afficher une invite ":" ou des commandes peuvent être entrées. Toutes les commandes tmux sont documentées dans la page de man. Dans ce cas, la commande "new-window" est nécessaire. Chaque commande possède un alias plus court qui peut-être utilisé à la place de taper le nom complet, pour "new-window" c'est "neww". Donc pour créer une nouvelle fenêtre exécutant

[tetrис\(6\)](#), tapez :

```
neww tetris
```

La nouvelle fenêtre sera fermée quand tetris sera terminé, ou sera forcement tuée en utilisant la combinaison de touches *Ctrl-b* <. Cela demandera en premier une confirmation et si elle est donnée, fermera la fenêtre et terminera le programme fonctionnant en elle.

Un autre besoin courant est de renommer une fenêtre. Cela peut-être fait avec la combinaison de touches *Ctrl-b* ,. La ligne d'état sera modifiée pour afficher une invite "(rename-window)" ou l'on fournira le nom de la nouvelle fenêtre. Renommer une fenêtre désactive le renommage automatique pour cette fenêtre, pour réactiver cette fonctionnalité, pressez *Ctrl-b* : pour obtenir l'invite de commande et entrez ce qui suit (plus d'infos sur ce que cela signifie dans le paragraphe suivant) :

```
setw -u automatic-rename
```

Une autre combinaison de touches importante à se souvenir : *Ctrl-b* ?. Cela affichera la liste de toutes les touches tmux et commandes qu'il exécute. Par exemple, *Ctrl-b* ? montre que la touche *c* est reliée à la commande "new-window" et la touche *n* à la commande "next-window".

Configurer tmux

Beaucoup d'utilisateurs voudront personnaliser l'aspect et le comportement de tmux. Cela peut-être réalisé à travers le fichier de configuration `~/.tmux.conf`. Ce fichier est une liste de commandes tmux qui sont exécutées quand tmux est démarré initialement, avant que la première session ne soit créée. Toutes les commandes tmux sont documentées dans la page man, mais quelques exemples courants que vous pourrez utiliser dans votre fichier de configuration sont expliqués ci-dessous.

La plupart des besoins courants sont la configuration des options. Il y a deux types d'options dans tmux : *options de session* et les *options de fenêtre*. Les options de session contrôlent la façon dont sont gérées les options de session et fenêtre d'une fenêtre individuelle. Pour l'ensemble il y a un ensemble d'options globales. Quand tmux vient à décider de la valeur d'une option pour une session ou fenêtre particulière, il regarde en premier les options locales de cette fenêtre; si l'option n'est pas configurée, la valeur de l'option globale est utilisée.

Les options de session sont configurées avec la commande "set-option" (alias "set") et les options de fenêtre avec la commande "set-window-option" (alias "setw"). Pour configurer une option globale, utilisez l'argument "-g", si cela n'est pas précisé l'option est configurée pour la fenêtre ou session courante. Ces commandes acceptent aussi quelques autres arguments, comme "-u" pour désactiver une option locale et permettre à une fenêtre ou session d'hériter à nouveau d'une option globale.

Dans le fichier de configuration, il est usuel de configurer des options globales. Regardez ces exemples

pour personnaliser la ligne d'état :

```
set -g status-bg blue
set -g status-right '#(sysctl vm.loadavg) '
setw -g window-status-current-attr underscore
```

Mettez ces trois commandes dans `.tmux.conf` et en redémarrant, tmux change l'arrière-plan de la ligne d'état en bleu, afficher la charge courante sur le coté droit et souligne la fenêtre courante. La ligne d'état peut-être désactivée complètement avec :

```
set -g status off
```

Il y a un grand nombre d'autres options; un autre exemple est de changer le style de combinaisons de touche pour ressembler à [vi\(1\)](#) à l'invite de commande et dans la liste des fenêtres et d'autres modes interactifs de tmux :

```
set -g status-keys vi
setw -g mode-keys vi
```

Les options courantes et leur valeurs seront affichées avec les commandes "show-options" et "show-window-options". Comme les commandes de configuration elles acceptent "-g" pour afficher les options globales.

Une autre tâche courante du fichier de configuration est d'ajouter ou de modifier des commandes de combinaisons de touches, ce sont les commandes qui sont exécutées après avoir pressé *Ctrl-b* avec une autre touche. Elles sont ajoutées ou modifiées avec la commande "bind-key" (alias "bind") et supprimées avec la commande "unbind-key" (alias "unbind"). Voici deux exemples de l'utilisation de "bind-key" :

```
bind C-d detach
bind / neww 'exec top'
```

La première crée une combinaison pour *Ctrl-b Ctrl-d* pour détacher tmux, la même que par défaut *Ctrl-b d*, et la seconde combinaison *Ctrl-b /* pour créer une nouvelle fenêtre exécutant top.

Plusieurs personnes aiment utiliser un préfixe différent de *Ctrl-b*. Cela peut-être réalisé en utilisant à la fois le "set-option" et les commandes de combinaison de touche pour altérer l'option de préfixe de touche et le changer pour qu'en pressant deux fois le préfixe continue de fournir la même combinaison de touche à travers la fenêtre. Pour changer le préfixe de touche à *Ctrl-a* :

```
set -g prefix C-a
unbind C-b
```

```
bind C-a send-prefix
```

La dernière chose utile à faire dans le fichier de configuration est de créer un ensemble de fenêtre initiale quand tmux démarre. C'est un peu plus compliqué que les exemples précédents. Dans tmux, une session ne peut pas avoir aucune fenêtre, et vous ne pouvez pas créer de fenêtre sans session. Donc, pour avoir des fenêtres créées par le fichier de configuration vous devez en premier créer une session qui les contient. Par exemple (remarquez que "new" est un alias for la commande "new-session") :

```
new -d 'exec top'
neww -d
neww -d
```

Ces commandes créent une nouvelle session avec "top" s'exécutant dans la première fenêtre, puis crée deux fenêtres additionnelles. L'argument "-d" indique à tmux de ne pas essayer d'attacher la nouvelle session ou de faire des nouvelles fenêtres la fenêtre courante. Avant de mettre ces lignes dans .tmux.conf, il y a un autre problème. Exécuté sans arguments, tmux exécute la commande "new-session", donc quand on lance tmux avec "tmux" du shell, le fichier de configuration indique à tmux de créer une session, et la commande du shell lui dit d'en créer une autre, donc vous avez finalement *deux* sessions. Pour éviter cela, tmux doit être démarré avec "tmux attach" quand il doit créer une session du fichier de configuration - cela signifie qu'il crée la session de .tmux.conf et l'attache immédiatement sans créer une seconde session.

Utilisation avancée de tmux

Ce paragraphe couvre brièvement certaines des fonctionnalités les plus avancées dans tmux. Consultez la [page man](#) pour plus d'information.

Dans tmux vous pouvez copier et coller du texte entre les fenêtres. C'est réalisé en copiant le texte en *mode copie* puis en le collant avec la commande coller. Pour entrer en mode copie, pressez *Ctrl-b [*. En mode copie (avec l'option de fenêtre "mode-keys" configurée pour emacs, pour les touches vi consultez la page man) les touches de flèches doivent être utilisées pour positionner le curseur, *Ctrl-Space* débute la sélection et *Ctrl-w* copie. Vous pouvez aussi utiliser *Page Up*, *Page Down*, *Ctrl-a* et *Ctrl-e* pour bouger le curseur. Pressez *q* ou *Escape* pour quitter le mode copie. Après cela, *Ctrl-b]* copiera le texte copié dans la fenêtre courante comme si vous aviez tapé le texte directement.

tmux est facilement scriptable, et la plupart des commandes qui sont entrées via l'invite de commande ou reliées à une touche peuvent être exécutées du shell. Presque toutes les commandes tmux acceptent un argument optionnel "-t" qui spécifie la session ou la fenêtre ou agir. Par exemple cette commande :

```
$ tmux kill-window -t0:1
```

tuera la fenêtre 1 dans la session 0. Et :


```
$ tmux new-window -tmysession
```

Crée une nouvelle fenêtre dans la session nommée 'mysession'. Plusieurs commandes acceptent d'autres arguments, par exemple la commande "new-window" accepte une option "-n" pour donner le nom de la nouvelle fenêtre, et "new-session" accepte plusieurs arguments pour spécifier les attributs de la fenêtre initiale créée avec la session. Ces arguments peuvent être naturellement utilisés quand une commande est liée à une touche ou exécutée de l'invite de commande bien évidemment.

Une autre fonctionnalité utile est la possibilité de couper une fenêtre unique en plusieurs sections, appelées *panes*. Vous pouvez découper une fenêtre verticalement (haut en bas) avec la combinaison de touche *Ctrl-b "*. Un pane peut être redimensionné en haut et bas avec *Ctrl-b Alt-Up* et *Ctrl-b Alt-Down* et le pane actif modifié avec *Ctrl-b o*. En plus, une fenêtre découpée de cette façon peut-être modifiée en une parmi plusieurs mises en page fixes, elles défilent avec *Ctrl-b Space* mais les panes utilisant ces mises en page ne peuvent pas être redimensionnées. En *-current*, le découpage a été étendu pour supporter le découpage horizontal (*Ctrl-b %*) et les mises en page fixes modifiées pour être appliquées une fois (avec la même combinaison de touche *Ctrl-b Space*) mais elles peuvent être librement redimensionnées et modifiées aussi bien verticalement que horizontalement.

[\[Index de La FAQ\]](#) [\[Section 6 - Administration réseau\]](#) [\[Section 8 - Questions diverses \]](#)



www@openbsd.org

\$OpenBSD: faq7.html,v 1.40 2010/05/23 16:20:41 ajacoutot Exp \$

8 - Questions Générales

Table des matières

- [8.1 - J'ai oublié mon mot de passe root..... que faire maintenant ?](#)
- [8.2 - X ne veut pas démarrer, et j'ai de nombreux messages d'erreur](#)
- [8.3 - Puis-je utiliser le langage de programmation "L" sous OpenBSD ?](#)
- [8.4 - Qu'est-ce que l'arborescence des Ports ?](#)
- [8.5 - Qu'est-ce que les Paquetages ?](#)
- [8.6 - Dois-je utiliser les Ports ou les Paquetages ?](#)
- [8.8 - Y'a t-il un moyen d'utiliser mon lecteur de disquette alors qu'il n'était pas connecté durant la phase de démarrage ?](#)
- [8.9 - Chargeur de démarrage OpenBSD \(*spécifique à i386 et amd64*\)](#)
- [8.10 - Utilisation de S/Key sur votre système OpenBSD](#)
- [8.12 - OpenBSD supporte-t-il le SMP \(Système Multiprocesseurs\) ?](#)
- [8.13 - Parfois, j'ai des erreurs d'entrées/sorties lorsque j'essaie d'utiliser mes périphériques tty](#)
- [8.14 - Quels sont les navigateurs Web disponibles pour OpenBSD ?](#)
- [8.15 - Comment utiliser l'éditeur mg ?](#)
- [8.16 - ksh\(1\) ne semble pas lire mon fichier .profile !](#)
- [8.17 - Pourquoi le fichier /etc/motd est-il écrasé alors que je l'ai modifié ?](#)
- [8.20 - Les polices anti-aliasées et "TrueType" avec X](#)
- [8.21 - Est-ce que OpenBSD supporte des systèmes de fichiers journalisés ?](#)
- [8.22 - Reverse DNS ou pourquoi cela prend autant de temps lorsque je me connecte ?](#)
- [8.23 - Pourquoi les pages internet d'OpenBSD ne sont-elles pas conformes au HTML4/XHTML ?](#)
- [8.24 - Mon horloge est décalée d'une vingtaine de secondes. Pourquoi ?](#)
- [8.25 - Mon horloge est décalée de plusieurs heures. Pourquoi ?](#)

8.1 - J'ai oublié mon mot de passe root..... Que faire maintenant ?

La procédure de base pour reprendre le contrôle de root consiste à démarrer en mode utilisateur unique, monter les partitions adéquates (/ et /usr), exécuter [passwd\(1\)](#) pour changer le mot de passe du super-utilisateur. Vous pouvez ensuite redémarrer et vous connecter normalement.

Voici la procédure détaillée :

- **Démarrer en mode utilisateur unique.** Cette partie de la procédure dépend de la [plate-forme](#). Dans le cas des plates-formes amd64 et i386, le [chargeur de démarrage de second niveau](#), [boot\(8\)](#), pause quelques instants afin de vous donner une chance de fournir des paramètres au noyau. L'invite de commandes ressemble à l'invite suivante :

```
probing: pc0 com0 com1 apm mem[636k 190M a20=on]
disk: fd0 hd0+
>> OpenBSD/i386 BOOT 3.02
boot>
```

A cet instant, saisissez "boot -s" pour démarrer en mode utilisateur unique :

```
boot> boot -s
```

La plupart des autres plates-formes envoient des paramètres au noyau à travers la ROM de démarrage.

Bien entendu, le problème que vous rencontrerez en premier consiste à arrêter le système. Très probablement, vous devrez appuyer sur le bouton de réinitialisation ou d'alimentation. Bien que non souhaitable, il n'y aucune autre alternative. Ne vous tracassez pas trop, le système de fichiers d'OpenBSD est très robuste.

- **Montez les partitions.** "/" et /usr devront être montées en lecture/écriture. En supposant que nous parlons bien de deux partitions distinctes (comme ça devrait être le cas), la série de commandes suivante vous permettra d'atteindre le but souhaité :

```
# fsck -p / && mount -uw /
# fsck -p /usr && mount /usr
```

- Exécutez [passwd\(1\)](#) pour modifier le mot de passe root. Etant donné que vous avez déjà les privilèges root (vu que vous êtes en mode utilisateur unique), on ne vous demandera pas de fournir votre mot de passe actuel.
- **Démarrez en mode multi-utilisateur.** Ceci peut être effectué soit en saisissant "CTRL-D" pour reprendre la séquence de démarrage normale ou en saisissant la commande [reboot\(8\)](#).

Si c'est une machine utilisée par plusieurs personnes, vous devriez probablement utiliser [sudo\(8\)](#) pour fournir à plusieurs personnes (de confiance) la possibilité d'exécuter des commandes en tant que super-utilisateur.

"Attendez. Ceci semble très facile ! Ce n'est pas très sécurisé !" Si un attaquant a un accès physique à votre système, il a gagné et ce quelque soit le système d'exploitation sur la machine. Il existe des méthodes pour obliger un utilisateur à saisir un mot de passe en mode utilisateur unique (voir [ttys\(5\)](#)), ou éliminer le temps d'attente sous i386/md64 (voir [boot.conf](#)), mais pour être franc, le contournement de ces astuces est aussi assez facile (*Une méthode* : démarrez à partir d'une disquette ou d'un CDROM, éditez ou remplacez le fichier contenant les mots de passe). Vous pouvez essayer d'éviter cela, mais quelqu'un pourrait potentiellement retirer le disque dur de votre système. Rendre l'exploitation de votre machine difficile n'est pas vraiment de la sécurité, et si vous n'avez pas sécurisé l'accès physique à la machine, vous n'avez pas de sécurité réelle.

Remarque : plusieurs systèmes de "gestion à distance" fournissent pratiquement toutes les fonctionnalités d'un accès physique à la machine, et ceci doit être pris en compte. Ne pensez pas que le système est sécurisé s'il y a un moyen pour l'attaquant de prendre le contrôle d'une console, insérer une disque virtuelle et forcer un redémarrage de la machine. Ceci équivaut à un accès physique. Le système de gestion de console n'est probablement pas aussi sécurisé qu'OpenBSD...

8.2 - X ne veut pas démarrer, et j'ai de nombreux messages d'erreur

La cause habituelle des problèmes avec X vient souvent de la configuration de `machdep.allowaperture` avec [sysctl\(8\)](#). Puisque cette fonctionnalité est désactivée par défaut sous OpenBSD, il s'agit certainement de la cause du problème.

Vous devez éditer `/etc/sysctl.conf` et positionner la valeur **`machdep.allowaperture=2`** (ou **`1`**, selon votre plate-forme). Cela permettra à X d'accéder au pilote d'ouverture (aperture) [xf86\(4\)](#), au prochain redémarrage. Cette valeur ne peut pas être changée après le démarrage. En revanche, elle peut être définie dès l'installation si vous répondez "Y" lorsque l'on vous demande si vous comptez utiliser X Window ("Do you expect to run the X Window System?").

Sous OpenBSD, le pilote d'ouverture (aperture) doit être activé sur les plates-formes alpha, amd64, i386, macppc et sparc64 afin de permettre l'accès à la carte vidéo. D'autres plates-formes utilisent un moyen plus sûr pour gérer le système vidéo et n'ont pas besoin de ceci (qui n'est pas inclus dans leur noyau). Si vous ne prévoyez pas d'utiliser X sur votre système, il est recommandé de ne pas activer le pilote d'ouverture (aperture).

Pour plus d'informations sur la configuration et l'utilisation de X sur votre plate-forme, consultez le

fichier/usr/X11R6/README de votre installation.

8.3 - Puis-je utiliser le langage de programmation "L" sous OpenBSD ?

Plusieurs langages de programmation communs sont supportés soit au niveau du système de base (et plus spécifiquement dans les jeux de fichiers `baseXX.tgz` et `compXX.tgz`), soit au niveau du [système de paquetages et de ports](#). Il est recommandé d'installer le jeu de fichiers nécessaire ou le paquetage contenant le compilateur spécifique que vous souhaitez employer au lieu de l'installation à partir des sources. Pour certains compilateurs, l'installation à partir des sources requièrent d'importantes ressources système. Cette opération est souvent inutile sauf si vous avez des besoins spécifiques ou [aucun paquetage n'est disponible](#).

Le tableau suivant cherche à fournir une vue d'ensemble des compilateurs pour différents langages, où vous pouvez les trouver et si des problèmes ou des limitations relatifs à leur utilisation existent. Certains de ces compilateurs ne sont disponibles que pour certaines plates-formes. Pour voir les plates-formes sur lesquelles fonctionne tel ou tel compilateur, il suffit d'examiner un [résultat de recherche](#) dans l'arborescence des ports et de noter ce qui est mentionné au niveau de "Archs". Une autre méthode consiste à inspecter directement le contenu du Makefile du port et d'y chercher `ONLY_FOR_ARCHS`, `NOT_FOR_ARCHS`, `BROKEN`, etc.

Remarque : Pour améliorer l'utilisation, cet article fournit une liste alphabétique sans distinguer entre différentes catégories de langages de programmation. Ceci n'est en aucun cas une liste exhaustive de tout ce qui est disponible ou peut fonctionner sous OpenBSD. Si vous relevez des inexactitudes ou des problèmes autres que ceux listés ci-dessous, prière de [nous en faire part](#).

Langage	Où ?	Remarques
Awk	base47.tgz , awk(1)	
	lang/gawk	GNU awk
C, C++	comp47.tgz , gcc(1)	Les compilateurs C/C++ dans le système de base ont été audités et contiennent plusieurs améliorations sécurité (telles que ProPolice) activées par défaut. Merci de bien vouloir consulter gcc-local(1) pour de plus amples détails. Ces compilateurs émettrons des alertes lors de l'utilisation de fonctions non sûres telles que <code>sprintf()</code> , <code>strcpy()</code> , <code>strcat()</code> , <code>tmpnam()</code> , etc. La plupart des plates-formes utilisent gcc 3.3.5. Les autres restent en gcc 2.95.3 pour le moment.

C, C++	lang/gcc	Ces compilateurs n'ont pas bénéficié d'un audit sécurité et ne contiennent pas les améliorations sécurité que possèdent les compilateurs du système de base. Les compilateurs sont renommés egcc, eg++, etc. pour éviter la confusion avec ceux du système de base.
Caml	lang/ocaml	Objective Caml
COBOL	lang/open-cobol	
Erlang	lang/erlang	
Fortran	comp47.tgz , g77 (1)	Support du Fortran 77 uniquement.
	lang/gcc	Fortran 95 est aussi supporté par egfortran dans gcc 4.0 et supérieur. Ce nouveau compilateur est disponible en tant que sous-paquetage (g95) de gcc.
Haskell	lang/ghc	
	lang/nhc98	
Java	devel/jdk	Sun JDK - seulement la 1.7 comme paquetage; pour des versions plus anciennes consultez les instructions ci-dessous.
	lang/classpath	bibliothèques class de base pour Java
	lang/kaffe	
	lang/jikes	Compilateur rapide, fonctionne bien. Il nécessite un "run-time jar", la version bytecode de toute l'API standard.
	devel/eclipse	IDE très fourni; Fonctionne avec Sun JDK
Lisp	lang/clisp	
Lua	lang/lua	Bibliothèques Lua additionnelles et outils auxiliaires sont disponibles dans l'arbre des ports.
Perl	base47.tgz , perl (1)	De nombreux modules Perl sont disponibles dans l'arbre des ports, regardez donc ceux-là avant d'installer des modules depuis CPAN.
PHP	www/php4	De nombreux sous-paquetages sont disponibles pour des modules PHP différents.
	www/php5	
Prolog	lang/swi-prolog	Environnement SWI-Prolog.
Python	lang/python	D'autres ports utilisent Python 2.5 par défaut.
Ruby	lang/ruby	
Scheme	lang/chicken	

	lang/scheme48
	lang/scm
	shells/scsh
Smalltalk	lang/squeak
Tcl	lang/tcl

Compilation du JDK Sun

A cause de la licence restrictive SCSL de Sun, OpenBSD n'est pas en mesure de livrer des paquets binaires du JDK < 1.7. En démarrant de la 1.7, OpenBSD a un port respectant la GPLv2 qui peut être installé comme paquetage. Les utilisateurs voulant un plugin pour le navigateur auront toujours besoin de compiler la version 1.5 ou 1.6 des ports tant que Sun ne fournira pas le code du plugin. Il est à remarquer qu'il faut beaucoup de RAM pour que la compilation réussisse.

Les ports du JDK sont dans le sous-répertoire `devel/jdk` de l'arbre des ports. Vous pouvez choisir entre les différentes versions de celui-ci, toutes dans leurs sous-répertoires. Lorsque vous tapez `make`, vous verrez un message vous demandant de récupérer manuellement les archives de distribution sur le site de Sun. Avant que vous ne puissiez faire cela, vous devez vous enregistrer sur ce site, et accepter la licence. C'est la raison pour laquelle le mécanisme des ports ne peut les télécharger automatiquement.

Une fois les archives de distribution et patches téléchargés, copiez les dans le répertoire `/usr/ports/distfiles`. Vous devez aussi disposer de X sur votre système. Débutez le processus de compilation en saisissant `make` dans le répertoire du port.

Le JDK requiert un compilateur Java 2 en état de fonctionnement afin d'agir en tant que bootstrap. Pour cela, depuis OpenBSD 4.0, le port JDK 1.5 utilise `kaffe`, ce qui permet à celui-ci d'être utilisé sur les architectures `i386` et `amd64` et ce qui réduit considérablement le temps de compilation.

Les anciennes versions du JDK nécessitent toujours une version Linux du JDK. L'émulation Linux sur OpenBSD est réservée aux systèmes `i386`, et le JDK ne peut ainsi être compilé que sur `i386`. Le mécanisme des ports devrait prendre en charge l'installation des fichiers nécessaires ainsi que le réglage `kern.emul.linux=1`. Pour davantage d'informations, consultez s'il vous plaît Linux emulation dans la page de manuel [compat_linux\(8\)](#), ainsi que [FAQ 9 - Exécution des binaires Linux sous OpenBSD](#). Notez que cette émulation Linux n'est nécessaire que lors de la compilation du JDK, aboutissant à un JDK OpenBSD natif. **Vous n'avez pas besoin de cette émulation pour utiliser le JDK natif.**

Après plusieurs heures, la compilation finira. Continuez simplement avec `make install` pour installer le JDK.

Si vous constatez des erreurs telles que "Could not reserve enough space for object heap", essayer d'augmenter les limites mémoire par processus en utilisant la commande shell `ulimit` avec le drapeau `-d`.

Autres outils de développement

Il existe plusieurs autres outils de développement dans le système de base ou en paquetages ou ports. Quelques exemples :

- Shells Unix : ksh et csh dans le système de base, plusieurs autres dans le sous-répertoire `shells` de l'arbre des ports.
- [lint\(1\)](#) : un programme de vérification C qui a été grandement amélioré à partir d'OpenBSD 3.9. Les versions compatibles lint ("linter versions") des bibliothèques système sont également fournies.
- Utilitaires "make" : Le traditionnel [make\(1\)](#) BSD est dans le système de base et l'arbre des ports contient d'autres versions qui sont nécessaires pour la compilation de certains logiciels.
- Outils graphiques : plusieurs outils graphiques populaires (tels que GTK+, Tk, Qt, wxWidgets, ...) ont été portés sur OpenBSD. Vous pourrez les trouver dans le sous-répertoire `x11` de l'arbre des ports.
- Systèmes de contrôle de versions : GNU CVS tel qu'il est utilisé par le projet OpenBSD est dans le système de base et l'arbre des ports en contient quelques autres. Notez qu'[OpenCVS](#) est en cours de développement.

8.4 - Qu'est-ce que l'arborescence des ports ?

Veillez consulter [FAQ 15, Utilisation des ports](#).

8.5 - Qu'est ce que les paquetages ?

Veillez consulter [FAQ 15, Gestion des paquetages](#).

8.6 - Dois-je utiliser les Ports ou les Paquetages ?

Veillez consulter [FAQ 15](#).

8.8 - Y'a t-il un moyen d'utiliser mon lecteur de disquette alors qu'il n'était pas connecté durant la phase de démarrage ?

Vous devez indiquer au noyau de toujours supposer que le lecteur de disquette est installé, même si il n'est pas détecté pendant la phase de démarrage. Pour ce faire, l'option `0x20` doit être positionnée dans

[fdc\(4\)](#). Ceci peut-être fait en utilisant [User Kernel Config](#) ou [config\(8\)](#) pour modifier votre noyau,

```
# config -e -f /bsd
OpenBSD 4.7 (GENERIC) #558: Thu Mar 17 20:46:15 MDT 2010
    deraadt@i386.openbsd.org:/usr/src/sys/arch/i386/compile/
GENERIC
Enter 'help' for information
ukc> change fd*
254 fd* at fdc0 drive -1 flags 0x0
change [n] y
drive [-1] ? ENTER
flags [0] ? 0x20
254 fd* changed
254 fd* at fdc0 drive -1 flags 0x20
ukc> q
Saving modified kernel.
#
```

8.9 - Chargeur de démarrage OpenBSD (spécifique à i386 et amd64)

Lorsque vous démarrez votre système OpenBSD, vous avez probablement remarqué l'invite de commande.

```
boot>
```

La plupart des utilisateurs n'auront pas à intervenir ici. Le système démarre même si aucune commande n'est entrée. Mais parfois surviennent des problèmes, ou certaines fonctions spéciales sont requises. C'est à cet endroit que ces options doivent être entrées. Avant de vous lancer, vous devriez lire la page de manuel [boot\(8\)](#). Ici, nous allons au-delà de la plupart des commandes utilisées pour le chargeur de démarrage.

Avant de débiter, si aucune commande n'est utilisée, le chargeur de démarrage essaiera automatiquement de démarrer **/bsd**. Si cela échoue, il essaiera **/obsd**, puis en cas de nouvelle échec **/bsd.old**. Vous pouvez spécifier un noyau en tapant :

```
boot> boot hd0a:/bsd
```

ou

```
boot> b /bsd
```

Ceci démarrera le noyau nommé `bsd` depuis la partition 'a' du premier disque reconnu par le BIOS.

Voici une brève liste d'options que vous pouvez utiliser avec le noyau OpenBSD.

- **-a** : Vous permet de spécifier un autre disque racine après le démarrage du noyau.
- **-c** : Vous permet d'entrer la planification de démarrage. Lisez la section [planification de démarrage](#) de la faq.
- **-s** : Vous permet de démarrer en mode utilisateur unique (single user).
- **-d** : Cette option est utilisée pour copier le noyau dans le ddb. Vous devez avoir DDB compilé dans le noyau.

Ces options sont saisies dans le format : **boot [image [-acds]]**

Pour plus d'informations, lisez la page de [manuel boot\(8\)](#).

8.10 - Utilisation de S/Key sur votre système OpenBSD

S/Key est un système d'authentification basé sur des mots de passe à usage unique. Il peut être utile pour les personnes n'ayant pas la possibilité d'utiliser un canal chiffré protégeant leurs identifiants en transit, tel qu'un canal établi par [ssh\(1\)](#).

ATTENTION : Les systèmes d'authentification basés sur des mots de passe à usage unique protègent uniquement l'information relative à l'authentification. Ils ne permettent pas de se prémunir contre l'usage des écoutes clandestines du réseau pour accéder à des informations confidentielles. De plus, si vous vous connectez à un système sûr A, il est recommandé de le faire depuis un autre système sûr B afin de s'assurer que personne n'accède au système A en enregistrant toutes vos saisies clavier ou en capturant et/ou en forgeant les entrées/sorties de vos terminaux.

Le système S/Key génère une séquence de mots passe à usage unique à partir de la *phrase secrète* d'un utilisateur ainsi qu'un défi transmis par le serveur à l'aide d'une *fonction de hachage* sûre. Le système est sûr uniquement si la phrase secrète n'est jamais transmise sur le réseau. **L'initialisation ou la modification de la phrase secrète DOIT donc se faire à travers un canal sûr**, tel que [ssh\(1\)](#) ou la console.

L'implémentation de S/Key sur OpenBSD peut utiliser plusieurs algorithmes de hachage à sens unique. Les algorithmes suivants sont disponibles :

- [md4](#)
- [md5](#)
- [sha1](#)

- [rmd160](#).

Mise en place de S/Key - Les premières étapes

Pour commencer, le répertoire `/etc/skey` doit exister. Si ce répertoire n'existe pas, le super utilisateur doit le créer. Cela peut simplement se faire en tapant :

```
# skeyinit -E
```

Une fois ce répertoire créé, vous pouvez initialiser votre S/Key. Pour cela vous devez utiliser [skeyinit\(1\)](#). Etant donné que `skeyinit(1)` vous demandera de saisir votre phrase secrète S/Key, vous devez l'exécuter **à travers un canal sûr**, tel que nous l'avons précédemment expliqué. Le programme vous rappellera même de le faire. Avec `skeyinit(1)`, votre mot de passe vous sera d'abord demandé. C'est le même mot de passe que vous utilisez pour vous connectez au système. Une fois authentifié avec votre mot de passe système, vous devrez saisir votre phrase secrète S/Key. Ce n'est **PAS** votre mot de passe système. votre phrase secrète devra comporter au moins 10 caractères. Nous vous suggérons d'utiliser des phrases de plusieurs mots comme mot de passe. Voici un exemple d'ajout d'utilisateur :

```
$ skeyinit
Reminder - Only use this method if you are directly
connected
           or have an encrypted channel. If you are using telnet,
           exit with no password and use skeyinit -s.
Password:
[Adding ericj with md5]
Enter new secret passphrase:
Again secret passphrase:

ID ericj skey is otp-md5 100 oshi45820
Next login password: HAUL BUS JAKE DING HOT HOG
```

La ligne `ID ericj skey is otp-md5 100 oshi45820` est particulièrement intéressante. Elle fournit beaucoup d'informations à l'utilisateur comme indiqué ci-après.

- `otp-md5` - Ceci indique quelle fonction à sens unique a été utilisée pour créer votre mot de passe à usage unique (otp).
- `100` - C'est votre numéro de séquence. Ce nombre part de 100 et se décrémente jusqu'à 0. Une fois ce numéro égal à 1, un autre mot de passe doit être généré, et ce grâce à [skeyinit\(1\)](#).
- `oshi45820` - C'est la clé.

Mais le plus important c'est votre prochain mot de passe. Il consiste en 6 petits mots, qui combinés,

constituent votre prochain mot de passe à usage unique, espaces compris. Ce mot de passe affiché par `skeyinit` ne peut être utilisé pour vous connecter (il existe une procédure pour l'utilisation du premier mot de passe à usage unique, lisez [skeyinit\(1\)](#)). Pour être capable de vous connecter, un mot de passe à usage unique correspondant à celui affiché par le processus d'identification doit être calculé en utilisant [skey\(1\)](#). La section suivante montre comment s'y prendre.

Utilisation de S/Key pour se connecter.

Maintenant votre `skey` est initialisé. Vous êtes prêt à vous connecter. Voici l'exemple d'une session utilisant S/Key pour se connecter : Pour exécuter une connexion avec S/Key, vous devez spécifier `:skey` comme votre identifiant.

```
$ ftp localhost
Connected to localhost.
220 oshibana.shin.ms FTP server (Version 6.5/OpenBSD) ready.
Name (localhost:ericj): ericj:skey
331- otp-md5 96 oshi45820
331 S/Key Password:
230- OpenBSD 4.7 (GENERIC) #558: Thu Mar 17 20:46:15 MDT
2010
230-
230- Welcome to OpenBSD: The proactively secure Unix-like
operating system.
230-
230- Please use the sendbug(1) utility to report bugs in
the system.
230- Before reporting a bug, please try to reproduce it
with the latest
230- version of the code. With bug reports, please try to
ensure that
230- enough information to reproduce the problem is
enclosed, and if a
230- known fix for it exists, include that as well.
230-
230 User ericj logged in.
Remote system type is UNIX.
Using binary mode to transfer files.
ftp> quit
221 Goodbye.
```

Notez que j'ai mis `:skey` pour mon nom d'utilisateur. Ceci indique à `ftpd` que je veux être identifié en utilisant S/Key. Certains d'entre vous ont dû remarquer que le numéro de séquence a été changé en `otp-`

md5 96 oshi45820. C'est parce que j'ai utilisé S/Key plusieurs fois pour m'identifier. Mais comment avoir votre mot de passe à usage unique ? Alors, pour calculer le mot de passe à usage unique, vous aurez besoin du numéro de séquence que vous utilisez et de votre clé. Vous être probablement en train de chercher comment vous rappeler le numéro de séquence.

Lorsque vous vous connectez, le processus d'identification affiche une ligne contenant les informations dont vous avez besoin, ce dont vous utilisez pour générer un mot de passe à usage unique à travers un autre canal sécurisé, en copiant la ligne dans une console :

```
otp-md5 96 oshi45820
```

Après avoir entré votre mot de passe, votre mot de passe à usage unique sera affiché. Vous pouvez le copier dans l'invite de S/Key lorsqu'il vous demande un mot de passe. Non seulement c'est *otp-md5* la description du hachage utilisé, mais c'est aussi un autre nom d'utilisateur pour la commande [skey\(1\)](#).

Si vous êtes connecté et souhaitez générer un autre mot de passe à usage unique pour la session suivante, utilisez [skeyinfo\(1\)](#). Il vous indiquera ce qu'il faudra utiliser pour la session suivante. Par exemple, je génère d'autres mots de passe à usage unique pour des utilisations ultérieures (souvenez-vous que je le fait depuis un canal sécurisé).

```
$ skeyinfo
95 oshi45820
```

La meilleure façon de le faire est d'utiliser **skeyinfo -v**, ce qui fournira une commande appropriée à exécuter dans une console. Par exemple :

```
$ skeyinfo -v
otp-md5 95 oshi45820
```

De même, la façon la plus simple de générer le mot de passe S/Key suivant est juste :

```
$ `skeyinfo -v`
Reminder - Do not use this program while logged in via
telnet.
Enter secret passphrase:
NOOK CHUB HOYT SAC DOLE FUME
```

Remarquez les caractères (```) dans l'exemple précédent.

Il est probable que beaucoup d'entre vous n'ont pas de connexion sécurisée ou une machine locale sécurisée pour créer les mots de passe, et leurs création à travers un canal non sécurisé n'est pas

admissible. Donc comment créer plusieurs mots de passe en une seule fois ? Vous pouvez indiquer à `skey(1)` le nombre de mot de passe que vous voulez créer. Vous pourrez alors les imprimer et les emporter avec vous.

```
$ otp-md5 -n 5 95 oshi45820
```

```
Reminder - Do not use this program while logged in via
telnet.
```

```
Enter secret passphrase:
```

```
91: SHIM SET LEST HANS SMUG BOOT
```

```
92: SUE ARTY YAW SEED KURD BAND
```

```
93: JOEY SOOT PHI KYLE CURT REEK
```

```
94: WIRE BOGY MESS JUDE RUNT ADD
```

```
95: NOOK CHUB HOYT SAC DOLE FUME
```

Il est à noter que le dernier mot de passe doit être le premier utilisé, car nous comptons de manière décroissante à partir de 100.

Utilisation de S/Key avec `ssh(1)` et `telnet(1)`

L'utilisation de S/Key avec `ssh(1)` ou `telnet(1)` est à peu près identique qu'avec `ftp` -- vous devez simplement rajouter `":skey"` à la fin de votre nom d'utilisateur. Exemple:

```
ssh -l ericj:skey localhost
```

```
otp-md5 98 oshi45821
```

```
S/Key Password: SCAN OLGA BING PUB REEL COCA
```

```
Last login: Thu Apr 7 12:21:48 on ttypl from 156.63.248.77
```

```
OpenBSD 4.7 (GENERIC) #558: Thu Mar 17 20:46:15 MDT 2010
```

```
Welcome to OpenBSD: The proactively secure Unix-like
operating system.
```

```
Please use the sendbug(1) utility to report bugs in the
system.
```

```
Before reporting a bug, please try to reproduce it with the
latest
```

```
version of the code. With bug reports, please try to ensure
that
```

```
enough information to reproduce the problem is enclosed,
and if a
```

```
known fix for it exists, include that as well.
```

```
You have mail.
```

§

8.12 - OpenBSD supporte-t-il le SMP (Système Multiprocesseurs) ?

SMP est supporté sur les plates-formes [OpenBSD/i386](#), [OpenBSD/amd64](#) [OpenBSD/mvme88k](#) et [OpenBSD/sparc64](#) (including the UltraSPARC T1 processors).

Un noyau SMP séparé, "bsd.mp", est fournit avec les paquetages d'installation, il peut être sélectionné pendant l'installation. Il est recommandé de tester ce noyau avant de le renommer en "bsd" pour en faire votre noyau par défaut.

Nous espérons que les autres plates-formes compatibles SMP soient supportées dans le futur. Sur la plupart des autres plates-formes, OpenBSD démarrera correctement sur un système SMP, mais n'utilisera qu'un seul processeur. L'exception est la plate-forme [SPARC](#) -- OpenBSD/sparc qui nécessitera parfois le retrait des modules MBus additionnels pour que le système démarre.

8.13 - Parfois, j'ai des erreurs d'entrées/sorties lorsque j'essaie d'utiliser mes périphériques tty

Vous devez utiliser /dev/cuaXX pour les connexions initialisées depuis le système OpenBSD, les périphériques /dev/ttyXX sont utilisables seulement pour les terminaux ou les connexions en dial. Alors qu'il était possible d'utiliser ces périphériques autrefois, le noyau OpenBSD n'est plus compatible avec ce type d'utilisation.

Extrait de [cua\(4\)](#):

For hardware terminal ports, dial-out is supported through matching device nodes called calling units. For instance, the terminal called /dev/tty03 would have a matching calling unit called /dev/cua03. These two devices are normally differentiated by creating the calling unit device node with a minor number 128 greater than the dial-in device node. *Whereas the dial-in device (the tty) normally requires a hardware signal to indicate to the system that it is active, the dial-out device (the cua) does not, and hence can communicate unimpeded with a device such as a modem.* This means that a process like [getty\(8\)](#) will wait on a dial-in device until a connection is established. Meanwhile, a dial-out connection can be established on the dial-out device (for the very same hardware terminal port) without disturbing anything else on the system. The [getty\(8\)](#) process does not even notice that anything is happening on the terminal port. If a connecting call comes in after the dial-out connection has finished, the [getty\(8\)](#) process will deal with it properly, without having noticed the intervening dial-out action.

8.14 - Quels sont les navigateurs Web disponibles pour OpenBSD ?

[Lynx](#), un navigateur en mode texte supportant le SSL est disponible dans le système de base. Les autres sont présents dans [l'arborescence des Ports](#), incluant (sans ordre particulier):

Navigateurs Graphique (X)

- [Konqueror](#) Installé avec [l'environnement de bureau KDE](#).
- [Konqueror-embedded](#) (konq-e) Konqueror, n'utilisant que les bibliothèques KDE plutôt que l'ensemble de KDE.
- [Links+](#) Un autre navigateur graphique rapide et léger (dispose également d'un mode console).
- [Firefox](#) et [SeaMonkey](#) sont des navigateurs disposant de beaucoup de fonctionnalités. SeaMonkey inclut beaucoup de fonctionnalités sortant du cadre d'un navigateur (client mail, client IRC, etc.), Firefox est simplement un navigateur basé sur Mozilla. Ils fonctionnent sur de nombreuses architectures.
- [Minimo](#), un navigateur Firefox léger dédié principalement aux systèmes tenant dans une main.
- [Opera](#) Navigateur commercial, i386 seulement.
- [Amaya](#) L'éditeur/navigateur du W3C.
- [Kazehakase](#), un navigateur léger capable d'utiliser les moteurs de rendu Gecko ou WebKit.
- [Midori](#), un navigateur basé sur WebKit du projet Xfce.
- [Chromium](#) la version open source du navigateur de Google Chrome, i386 seulement (actuellement).
- [XXXTerm](#) un navigateur minimaliste avec des combinaisons de touches au clavier à la vi en plus des opérations de navigation traditionnelles

Navigateurs en mode console

- [elinks](#) Riche en fonctionnalités, capable d'afficher les cadres et les tableaux, hautement configurable.
- [w3m](#) Supporte les tableaux et les frames. (dispose aussi d'un mode graphique).
- [links](#) Supporte les tables.

Vous les trouverez tous dans la [collection de paquetages](#). Ils sont tous présents dans `/usr/ports/www/` après l'installation de l'arborescence des ports. Beaucoup sont aussi disponibles en [paquetages](#) précompilés, présents sur les [serveurs FTP](#) et sur le [CD-ROM](#). Etant donné que la plupart des navigateurs graphiques pèsent lourd et prennent beaucoup de temps à télécharger et à compiler, vous devriez *sérieusement* [prendre en compte](#) l'utilisation des paquetages précompilés quand ils sont disponibles.

8.15 - Comment utiliser l'éditeur mg ?

MG est un éditeur de texte minimal dans le style d'Emacs, inclus dans OpenBSD. Minimal signifie qu'il est léger (Emacs est très lourd). Pour les bases, lisez la page de manuel de [mg\(1\)](#) ainsi que le [tutoriel](#), inclus dans le code source.

Notez que mg est une petite implémentation de Emacs, qui est souvent similaire aux fonctionnalités d'édition de texte de Emacs 17, et n'implémente donc pas les autres fonctionnalités (comprenant les outils de mail et de news, aussi bien que les modes pour Lisp, C++, Lex, Awk, Java, etc...)

8.16 - ksh(1) ne semble pas lire mon fichier .profile !

Il y a deux raisons possibles pour que [ksh\(1\)](#) semble ignorer le fichier `.profile` d'un utilisateur.

- `.profile` n'appartient pas à l'utilisateur. Pour corriger pour **utilisateur**,

```
# chown utilisateur ~utilisateur/.profile
```

- Vous utilisez ksh(1) depuis X Window System

dans une console [xterm\(1\)](#), `argv[0]` pour ksh(1) n'est pas précédé d'un trait d'union ("-"). Ajouter un trait d'union pour `argv[0]` précise à csh(1) et ksh(1) qu'ils doivent interpréter leurs fichiers de connexion. (Pour csh(1) c'est `.login`, avec un `.cshrc` séparé qui est toujours interprété quand csh(1) est lancé. Avec ksh(1), cela est plus perceptible car il n'y a qu'un fichier de démarrage, `.profile`. Ce fichier est ignoré si le shell n'est pas un shell de connexion.)

Pour corriger cela, ajoutez la ligne `"XTerm*loginShell: true"` au fichier `.Xdefaults` de votre dossier home. Note, ce fichier n'existe pas par défaut, vous devrez le créer.

```
$ echo "XTerm*loginShell: true" >> ~/.Xdefaults
```

Vous ne devriez pas avoir à faire cela sur d'autres systèmes, étant donné que certaines installations de X Window System viennent avec leur paramétrage par défaut. OpenBSD a choisi de suivre la ligne de conduite de X.

8.17 - Pourquoi le fichier /etc/motd est-il écrasé alors que je l'ai modifié ?

Le fichier `/etc/motd` est édité à chaque démarrage du système, remplaçant tout jusqu'à la première

ligne vide du fichier, exclue, par les informations de version du noyau. Lorsque vous éditez ce fichier, soyez sûr que vous commencez après cette ligne vide, pour empêcher `/etc/rc` de supprimer ces lignes de `/etc/motd` au démarrage.

8.20 - Les polices anti-aliasées et "TrueType" avec X

Voyez [ce document](#).

8.21 - Est-ce que OpenBSD supporte des systèmes de fichiers journalisés ?

Non il n'en supporte aucun. Nous utilisons un mécanisme différent pour obtenir des résultats similaires, celui-ci est appelé Soft Updates. Veuillez lire la [FAQ 14 - Soft Updates](#) pour obtenir plus de détails.

8.22 - Reverse DNS

- ou -

pourquoi cela prend autant de temps lorsque je me connecte ?

Beaucoup de nouveaux utilisateurs sur OpenBSD rencontrent un délai de connexion de deux minutes lorsqu'ils utilisent des services tels que [ssh](#) ou [ftp](#). Cela peut aussi être rencontré lorsque l'on utilise un proxy comme [ftp-proxy](#), ou encore pendant l'envoi d'un email depuis une station de travail avec [sendmail](#).

Ceci est le plus souvent dû à un problème de reverse-DNS. DNS est le Serveur de Nom de Domaine (Domaine Name Server), le système qu'Internet utilise pour convertir un nom, comme "www.openbsd.org" en une adresse numérique. Une autre possibilité de DNS est de pouvoir prendre une adresse numérique et la reconvertir en "nom", c'est ce qu'on appelle le "Reverse-DNS".

Pour fournir un meilleur système de connexion, OpenBSD réalise un reverse-DNS sur chaque machine qui lui est attachée de différentes façons, incluant [ssh](#), [ftp](#), [sendmail](#) ou [ftp-proxy](#). Malheureusement dans certains cas, la machine créant la connexion ne possède pas une entrée reverse DNS correcte.

Un exemple de cette situation :

Un utilisateur met en place une machine OpenBSD comme pare-feu et passerelle pour son réseau local, connectant tous ses ordinateurs interne au travers d'une unique adresse en utilisant [NAT](#). Celui-ci peut aussi l'utiliser comme un relais de messagerie sortant. Il suit les notes d'installation, et est satisfait du résultat, à l'exception d'une petite chose -- chaque fois qu'il tente une requête sur la machine de quelque manière que ce soit, il doit attendre deux minutes avant qu'il ne se passe quoi ce soit.

Ce qu'il se passe :

Pour une station de travail derrière le NAT de la passerelle avec une adresse [IP non enregistrée](#) étant 192.168.1.35, l'utilisateur utilise [ssh](#) pour accéder au système de la passerelle. Le client [ssh](#) demande le nom d'utilisateur et le mot de passe, et les envoie à la machine passerelle. La passerelle tente ensuite de savoir qui a essayé de se connecter et réalise une requête reverse DNS de 192.168.1.35. Le problème est que les adresses 192.168.0.0 sont des adresses réservées à un usage privé, et un serveur DNS hors de votre réseau sait qu'il ne doit pas avoir d'information sur ces adresses. Certains vont rapidement retourner un message d'erreur, dans ce cas, OpenBSD comprendra qu'il n'y a pas plus d'informations à obtenir, et continuera rapidement en acceptant l'utilisateur. D'autres serveurs DNS ne retourneront AUCUNE réponse. Dans ce cas, vous vous retrouverez à attendre que le résolveur d'adresses OpenBSD dépasse son délai d'attente, ce qui prendra deux minutes avant que la connexion soit autorisée. Dans le cas de [ftp-proxy](#), certains clients vont dépasser le délai imparti avant que la requête DNS ne dépasse le sien, donnant l'impression que ftp-proxy ne fonctionne pas.

Cela peut être relativement ennuyant. Heureusement, il existe une solution simple pour corriger cela.

Correction, en utilisant `/etc/hosts` :

La solution la plus simple est de partager votre fichier [/etc/hosts](#) entre toutes les stations de travail de votre réseau et vous assurer que le fichier [/etc/resolv.conf](#) contient bien la ligne `lookup file bind` qui confirme que le résolveur va bien considérer en premier lieu le fichier [/etc/hosts](#), avant d'interroger les serveurs DNS spécifiques indiqués par les lignes "nameserver" dans votre fichier [/etc/resolv.conf](#).

Votre fichier `/etc/hosts` ressemblera à quelque chose comme ceci :

```
:::1 localhost.in.example.org localhost
127.0.0.1 localhost.in.example.org localhost
192.168.1.1 gw.in.example.org gw
192.168.1.20 scrappy.in.example.org scrappy
192.168.1.35 shadow.in.example.org shadow
```

Votre fichier `resolv.conf` ressemblera à quelque chose comme ceci :

```
search in.example.org
nameserver 24.2.68.33
nameserver 24.2.68.34
lookup file bind
```

Une objection courante est de dire "Mais, j'utilise DHCP pour mon réseau local ! Comment puis-je configurer mon `/etc/hosts` ?" Très simple, en réalité. Entrez juste les lignes pour toutes les adresses que votre serveur DHCP va délivrer, ainsi que tous les périphériques statiques :

```
:::1 localhost.in.example.org localhost
127.0.0.1 localhost.in.example.org localhost
192.168.1.1 gw.in.example.org gw
192.168.1.20 scrappy.in.example.org scrappy
192.168.1.35 shadow.in.example.org shadow
192.168.1.100 d100.in.example.org d100
192.168.1.101 d101.in.example.org d101
192.168.1.102 d102.in.example.org d102
    [... snip ...]
192.168.1.198 d198.in.example.org d198
192.168.1.199 d199.in.example.org d199
```

Dans ce cas, je présume que votre plage DHCP est définie entre 192.168.1.100 et 192.168.1.199, ainsi que trois définitions statiques au début du fichier.

Si votre passerelle doit utiliser DHCP pour sa configuration, vous devriez rencontrer un problème -- [dhclient](#) va écraser votre fichier `/etc/resolv.conf` à chaque renouvellement de paramètres, cela supprimera la ligne "lookup file bind". Cela peut être résolu en ajoutant la ligne "lookup file bind" au fichier `/etc/resolv.conf.tail`.

Correction, en utilisant un serveur DNS local

Les détails sur cette mise en place, sortent du cadre de ce document, mais la procédure basique est de configurer votre serveur DNS favori, et de vous assurer qu'il a autorité pour les résolutions "forward" et "reverse" pour tous les points du réseau, et que vos ordinateurs (ainsi que votre passerelle) utilisent ce serveur DNS.

8.23 - Pourquoi les pages web d'OpenBSD ne sont-elles pas conformes au HTML4/XHTML ?

Les présentes pages Web ont été écrites avec précaution pour fonctionner sur une large variété de navigateurs actuels, pour les versions 4.0 et suivantes. Nous ne souhaitons pas rendre ces anciennes pages conformes aux normes HTML4 ou XHTML tant que nous ne serons pas sur qu'elles fonctionneront aussi avec les anciens navigateurs ; ce n'est tout simplement pas une priorité. Nous accueillons les nouveaux contributeurs, mais nous vous suggérons de travailler à écrire du code, ou documenter de nouveaux aspects du système, et non de vous concentrer sur la conformité aux nouveaux standards des pages existantes.

8.24 - Mon horloge est décalée de plusieurs heures. Pourquoi ?

Lorsque vous utilisez [rdate\(8\)](#) pour synchroniser votre horloge à un serveur NTP, vous pourrez observer que votre horloge avance/retarde d'une vingtaine de secondes par rapport à votre définition locale de temps.

Ceci est causé par une différence entre le temps UTC (Coordinated Universal Time, basé sur des observations astronomiques) et le temps TAI (International Atomic Time, basé sur des horloges atomiques). Pour compenser les variations dans la rotation de la Terre, des secondes sont insérées dans UTC, mais TAI n'est pas ajusté. Pour une description plus détaillée, cherchez "leap seconds UTC TAI" sur le Web.

Corriger le problème est simple. Dans beaucoup de pays, vous obtiendrez la bonne heure si vous utilisez le paramètre "-c" à [rdate\(8\)](#) et utilisez une zone dans le dossier `/usr/share/zoneinfo/right/`. Par exemple, si vous habitez en Allemagne, vous pouvez utiliser ces commandes :

```
# cd /etc && ln -sf /usr/share/zoneinfo/right/CET localtime
# rdate -ncv ptbtime1.ptb.de
```

Dans d'autres pays, les règles peuvent différer.

8.25 - Pourquoi mon horloge avance/retarde-t-elle de plusieurs heures ?

Par défaut, OpenBSD suppose que l'horloge matérielle de votre équipement est paramétrée de telle façon à indiquer l'heure UTC ("Universal Coordinated Time" - Temps universel coordonné) au lieu de l'heure locale comme l'assument certains systèmes d'exploitation ; cela peut causer certains problèmes en cas de [Multiboot](#).

La plupart des autres systèmes d'exploitation peuvent être configurés de la même façon ce qui permet d'éviter tout problème.

Si avoir l'horloge paramétrée sur UTC représente un problème, vous pouvez changer le comportement par défaut à l'aide de [config\(8\)](#). Par exemple, pour configurer OpenBSD de telle façon à utiliser une horloge matérielle paramétrée en US/Eastern (5 heures de moins que UTC, c'est-à-dire 300 minutes) :

```
# config -ef /bsd
OpenBSD 4.7 (GENERIC) #558: Thu Mar 17 20:46:15 MDT 2010
  deraadt@i386.openbsd.org:/usr/src/sys/arch/i386/compile/
GENERIC
Enter 'help' for information
ukc> timezone 300
timezone = 300, dst = 0
ukc> quit
Saving modified kernel.
```

Veillez consulter [options\(4\)](#) et cherchez l'option "TIMEZONE=value" pour plus d'informations.

Normalement, le fuseau horaire est paramétré à l'installation. Si vous avez besoin de modifier le fuseau horaire, vous pouvez créer un nouveau lien symbolique vers le fichier de fuseau horaire adéquat dans /usr/share/zoneinfo. Par exemple, pour paramétrer la machine de telle façon à utiliser EST5EDT comme le nouveau fuseau horaire local :

```
# ln -fs /usr/share/zoneinfo/EST5EDT /etc/localtime
```

Veillez consulter également :

- [date\(1\)](#)
- [8.24 - Mon horloge est décalée de plusieurs heures. Pourquoi ?](#)
- [OpenNTPD](#)

[\[Index de la FAQ\]](#) [\[Section 7 - Contrôle du clavier et de l'affichage\]](#) [\[Section 9 - Migrer vers OpenBSD\]](#)



www@openbsd.org

\$OpenBSD: faq8.html,v 1.79 2010/06/19 07:38:08 ajacoutot Exp \$

[\[Index de La FAQ\]](#) [\[Section 8 - Questions Générales\]](#) [\[Section 10 - Gestion du Système\]](#)

9 - Astuces et conseils pour les utilisateurs de Linux (et d'autres OS libres Unix-like)

Table Des Matières

- [9.1 - Astuces et conseils pour les utilisateurs d'autres systèmes d'exploitation Unix-like](#)
- [9.2 - Double démarrage de Linux et d'OpenBSD](#)
- [9.3 - Convertir votre fichier de mots de passe de Linux \(ou de tout autre système de type "Sixth Edition"\) au format BSD](#)
- [9.4 - Exécution des binaires Linux sous OpenBSD](#)
- [9.5 - Accéder à vos fichiers Linux depuis OpenBSD](#)

Les utilisateurs Linux pourront trouver des informations supplémentaires a l'adresse suivante : <http://sites.inka.de/mips/unix/bsdlinux.html>.

9.1 - Astuces et conseils pour les utilisateurs d'autres systèmes d'exploitation Unix-like

Bien qu'OpenBSD soit un système d'exploitation Unix-like très traditionnel et très familier pour les personnes ayant utilisé d'autres systèmes d'exploitation Unix-like, il existe d'importantes différences. Les nouveaux utilisateurs d'OpenBSD doivent se baser sur leur propre expérience : si votre connaissance d'Unix se limite à une certaine expérience avec une variante de Linux, vous pourrez trouver OpenBSD "étrange". Soyez rassuré, Linux paraît aussi "étrange" pour quelqu'un qui a commencé son expérience d'Unix avec OpenBSD. Vous devez faire la distinction entre le "standard" et votre expérience.

Si vous avez appris Unix à l'aide d'un des [bons ouvrages](#) sur Unix en général, et si vous avez saisi la "philosophie Unix" puis étendu votre connaissance à un plate-forme particulière, vous trouverez qu'OpenBSD est un Unix "vrai" et très familier. Si vous avez appris Unix en utilisant un procédé de type "saisis ceci au clavier pour faire cela" ou un livre tel que "Apprendre PinkBeenie v8.3 en 31.4 Heures", puis vous vous êtes dit que vous "connaissez Unix", vous trouverez certainement OpenBSD très

différent.

Une différence importante entre OpenBSD et plusieurs autres systèmes d'exploitation est la documentation. Les développeurs OpenBSD sont très fiers des [pages de manuel](#) du système. Les pages de manuel sont *les* sources de référence de la documentation OpenBSD -- ce qui n'est pas le cas de cette FAQ ou des pages maintenues indépendamment du projet ou des "HOWTO"s etc. Lorsqu'un développeur fait une modification au niveau système, on attend de sa part qu'il mette à jour les pages du manuel en même temps que la modification du code système, et pas "après" ou "lorsqu'il aura le temps de le faire" ou lorsque "quelqu'un se plaint". Une page de manuel existe pour virtuellement chaque programme, utilitaire, pilote, fichier de configuration et ainsi de suite dans le système de base. On attend de la part de l'utilisateur qu'il prenne le temps d'effectuer des recherches dans les pages du manuel avant de demander de l'aide sur les [listes de diffusion](#).

Voici quelques unes des différences les plus communes entre OpenBSD et d'autres variantes Unix.

- OpenBSD est un Unix du style "BSD" assez pur, qui suit la conception 4.4BSD de manière rapprochée. Linux et SCO Unix sont des systèmes de type "System V". Certains systèmes d'exploitation Unix-like (y compris quelques distributions Linux) font un mélange de plusieurs caractéristiques propres à SysV et à BSD. Cela prête à confusion particulièrement dans les [scripts de démarrage](#), pour lesquels OpenBSD utilise le système traditionnel [rc\(8\)](#) 4.4BSD.
- OpenBSD est un *système* complet, destiné à être maintenu de manière homogène. Ce n'est pas un "noyau plus quelques utilitaires" qui peuvent être mis à jour séparément. Si vous n'arrivez pas à maintenir votre système (noyau, utilitaires et applications) homogène, de mauvaises choses peuvent arriver.
- Etant donné que plusieurs applications ne sont pas développées pour être compilées directement et s'exécuter dans un environnement OpenBSD, OpenBSD a une [arborescence de ports](#), un système qui permet aux utilisateurs de facilement acquérir du code, le modifier pour qu'il soit compilable sur OpenBSD, installer des dépendances, le compiler, et l'installer et le supprimer de manière standardisée et maintenable. Des [paquetages](#) pré-compilés sont créés et distribués par l'équipe de portage OpenBSD. Les utilisateurs sont [encouragés](#) à utiliser des paquetages au lieu de compiler les leurs.
- OpenBSD utilise CVS pour suivre les modifications des sources. OpenBSD a été un pionnier d'[anonymous CVS](#), qui permet à n'importe qui d'extraire l'arborescence complète des sources pour n'importe quelle version d'OpenBSD (à partir de 2.0 jusqu'à la version actuelle, y compris n'importe quelle révision de n'importe quel fichier située entre ces deux versions) à n'importe quel moment, et vous pouvez accéder aux modifications les plus récentes à peine quelques heures après qu'elles aient été effectuées. Il y a aussi une [interface web pour CVS](#) très utile et facile d'utilisation.
- OpenBSD produit une version officielle disponible sur [CD](#) et par [FTP](#) tous les six mois selon un [agenda prédéfini](#). Des snapshots pour toutes les plates-formes supportées sont créés de manière semi-régulière à partir du code de développement le plus récent. Un des buts du projet OpenBSD

est de garder l'arborescence des sources compilable à tout moment et de faire en sorte que le système issu de la compilation de cette arborescence soit utilisable. L'arborescence peut connaître des problèmes de compilation mais c'est à titre exceptionnel et ces problèmes sont résolus rapidement.

- OpenBSD contient [une cryptographie forte](#), qui ne peut être incluse dans des systèmes d'exploitation provenant de certains pays.
- OpenBSD fait l'objet d'un audit sécurité lourd et continu pour s'assurer de la qualité (et donc, de la sécurité) du code.
- Le noyau d'OpenBSD est `/bsd`.
- Les noms des disques durs sont du type `/dev/wd` (IDE) et `/dev/sd` (SCSI ou équipements à émulation SCSI).
- [/sbin/route](#) sans arguments sous Linux fournit l'état de toutes les routes actives, mais sous OpenBSD (et plusieurs autres systèmes d'exploitation), vous aurez besoin du paramètre "*show*" ou utilisez la commande "[netstat -r](#)".
- OpenBSD NE supporte pas les Systèmes de Fichiers à Journaux tels que ReiserFS, JFS d'IBM ou XFS de SGI. Au lieu de cela, nous utilisons la fonctionnalité [Soft Updates](#) du très robuste Unix Fast File System (FFS) pour atteindre les objectifs de performance et de stabilité.
- OpenBSD inclut [Packet Filter \(PF\)](#), et non pas `ipfw`, `ipchains`, `netfilter`, `iptables` ou `ipf`. Ce qui veut dire que la Traduction d'Adresses IP (connu sous le nom d'IP- Masquerading sous Linux), la gestion de la bande passante et le filtrage est effectué via [pfctl\(8\)](#), [pf\(4\)](#) et [pf.conf\(5\)](#). Consultez le [Guide de l'Utilisateur PF](#) pour des informations détaillées sur la configuration.
- Les adresses des interfaces sont stockées dans les fichiers [/etc/hostname.<interfacename>](#) (par exemple, `/etc/hostname.dc0` pour une interface réseau utilisant le pilote [dc\(4\)](#)). Ces fichiers peuvent contenir un nom d'hôte (résolu à partir de [/etc/hosts](#)) au lieu d'une adresse IP.
- Le nom de la machine est dans le fichier [/etc/myname](#).
- La passerelle par défaut est stockée dans le fichier [/etc/mygate](#).
- Le shell par défaut d'OpenBSD est [/bin/ksh](#), qui est [pdksh](#), le shell Korn du domaine public. Les autres shells inclus sont [csh](#) et [sh](#). Les shells tels que *bash* et *tcsh* peuvent être ajoutés à partir des [paquetages](#) ou installés à partir des [ports](#). Les utilisateurs familiers avec *bash* sont encouragés à [essayer ksh\(1\)](#) avant d'installer *bash* sur leur système -- il a la plupart des fonctionnalités que les utilisateurs utilisent dans *bash*.
- La gestion des mots de passe est différente de celle de la plupart des autres systèmes d'exploitation Unix-like. Les mots de passe sont stockés dans le fichier [master.passwd\(5\)](#) qui ne peut être lu que par *root*. Ce fichier ne doit être modifié que par le programme [vipw](#).
- Les périphériques sont appelés selon le pilote et pas selon le type. Par exemple, il n'existe pas de périphériques `eth*`. Pour une carte Ethernet NE2000, ça serait `ne0` et `xl0` pour des périphériques Ethernet 3Com Etherlink XL ou Fast Etherlink XL etc. Tous ces pilotes ont des pages de manuel dans la section 4. Ainsi, pour en savoir plus sur les messages que votre pilote 3c905 affiche, vous pouvez faire "[man 4 xl](#)".
- OpenBSD/i386, amd64, et d'autres plates-formes utilisent un système de partitionnement de disque à "deux couches", où la première couche est la partition [fdisk](#), visible depuis le BIOS, et

avec laquelle la plupart des utilisateurs d'ordinateurs compatibles IBM sont familiers. La seconde couche est le [disklabel](#), un système de partitionnement BSD traditionnel. OpenBSD supporte un maximum de 15 partitions disklabel sur un disque, toutes dans la même partition fdisk. Ceci permet à OpenBSD de coexister avec d'autres systèmes d'exploitation, y compris d'autres systèmes d'exploitation Unix-like. OpenBSD doit faire partie des 4 partitions "primaires".

- Certains systèmes d'exploitation vous encouragent à adapter votre noyau à votre machine. Les utilisateurs d'OpenBSD sont [encouragés](#) à utiliser le noyau standard GENERIC fourni et testé par les développeurs. Les utilisateurs souhaitant "adapter" et "optimiser" causent souvent plus de problèmes qu'ils n'en résolvent et aucune aide ne leur sera fournie par les développeurs.
- L'équipe de développement OpenBSD travaille dur pour maintenir la [politique de copyright](#) et la [sécurité](#) du projet. Pour cette raison, certaines nouvelles versions de logiciels qui ne remplissent pas les objectifs de licence ou de sécurité ne sont pas intégrées à OpenBSD, et pourraient ne jamais l'être. La sécurité et les licences libres ne seront jamais négligées au profit d'un numéro de version plus grand.

9.2 - Double démarrage de Linux et de OpenBSD

Oui c'est possible !

Lisez [INSTALL.linux](#).

9.3 - Convertir votre fichier de mots de passe de Linux (ou de tout autre système de type "Sixth Edition") au format BSD

Tout d'abord, déterminez si votre fichier de mots de passe Linux est en mode shadow ou pas. Si c'est le cas, installez [John the Ripper](#) à partir de [paquetages ou ports](#) (`security/john`) et utilisez son utilitaire unshadow pour faire fusionner les fichiers `passwd` et `shadow` en un seul fichier type "Sixth Edition".

En utilisant votre fichier de mots de passe Linux, que nous allons appeler `linux_passwd`, vous devez rajouter `:::0:0` entre les champs quatre et sept. [awk\(1\)](#) peut faire cela pour vous.

```
# cat linux_passwd | awk -F : # '{printf("%s:%s:%s:%s:::0:0:%s:%s:%s\n", \
> $1,$2,$3,$4,$5,$6,$7); }' > new_passwd
```

A partir de là, vous devriez éditer le fichier `new_passwd` et enlever les entrées correspondantes à root et d'autres entités système qui sont déjà présentes dans le fichier de mots de passe OpenBSD ou ne sont pas applicables à OpenBSD (toutes). De même, assurez vous qu'il n'y a pas des noms d'utilisateurs ou des ID utilisateurs dupliqués entre `new_passwd` et le fichier `/etc/passwd` de la machine OpenBSD.

La manière la plus facile consiste à utiliser un nouveau `/etc/passwd`.

```
# cat new_passwd >> /etc/master.passwd
# pwd_mkdb -p /etc/master.passwd
```

La dernière étape, `pwd_mkdb`, est nécessaire pour reconstruire les fichiers `/etc/spwd.db` et `/etc/pwd.db`. Cette commande crée aussi un fichier de mots de passe au format "Sixth Edition" (sans les mots de passe cryptés) dénommé `/etc/passwd` pour les programmes qui l'utilisent. OpenBSD utilise un algorithme de cryptage plus fort, `blowfish`, qui est rarement présent dans d'autres systèmes qui utilisent des fichiers de mots de passe au format "Sixth Edition". Pour utiliser cet algorithme plus solide, dites aux utilisateurs d'utiliser `'passwd'` et de changer leur mot de passe. Le nouveau mot de passe sera crypté avec l'algorithme par défaut (`blowfish` si vous n'avez pas édité `/etc/passwd.conf`). Ou, en utilisateur `root`, vous pouvez utiliser `passwd username`.

9.4 - Exécution des binaires Linux sous OpenBSD

OpenBSD/i386 est capable d'exécuter des binaires Linux lorsque le noyau est compilé avec l'option `COMPAT_LINUX` et le paramètre `sysctl kern.emul.linux` positionné. Si vous utilisez le noyau `GENERIC` (ce qui devrait être le cas normalement), l'option `COMPAT_LINUX` est incluse et vous aurez juste besoin de positionner le paramètre `sysctl` précité comme suit :

```
# sysctl kern.emul.linux=1
```

Pour que cette modification soit prise en compte à chaque redémarrage de la machine, supprimez le caractère `"#"` (commentaire) au début de la ligne

```
#kern.emul.linux=1      # enable running Linux binaries
```

dans le fichier `/etc/sysctl.conf`. Vous devez alors obtenir :

```
kern.emul.linux=1      # enable running Linux binaries
```

puis redémarrez votre système pour que cette modification puisse prendre effet.

Pour utiliser des binaires Linux qui ne sont pas statiquement liés (la plupart d'entre eux), vous devez suivre les instructions de la page du manuel [compat_linux\(8\)](#).

Un moyen simple d'obtenir la plupart des bibliothèques Linux les plus communes est d'installer `fedora/base` à partir du site miroir FTP le plus proche. Pour plus d'informations concernant les paquetages et le système de ports, veuillez consulter [FAQ 15 - Le système de ports et de paquetages d'OpenBSD](#). Pour installer le paquetage mentionné ci-dessus, utilisez les commandes suivantes :

```
# export PKG_PATH=ftp://your.ftp.mirror/pub/OpenBSD/4.7/  
packages/i386  
# pkg_add -i fedora_base
```

Veillez noter que [pkg_add\(1\)](#) exécutera automatiquement `sysctl` pour paramétrer correctement `kern.emul.linux` à la bonne valeur après avoir ajouté le paquetage. Cependant, il ne modifie pas `/etc/sysctl.conf`. Si vous voulez activer l'émulation Linux par défaut, vous devez modifier la variable `kern.emul.linux` dans ce fichier.

9.5 - Accéder à vos fichiers Linux depuis OpenBSD

OpenBSD supporte le système de fichiers EXT2FS. Pour plus d'informations, veuillez consulter la [FAQ 14](#).

[\[Index de La FAQ\]](#) [\[Section 8 - Questions Générales\]](#) [\[Section 10 - Gestion du système\]](#)



www@openbsd.org

\$OpenBSD: faq9.html,v 1.58 2010/05/23 16:20:41 ajacoutot Exp \$

OpenBSD

[\[Index de la FAQ\]](#) [\[Section 9 - Migrer vers OpenBSD\]](#) [\[Section 11 - Le système X Window\]](#)

10 - Gestion du Système

Table des matières

- [10.1 - Quand j'essaie de passer root à l'aide de su, on me dit que je suis dans le mauvais groupe.](#)
- [10.2 - Comment dupliquer un système de fichiers ?](#)
- [10.3 - Comment démarrer des services en même temps que le système ? \(Vue d'ensemble de rc\(8\)\)](#)
- [10.4 - Pourquoi les utilisateurs sont interdits de relais quand ils envoient des mails à distance à travers mon système OpenBSD ?](#)
- [10.5 - J'ai mis en place POP, mais j'ai des erreurs quand j'accède à ma messagerie via POP. Que puis-je faire ?](#)
- [10.6 - Pourquoi Sendmail ignore-t-il le fichier /etc/hosts ?](#)
- [10.7 - Configurer HTTP en mode sécurisé à l'aide de ssl\(8\)](#)
- [10.8 - J'ai effectué des changements dans /etc/passwd avec vi\(1\), mais les changements ne semblent pas être pris en compte. Pourquoi ?](#)
- [10.9 - Comment je crée un compte utilisateur ? Ou je supprime un compte utilisateur ?](#)
- [10.10 - Comment puis-je créer un compte pour ftp uniquement ?](#)
- [10.11 - Mise en place des quotas](#)
- [10.12 - Mise en place de Clients et Serveurs KerberosV](#)
- [10.13 - Mise en place d'un serveur FTP Anonyme](#)
- [10.14 - Confiner les utilisateurs à leur répertoire HOME avec ftpd\(8\)](#)
- [10.15 - Appliquer des correctifs sous OpenBSD](#)
- [10.16 - Parlez moi de chroot\(2\) Apache ?](#)
- [10.17 - Puis-je changer le shell de l'utilisateur root ?](#)
- [10.18 - Que puis-je faire d'autre avec ksh ?](#)
- [10.19 - Services d'annuaires](#)
 - [10.19.1 - Quels sont les services d'annuaires disponibles ?](#)
 - [10.19.2 - Considérations sur la sécurité de YP](#)
 - [10.19.3 - Configurer un serveur YP](#)
 - [10.19.4 - Configurer un client YP](#)

10.1 - Quand j'essaie de passer root à l'aide de su, on me dit que je suis dans le mauvais groupe

Les utilisateurs existant sur le système doivent être rajoutés au groupe "wheel" à la main. Ceci est fait pour des raisons de sécurité, et vous devriez apporter une attention toute particulière lorsque vous donnez l'accès à ce groupe à des utilisateurs. Sous OpenBSD, les utilisateurs appartenant au groupe wheel sont autorisés à utiliser le programme [su\(1\)](#) pour devenir root. Les utilisateurs n'appartenant pas au groupe "wheel" ne peuvent pas utiliser su(1). Voici un exemple d'une entrée /etc/group pour mettre l'utilisateur **ericj** dans le groupe "wheel".

Si vous ajoutez un utilisateur avec [adduser\(8\)](#), vous pouvez le mettre dans le groupe wheel en répondant wheel à la question "Invite user into other groups:". Ceci aura pour effet de rajouter l'entrée correspondante dans /etc/group qui ressemble à la ligne suivante :

```
wheel:*:0:root,ericj
```

Si vous cherchez un moyen pour limiter l'accès des utilisateurs aux privilèges du super utilisateur, sans pour autant les mettre dans le groupe "wheel", utilisez [sudo\(8\)](#).

10.2 - Comment dupliquer un système de fichiers ?

Pour dupliquer votre système de fichiers, utilisez [dump\(8\)](#) et [restore\(8\)](#). Par exemple, pour dupliquer tout ce qu'il y a sous le répertoire SRC vers le répertoire DST, faites un :

```
# cd /SRC; dump 0f - . | (cd /DST; restore -rf - )
```

dump est conçu pour vous fournir beaucoup de possibilités de sauvegarde, et c'est peut-être trop si vous voulez juste dupliquer une partie d'un système de fichiers (entier). La commande [tar\(1\)](#) peut être plus rapide pour ce genre d'opération. Le format est très similaire à celui de dump :

```
# cd /SRC; tar cf - . | (cd /DST; tar xpf - )
```

10.3 - Comment démarrer des services en même temps que le système ? (Vue d'ensemble de rc(8))

OpenBSD utilise un démarrage de type [rc\(8\)](#). Il utilise seulement quelques fichiers clés pour le démarrage.

- /etc/rc - Script principal. Ne doit pas être édité.
- /etc/rc.conf - Fichier de configuration utilisé par /etc/rc pour activer les paramètres démarrés en

même temps que le système. Ne doit pas être édité.

- `/etc/rc.conf.local` - Fichier de configuration servant à écraser `/etc/rc.conf`, ainsi vous ne touchez pas à `/etc/rc.conf`, ce qui est important lors de la mise à jour du système.
- `/etc/netstart` - Script pour initialiser le réseau. Ne devrait pas être édité.
- `/etc/rc.local` - Script utilisé pour l'administration locale. C'est là où les informations relatives à de nouveaux services ou des informations spécifiques à l'hôte doivent être stockées.
- `/etc/rc.securelevel` - Script utilisé pour exécuter des commandes qui doivent être exécutées avant que le niveau de sécurité ne change. Voir [init\(8\)](#)
- `/etc/rc.shutdown` - Script exécuté lors de l'arrêt de la machine. Mettez tout ce que vous voulez exécuter avant l'arrêt de la machine dans ce fichier. Voir [rc.shutdown\(8\)](#)

Comment fonctionne rc(8) ?

`/etc/rc.conf` (comme guide), `/etc/rc.conf.local` (pour les changements), `/etc/rc.local` et `/etc/rc.shutdown` sont les principaux fichiers à connaître par l'administrateur système. Pour comprendre le fonctionnement de la procédure `rc(8)`, en voici le déroulement :

`/etc/rc` est appelé après le démarrage du noyau :

- Les systèmes de fichiers sont vérifiés.
- Les variables de configuration par défaut sont lues à partir de `/etc/rc.conf` et ensuite les changements locaux dans `/etc/rc.conf.local`. Les paramètres dans `rc.local.conf` vont surpasser ceux se trouvant dans `rc.conf`.
- Les systèmes de fichiers sont montés.
- `/tmp` est nettoyé et les fichiers d'éditeurs sont préservés.
- Le réseau est configuré à l'aide de `/etc/netstart`
 - Les interfaces réseau sont montées.
 - Le nom d'hôte et le nom de domaine (ainsi que d'autres paramètres) sont positionnés.
- Les services système sont démarrés.
- Diverses vérifications sont effectuées (quota, savecore, etc).
- Les services locaux sont démarrés à partir de `/etc/rc.local`.

Démarrage des services fournis avec OpenBSD

La plupart des services fournis avec OpenBSD sont lancés au démarrage simplement par des variables définies dans le fichier de configuration `/etc/rc.conf`. Pour commencer, jetez un coup d'oeil au fichier [/etc/rc.conf](#) par défaut. Vous verrez des lignes similaires à la ligne suivante :

```
ftpd_flags=NO          # for non-inetd use: ftpd_flags="-D"
```

Une ligne telle que celle-ci montre que [ftpd\(8\)](#) n'est pas lancé au démarrage du système (du moins pas à travers `rc(8)`); `ftpd` est souvent démarré via [inetd\(8\)](#) lisez la [FAQ Serveur FTP Anonyme](#) pour plus

d'informations). Chaque ligne est dotée d'un commentaire qui vous montrent les drapeaux utilisés dans le cadre d'une utilisation courante du service ou du daemon. Cela ne veut pas dire que vous devez appeler ce service avec ces mêmes drapeaux. Lisez la page man correspondante pour savoir comment démarrer un service ou daemon donné de la manière que vous souhaitez.

Nous vous suggérons fortement de ne jamais toucher au fichier */etc/rc.conf* lui-même. A la place, créez un fichier */etc/rc.conf.local*, copiez juste les lignes que vous désirez modifier de */etc/rc.conf* et ajustez les à votre convenance. Cela permettra de faire les futures mises à jour simplement -- tous les changements seront dans un seul fichier. En fait, le [processus de mise à jour](#) standard considère que vous n'avez pas modifié */etc/rc.conf*, et l'écrasera avec la nouvelle version.

Par exemple, voici la ligne par défaut concernant `httpd(8)` :

```
httpd_flags=NO          # for normal use: "" (or "-DSSL"
after reading ssl(8))
```

D'après cet exemple, vous pouvez voir qu'aucun drapeau n'est nécessaire pour démarrer `httpd` normalement. Ainsi, la ligne `httpd_flags=""` ajoutées à */etc/rc.conf.local* suffit. Mais pour démarrer `httpd` avec le support `ssl` (Reportez vous à la [FAQ SSL](#) ou à [ssl\(8\)](#)), vous devez démarrer `httpd` avec une ligne comme celle-ci : `httpd_flags="-DSSL"`, et vous pouvez aussi ajouter d'autres paramètres pour d'autres raisons.

Démarrage et configuration des services locaux

Pour les services que vous installez via les paquetages ou d'autres méthodes, vous devez utiliser le fichier */etc/rc.local*. Par exemple, j'ai installé un service fourni par l'appliquatif */usr/local/sbin/daemonx*. Je souhaite que ce service soit lancé au démarrage. Pour cela, je rajoute les lignes suivantes dans */etc/rc.local* :

```
if [ -x /usr/local/sbin/daemonx ]; then
    echo -n ' daemonx';      /usr/local/sbin/daemonx
fi
```

(Si le service ne se détache pas automatiquement lors de son démarrage, souvenez-vous de rajouter `"&"` à la fin de la commande.)

A partir de là, ce service sera lancé au démarrage. Vous pourrez voir toutes les erreurs au démarrage. Un démarrage normal sans erreurs affichera le message suivant :

```
Starting local daemons: daemonx.
```

rc.shutdown

`/etc/rc.shutdown` est un script exécuté à l'arrêt de la machine. Toutes les tâches à effectuer avant l'arrêt du système devront être ajoutées à ce fichier. Si vous avez `apm`, vous pouvez aussi positionner `"powerdown=YES"`. C'est l'équivalent de `"shutdown -p"`.

10.4 - Pourquoi les utilisateurs sont interdits de relais quand ils envoient des mails à distance à travers mon système OpenBSD ?

Essayez ceci :

```
# grep relay-domains /etc/mail/sendmail.cf
```

Le résultat ressemblerait à la ligne suivante :

```
FR-o /etc/mail/relay-domains
```

Si ce fichier n'existe pas, créez le. Vous devez saisir les hôtes qui envoient des messages à distance en respectant la syntaxe suivante :

```
.domain.com      #Allow relaying for/to any host in domain.com
sub.domain.com   #Allow relaying for/to sub.domain.com and any
host in that domain
10.2             #Allow relaying from all hosts in the IP net
10.2.*.*
```

N'oubliez pas d'envoyer un signal 'HangUP' à `sendmail` (signal qui notifie la plupart des processus de relire leur fichier de configuration) :

```
# kill -HUP `head -1 /var/run/sendmail.pid`
```

Pour plus d'informations

- <http://www.sendmail.org/~ca/email/relayingdenied.html>
- <http://www.sendmail.org/tips/relaying.html>
- <http://www.sendmail.org/antispam.html>

10.5 - J'ai mis en place POP, mais j'ai des erreurs quand j'accède à ma messagerie via POP. Que puis-je faire ?

La plupart des problèmes rencontrés avec POP sont liés aux fichiers temporaires et aux fichiers verrous. Si votre serveur POP renvoie une erreur du type :

```
-ERR Couldn't open temporary file, do you own it?
```

Essayez de positionner les permissions comme suit :

```
permission in /var
drwxrwxr-x  2 bin      mail      512 May 26 20:08 mail

permissions in /var/mail
-rw-----  1 username  username  0 May 26 20:08
username
```

Vérifiez aussi que l'utilisateur possède son propre fichier /var/mail. Bien évidemment, ceci devrait être le cas (comme par exemple l'utilisateur joe qui possède /var/mail/joe) mais si ça n'a pas été configuré proprement, le problème viendrait de là !

Bien entendu, si vous donner l'accès à /var/mail en écriture au groupe mail, vous allez probablement vous exposer à des vagues et obscurs problèmes de sécurité. Il se pourrait que ça ne pose aucun problème mais on ne sait jamais (et particulièrement si vous êtes un site de haut vol, un FAI,...) ! Il existe plusieurs services POP de la collection de ports OpenBSD. Si possible, utilisez [popa3d\(8\)](#) disponible dans le système de base d'OpenBSD. Ou peut-être vous avez sélectionné les mauvaises options pour votre programme POP serveur (comme le dot locking). Ou vous avez peut-être simplement besoin de changer le répertoire dans lequel les verrous sont créés (bien que les opérations de verrouillage ne devraient être bénéfiques qu'au service POP).

Note : Il est à noter que OpenBSD n'a pas de groupe "mail". Vous devez en créer un, si nécessaire, dans le fichier */etc/group*. La ligne suivante devrait suffire :

```
mail:*:6:
```

10.6 - Pourquoi Sendmail ignore-t-il le fichier /etc/hosts ?

Par défaut, Sendmail utilise le DNS pour la résolution de nom, non le fichier */etc/hosts*. Ce comportement peut être changé par l'usage du fichier */etc/mail/service.switch*.

Si vous désirez interroger le fichier d'hôtes avant les serveurs DNS, créez un fichier */etc/mail/service.switch* contenant les lignes suivantes :

```
hosts      files dns
```

Si vous désirez n'interroger QUE le fichier d'hôtes, utilisez ce qui suit :

```
hosts      files
```

Envoyez un signal HUP à Sendmail :

```
# kill -HUP `head -1 /var/run/sendmail.pid`
```

et les changements prendront effet.

10.7 - Configurer HTTP en mode sécurisé à l'aide de SSL(8)

OpenBSD est fourni avec des bibliothèques RSA et un service httpd supportant SSL. Pour utiliser SSL avec [httpd\(8\)](#), vous devez d'abord créer un certificat. Ce certificat sera stocké dans `/etc/ssl` avec la clef correspondante dans `/etc/ssl/private/`. Les étapes décrites ici sont en partie prises de la page de manuel [ssl\(8\)](#). Lisez la pour plus d'informations. Cette partie de la FAQ s'intéresse seulement à la génération d'un certificat RSA pour les serveurs Web. Elle ne décrit pas les certificats serveur DSA. Pour plus d'informations à ce sujet, lisez la page de manuel [ssl\(8\)](#).

Pour commencer, vous aurez besoin de créer votre clé serveur et le certificat en utilisant OpenSSL :

```
# openssl genrsa -out /etc/ssl/private/server.key 1024
```

Ou si vous voulez que la clé soit cryptée avec un mot de passe que vous devez saisir à chaque démarrage des serveurs

```
# openssl genrsa -des3 -out /etc/ssl/private/server.key 1024
```

La prochaine étape consiste à générer une requête de signature de certificat qui est utilisée pour permettre à une autorité de certification (CA) de signer votre certificat. Pour cela, utilisez la commande suivante :

```
# openssl req -new -key /etc/ssl/private/server.key -out /etc/ssl/private/server.csr
```

Le fichier `server.csr` pourra alors être communiqué à une autorité de certification qui signera la clé. Une de ces autorités est **Thawte Certification** que vous pourrez joindre à l'adresse <http://www.thawte.com/>.

Si vous ne pouvez pas vous permettre un tel service, ou si vous voulez auto signer le certificat, vous pouvez utiliser la commande suivante :

```
# openssl x509 -req -days 365 -in /etc/ssl/private/server.csr
\
  -signkey /etc/ssl/private/server.key -out /etc/ssl/
```

server.crt

Avec `/etc/ssl/server.crt` et `/etc/ssl/private/server.key`, vous devez être désormais capable de démarrer [httpd\(8\)](#) avec le drapeau `-DSSL` (consultez la [section à propos de rc\(8\) dans cette faq](#)), activant ainsi les transactions https sur le port 443 de votre machine.

10.7 - J'ai effectué des changements dans `/etc/passwd` avec `vi(1)`, mais les changements ne semblent pas être pris en compte. Pourquoi ?

Si vous éditez `/etc/passwd`, vos modifications seront perdues. OpenBSD génère `/etc/passwd` dynamiquement avec [pwd_mkdb\(8\)](#). Le fichier principal de mots de passe sous OpenBSD est `/etc/master.passwd`. D'après `pwd_mkdb(8)`,

```
FILES
    /etc/master.passwd  fichier courant de mots de passe
    /etc/passwd         fichier de mots de passe au style
"6th Edition"
    /etc/pwd.db         fichier non sécurisé de mots de
passe au format base de données
    /etc/pwd.db.tmp    fichier temporaire
    /etc/spwd.db       fichier sécurisé de mots de passe au
format base de données
    /etc/spwd.db.tmp   fichier temporaire
```

Dans un fichier de mots de passe Unix traditionnel, toutes les informations y compris le mot de passe crypté de l'utilisateur sont à la disposition de n'importe quel utilisateur du système (et c'est la cible principale de programmes tels que Crack). 4.4BSD a introduit le fichier `master.passwd` qui a un format étendu (avec les options additionnelles par rapport à `/etc/passwd`). Ce fichier n'est accessible que pour root. Pour un accès plus rapide aux données, les appels à la bibliothèque qui utilisent ce type d'informations accèdent normalement à `/etc/pwd.db` et à `/etc/spwd.db`.

OpenBSD met à votre disposition un outil qui vous permet d'éditer le fichier de mots de passe. Cet outil s'appelle [vipw\(8\)](#). `vipw` utilisera `vi` (ou votre éditeur favori tel que défini par `$EDITOR`) pour éditer `/etc/master.passwd`. Suite à vos modifications, `vipw` recréera `/etc/passwd`, `/etc/pwd.db` et `/etc/spwd.db` qui tiendront compte de vos modifications. `vipw` verrouille aussi l'accès à ces fichiers de telle façon à en interdire l'accès à quiconque essaie d'en changer le contenu en même temps que vous.

10.8 - Comment je crée un compte utilisateur ? Ou je supprime un compte utilisateur ?

OpenBSD offre deux commandes pour facilement créer des comptes utilisateurs sur le système :

- [adduser\(8\)](#)
- [user\(8\)](#)

Il est toujours possible de créer des utilisateurs à la main en utilisant [vipw\(8\)](#), mais cela complique la plupart des étapes.

La manière la plus facile pour créer un compte utilisateur sous OpenBSD est d'utiliser le script [adduser\(8\)](#). Ce script est paramétrable à travers le fichier `/etc/adduser.conf`. `adduser(8)` permet d'effectuer des vérifications sur la cohérence de `/etc/passwd`, `/etc/group` et les bases de données shell. `adduser(8)` crée pour vous les entrées correspondantes et les répertoires `$HOME`. Il peut aussi envoyer un message de bienvenue aux utilisateurs. Le comportement de ce programme peut être adapté à vos besoins. Pour illustrer notre propos, prenons comme exemple la création du compte **testuser**. Le répertoire de cet utilisateur sera `/home/testuser`. L'utilisateur fera partie du groupe **guest** comme groupe et aura un shell `/bin/ksh`.

```
# adduser
```

```
Use option ``-silent'' if you don't want to see all warnings
and questions.
```

```
Reading /etc/shells
Reading /etc/login.conf
Check /etc/master.passwd
Check /etc/group
```

```
Ok, let's go.
```

```
Don't worry about mistakes. I will give you the chance later
to correct any input.
```

```
Enter username []: testuser
```

```
Enter full name []: Test FAQ User
```

```
Enter shell csh ksh nologin sh [sh]: ksh
```

```
Uid [1002]: Entrée
```

```
Login group testuser [testuser]: guest
```

```
Login group is ``guest''. Invite testuser into other groups:
guest no
```

```
[no]: no
```

```
Login class auth-defaults auth-ftp-defaults daemon default
staff
```

```
[default]: Entrée
```

```
Enter password []: Type password, then Enter
```

```
Enter password again []: Type password, then Enter
```

```

Name:          testuser
Password:      ****
Fullname:      Test FAQ User
Uid:          1002
Gid:          31 (guest)
Groups:        guest
Login Class:   default
HOME:         /home/testuser
Shell:        /bin/ksh
OK? (y/n) [y]: y
Added user ``testuser''
Copy files from /etc/skel to /home/testuser
Add another user? (y/n) [y]: n
Goodbye!

```

Pour supprimer des comptes utilisateurs, utilisez la commande [rmuser\(8\)](#). Cette commande supprimera toute chose relative à l'utilisateur. Elle supprimera son entrée [crontab\(1\)](#), son répertoire \$HOME (s'il lui appartient) et son courrier. Bien évidemment, cette commande supprimera aussi les entrées correspondantes dans */etc/passwd* et */etc/group*. Comme exemple, nous allons utiliser cette commande pour supprimer le compte utilisateur précédemment créé. Notez que la commande vous demande l'identifiant du compte et si oui ou non elle doit supprimer le répertoire home de l'utilisateur.

```

# rmuser
Enter login name for user to remove: testuser
Matching password entry:

testuser:$2a$07$ZWnBOsbqMJ.
ducQBfsTKUe3PL97Ve1AHWJ0A4uLamniLNXLeyrEie:1002
:31::0:0:Test FAQ User:/home/testuser:/bin/ksh

Is this the entry you wish to remove? y
Remove user's home directory (/home/testuser)? y
Updating password file, updating databases, done.
Updating group file: done.
Removing user's home directory (/home/testuser): done.

```

Créer des comptes utilisateurs via user(8)

Ces outils sont moins interactifs que la commande [adduser\(8\)](#), ce qui en facilite l'usage dans des scripts.

La liste complète des outils est :

- [group\(8\)](#)
- [groupadd\(8\)](#)
- [groupdel\(8\)](#)
- [groupinfo\(8\)](#)
- [groupmod\(8\)](#)
- [user\(8\)](#)
- [useradd\(8\)](#)
- [userdel\(8\)](#)
- [userinfo\(8\)](#)
- [usermod\(8\)](#)

Création effective des comptes utilisateurs

Etant donné que la commande `user(8)` n'est pas interactive, la manière la plus simple et la plus efficace pour créer des comptes utilisateurs est d'utiliser la commande `adduser(8)`. La commande `/usr/sbin/user` est seulement une interface aux autres commandes `/usr/sbin/user*`. Ainsi, dans l'exemple qui suit il est possible d'utiliser soit **user add** soit **useradd**. Le choix est votre et ne change rien au résultat.

Dans cet exemple, nous allons créer un compte avec les mêmes spécificités que le compte créé [précédemment](#). `useradd(8)` est bien plus facile à utiliser si vous connaissez les paramètres par défaut avant de créer un compte utilisateur. Ces paramètres se trouvent dans le fichier `/etc/usermgmt.conf` et peuvent être visualisés comme suit :

```
$ user add -D
group          users
base_dir      /home
skel_dir      /etc/skel
shell         /bin/csh
inactive      0
expire        Null (unset)
range         1000..60000
```

Ces paramètres vont être appliqués à chaque nouveau compte si vous ne changez pas leur valeur en utilisant des options en ligne de commande. Par exemple, dans notre cas nous voulons que l'utilisateur appartienne au groupe **guest** et non pas à **users**. Il est à noter que lors de la création des comptes utilisateurs, les mots de passe doivent être spécifiés sous leur forme cryptée en ligne de commande. Vous devez donc utiliser, au préalable, l'utilitaire [encrypt\(1\)](#) pour créer le mot de passe. Par exemple : Les mots de passe par défaut sous OpenBSD utilisent l'algorithme Blowfish avec 6 répétitions. Voici un exemple d'utilisation de la commande `encrypt` :

```
$ encrypt -p -b 6
Enter string:
```

```
$2a$06$YodOZM3.4m6MObBXjeZtBOWArqC2.uRJZXUkOghbieIvSWXVJRz1q
```

Maintenant que nous avons le mot de passe crypté, nous sommes prêts à créer le compte utilisateur :

```
# user add -p '$2a$06$YodOZM3.4m6MObBXjeZtBOWArqC2.
uRJZXUkOghbieIvSWXVJRz1q' -u 1002 \
-s /bin/ksh -c "Test FAQ User" -m -g guest testuser
```

Remarque : Assurez vous d'utiliser " pour englober le mot de passe. L'utilisation de "" ne permet pas d'empêcher le shell d'interpréter le jeu de caractères correspondant au mot de passe avant de les communiquer à user(8). De même, assurez vous d'utiliser l'option **-m** si vous voulez créer le répertoire \$HOME de l'utilisateur et copier les fichiers à partir de */etc/skel* vers \$HOME.

Pour voir si le compte utilisateur a été correctement créé, nous pouvons recourir à plusieurs utilitaires. Voici quelques commandes pour vérifier rapidement que tout s'est bien passé :

```
$ ls -la /home
total 14
drwxr-xr-x  5 root      wheel   512 May 12 14:29 .
drwxr-xr-x 15 root      wheel   512 Apr 25 20:52 ..
drwxr-xr-x 24 ericj     wheel  2560 May 12 13:38 ericj
drwxr-xr-x  2 testuser  guest   512 May 12 14:28 testuser
$ id testuser
uid=1002(testuser) gid=31(guest) groups=31(guest)
$ finger testuser
Login: testuser                Name: Test FAQ User
Directory: /home/testuser      Shell: /bin/ksh
Last login Sat Apr 22 16:05 (EDT) on ttyC2
No Mail.
No Plan.
```

En plus de ces commandes, user(8) fournit son propre utilitaire, appelé userinfo(8), qui permet d'afficher les caractéristiques d'un compte utilisateur :

```
$ userinfo testuser
login    testuser
passwd   *
uid      1002
groups   guest
change   Wed Dec 31 19:00:00 1969
class
gecos    Test FAQ User
dir      /home/testuser
```



```
shell    /bin/ksh
expire  Wed Dec 31 19:00:00 1969
```

Suppression des comptes utilisateurs

Pour supprimer des comptes utilisateurs avec la hiérarchie de commandes `user(8)`, vous devez utiliser `userdel(8)`. Cette commande est simple et efficace. Pour supprimer le compte précédemment créé, utilisez :

```
# userdel -r testuser
```

Notez bien l'option `-r` qui doit être spécifiée si vous voulez supprimer les répertoires `$HOME` aussi. Si vous voulez juste bloquer l'accès au compte sans supprimer des informations liées au compte, utilisez `-p` au lieu de `-r`.

10.10 - Comment puis-je créer un compte pour ftp uniquement ?

Il y a plusieurs méthodes pour effectuer cette opération. Une des manières les plus communes est d'ajouter `/usr/bin/false` à `/etc/shells`. A partir de là, lorsque vous affectez `/usr/bin/false` à un utilisateur, il ne sera plus capable d'ouvrir une session interactive sur le système, néanmoins il pourra utiliser le service ftp. Vous souhaitez peut-être aussi restreindre l'accès en [Confiner les utilisateurs à leur répertoire HOME avec ftpd\(8\)](#).

10.11 - Mise en place des quotas

Les quotas sont utilisés pour limiter l'espace disque disponible pour les utilisateurs. Ce système peut être très utile si vous avez des ressources limitées. Les quotas peuvent être configurés par utilisateur et/ou par groupe.

La première étape pour configurer les quotas est de s'assurer que l'option `QUOTA` est présente dans votre [configuration noyau](#). Cette option est incluse dans le noyau `GENERIC`. Ensuite, vous aurez besoin de marquer les systèmes de fichiers où les quotas sont utilisés dans le fichier `/etc/fstab`. Les mots clés `userquota` et `groupquota` doivent être utilisés pour marquer chaque système de fichiers où les quotas sont activés. Par défaut, les fichiers `quota.user` et `quota.group` seront créés à la racine des systèmes de fichiers où les quotas sont utilisés pour stocker les informations relatives à ces derniers. Si vous voulez les créer ailleurs, spécifiez un fichier avec l'option des quotas dans `/etc/fstab`, par exemple `"userquota=/var/quotas/quota.user"`. Voici un exemple de `/etc/fstab` avec un système de fichiers avec quotas activés et le fichier de quotas se trouvant dans un endroit non-standard :

```
/dev/wd0a / ffs rw,userquota=/var/quotas/quota.user 1 1
```

Maintenant, il faut configurer les quotas par utilisateur. A cette fin, nous utilisons la commande [edquota](#)

(8). Une utilisation simple est "edquota <user>". edquota(8) va utiliser vi(1) pour éditer les quotas à moins que la variable d'environnement EDITOR est positionnée pour charger un autre éditeur. Par exemple la commande :

```
# edquota ericj
```

Affichera un résultat similaire à :

```
Quotas for user ericj:
/: KBytes in use: 62, limits (soft = 0, hard = 0)
   inodes in use: 25, limits (soft = 0, hard = 0)
```

Pour ajouter des limites, éditer là pour donner un résultat similaire à :

```
Quotas for user ericj:
/: KBytes in use: 62, limits (soft = 1000, hard = 1050)
   inodes in use: 25, limits (soft = 0, hard = 0)
```

Notez que l'allocation de quotas est en blocs de 1k. Dans ce cas-ci, softlimit est fixé à 1000k et hardlimit à 1050k. softlimit est une limite qui permet au système de prévenir les utilisateurs quand ils l'ont dépassé. Ils auront alors jusqu'à la fin de leur période de grâce pour redescendre en dessous de cette limite. Les périodes de grâce peuvent être configurées à l'aide de l'option **-t** de edquota(8). Après la fin de la période de grâce, softlimit est géré comme hardlimit. Ce qui cause un échec d'allocation.

Une fois les quotas configurés, il faut les activer. Pour cela, utilisez la commande [quotaon\(8\)](#). Par exemple :

```
# quotaon -a
```

Cette commande analysera le contenu de /etc/fstab et activera les quotas sur les systèmes de fichiers où les options de quota sont positionnées. Maintenant que les quotas sont activés, vous pouvez les visualiser à l'aide de [quota\(1\)](#). Ainsi, la commande "quota <user>" fournit les informations concernant cet utilisateur. Si aucun argument n'est utilisé, quota vous fournira des statistiques sur les quotas. Par exemple :

```
# quota ericj
```

Afficherait :

```
Disk quotas for user ericj (uid 1001):
   Filesystem blocks  quota  limit  grace  files
quota  limit  grace
```

```
0 0 / 62 1000 1050 27
```

Par défaut, les quotas positionnés dans `/etc/fstab` seront activés au démarrage. Pour les désactiver, utilisez :

```
# quotaoff -a
```

10.12 - Mise en place de Clients et Serveurs KerberosV

OpenBSD inclut KerberosV comme un composant pré-installé sur le système par défaut.

Pour plus d'information concernant KerberosV, sur votre système OpenBSD, utilisez la commande :

```
# info heimdal
```

10.13 - Mise en place d'un serveur FTP Anonyme

Le mode FTP anonyme permet à des utilisateurs sans compte d'accéder aux fichiers sur votre machine en utilisant le protocole de transfert de fichiers. Ce chapitre a pour but de fournir une vue d'ensemble de la configuration d'un serveur FTP anonyme, les logs générés, etc...

Création du compte FTP

La première étape consiste à créer un compte *ftp* sur votre système. Ce compte ne doit pas avoir de mot de passe utilisable. Dans cet exemple, nous allons considérer que `/home/ftp` est le répertoire correspondant au compte "ftp" mais vous n'êtes pas obligé de choisir la même chose. Quand le mode anonyme est utilisé, le service ftp va se confiner au répertoire HOME de l'utilisateur *ftp* (dans notre cas, ce répertoire est `/home/ftp`). Pour en savoir plus, lisez les pages du manuel [ftpd\(8\)](#) et [chroot\(2\)](#). Voici un exemple de création du compte *ftp* en utilisant la commande [adduser\(8\)](#). Au préalable, nous avons besoin d'ajouter `/usr/bin/false` au fichier `/etc/shells`. C'est le shell que nous allons attribuer à l'utilisateur *ftp*. Il ne permettra pas de connexion en login à ce compte même si nous configurons un mot de passe vide. Pour effectuer cette opération, il suffit de faire

```
echo /usr/bin/false >> /etc/shells
```

Ensuite vous êtes prêt pour ajouter l'utilisateur *ftp*.

```
# adduser
```

```
Use option ``-silent'' if you don't want to see all warnings  
and questions.
```

```

Reading /etc/shells
Reading /etc/login.conf
Check /etc/master.passwd
Check /etc/group

```

Ok, let's go.

Don't worry about mistakes. I will give you the chance later to correct any input.

Enter username []: **ftp**

Enter full name []: **anonymous ftp**

Enter shell csh false ksh nologin sh tcsh zsh [sh]: **false**

Uid [1002]: *Entrée*

Login group ftp [ftp]: *Entrée*

Login group is ``ftp''. Invite ftp into other groups: guest
no

[no]: **no**

Login class auth-defaults auth-ftp-defaults daemon default
staff

[default]: *Entrée*

Enter password []: *Entrée*

Set the password so that user cannot logon? (y/n) [n]: **y**

```

Name:          ftp
Password:      ****
Fullname:      anonymous ftp
Uid:           1002
Gid:           1002 (ftp)
Groups:        ftp
Login Class:   default
HOME:          /home/ftp
Shell:         /usr/bin/false
OK? (y/n) [y]: y
Added user ``ftp''
Copy files from /etc/skel to /home/ftp
Add another user? (y/n) [y]: n
Goodbye!

```

Configuration du répertoire

L'opération a créé, en plus de l'utilisateur, le répertoire */home/ftp*. C'est ce que nous voulons mais nous avons besoin d'effectuer quelques modifications pour préparer le système à héberger le service FTP anonyme. Ces modifications sont expliquées dans la page du manuel [ftpd\(8\)](#).

Vous n'avez pas besoin de créer un répertoire `/home/ftp/usr` ou `/home/ftp/bin`.

- `/home/ftp` - C'est le répertoire principal. Il doit être possédé par root avec les permissions 555.
- `/home/ftp/etc` - Ce répertoire est optionnel et non recommandé. Son seul but est de fournir des informations sur les comptes utilisateurs existant. Si vous voulez que les fichiers de votre répertoire de ftp soient associés à de vrais utilisateurs, vous devez copier `/etc/pwd.db` et `/etc/group` dans ce répertoire. Les permissions du répertoire doivent être 511. Les permissions sur les deux fichiers doivent être 444. Ils sont utilisés pour fournir une correspondance entre des nombres et les noms attribués aux comptes utilisateurs et groupes. Il n'y a pas de mots de passe dans `pwd.db`. Tous les mots de passe sont stockés dans `spwd.db` alors ne copiez pas ce fichier.
- `/home/ftp/pub` - C'est le répertoire standard pour mettre les fichiers que vous voulez partager. Ce répertoire doit avoir les permissions 555.

Il est à noter que tous ces répertoires doivent être la propriété de "root". Voici à quoi doivent ressembler les répertoires après leur création :

```
# pwd
/home
# ls -laR ftp
total 5
dr-xr-xr-x  5 root  ftp    512 Jul  6 11:33 .
drwxr-xr-x  7 root  wheel  512 Jul  6 10:58 ..
dr-x--x--x  2 root  ftp    512 Jul  6 11:34 etc
dr-xr-xr-x  2 root  ftp    512 Jul  6 11:33 pub

ftp/etc:
total 43
dr-x--x--x  2 root  ftp    512 Jul  6 11:34 .
dr-xr-xr-x  5 root  ftp    512 Jul  6 11:33 ..
-r--r--r--  1 root  ftp    316 Jul  6 11:34 group
-r--r--r--  1 root  ftp  40960 Jul  6 11:34 pwd.db

ftp/pub:
total 2
dr-xr-xr-x  2 root  ftp    512 Jul  6 11:33 .
dr-xr-xr-x  5 root  ftp    512 Jul  6 11:33 ..
```

Démarrage du serveur et logs

Vous pouvez choisir d'exécuter ftpd soit à partir de [inetd\(8\)](#) soit de le lancer directement via les scripts [rc](#). Les exemples suivants vous montreront le service lancé via `inetd.conf`. Tout d'abord, nous devons nous familiariser avec quelques options de ftpd. La ligne par défaut dans `/etc/inetd.conf` est :

```
ftp          stream tcp    nowait  root    /usr/libexec/
ftpd         ftpd    -US
```

Comme vous pouvez le voir, ftpd est invoqué avec `-US`. Ces options vont permettre de loguer les connexions anonymes dans `/var/log/ftpd` et les sessions courantes dans `/var/run/utmp`. Ce qui permet de voir ces sessions via `who(1)`. Dans certains cas, on souhaitera fournir un accès anonyme et désactiver ftp pour les utilisateurs du système. Pour cela, il faut utiliser l'option `-A` de ftpd. Voici une ligne d'invocation de ftpd en mode anonyme exclusif. On utilise aussi `-ll` qui logue chaque connexion vers syslog en plus des commandes ftp get, retrieve, etc...

```
ftp          stream tcp    nowait  root    /usr/libexec/
ftpd         ftpd    -llUSA
```

Note : Les personnes gérant des serveurs ftp à HAUT trafic ne devraient pas invoquer ftpd à partir de `inetd.conf`. La meilleure option consiste à commenter la ligne correspondant à ftpd dans `/etc/inetd.conf` et à démarrer ftpd à partir de `rc.conf.local` avec l'option `-D`. Ce qui va démarrer ftpd en tant que service. Ce mode de fonctionnement est beaucoup moins coûteux et plus rapide que le démarrage via `inetd`. La ligne correspondant à ftpd dans `rc.conf.local` ressemblerait à :

```
ftpd_flags="-DllUSA"          # for non-inetd use:
ftpd_flags="-D"
```

Bien évidemment, cette méthode ne fonctionnera que si ftpd est commenté dans `/etc/inetd.conf` et en veillant qu'`inetd` ait bien relu son fichier de configuration.

Autres fichiers importants

- `/etc/ftpwelcome` - Ce fichier contient le message de bienvenue qui sera affiché aux personnes qui se connectent sur votre serveur ftp.
- `/etc/motd` - Ce fichier contient le message qui sera affiché aux utilisateurs une fois authentifiés sur votre serveur ftp.
- `.message` - Ce fichier peut être mis dans n'importe quel répertoire. Il contient un message qui sera affiché lorsque l'utilisateur entre dans le répertoire où ce fichier se trouve.

10.13 - Confiner les utilisateurs à leur répertoire HOME avec ftpd (8)

Par défaut, lorsque les utilisateurs se connectent en ftp, ils peuvent aller dans n'importe quel répertoire du système, dans la mesure où les contrôles d'accès leur permettent. Dans certains cas, ce comportement peut ne pas être souhaitable. Il est possible de restreindre les utilisateurs ftp à leur répertoire HOME en utilisant "chroot".

Si vous voulez autoriser les connexions ftp en chroot, utilisez l'option **-A** de [ftpd\(8\)](#).

Si vous voulez utiliser chroot de manière plus fine, consultez "[login capability infrastructure](#)" et [ftpd\(8\)](#)

Les utilisateurs appartenant à une classe de connexion où la variable `ftp-chroot` est positionnée seront automatiquement mis dans un chroot. De plus, vous pouvez ajouter un nom d'utilisateur au fichier `/etc/ftpchroot` pour mettre ces utilisateurs dans un chroot. Un utilisateur a uniquement besoin d'être listé dans un de ces endroits.

10.15 - Appliquer des correctifs sous OpenBSD

Même avec OpenBSD, des bogues apparaissent de temps à autre. Certaines bogues causent des problèmes de fiabilité (par exemple, quelque chose peut amener le système à ne plus fonctionner correctement). D'autres bogues causent des problèmes de sécurité (qui peuvent permettre à d'autres personnes d'utiliser votre machine de façon inattendue). Lorsqu'un bogue critique est trouvé, le correctif sera mis en place au niveau de l'arborescence du code source *-current*. Ce correctif sera ensuite propagé vers les [versions maintenues](#) d'OpenBSD. Ces correctifs apparaissent sur la [page web des errata](#). Ils sont séparés en correctifs "communs" applicables à toutes les [plates-formes](#), et en correctifs applicables à une ou plusieurs plates-formes mais pas toutes.

Cependant, il est à noter qu'il n'y a pas de correctifs pour les nouvelles fonctionnalités ajoutées à OpenBSD. Les correctifs corrigent uniquement des problèmes de stabilité ou de sécurité qui doivent être réglés très rapidement sur les systèmes impactés (mais pas tous les systèmes vu que l'application de tel ou tel correctif dépend des applications utilisées).

Il existe trois façons d'installer les correctifs sur votre système :

- **Mettez à jour votre système en [-current](#).** Vu que tous les correctifs sont appliqués systématiquement au code source de la branche *-current*, la mise à jour de votre système avec le dernier snapshot disponible est une excellente façon d'installer les correctifs. Cependant, l'utilisation de la branche *-current* n'est pas adaptée à tout le monde.
- **Mettez à jour votre système en [-stable](#).** Vous pouvez faire ceci en téléchargeant votre arborescence du code source ou en la mettant à jour en utilisant la branche *-stable* appropriée puis en recompilant le noyau et le reste du système d'exploitation. C'est probablement la méthode la plus aisée mais la recompilation de tout le système prend du temps. De plus, le téléchargement complet du code source peut prendre du temps aussi si vous avez une bande passante limitée.
- **Corrigez, compilez et installez les fichiers impactés individuellement.** C'est la méthode que nous allons aborder ci-après de manière détaillée. Bien que cette méthode nécessite peu de bande passante et prend, typiquement, peu de temps par rapport à un téléchargement ou une mise à jour via `cvs(1)` et une compilation complète de tout le système, elle peut parfois être la plus difficile à utiliser. En effet, il n'y a pas une procédure universelle à suivre pour la mise à jour. Certaines fois

vous devez mettre en oeuvre un correctif pour une application donnée puis recompiler et installer cette dernière. D'autres fois, vous aurez peut-être à recompiler des sections entières de l'arborescence des sources si le problème est localisé au niveau d'une bibliothèque.

Encore une fois, la correction de fichiers individuels n'est pas toujours simple. Pensez à utiliser la deuxième méthode décrite plus haut et suivre la branche [-stable](#) (dite aussi la branche "des correctifs") d'OpenBSD. L'utilisation combinée de ces différentes méthodes est possible si vous comprenez exactement comment ça fonctionne. Les nouveaux utilisateurs devront choisir une seule méthode.

Quelle est la différence entre les correctifs de la page des errata et ce qui existe au niveau de la base de données CVS ?

Tous les correctifs postés sur la [page web des errata](#) concernent uniquement l'arborescence des sources de la version indiquée dans cette page. Les autres correctifs qui concernent l'arborescence actuelle de CVS peuvent contenir certaines modifications qui ne sont peut-être pas désirables sur la version de production. Ceci est important : Si vous avez installé un snapshot et que vous avez téléchargé les arborescences du code source au moment où vous avez obtenu le snapshot, il se peut que si vous essayer d'appliquer un des correctifs publiés, ça ne marche pas à cause d'une modification du code source.

Application des correctifs

Les correctifs d'OpenBSD sont distribués sous la forme de fichiers diff unifiés. Ces fichiers sont des fichiers texte qui contiennent les différences par rapport au code source d'origine. Ils ne sont **PAS** distribués sous forme binaire. Cela veut dire que pour appliquer les correctifs, vous devez avoir à disposition sur votre système le code source de la version **RELEASE** d'OpenBSD. De manière générale, vous devriez avoir à disposition l'arborescence complète du code source. Si votre machine héberge une version à partir d'un CDROM officiel, l'arborescence du code source est disponible sur le disque 3. Elle est aussi disponible sous forme de fichiers à partir des [serveurs FTP](#). Nous allons désormais supposer que vous avez l'arborescence complète à votre disposition.

A titre d'exemple, nous allons appliquer le correctif 001 pour OpenBSD qui corrige un bogue au niveau du pilote [st\(4\)](#) qui gère les lecteurs de bandes magnétiques. Sans ce correctif, la restauration de sauvegardes est assez difficile. Les personnes utilisant un lecteur de bandes magnétiques ont *besoin* de ce correctif. Les autres personnes peuvent s'en passer. Jetons un coup d'oeil à ce correctif :

```
# more 001_st.patch
Apply by doing:
    patch -p0 < 001_st.patch
```

```
Rebuild your kernel.
Index: sys/scsi/st.c
```

```
=====
```



```

RCS file: /cvs/src/sys/scsi/st.c,v
retrieving revision 1.41
retrieving revision 1.41.2.1
diff -u -p -r1.41 -r1.41.2.1
--- sys/scsi/st.c      1 Aug 2004 23:01:06 -0000      1.41
+++ sys/scsi/st.c      2 Nov 2004 01:05:50 -0000
1.41.2.1
@@ -1815,7 +1815,7 @@ st_interpret_sense(xs)
     u_int8_t skey = sense->flags & SSD_KEY;
     int32_t info;

-     if (((sense->flags & SDEV_OPEN) == 0) ||
+     if (((sc_link->flags & SDEV_OPEN) == 0) ||
         (serr != 0x70 && serr != 0x71))
         return (EJUSTRETURN); /* let the generic code
handle it */

```

Comme vous pouvez le constater, le début du patch inclut de brèves instructions sur la façon de l'appliquer. Nous admettons que vous avez placé ce patch dans le répertoire `/usr/src` auquel cas les étapes suivantes seront utilisées :

```

# cd /usr/src
# patch -p0 < 001_st.patch
Hmm... Looks like a unified diff to me...
The text leading up to this was:

```

```

-----
|Apply by doing:
|       cd /usr/src
|       patch -p0 < 001_st.patch
|

```

```

|Rebuild your kernel.
|

```

```

|Index: sys/scsi/st.c
|

```

```

=====
|RCS file: /cvs/src/sys/scsi/st.c,v
|retrieving revision 1.41
|retrieving revision 1.41.2.1
|diff -u -p -r1.41 -r1.41.2.1
|--- sys/scsi/st.c      1 Aug 2004 23:01:06 -0000      1.41
|+++ sys/scsi/st.c      2 Nov 2004 01:05:50 -0000
|1.41.2.1
|

```

```

-----
Patching file sys/scsi/st.c using Plan A...

```

```
Hunk #1 succeeded at 1815.
indique le
succès de l'opération
done
```

```
<-- Ce message
```

Le message "Hunk #1 succeeded" indique que le correctif a été appliqué avec succès. Plusieurs correctifs sont plus complexes que l'exemple utilisé, et impliqueront de multiples "hunks" et fichiers. Dans ce cas, il faudrait que vous vous assuriez que tous les "hunks" ont été appliqués avec succès pour tous les fichiers concernés. Si ce n'est pas le cas, ceci veut normalement dire que votre arborescence du code source n'est pas bonne, que vous n'avez pas suivi les instructions, ou que le correctif ne correspond pas au correctif originel. Les correctifs sont très sensibles aux espaces blancs -- un copier/coller depuis votre navigateur changera la plupart du temps les caractères de tabulation en espaces ou modifiera les espaces blancs de telle façon à rendre le correctif inutilisable.

Vous pouvez maintenant [recompiler le noyau](#) comme d'habitude, l'installer et redémarrer le système.

Les correctifs ne s'appliquent pas systématiquement au noyau. Dans certains cas, vous devrez recompiler des utilitaires. Dans d'autres, vous devrez recompiler tous les utilitaires liés statiquement à une bibliothèque sujette à correction. Suivez les instructions dans l'en-tête du correctif, et si vous n'êtes pas certain, recompilez tout le système.

Les correctifs qui n'impactent pas directement votre utilisation du système n'ont pas besoin d'être appliqués normalement. Par exemple, si vous n'aviez pas de lecteur de bande magnétique dans votre système, vous ne bénéficierez pas du correctif ci-dessus. Cependant, les correctifs sont supposés être appliqués dans l'ordre. Il est donc possible qu'un correctif ultérieur dépend d'un correctif apparu plutôt. Soyez conscient de ce mode de fonctionnement si vous sélectionnez la méthode consistant à choisir vous-même vos correctifs. Si vous avez un doute, appliquez-les tous, dans l'ordre de leur mise à disposition.

10.16 - Parlez moi de chroot(2) Apache ?

Sous OpenBSD, le serveur [httpd\(8\)](#) d'Apache est [chroot\(2\)](#)é par défaut. Etant un grand pas en avant du point de vue de la sécurité, cela peut créer des problèmes si vous n'y êtes pas préparé.

Qu'est-ce qu'un chroot ?

Une application [chroot\(2\)](#)ée est bloquée dans un répertoire spécifique et ne peut errer dans les autres répertoires de l'arbre du système de fichiers et voit ce répertoire comme son répertoire / (root). Dans le cas d'[httpd\(8\)](#), le programme démarre, ouvre ses fichiers log, ouvre ses ports TCP (bien qu'il n'accepte pas encore de données), et lit son fichier de configuration. Ensuite, il se fixe lui-même dans le répertoire `/var/www` et baisse ses privilèges. Ce qui veut dire que tous les fichiers servis et utilisés par Apache, doivent être dans le répertoire `/var/www`. Dans la configuration par défaut d'OpenBSD, tous les fichiers du répertoire `/var/www` sont en lecture seule pour l'utilisateur qui fait tourner Apache, `www`. Cela aide

considérablement la sécurité ; si il devait y avoir un problème de sécurité, les dégâts seraient confinés dans un seul répertoire avec les permissions de "lecture seule" et aucune ressource pour causer des problèmes.

Qu'est-ce que cela signifie pour l'administrateur ?

En gros, chroot(2)er Apache est quelque chose qui n'est pas adopté par la plupart des autres systèmes d'exploitation. Beaucoup d'applications et de configurations système ne fonctionneront plus comme avant sans aménagement. De plus, il faut se souvenir que sécurité et commodité sont souvent incompatibles. L'implémentation d'Apache sous OpenBSD ne sacrifie pas la sécurité aux profit des capacités ou de la "facilité".

- **Hiérarchie historique du système de fichiers** : Les serveurs mis à jour depuis d'anciennes versions d'OpenBSD peuvent avoir les fichiers web situés dans les répertoires HOME des utilisateurs, ce qui clairement ne fonctionnera pas dans un environnement chroot(2)é étant donné qu'htpd(8) ne peut atteindre le répertoire */home*. Les administrateurs découvriront peut-être que leur partition */var/www* existante est trop petite pour accueillir tous les fichiers web. Vos options sont dès lors de restructurer votre système ou de ne pas utiliser l'option du chroot(2). Vous pouvez, bien évidemment, utiliser des liens symboliques dans le répertoire HOME de l'utilisateur pointant vers les sous-répertoires dans */var/www*, mais vous ne pouvez PAS utiliser des liens dans */var/www* pointant vers une autre partie du système de fichier -- ceci ne peut fonctionner dû au chroot(2). Notez que si vous voulez que vos utilisateurs aient un [accès FTP chroot\(2\)é](#), ceci ne fonctionnera pas plus étant donné que le chroot FTP va (à nouveau) vous empêcher d'accéder aux destinations de ces liens symboliques. Une solution est donc, de ne pas utiliser */home* comme répertoire HOME pour ces utilisateurs mais plutôt quelque chose similaire à */var/www/users*. Les liens symboliques sont totalement utilisables à l'intérieur du chroot(2) mais ceux-ci doivent être relatifs et non absolus.
- **Rotation des fichiers log** : Normalement, une rotation des fichiers log est réalisée en renommant les anciens fichiers, ensuite en envoyant à htpd(8) un signal SIGUSR1 qui forcera Apache à fermer ses anciens fichiers logs et à en ouvrir de nouveaux. Ceci n'est désormais plus possible, étant donné que htpd(8) n'a pas la possibilité d'ouvrir les fichiers de log en écriture une fois que ses privilèges ont été abaissés. htpd(8) doit donc être stoppé et relancé. Cela demande parfois quelques secondes pour que tous les processus enfants se terminent, ce qui est obligatoire avant que htpd(8) ne puisse redémarrer ; une possibilité de rotation des logs pourrait être :

```
# apachectl stop
  rename your log files
# apachectl start ; sleep 10 ; apachectl start
```

Oui, la dernière ligne tente de redémarrer Apache immédiatement et dans le cas où cela ne fonctionnerait pas, on attend quelques secondes puis on réessaye. Et oui, cela signifie bien que pendant quelques secondes chaque fois que vous effectuerez une rotation des logs, votre serveur web sera inaccessible. Bien que cela puisse poser des problème, n'importe quelle tentative pour permettre à htpd(8) de rouvrir des fichiers après avoir chroot(2)é ira à l'encontre de l'intérêt même

du chroot ! Il existe néanmoins d'autres techniques, notamment loguer vers un [pipe\(2\)](#), et utiliser un programme extérieur afin de réaliser une rotation des fichiers à la fin du pipe.

- **Modules Apache existants** : Pratiquement, ils se lanceront tous, cependant, certains pourraient ne pas fonctionner correctement dans le chroot(2), et pourraient causer des problèmes lors de "[apachectl restart](#)", générant une erreur, causant httpd(8) à se stopper.
- **CGIs existants** : La plupart ne fonctionneront pas tels quels. Ils auront certainement besoin de programmes ou de bibliothèques se trouvant hors de */var/www*. Certains peuvent être corrigés en étant compilés statiquement (n'ayant pas besoin de bibliothèques se trouvant dans un autre répertoire), la plupart peuvent être corrigés en copiant dans */var/www* les fichiers nécessaires à l'application, bien que cette manœuvre soit non triviale et requiert une certaine connaissance du programme.
- **Options de montage du système de fichiers** : Par défaut sous OpenBSD, votre partition */var* est monté avec les options `nosuid` et `nodev`. Si vous tentez d'utiliser une application à l'intérieur du chroot, vous serez peut-être amené à changer ces options. Bien entendu, il est possible que vous ayez besoin d'effectuer ces changements même si vous n'utilisez pas l'option du chroot.
- **Résolution de noms** : httpd(8) à l'intérieur de chroot(2) NE sera pas capable d'utiliser */etc/hosts* ou */etc/resolv.conf*. Par conséquent, si vous avez applications qui nécessitent la résolution de nom, vous devrez remplir */var/www/etc/hosts* et/ou */var/www/etc/resolv.conf* dans l'environnement chroot(2). Il est à noter que certaines applications ont besoin de faire la résolution de "localhost" pour fonctionner.

Dans certains cas, les applications ou les configurations peuvent être changées pour fonctionner dans le chroot(2). Dans d'autres cas, vous devrez tout simplement retirer cette option en utilisant l'option `-u` de httpd(8) dans [/etc/rc.conf.local](#).

Exemple de chroot(2) d'une application : wwwcount

Voyons comment mettre en place chroot(2) pour une application à travers un exemple. Cet exemple se base sur `wwwcount`, un compteur tout simple d'accès aux pages web disponible dans les [paquetages](#). Bien qu'il soit un programme très efficace, `wwwcount` ne sait rien d'Apache chroot(2)é et ne fonctionnera pas dans un environnement chroot(2)é avec sa configuration de base.

Tout d'abord, nous installons le paquetage [wwwcount](#). Nous le configurons et le testons et c'est là que nous en déduisons qu'il ne semble pas fonctionner : Apache nous affiche le message "Internal Server Error". La première étape consiste à arrêter Apache et à le redémarrer avec l'option `-u` pour vérifier que le problème vient bien du chroot(2) et pas de la configuration système.

```
# apachectl stop
/usr/sbin/apachectl stop: httpd stopped
# httpd -u
```

Après avoir fait cela, nous constatons que le compteur fonctionne correctement, du moins après avoir

changé les droits d'un répertoire afin qu'Apache (et les CGIs qu'il exécute) puisse écrire à des fichiers. Ainsi, nous sommes maintenant certains que le problème vient du chroot. Nous arrêtons alors et redémarrons Apache à nouveau en utilisant le chroot par défaut :

```
# apachectl stop
/usr/sbin/apachectl stop: httpd stopped
# httpd
```

Un bon point pour commencer serait de supposer que wwwcount utilise des bibliothèques et d'autres fichiers qu'il ne peut accéder une fois dans le chroot. On peut utiliser à cet effet la commande [ldd\(1\)](#) pour trouver les dépendances dynamiques dont le CGI a besoin :

```
# cd /var/www/cgi-bin/
# ldd Count.cgi
Count.cgi:
          Start      End          Type Open Ref GrpRef Name
          1c000000 3c007000  exe   1    0   0      /var/www/cgi-
bin/Count.cgi
          0c085000 2c0be000  rlib  0    1   0      /usr/lib/libc.
so.50.1
          08713000 08713000  rtld  0    1   0      /usr/libexec/
ld.so
```

Ah ! voilà un problème. Deux fichiers ne sont pas disponibles dans l'environnement chroot(2). Alors, on les copie dans cet environnement :

```
# mkdir -p /var/www/usr/lib /var/www/usr/libexec
# cp /usr/lib/libc.so.50.1 /var/www/usr/lib
# cp /usr/libexec/ld.so /var/www/usr/libexec
```

Puis nous essayons le compteur à nouveau.

Au moins, le programme s'exécute maintenant et nous affiche des messages d'erreur directement : "Unable to open config file for reading". Nous avons bien progressé mais nous n'avons pas encore fini. Le fichier de configuration se trouve normalement dans le répertoire `/var/www/wwwcount/conf`, mais au sein de l'environnement chroot, le programme devrait le voir sous `/wwwcount/conf`. Nous avons donc deux options. Soit nous recompilons le programme pour qu'il tienne compte du nouveau fichier de configuration par défaut (où plutôt du chemin pour l'atteindre) soit nous déplaçons les fichiers de données. Vu que nous avons installé le programme à partir d'un paquetage, nous prenons l'option 2, à savoir le déplacement des fichiers de données. Afin que nous puissions utiliser exactement la même configuration dans un environnement chroot(2)é ou pas, nous utiliserons un lien symbolique :

```
# mkdir -p /var/www/var/www
```

```
# cd /var/www/var/www
# ln -s ../../wwwcount wwwcount
```

Remarquez que le lien symbolique a été pensé pour fonctionner au sein du chroot. Nous testons notre programme à nouveau et nous voilà avec un autre problème. Maintenant wwwcount se plaint de ne pas trouver les fichiers "strip image" qu'il utilise pour afficher des messages. Après quelques recherches, nous nous rendons compte que ces fichiers sont stockés dans /usr/local/lib/wwwcount, nous devons donc les copier dans le chroot aussi.

```
# tar cf - /usr/local/lib/wwwcount | (cd /var/www; tar xpf - )
```

Nous testons à nouveau ... et ça marche !

Notez que nous n'avons copié que les fichiers strictement nécessaires au bon fonctionnement. En général, seuls les fichiers nécessaires à l'application doivent être copiés dans le chroot.

Dois-je utiliser chroot ?

Dans l'exemple précédent, le programme est relativement simple et pourtant nous avons rencontré différents types de problèmes.

Toutes les applications ne peuvent ou ne devraient pas être chroot(2)ées.

Le but est la mise en place d'un serveur web sécurisé et le chroot(2)age n'en est qu'un outil, pas un but en soi. Souvenez-vous, la configuration initiale de l'Apache chroot(2)é sous OpenBSD fait en sorte que l'utilisateur sous lequel le programme httpd(8) tourne ne peut pas lancer de programme, ne peut pas modifier de fichiers et ne peut pas prendre l'identité d'un autre utilisateur. Réduire ces restrictions diminuera votre sécurité, chroot ou pas.

Certaines applications sont relativement simples et les mettre dans un chroot(2) n'a aucun sens. D'autres sont très complexes. Elles ne valent pas les efforts nécessaires pour les mettre en chroot(2) et même si c'était le cas, vous perdriez les avantages du chroot(2) après avoir copié la masse importante de fichiers dont elles ont besoin pour fonctionner. Ainsi le programme OpenWebMail a besoin de pouvoir lire et écrire dans le répertoire mail, le répertoire home de l'utilisateur et doit pouvoir travailler en tant que n'importe quel utilisateur du système. Essayer le mettre cette application en chroot serait inutile car vous serez obligé de désactiver tous les bénéfices du chroot(2)age. Même avec des applications aussi simples que le compteur précédent, il doit pouvoir écrire sur le disque (pour garder la trace de ses compteurs) et donc, une *partie* du bénéfice du chroot(2) est perdue.

Toute application qui doit fonctionner sous root ne vaut pas le coup d'être chroot(2)ée puisque root peut généralement s'échapper du chroot(2).

N'oubliez pas, si la procédure de chroot pour votre application est trop complexe vous pourriez ne pas mettre à jour votre système aussi souvent qu'il le faudrait. Ceci pourrait rendre votre système MOINS sécurisé qu'un système plus facilement administrable et dont le chroot est désactivé.

10.17 - Puis-je changer le shell de l'utilisateur root ?

Il est parfois dit qu'il ne faut pas changer le shell de l'utilisateur root, bien qu'il n'y ait aucune raison de ne pas le faire sous OpenBSD.

Le shell par défaut sur OpenBSD de l'utilisateur *root* est [ksh](#).

Une directive Unix traditionnelle est d'utiliser pour l'utilisateur root des shells compilés statiquement, car si votre système démarre en mode utilisateur unique, les partitions non-root ne seront pas montées et les shells liés dynamiquement ne seront pas capable d'accéder aux bibliothèques situées dans la partition /usr. Ceci n'étant pas très important pour OpenBSD, car le système vous demandera de choisir un shell lors d'un démarrage en mode utilisateur unique, le shell par défaut étant [sh](#). Les trois shells standards sous OpenBSD ([csh](#), [sh](#) et [ksh](#)) sont liés statiquement et donc utilisables en mode utilisateur unique.

10.18 - Que puis-je faire d'autre avec *ksh* ?

Sous OpenBSD, [ksh](#) est [pdksh](#), le Shell Korn du Domaine Public (Public Domain Korn Shell), et est le même binaire que [sh](#).

Les utilisateurs qui sont à l'aise avec *bash*, souvent utilisé sur les systèmes Linux, trouveront probablement [ksh](#) très familier. Ksh(1) offre la plupart des options habituellement utilisées avec *bash*, notamment l'achèvement des commandes avec la touche tab, l'édition de la ligne de commande et l'historique via les touches fléchées, et CTRL-A/CTRL-E pour aller au début/à la fin de la ligne de commande. Si vous désirez d'autres options de *bash*, *bash* peut être installé soit via les [paquetages](#) ou soit via les [ports](#).

Le prompt de *ksh* peut être facilement changé de manière à fournir plus d'informations que le simple "\$ " par défaut en modifiant la variable PS1. Par exemple, en insérant la ligne suivante :

```
export PS1='$PWD $ '
```

dans votre `/etc/profile`, cela produira le prompt suivant :

```
/home/nick $
```

Consultez le fichier [/etc/ksh.kshrc](#), qui inclut plusieurs options utiles ainsi que des exemples, et qui peut être invoqué dans les fichiers `.profile` de vos utilisateurs.

ksh(1) sous OpenBSD a été amélioré. Un certain nombre de caractères spéciaux ont été ajoutés au niveau de la chaîne primaire de l'invite de commandes, PS1. Ces caractères sont similaires à ceux présents dans bash. Par exemple :

- \e - Insertion d'un caractère d'échappement ASCII.
- \h - Le nom d'hôte sans le nom de domaine.
- \H - Le nom d'hôte complet, avec le nom de domaine.
- \n - Insertion d'un caractère de retour à la ligne.
- \t - L'heure actuelle, sur 24 heures, au format HH:MM:SS.
- \u - Le nom de l'utilisateur actuel.
- \w - Le répertoire de travail actuel. \$HOME est abrégé en '~'.
- \W - La racine du répertoire de travail actuel.
- \\$ - Affiche "#" pour les super-utilisateurs, et "\$" pour les autres

(consultez la page du manuel [ksh\(1\)](#) pour plus de détails , et beaucoup, beaucoup plus de caractères spéciaux ! Veuillez noter que le caractère "\$" a une signification spéciale à l'intérieur des double quotes. Il est donc à manipuler avec précaution)

Vous pouvez par exemple utiliser la commande suivante :

```
export PS1="\n\u@\H\n\w \\\$ "
```

pour disposer d'une invite de commandes très parlante mais dont l'utilité n'est que relative.

10.19 - Services d'Annuaire

OpenBSD peut-être utilisé aussi bien comme client que serveur de bases de données contenant l'identification de l'utilisateur, l'information sur les groupes et d'autres données liées au réseau.

10.19.1 - Quels sont les services d'annuaire disponibles ?

Evidemment vous pouvez utiliser différents services d'annuaire sur OpenBSD. Mais YP est le seul qui peut-être accessible directement en utilisant les fonctions standards de la librairie C comme [getpwent\(3\)](#), [getgrent\(3\)](#), [gethostbyname\(3\)](#) et bien d'autres. Donc, si vous conservez vos données dans une base de données YP, vous n'avez pas besoin de les copier dans les fichiers de configuration locaux comme [master.passwd\(5\)](#) avant de l'utiliser, par exemple pour authentifier des utilisateurs système.

YP est un service d'annuaire compatible avec Sun Microsystems NIS (Network Information System). Regardez [yp\(8\)](#) pour un aperçu des pages de manuel disponibles. Soyez prudent, certains systèmes d'exploitation possèdent des services d'annuaire qui contiennent des noms similaires mais qui sont incompatibles, comme par exemple NIS+.

Pour utiliser d'autres services d'annuaires à l'exception de YP, vous avez besoin de remplir les fichiers de configuration locaux de l'annuaire ou vous avez besoin d'un frontal YP pour l'annuaire. Par exemple, vous pouvez utiliser le port `sysutils/login_ldap` quand vous choisissez le premier, alors que le daemon [ypldap\(8\)](#) fournit ce dernier.

Pour certaines applications, synchroniser simplement un petit nombre de fichiers de configuration sur un groupe de machines en utilisant des outils comme [cron\(8\)](#), [scp\(1\)](#) ou `rsync` (disponible dans les ports) constitue une alternative robuste et facile à un service d'annuaire complet.

10.19.2 - Considérations sur la sécurité de YP

Pour des raisons de compatibilité, toutes les fonctionnalités de sécurité de OpenBSD construites dans l'implémentation de YP sont *désactivées* par défaut. Même quand elles sont toutes activées, le protocole NIS reste intrinsèquement non sécurisé pour deux raisons : Toutes les données, incluant les données sensibles comme les hashes de mot de passe, sont transmises non chiffrées sur le réseau, et ni le client ou le serveur ne peut vérifier de manière fiable l'identité de l'autre.

Donc, avant de mettre en place un serveur YP, vous devez considérer ces faiblesses de sécurité intrinsèque comme acceptable dans votre contexte. En particulier, YP n'est pas adapté si des attaquants potentiels ont un accès physique à votre réseau. Quiconque obtient l'accès root d'un des ordinateurs connecté sur l'un de vos segment réseau faisant transiter du trafic YP peut se connecter sur votre domaine YP et récupérer des données. Dans certains cas, faire transiter du trafic YP à travers des tunnels SSL ou IPSec peut-être une option, ou vous devrez considérer de combiner YP avec une authentification [kerberos\(8\)](#).

10.19.3 - Configurer un serveur YP

1. Un serveur YP sert un groupe de clients appelé un "domaine". Vous devez d'abord choisir un nom de domaine; cela peut-être une chaîne arbitraire et ne doit pas avoir de lien avec les noms de domaines DNS. Choisir un nom aléatoire comme "Eepoo5vi" peut améliorer de façon marginale la sécurité, avec l'effet d'être plutôt de la sécurité par l'obscurité. Dans le cas où vous devez maintenir plusieurs domaines YP distincts, il est sûrement meilleur de choisir des noms de description comme "ventes", "marketing" et "recherche" pour ne pas avoir d'erreurs d'administration système causées par l'obscurité. Il faut aussi remarquer que certaines versions de SunOS doivent utiliser le nom de domaine DNS de l'hôte, donc votre choix est plutôt restreint dans un réseau incluant de tels hôtes.

Utilisez l'utilitaire [domainname\(1\)](#) pour configurer le nom de domaine et le mettre dans le fichier [defaultdomain\(5\)](#) pour qu'il soit automatiquement configuré au démarrage du système.

```
echo "puffynet" > /etc/defaultdomain
domainname `cat /etc/defaultdomain`
```

2. Initialiser le serveur YP en utilisant la commande interactive :

```
ypinit -m
```

A ce stade, il n'est plus nécessaire de spécifier les serveurs esclaves. Pour ajouter des serveurs esclaves, vous pourrez relancer [ypinit\(8\)](#) plus tard, en utilisant l'option `-u`. Configurer au moins un serveur esclave pour chaque domaine est utile pour éviter les interruptions de service, le serveur maître pourrait s'arrêter ou perdre sa connectivité réseau, en particulier les processus clients essaieront d'accéder aux blocs de tables YP indéfiniment tant qu'ils ne recevront pas les informations demandées. Donc, les interruptions de service YP rendront typiquement les hôtes clients complètement inutilisables tant que le service YP n'est pas de retour.

3. Il faut décider où stocker les fichiers sources pour générer vos tables YP initiales. Garder distincte la configuration du serveur de la configuration du service aide à contrôler quelle information sera donnée de celle qui ne le sera pas, donc le répertoire par défaut `/etc` n'est pas souvent le meilleur choix.

Le seul inconvénient causé par le changement de répertoire source est que vous ne pourrez plus ajouter, supprimer ou modifier des utilisateurs et des groupes dans le domaine YP en utilisant des utilitaires comme [user\(8\)](#) et [group\(8\)](#). A la place, vous devrez éditer les fichiers de configuration avec un éditeur de texte.

Pour définir le répertoire source, éditez le fichier `/var/yp/`domainname`/Makefile.y` et modifiez la variable `DIR`, par exemple :

```
DIR=/etc/yp/src/puffynet
```

4. Considérez la possibilité de personnalisation d'autres variables dans `/var/yp/`domainname`/Makefile`. Regardez [Makefile.y\(8\)](#) pour plus de détails.

Par exemple, même dans le cas où vous utilisez le répertoire source par défaut `/etc`, vous n'avez généralement pas besoin de tous les comptes et groupes existants sur le serveur pour tous les hôtes clients. En particulier, ne pas fournir de compte `root` et garder le hash du compte `root` confidentiel est souvent bénéfique pour la sécurité. Étudiez les valeurs des variables `MINUID`, `MAXUID`, `MINGID` et `MAXGID` et ajustez les selon vos besoins.

Si tous vos clients YP utilisent OpenBSD ou FreeBSD, excluez les mots de passe chiffrés de la table `passwd` en configurant `UNSECURE=""` dans `/var/yp/`domainname`/Makefile.y`.

La pratique courante d'éditer le fichier modèle `/var/yp/Makefile.y` n'est plus recommandé. Les changements dans ce fichier affectent tous les domaines initialisés après le changement, mais n'affecte pas les domaines initialisés avant le changement, il s'agit donc d'erreurs de toute façon :

vous risquez que les changements attendus ne soient pas pris en compte, et vous risquez de les oublier et qu'ils affectent d'autres domaines plus tard alors que cela n'était pas voulu.

5. Créez le répertoire source et le remplir avec les fichiers de configuration dont vous avez besoin. Regardez [Makefile.ypp\(8\)](#) pour savoir quel table YP a besoin de quel fichier source. Pour le format de fichier de configuration individuel, consultez [passwd\(5\)](#), [group\(5\)](#), [hosts\(5\)](#) voir plus, et regardez les exemple dans `/etc`.
6. Créez la version initiale de votre table YP en utilisant les commandes

```
cd /var/yp
make
```

Ne vous inquiétez pas des messages d'erreurs de [yppush\(8\)](#) maintenant. Le serveur YP ne fonctionne pas encore.

7. YP utilise les [rpc\(3\)](#) (remote procedure calls) pour communiquer avec ces clients, il est donc nécessaire d'activer [portmap\(8\)](#). Pour le faire, éditez [rc.conf.local\(8\)](#) et configurez `portmap=YES`. Cela démarrera le portmapper au prochain redémarrage. Vous pouvez aussi éviter de redémarrer en le démarrant manuellement :

```
echo "portmap=YES" >> /etc/rc.conf.local
portmap
```

8. Pensez à utiliser soit le [securenet\(5\)](#) ou les fonctionnalités de sécurité de votre serveur daemon YP dans [ypserv.acl\(5\)](#). Mais sachez que les deux ne fournissent seulement que du contrôle d'accès IP. Ainsi, ils aident aussi longtemps que des agresseurs potentiels n'ont ni l'accès physique au matériel du réseau transportant des segments de votre trafic YP, ni l'accès root de toute machine connectée sur ces segments réseaux.
9. Finalement démarrez le daemon serveur YP :

```
ypserv
```

Il démarrera automatiquement à chaque redémarrage aussi longtemps que le répertoire `/var/yp/`domainname`` continuera d'exister.

10. Pour tester le nouveau serveur, faite le devenir son propre client en suivant les instructions de la première partie de la section suivante. Dans le cas ou vous ne voulez pas que le serveur utilise ces propres tables, vous pouvez désactiver la partie cliente après le test avec les commandes suivantes :

```
pkill ypbind
rm -rf /var/yp/binding
```

11. Si vous désirez permettre aux utilisateurs de changer leurs mots de passe des machines clientes, vous devez activer [yppasswdd\(8\)](#) :

```
echo 'yppasswdd_flags="-d /etc/yp/src/puffynet"' >> /etc/
rc.conf.local
rpc.yppasswdd
```

Dans le cas où vous utilisez le répertoire source par défaut dans `/etc`, utilisez juste `yppasswdd_flags=""`.

- Rappelez-vous que chaque fois que vous changez un fichier source d'une table YP, vous devez régénérer vos tables YP.

```
cd /var/yp
make
```

Cela va mettre à jour tous les fichiers de bases de données dans `/var/yp/`domainname``, avec une exception : le fichier `ypservers.db`, contenant tous les serveurs YP maîtres et esclaves associés avec le domaine, qui est créé directement avec `ypinit -m` et modifié exclusivement par `ypinit -u`. Dans le cas où vous l'avez supprimé accidentellement, exécutez `ypinit -u` pour le recréer de zéro.

10.19.4 - Configurer un client YP

Configurer un client YP comprend deux parties distinctes. En premier, vous devez exécuter le daemon YP client, connecter votre hôte client au serveur YP. Finaliser les points suivants vous permettra de récupérer les données du serveur YP, mais ces données ne seront pas encore utilisables par le système :

- Comme sur le serveur, vous devez configurer le nom de domaine et activer le portmapper:

```
echo "puffynet" > /etc/defaultdomain
domainname `cat /etc/defaultdomain`
echo "portmap=YES" >> /etc/rc.conf.local
portmap
```

- Il est recommandé de fournir une liste de serveurs YP dans le fichier de configuration `/etc/yp/`domainname``. Sinon le daemon client YP utilisera des broadcasts réseaux pour trouver les serveurs YP pour son domaine. Spécifier de manière explicite les serveurs est plus robuste et aussi marginalement moins ouvert aux attaques. Si vous n'avez pas configuré de serveurs esclaves, mettez juste le nom du serveur maître dans `/etc/yp/`domainname``.
- Le daemon client YP est appelé [ypbind\(8\)](#). Le démarrer manuellement créera le répertoire `/var/yp/binding`, comme cela il sera automatiquement redémarré au démarrage.

```
ypbind
```

- Si tout se passe bien vous devez être capable de faire une requête sur le serveur YP en utilisant



www@openbsd.org

\$OpenBSD: faq10.html,v 1.74 2010/05/23 16:20:39 ajacoutot Exp \$



[\[Index de La FAQ\]](#) [\[Section 10 - Gestion du Système\]](#) [\[Section 12 - Questions Spécifiques Aux Plates-Formes Et Au Matériel\]](#)

11 - Le système X Window

Table des matières

- [11.1 - Introduction à X](#)
 - [11.2 - Configurer X](#)
 - [11.3 - Configurer X sur amd64 et i386](#)
 - [11.4 - Démarrer X](#)
 - [11.5 - Personnaliser X](#)
-

11.1 - Introduction à X

Le système X Window (parfois dénommé simplement "X" ou, incorrectement "X Windows") est l'environnement qui fournit les services graphiques à OpenBSD et les autres systèmes de type Unix. Cependant, seul, X n'apporte que très peu de fonctionnalités : il faut également avoir un gestionnaire de fenêtres afin d'offrir une interface utilisateur. La plupart de ce que l'on percevra de X viendra du gestionnaire de fenêtres et non de X lui-même. OpenBSD fournit une version libre du gestionnaire de fenêtres [fvwm\(1\)](#), bien que vous puissiez également utiliser un des autres gestionnaires présents dans les [paquetages](#). Effectuez une recherche avec le mot-clef "window manager" afin d'obtenir une liste des gestionnaires de fenêtres disponibles.

X est considéré comme un protocole fonctionnant sur le mode "client-server" mais cette terminologie peut parfois prêter à confusion. L'ordinateur avec l'affichage graphique à l'écran est le "serveur X". L'application qui définit ce que le serveur X doit afficher à l'écran est le "client X" même s'il s'agit d'une machine bien plus puissante dans un centre d'hébergement. Ce modèle peut être utilisé afin d'avoir des applications gourmandes en ressources processeur (clients X) tournant sur une machine très puissante mais utilisant un serveur X tournant sur une petite machine de faible puissance sur votre bureau pour afficher leurs interfaces.

Il est possible de faire tourner des clients X sur un système sans support graphique. Ainsi, on pourrait avoir une application (le client X) tournant sur un mvme88k et affichant son interface sur un alpha (le serveur X). Puisque X est un protocole bien défini et multiplate-forme il est même possible d'avoir une application X tournant (par exemple) sur une machine Solaris et utiliser une machine OpenBSD pour l'afficher.

Le client et le serveur peuvent également tourner sur la même machine et pour la majorité de cette documentation, c'est ce que nous supposons.

11.1.1 - Quelle puissance d'ordinateur est nécessaire pour faire tourner X ?

X en lui-même est un programme très lourd et un ordinateur rapide sera idéal si vous le lancez et l'arrêtez régulièrement. Cependant, une fois démarré, il reste très réactif même sur une machine modeste. Afin d'avoir une meilleure réactivité sur certaines plates-formes même si vous ne travaillez qu'avec du texte, il vous sera nécessaire de faire tourner X. Ces plates-formes, telles que [sparc](#) et [sparc64](#) ont été prévues pour fonctionner avec une interface graphique, et le mode console est très lent.

Ceci dit, l'utilité de faire tourner X est généralement d'utiliser des applications X. Certaines de ces applications sont très peu gourmandes, d'autres vous sembleront utiliser tout le processeur et la RAM que vous leur octroierez. Evidemment, certains utilisateurs n'utilisent X qu'afin de bénéficier de nombreux [xterm\(1\)](#)s, ce qui reste possible même sur du matériel modeste.

11.1.2 - Puis-je bénéficier d'une interface graphique sans X ?

Si l'on considère que les [graphiques ASCII](#) ne sont pas satisfaisants, cela nécessiterait un pilote framebuffer pour console. Quelques systèmes d'exploitation offrent un tel pilote ce qui n'est pas le cas pour OpenBSD et il n'y a pas vraiment d'intérêt pour cela parmi les développeurs.

11.2 - Configurer X

La configuration de X varie considérablement d'une plate-forme à l'autre. Dans tous les cas, il y aura des instructions et des informations propres à la plate-forme dans le fichier `/usr/X11R6/README` du système sur lequel il est installé.

Plusieurs plates-formes requièrent le pilote d'aperture [xf86\(4\)](#) qui offre un accès à la mémoire et aux ports d'entrée/sortie de la carte VGA ainsi qu'aux registres de configuration PCI nécessaires. Ce pilote doit être activé avant d'être utilisé soit en répondant "yes" à la question suivante posée durant l'installation du système :

```
Do you expect to run the X window System [no]
```


soit en changeant la valeur de `machdep.allowaperture` en une valeur non nulle appropriée à votre plate-forme dans le fichier `/etc/sysctl.conf` et en redémarrant la machine (pour raisons de sécurité, cette valeur `sysctl` ne peut pas être changée après le lancement de la machine). Il existe des conséquences sur la sécurité, ne le faites pas si vous n'en avez pas vraiment besoin.

11.2.1 - alpha

[/usr/X11R6/README](#) pour alpha.

Mettez `machdep.allowaperture=1` dans `/etc/sysctl.conf`.

Les cartes TGA ainsi que certaines VGA sont supportées. Aucune configuration supplémentaire ne devrait être nécessaire.

11.2.2 - amd64

[/usr/X11R6/README](#) pour amd64.

Mettez `machdep.allowaperture=2` dans `/etc/sysctl.conf`.

Dans la plupart des cas, X sous amd64 se configure automatiquement et aucune manipulation supplémentaire ne devrait être nécessaire. Si une configuration spécifique est nécessaire, utilisez la procédure [X -configure](#) comme expliquée [plus bas](#).

11.2.3 - armish

Pas de serveur X, seulement des clients.

11.2.4 - hp300

[/usr/X11R6/README](#) pour hp300.

11.2.5 - hppa

Pas de serveur X, seulement des clients.

11.2.6 - i386

[/usr/X11R6/README](#) pour i386.

Mettez `machdep.allowaperture=2` dans `/etc/sysctl.conf`.

Du fait du choix énorme de cartes vidéos, souris, claviers et autres matériels pour cette plate-forme, configurer X sous un système i386 peut s'avérer excitant. Suffisamment excitant d'ailleurs pour qu'une [section séparée](#) soit proposée.

Fort heureusement, les choses ne sont pas toujours aussi noires qu'elles le semblent ; dans la plupart des cas, X "fonctionne tout simplement" en tapant "startx". Dans ce cas, votre matériel sera détecté et interrogé sur ses capacités ce qui permettra à X de très bien fonctionner.

11.2.7 - landisk

Pas de serveur X, seulement les clients.

11.2.8 - loongson

Pas besoin de configuration.

11.2.9 - luna88k

Pas de serveur X, seulement les clients.

11.2.10 - macppc

[/usr/X11R6/README](#) pour macppc.

Mettez `machdep.allowaperture=2` dans `/etc/sysctl.conf`.

Les systèmes PPC Macintosh peuvent fonctionner sous deux modes différents : "accéléré" et "framebuffer" (non accéléré).

En mode "framebuffer", le système tournera avec 8 bits par pixel et la résolution vidéo sera contrôlée par l'environnement Macintosh, vous devrez donc garder une petite partition sous MacOS sur votre disque afin de pouvoir changer la configuration. Ce mode à l'avantage de "fonctionner tout simplement" mais il peut apparaître comme statique et frustrant (par exemple, changer la résolution peut demander un redémarrage sous MacOS).

Si votre Macintosh possède un système vidéo basé sur de l'ATI il peut tourner en mode X accéléré ce qui offre de meilleures performances et plus de contrôle sous l'environnement OpenBSD. Les cartes vidéo NVIDIA présentes dans certains systèmes macppc devraient également fonctionner dans la plupart des

cas. Le fichier README contient des détails sur la configuration du pilote accéléré, basez-vous sur l'exemple qu'il fournit.

Bien que le fichier README contienne des détails pour utiliser une souris Apple standard à un bouton sous X, à moins d'utiliser un portable, il vous est vivement recommandé d'acheter une souris tierce plus moderne.

iMacs CRT et X

Les iMacs ont très rarement un CRT (de nos jours), en ce sens qu'il a un balayage de fréquence horizontale fixe. La tentative d'utilisation d'un balayage horizontal de vitesse en dehors d'une fourchette très étroite fera que le moniteur ne s'allume pas. Les lignes suivantes, ajoutées à xorg.conf dans la Section "Monitor" semble faire que beaucoup d'iMacs CRT dans la plage 333MHz à 500MHz (et probablement plus !) fonctionnent bien :

```

    HorizSync      60.0 - 60.0
    VertRefresh    43.0 - 117.0

```

Vous pouvez restreindre la limite inférieure de `VertRefresh` pour des valeurs que vous êtes plus susceptibles de trouver acceptables, par exemple 70.

11.2.11 - mvme68k

Pas de serveur X, seulement les clients.

11.2.12 - mvme88k

Pas de serveur X, seulement les clients.

11.2.13 - sgi

X fonctionne sur les systèmes framebuffer O2 en mode non accéléré.

11.2.14 - sparc

</usr/X11R6/README> pour sparc.

Avec un seul framebuffer supporté il n'y a pas besoin de configuration. Si vous souhaitez utiliser une configuration multi-affichage, lisez le README ci-dessous pour plus de détails.

La résolution est contrôlée par le firmware avant de démarrer OpenBSD.

11.2.15 - sparc64

[/usr/X11R6/README](#) pour sparc64.

La plupart des configurations simple "fonctionne" normalement. Si la vôtre ne fonctionne pas ou si vous souhaitez obtenir de la fantaisie, consultez le fichier README ci-dessus.

11.2.16 - vax

Lisez [/usr/X11R6/README](#) pour vax.

Le serveur X ne fonctionne pour l'instant que sur les modèles VAXstation 4000 en utilisant soit le framebuffer [lcg\(4\)](#) soit le framebuffer [lcspix\(4\)](#).

11.2.17 - zaurus

[/usr/X11R6/README](#) pour zaurus.

Pas de configuration nécessaire, X "fonctionne tout simplement".

11.3 - Configurer X sur amd64 et i386

La variété de matériel présent sur ces plates-formes rend la configuration de X assez "laborieuse".

11.3.1 - Configuration de X.Org

X.Org a fait de gros progrès pour permettre à leurs serveurs de "fonctionner tout simplement". Dans la plupart des cas tout fonctionne sans fichier `/etc/X11/xorg.conf`. Mais pas toujours et parfois vous aurez besoin de configurer quelque chose qui ne fonctionne pas.

Un bon endroit pour démarrer (et quelquefois le seul bon endroit !) est d'exécuter [Xorg\(1\)](#) lancé avec le mode "X -configure" chargera tous les pilotes vidéo disponibles, détectera le matériel et se basant sur ce qu'il trouve, écrira un fichier `xorg.conf` qui pourra ou non fonctionner, mais même s'il ne fonctionne pas il représentera un bon point de départ à une configuration manuelle.

Une autre façon de configurer X mais qui demande du temps est d'utiliser votre moteur de recherche favori afin de trouver quelqu'un d'autre qui aurait rencontré et résolu votre problème. Il faut remarquer que les fichiers `xorg.conf` des autres OS Unix-like fourniront souvent une aide très utile dans la

manière de faire fonctionner X sur un matériel similaire. Bien que cette méthode ne soit pas mauvaise, nous ne nous y attarderons pas plus.

11.3.2 - Notre machine d'exemple

Pour une démonstration de configuration X, nous utiliserons un vieux système équipé d'un Celeron 400MHz et d'un port AGP. La carte vidéo sera une ancienne carte AGP visible dans le dmesg à la ligne :

```
vga1 at pci1 dev 0 function 0 "3DFX Interactive Banshee"
rev 0x03
```

Il s'agit d'une ancienne carte haut-de-gamme qui n'est à présent plus vraiment supportée par les nouvelles versions des systèmes d'exploitation "principaux". L'écran relié au système est un moniteur CRT "Sony Multiscan G400 19" et il serait plaisant de pouvoir afficher une résolution de 1280x1024 sur celui-ci avec une fréquence de rafraîchissement décente et des couleurs 24 bit.

En premier lieu, après avoir installé OpenBSD avec X (et être certain que le pilote d'aperture est activé dans le noyau), voyons ce que la détection et la configuration automatique de X.Org nous propose, après tout nous pourrions être chanceux. Nous allons nous identifier puis utiliserons donc la commande [startx \(1\)](#). L'écran devient blanc quelques secondes puis nous obtenons le fond d'écran en "échiquier" de X, le curseur X et enfin une fenêtre xterm.

Ca marche !

Plus ou moins. Bien que le système soit pleinement fonctionnel, il ne paraît pas avoir détecté une quelconque des fonctionnalités de l'écran et tourne à très basse résolution (640x480). Nous espérons pouvoir faire mieux que cela. Bien mieux en fait. Cependant, cela signifie que nous aurons besoin de créer notre propre fichier de configuration xorg.conf.

Servons-nous de la commande "X -configure" afin de créer un fichier xorg.conf de base. Vous devrez lancer la commande suivante en tant qu'utilisateur root :

```
# X -configure
[...]
Your xorg.conf file is /root/xorg.conf.new
```

```
To test the server, run 'X -config /root/xorg.conf.new'
```

Au fait, ce message est sérieux ; utilisez le chemin complet vers votre fichier xorg.conf.new même s'il repose dans votre répertoire actuel par défaut. Sinon, [X\(7\)](#) ne trouvera pas le fichier et il utilisera la configuration par défaut qui peut n'avoir aucun rapport avec le fichier sur lequel vous affiner

actuellement la configuration. Cela permettra de vous éviter de déboguer le pourquoi cela ne fonctionne pas. Faites-nous confiance là-dessus.

Donc, faisons comme il dit et voyons ce que nous obtenons :

```
# X -config /root/xorg.conf.new
```

A présent, tout ce que nous avons est un écran noir. Les choses avaient pourtant si bien commencées...

Le moment est peut-être venu de vous parler des différentes manières de sortir de X lorsqu'il démarre de cette façon. Par ordre de préférence :

- **CTRL-ALT-Backspace** : cette combinaison devrait à priori terminer X ainsi que toutes les applications X immédiatement. Bien entendu, durant la procédure de configuration vous n'avez aucune application qui tourne et donc, ce n'est pas un problème (et en fait, actuellement, il s'agit de votre *meilleur* moyen de sortir de X).
- Connectez vous en **SSH vers votre machine**, et tapez "[pkill](#) Xorg", ce qui devrait terminer X et vous ramener sur une console utilisable.
- Connectez vous en **SSH vers votre machine**, et redémarrez-la.
- **Reset du bouton d'alimentation**. Parfois, les choses se passent vraiment mal. Oui, il est généralement important de configurer X avant de charger des applications critiques dans le système. Parfois une mauvaise configuration de X fera planter totalement la machine au point que seul un reset matériel résoudra le problème.

Heureusement pour nous, CTRL-ALT-Backspace fonctionne bien et nous voici revenu à l'invite de commandes. Bien, à présent nous devons regarder si nous pouvons comprendre ce qui s'est mal passé. Tout d'abord, nous devrions jeter un oeil sur ce à quoi X s'imagine et qui est enregistré dans le fichier `/var/log/Xorg.0.log`. Dans ce cas, il semble que X pense que tout tourne bien et il n'y a pas d'erreurs flagrantes dans le journal d'événements (les lignes commençant par "`(EE)`" sont des erreurs).

Maintenant, il est utile de connaître votre matériel. Brancher ce système sur un moniteur différent alors que l'écran noir est toujours présent affiche le message "Sync. Out of Range" sur l'écran. Apparemment donc, la configuration que X nous a fourni ne fonctionnera pas sur ce moniteur et peut-être même sur AUCUN si le mode sélectionné ne fonctionne pas avec cette carte particulière (souvenez-vous que X détecte la puce sur la carte et ce qu'elle est potentiellement capable de faire, pas la façon dont le fabricant a intégré tous les éléments). Différents moniteurs réagiront différemment lorsque le rafraîchissement est mauvais, certains tenteront tout de même d'afficher ce qu'ils peuvent, d'autres passeront en mode de conservation d'énergie alors que d'autres émettront d'horribles bruits ou afficheront des messages informatifs à l'écran. Ce moniteur ne semble rien faire de tout cela. On notera de ne PAS utiliser cet écran pour une configuration initiale de X dans le futur.

En jetant un oeil au fichier `xorg.conf.new` l'on peut voir ceci :

```

Section "Monitor"
    #DisplaySize      370   270       # mm
    Identifier        "Monitor0"
    VendorName        "SNY"
    ModelName         "SONY CPD-G400"
    ### Comment all HorizSync and VertSync values to use DDC:
    HorizSync         30.0 - 107.0
    VertRefresh       48.0 - 120.0
    Option             "DPMS"
EndSection

```

Pour le test, essayons d'utiliser DDC ("Data Display Channel", une façon que possède le moniteur de renseigner l'ordinateur et la carte vidéo sur ses capacités) et voyons ce qu'il se passe. Cette fois, nous obtenons le maillage X et un curseur de souris mobile ce qui est tout ce à quoi nous pouvons nous attendre lorsque nous démarrons X de cette façon (nous quittons X en utilisant CTRL-ALT-Backspace comme expliqué plus haut). Cela reste (encore) dans une très basse résolution, mais cela fonctionne ce qui signifie certainement que la résolution et la fréquence de rafraîchissement posent problème. Nous restaurons "HorizSync" et "VertRefresh" à leur valeur précédente car nous avons vérifié les spécifications de ce moniteur en recherchant sur Internet.

Essayons de forcer Xorg à une résolution spécifique et voyons si nous avons de la chance. Dans la partie Section "Screen" du fichier xorg.conf, nous ajouterons deux lignes. Celles-ci sont montrées en gras :

```

Section "Screen"
    Identifier "Screen0"
    Device     "Card0"
    Monitor    "Monitor0"
DefaultDepth 24
    SubSection "Display"
        Viewport 0 0
        Depth    1
    EndSubSection
    SubSection "Display"
        Viewport 0 0
        Depth    4
    EndSubSection
    SubSection "Display"
        Viewport 0 0
        Depth    8
    EndSubSection
    SubSection "Display"

```

```

        Viewport    0 0
        Depth      15
    EndSubSection
    SubSection "Display"
        Viewport    0 0
        Depth      16
    EndSubSection
    SubSection "Display"
        Viewport    0 0
        Depth      24
        Modes      "1280x1024"
    EndSubSection
EndSection

```

Ces deux changements indiquent que nous souhaitons faire tourner X à une profondeur d'affichage de 24 bit et, pour cette profondeur, bénéficier d'une résolution de 1280x1024. Puisqu'aucune autre résolution n'est précisée pour le mode "Depth 24", le système sera forcé de l'utiliser.

Nous testons nos changements et... **SUCCES !** Nous obtenons ce qui semble être un très bel affichage en haute résolution. Notez bien que TOUT ce à quoi vous devez vous attendre est un maillage X (appelé "[rootweave](#)", il est parfait pour voir la qualité de votre moniteur et pour calibrer les écrans LCD) et un curseur mobile. A ce point, rien ne se doit d'être fonctionnel.

A présent, installons ce fichier `xorg.conf` afin de voir comment cela fonctionne avec un X fonctionnel qui tourne.

```
# cp xorg.conf.new /etc/X11/xorg.conf
```

Nous pouvons maintenant utiliser la commande normale `startx(1)`. Ca marche !

Il serait probablement bon de vérifier que nous sommes bien à la résolution et à la profondeur que nous désirons et que nous tournons à une fréquence de rafraîchissement correcte. Nous pouvons le vérifier avec les commandes [xrandr\(1\)](#) et [xdpyinfo\(1\)](#). Entre autres, `xdpyinfo(1)` nous dit :

```

[... ]
screen #0:
  print screen:      no
  dimensions:       1280x1024 pixels (433x347 millimeters)
  resolution:       75x75 dots per inch
  depths (7):       24, 1, 4, 8, 15, 16, 32
  root window id:   0x44
  depth of root window: 24 planes

```


[. . .]

Donc oui, nous tournons bien à 1280x1024 avec une profondeur de 24 planes (bits).

xrandr(4) nous dit :

```

SZ:      Pixels      Physical      Refresh
*0      1280 x 1024    ( 433mm x 347mm ) *85   75   60
 1      1280 x  960    ( 433mm x 347mm )  85   60
[ . . . ]

```

ce qui signifie que nous tournons avec une fréquence de rafraîchissement de 85Hz, ce qui devrait être confortable pour la plupart des utilisateurs.

11.3.3 - Et si ce n'est pas si "facile" ?

Parfois, les choses ne vont pas de soi. Voici quelques conseils.

- Référez-vous à la page de manuel du serveur X que vous utilisez. Dans notre exemple, nous pouvons voir dans `/var/log/Xorg.0.log` que X utilise le pilote TDFX et donc notre manuel sera [tdfx\(4\)](#). Vous trouverez souvent des astuces, limitations et options pour configurer votre carte vidéo. Celles-ci varient selon les pilotes donc n'oubliez pas que vous n'avez pas besoin de lire la page de manuel de votre pilote actuel sous prétexte d'en avoir lu une différente avant.
- Testez des écrans différents. Comme nous l'avons vu dans notre exemple précédent, différents moniteurs donnent souvent des indices différents sur les problèmes que vous pouvez rencontrer.
- Essayez le pilote X [vesa\(4\)](#). Pour des raisons de performance, il s'agit de votre "dernier choix", mais il fonctionne avec presque toutes les cartes vidéo même celles dont aucun des "meilleurs" pilotes pour serveur X ne fonctionne.
- Utilisez un autre matériel. Si vous avez le choix du type de carte vidéo à utiliser, essayez-en d'autres.

11.4 - Démarrer X

Il existe deux méthodes génériques pour démarrer X :

11.4.1 - A la demande :

Identifiez-vous sous une console en tant qu'utilisateur puis lancez [startx\(1\)](#).

11.4.2 - Démarrer directement sous X :

Cela se fait en utilisant le gestionnaire de connexion graphique ("X Display Manager") [xdm\(1\)](#). Normalement, xdm(1) est démarré en tant que root par [rc](#) et présente une invite d'identification. En cas d'identification correcte, il démarre une session X pour cet utilisateur. Si, ou lorsque cette session X doit se terminer (cela inclut l'enchaînement CTRL-ALT-Backspace), xdm(1) réapparaîtra avec son invite d'identification. C'est pour cette raison que vous ne devez PAS démarrer xdm(1) à partir du fichier `/etc/rc.conf.local` avant d'être certain d'avoir vérifié que X fonctionne comme vous le souhaitez sinon votre machine risque de devenir difficile à gérer ! (dans le pire des cas : démarrez en mode "single user" comme si vous aviez [perdu votre mot de passe](#) et commentez la ligne `xdm_flags` de votre fichier `/etc/rc.conf.local`).

Sur certaines plates-formes, vous aurez besoin de désactiver la console [getty\(8\)](#) pour utiliser xdm(1).

11.5 - Personnaliser X

11.5.1 - Introduction

L'environnement X par défaut de OpenBSD est complètement fonctionnel, mais vous désirez peut-être le personnaliser. Vous pouvez avoir envie de changer l'image ou la couleur du fond d'écran, ou vous pouvez avoir envie de changer de Gestionnaire de Fenêtres (le programme qui définit le plus votre environnement X) ou changer les applications qui sont démarrées quand X démarre.

Le gestionnaire de fenêtres par défaut dans OpenBSD est [fvwm\(1\)](#). Fvwm est un bon gestionnaire de fenêtres généraliste, mais c'est difficilement votre seul choix; ce n'est même pas le seul gestionnaire de fenêtres fourni avec OpenBSD (voir [cwm\(1\)](#) et [twm\(1\)](#)). Un grand nombre de gestionnaires de fenêtres est également disponibles via les [paquetages](#).

De manière similaire au [script de démarrage système](#), X possède un processus qu'il exécute pour mettre en place l'environnement de l'utilisateur. Plus concrètement, il possède plus d'un processus qui sont utilisés en fonction de la façon dont vous [démarez X](#). La compréhension de la façon dont X démarre vous aidera à comprendre comment personnaliser votre environnement de travail comme vous le désirez.

Il faut remarquer que vous pouvez personnaliser l'environnement pour tous les utilisateurs ou par utilisateur. Il est préférable de faire des changements au niveau utilisateur plutôt que de changer les options du système par défaut, comme les scripts utilisateurs sont stockés dans le repertoire home de l'utilisateur, vous aurez moins de fichiers à fusionner quand vous mettrez à jour votre version d'OpenBSD. Les options par défaut générales du système se trouvent dans `/etc/X11` et sont chargées initialement de [xetcXX.tgz](#), qui n'est pas rechargé par une [procédure de mise à jour suggérée](#), donc si

vous faites des changements sur le système général, ils persisteront, mais vous devrez fusionner ces changements dans les nouvelles versions de ces fichiers.

11.5.2 - démarrage de startx(1)

startx(1) recherche le fichier `.xinitrc` dans le repertoire home de l'utilisateur. `.xinitrc` est habituellement un script shell qui peut démarrer autant de programme "client" X (applications qui utilisent X) que désiré. Quand le script se termine, le serveur X s'arrête aussi. Généralement, la plupart des programmes exécutés par ce script doivent être exécutés en tache de fond, mais le dernier doit fonctionner en avant-plan (typiquement le gestionnaire de fenêtres); quand il se termine, le script se terminera et X s'arrêtera.

Dans le cas le plus simple, cela peut-être aussi petit que le nom du gestionnaire de fenêtres que vous invoquez :

```
cwm
```

Ou vous pouvez y mettre un peu plus de fantaisie :

```
xconsole -geometry -0+0 -fn 5x7 &
oclock -geometry 75x75-0-0 &
xsetroot -solid grey &
cwm
```

Cela démarrera la [xconsole\(1\)](#) qui fournit une copie de tout le texte que le noyau peut avoir envoyé à la console (qui est maintenant couverte par l'écran graphique), une horloge analogique, [oclock\(1\)](#), et configurer le fond d'écran avec du gris via [xsetroot\(1\)](#), tout cela avant d'invoquer le gestionnaire de fenêtres [cwm\(1\)](#). Remarquez que seulement le gestionnaire de fenêtres n'est pas "mis en tache de fond" avec un caractère "&". Cela signifie que X restera exécuté jusqu'a ce que `cwm(1)` se termine.

Si le repertoire home de l'utilisateur ne possède pas de fichier `.xinitrc`, le fichier système `/etc/X11/xinit/xinitrc` est utilisé. Ce fichier peut vous fournir d'autres idées pour votre script `.xinitrc`.

11.5.3 - démarrage de xdm(1)

[xdm\(1\)](#) est habituellement démarré par les [scripts de démarrage système](#), mais pour des raisons de test (recommandé tant que vous n'avez pas votre config X correcte !), il peut-être exécuté sous root.

`xdm(1)` possède beaucoup d'autres fonctionnalités dont nous ne parlerons pas ici, mais pour nos besoins, `xdm` affichera l'utilisateur avec un écran de connexion, récupérera et vérifiera le nom de l'utilisateur et le mot de passe, et fournira à l'utilisateur son environnement X. Quand X s'arrête, de façon délibérée ou

accidentelle, xdm le redémarrera de lui-même. C'est pour cette raison que vous devez configurer correctement X avant d'utiliser xdm(1) et avant d'avoir xdm(1) démarrer automatiquement au démarrage de la machine, autrement, vous aurez des difficultés à prendre le contrôle de la machine.

Quand xdm(1) démarre, il exécute `/etc/X11/xdm/Xsession`, qui vérifiera si l'utilisateur possède un fichier `.xsession` dans son répertoire home. Mais, si vous désirez changer votre gestionnaire de fenêtres par défaut, invoquez le simplement (et peut-être d'autres choses) dans `.xsession`. Encore un fois, tous les programmes que vous désirez démarrer avec X (par exemple, peut-être que trois `xterm(1)`s peuvent être placés ici, mais ils doivent tous être en tâche de fond excepté votre gestionnaire de fenêtre, qui encore une fois, quand il se terminera, terminera votre session X. Dans ce cas, xdm(1) redémarrera X et retournera à l'écran de connexion.

11.5.4 - Essayer un nouveau gestionnaire de fenêtre

Vous pouvez invoquer un gestionnaire de fenêtre particulier quand vous démarrer X sans altérer les paramètres par défaut comme ceci :

```
$ startx /usr/local/bin/fluxbox
```

Plusieurs gestionnaires de fenêtres (incluant `cwm(1)` et `fvwm(1)`) offrent la possibilité de changer de gestionnaire de fenêtres à la volée, sans redémarrer X ou vos applications. Votre nouveau gestionnaire de fenêtres remplace le précédent; quitter le nouveau gestionnaire de fenêtres terminera X, il ne retournera pas à votre précédent gestionnaire de fenêtres. `fvwm(1)` vous permet de démarrer un gestionnaire de fenêtres différent en cliquant à gauche sur le fond d'écran ("root window"), choisir "(Re) Start", puis sélectionnez votre gestionnaire de fenêtre préféré (cependant, il faut remarquer que vous devez ajouter votre gestionnaire de fenêtres alternatif à votre fichier `.fvwmrc` (celui par défaut pour le système est `/usr/X11R6/lib/X11/fvwm/.fvwmrc`)). `cwm(1)` vous permet d'invoquer un autre gestionnaire de fenêtre en tapant `Ctrl-Alt-w` puis en sélectionnant le gestionnaire dans lequel vous voulez aller.

Des que vous avez trouvé un gestionnaire de fenêtre que vous aimez, vous pouvez le configurer comme le programme final exécuté par vos scripts de démarrage comme décrit ci-dessus.

[\[Index de La FAQ\]](#) [\[Section 10 - Gestion du Système\]](#) [\[Section 12 - Questions Spécifiques Aux Plates-Formes Et Au Matériel\]](#)



www@openbsd.org

\$OpenBSD: faq11.html,v 1.50 2010/06/19 07:38:08 ajacoutot Exp \$

12 - Questions Spécifiques Aux Plates-Formes Et Au Matériel

Table des matières

- [12.1 - Remarques Générales sur le Matériel](#)
 - [12.1.1 - PCI](#)
 - [12.1.2 - ISA](#)
 - [12.1.3 - Un périphérique est "reconnu" mais il est marqué comme "not configured" dans le dmesg](#)
 - [12.1.4 - J'ai une carte listée dans le matériel "supporté" mais elle ne fonctionne pas !](#)
 - [12.1.5 - Est-ce que les WinModems sont supportés ?](#)
 - [12.1.6 - Qu'est-il arrivé au support RAID des cartes Adaptec \(aac\)?](#)
 - [12.1.7 - Ma carte ami\(4\) ne supporte qu'un seul disque logique !](#)
 - [12.1.8 - Comment est-ce que j'active ma carte accélératrice crypto ?](#)
- [12.2 - DEC Alpha](#)
- [12.3 - AMD 64](#)
 - [12.3.1 - Puis-je faire fonctionner OpenBSD/amd64 sous mon Intel P4 ?](#)
 - [12.3.2 - Puis-je faire fonctionner mon binaire i386 sous OpenBSD/amd64 ?](#)
 - [12.3.3 - Est-ce que c'est toujours mieux d'utiliser OpenBSD/amd64 sur les processeurs qui peuvent le supporter ?](#)
- [12.4 - ARM-based appliances](#)
- [12.5 - HP 9000 series 300, 400](#)
- [12.6 - HP Precision Architecture \(PA-RISC\)](#)
- [12.7 - i386](#)
 - [12.7.1 - Cartes Réseau ISA](#)
 - [12.7.2 - OpenBSD ne fonctionne pas sur mon système 80386/80386SX/80486SX](#)
 - [12.7.3 - Mon dmesg affiche plusieurs périphériques partageant la même interruption](#)
 - [12.7.4 - Mon clavier / ma souris n'arrête pas de se bloquer \(ou d'avoir un comportement complètement erratique\) !](#)

- [12.7.5 - Mon Soekris a de faible performance](#)
 - [12.7.6 - J'obtiens des erreurs "missing interrupt" sur mon périphérique CF](#)
 - [12.7.7 - Est-ce que ce nouveau CPU est supporté ?](#)
 - [12.8 - Landisk](#)
 - [12.9 - Luna88k](#)
 - [12.11 - MacPPC](#)
 - [12.11.1 - Pourquoi ma carte réseau bm\(4\) ne fonctionne pas ?](#)
 - [12.12 - MVME68k](#)
 - [12.13 - MVME88k](#)
 - [12.14 - SGI](#)
 - [12.15 - SPARC](#)
 - [12.16 - UltraSPARC](#)
 - [12.16.1 - Mon système UltraSPARC ne veut pas démarrer à partir de l'image sur disquette](#)
 - [12.16.2 - J'obtiens le message "partition extends past end of unit" dans disklabel](#)
 - [12.17 - DEC VAX](#)
 - [12.17.1 - Puis-je utiliser le simulateur VAX SIMH ?](#)
 - [12.18 - Sharp Zaurus](#)
 - [12.18.1 - Les périphériques USB ne fonctionnent pas correctement](#)
-

12.1 - Notes Générales sur le Matériel

12.1.1 - Périphériques PCI

- Les périphériques PCI sont pratiquement tous auto- configurés -- la machine et le système d'exploitation vont allouer des ressources aux cartes si nécessaire.
- Les interruptions peuvent être partagées sur le bus PCI. Non seulement ils le peuvent, mais le système va souvent être plus performant lorsque les IRQs sont partagés, et particulièrement sur les systèmes i386.
- Il existe plusieurs standard PCI différents. Occasionnellement, vous trouverez une carte de spécification PCI2.2 qui va juste fonctionner sur un système de spécification 2.1. De même, plusieurs cartes avec des ponts intégrés (telles que les cartes réseaux multi-port) ne fonctionneront pas correctement sur d'anciens systèmes.
- Le bus PCI supporte deux niveaux de signalisation, 3.3V et 5V. Les cartes fonctionnant avec la signalisation 3.3V ont une seconde "coupure" au niveau de leur connecteur PCI. La plupart des cartes PCI utilisent la signalisation 5V, utilisée par la plupart des ordinateurs. Les ordinateurs à carte unique Soekris (Net45x1 et Net4801) sont des machines communes qui ne supportent que la signalisation 3.3V.

12.1.2 - Périphériques ISA

- Les périphériques ISA ne peuvent partager de ressources et de manière générale, doivent être configurés manuellement en utilisant des paramètres qui ne rentrent pas en conflit avec d'autres périphériques système.
- Quelques périphériques ISA sont "Plug and Play" ([isapnp\(4\)](#)) -- si vous avez des problèmes avec de tels périphériques, vérifiez leur configuration dans votre [dmesg\(8\)](#), ISAPNP ne fonctionne pas toujours comme on le souhaite.
- De manière générale, si vous avez le choix évitez les cartes ISA au profit de cartes PCI. Les cartes ISA sont plus difficiles à configurer et ont un impact négatif plus grand sur les performances système.

12.1.3 - Un périphérique est "reconnu" mais il est marqué comme "not configured" dans le dmesg

Ceci veut clairement dire que votre périphérique n'est pas supporté par le noyau que vous utilisez. Vous ne serez donc pas en mesure de l'utiliser.

PCI et de nombreux autres types de périphériques offrent des informations d'identification afin que le système d'exploitation puisse les reconnaître correctement et activer s'il y a lieu le support de ces périphériques. Ajouter ces informations est chose facile, ajouter le support d'un périphérique ne l'est pas. Souvent. Voici une partie d'un dmesg avec deux exemples de périphériques "not configured" :

```
...
vendor "Intel", unknown product 0x5201 (class network
subclass
ethernet,
rev 0x03) at pci2 dev 9 function 1 not configured
...
"Intel EE Pro 100" rev 0x03 at pci2 dev 10 function 0 not
configured
...
```

Dans le premier exemple (une carte réseau), le code du constructeur a été identifié ainsi que le type générique de la carte. Cependant, le modèle exact de cette carte n'a pu être identifié. Le second exemple montre une autre carte réseau qu'un développeur a vu et a saisi dans le fichier d'identification utilisée pour identifier la carte. Cependant, les cartes ne seront pas fonctionnelles dans les deux cas vu qu'elles sont affichées comme "not configured", ce qui veut dire qu'aucun pilote ne leur a été affecté.

Que puis-je faire dans le cas d'un périphérique not configured ?

- Si le périphérique ou la carte que vous avez ne vous est pas utile, vous pouvez ignorer les périphériques "not configured" puisqu'ils ne causeront aucun dommage à votre système. Certains périphériques spéciaux ne sont pas configurés à bon escient afin que le BIOS du système puisse les gérer.
- Dans certains cas, c'est juste une variante d'un périphérique supporté. Si c'est le cas, il serait relativement facile pour un développeur d'ajouter le support de cette nouvelle carte. Dans d'autres cas, c'est peut-être une puce ou une implémentation qui n'est absolument pas supportée (telle que dans les exemples précédents). Dans ce cas, un nouveau pilote devra être écrit mais ce n'est pas toujours possible. Des fois, le périphérique n'est pas entièrement documenté. Vous êtes bien entendu invité à écrire un pilote pour le périphérique vous-même si vous en avez les compétences.
- Si vous utilisez un noyau d'installation, le périphérique n'est peut-être pas supporté par la méthode d'installation que vous avez utilisé mais supporté par un disque de démarrage différent. C'est chose commune pour des utilisateurs de certaines cartes SCSI populaires qui ont mal lu les notes de bas de page sur la page consacrée à la [plate-forme i386](#) et qui essaient de démarrer avec des disquettes de démarrage ne supportant pas leur carte SCSI au lieu d'utiliser la disquette adéquate.
- Si vous utilisez un noyau modifié, vous avez peut-être supprimé le support du périphérique dont vous avez besoin. De manière générale, la suppression de périphériques dans le noyau est une [mauvaise idée](#).
- Avant de faire un rapport sur un périphérique "not configured", assurez-vous d'abord que vous avez utilisé le dernier [snapshot](#) disponible pour voir si le support de ce périphérique n'a pas été récemment ajouté. Puis vérifiez les [archives des listes de diffusion](#) pour voir si le problème n'a pas été discuté. Rappelez-vous cependant que si vous utilisez une ancienne version d'OpenBSD, vous devriez mettre à jour votre système pour bénéficier des derniers pilotes.

12.1.4 - J'ai une carte listée dans le matériel "supporté" mais elle ne fonctionne pas !

Malheureusement, beaucoup de constructeurs utilisent les numéros de modèle pour indiquer leur position sur le marché, au lieu de la nature technique du produit. Pour cette raison, vous avez peut-être acheté un produit avec le même nom et du même modèle qu'un produit listé dans les [pages consacrées aux plates-formes](#), alors qu'en réalité c'est un produit entièrement différent qui ne fonctionne pas sous OpenBSD. Par exemple, plusieurs cartes réseau sans-fil sorties mises sur le marché il y a un certain temps utilisaient la puce Prism2 donc le pilote ([wi\(4\)](#)). Mais plus tard, lorsque des puces à bas prix sont devenues disponibles, plusieurs fabricants ont changé leur produit pour utiliser des puces pour lesquelles aucun pilote libre n'existait. En revanche, ils n'ont pas changé le nom de leur produit. Les cartes réseau sans-fil sont malheureusement loin d'être les seules dans ce cas.

12.1.5 - Est-ce que les WinModems sont supportés ?

Les WinModems sont des modems à bas prix qui s'appuient sur le processeur pour gérer le traitement du signal qui est normalement effectué au niveau matériel dans un "vrai" modem. Vu le nombre de composants WinModem incompatibles et typiquement non documentés, OpenBSD ne supporte pas les WinModems et ce n'est pas prêt de changer.

12.1.6 - Qu'est-il arrivé au support RAID des cartes Adaptec (aac)?

Adaptec a refusé de fournir une documentation précise et utile pour leurs contrôleurs RAID ([aac\(4\)](#)), à base de FSA. Ces contrôleurs RAID semblent être très buggés. Cette documentation est donc nécessaire pour créer un pilote utilisable. Vu que le pilote était très peu fiable, il a été supprimé du noyau GENERIC.

Je peux toujours compiler mon propre noyau avec le support aac(4), n'est-ce pas ?

Bien sûr. Mais quelle partie de l'expression "très peu fiable" n'avez-vous pas compris ? Ce n'est pas une fonctionnalité "expérimentale" mais un pilote connu pour être défectueux. Il fonctionnera peut-être correctement avec certaines variations de matériel mais nous ne vous recommandons pas de parier vos données sur ce pilote.

12.1.7 - Ma carte ami(4) ne supporte qu'un seul disque logique !

[ami\(4\)](#) contient un bogue connu qui cause une corruption de données si vous utilisez plus d'un seul volume sur certains contrôleurs. Pour les contrôleurs ayant ce problème, OpenBSD vous limitera à un seul disque logique. Vous aurez alors un message dans votre dmesg qui ressemble au message suivant :

```
ami0: FW A.04.03, BIOS vA.04.03, 4MB RAM
ami0: 3 channels, 16 targets, 2 logical drives
ami0: firmware buggy, limiting access to first logical disk
scsibus0 at ami0: 1 targets
```

12.1.8 - Comment est-ce que j'active ma carte accélératrice crypto ?

Le matériel accélérateur crypto supporté, comme les cartes basées sur le [hifn\(4\)](#) fonctionneront "telles quelles" quand elles seront installées dans un système OpenBSD. Pas d'activation ou d'autre configuration est nécessaire.

L'ajout de complexité pour dialoguer avec l'un de ces périphériques peut être parfois plus négatif que positif dans l'avantage de décharger le processeur de ces tâches de chiffrement. Vos résultats varieront fortement en fonction de la tâche que vous avez à accomplir.

12.2 - DEC Alpha

[aucune information pour le moment]

12.3 - AMD 64

12.3.1 - Puis-je faire fonctionner OpenBSD/amd64 sous mon Intel P4 ?

Dans le cas de plusieurs processeurs récents, la réponse est "oui". Malheureusement, il est difficile de distinguer entre les variantes de processeurs Intel qui supportent le jeu d'instructions amd64 et celles qui ne le supportent pas. Il est généralement plus simple d'essayer et de voir si ça fonctionne.

12.3.2 - Puis-je faire fonctionner mon binaire i386 sous OpenBSD/amd64 ?

Non.

OpenBSD/amd64 et OpenBSD/i386 sont deux plates-formes distinctes. A l'heure actuelle, aucune compatibilité binaire n'est offerte. OpenBSD encourage les applications libres mais il n'y a pas d'intérêt fort à offrir cette possibilité pour les développeurs.

Veillez noter que les amorçeurs de démarrage OpenBSD/amd64 et OpenBSD/i386 ont la possibilité de charger les noyaux de l'autre plate-forme rendant plus facile la réinstallation d'un système avec l'une ou l'autre plate-forme. Cependant, cela doit être un effacement total et une réinstallation. Tout binaire non supprimé et correspondant à l'installation précédente vous rendra la vie dure.

12.3.3 - Est-ce que c'est toujours mieux d'utiliser OpenBSD/amd64 sur les processeurs qui peuvent le supporter ?

Pas toujours.

Il existe un certain nombre de raisons pour lesquelles une personne préférerait utiliser OpenBSD/i386 au lieu de OpenBSD/amd64, même sur le matériel qui supporte du code amd64 :

- Besoin d'une compatibilité binaire i386 ou de compatibilité avec un autre système d'exploitation.
- Besoin d'utiliser des applications qui ne sont pas "64 bit clean".
- Besoin de mettre des disques sur une autre machine qui ne supporte pas OpenBSD/amd64.
- Pour certaines applications et certains matériels, OpenBSD/i386 peut s'avérer plus performant qu'OpenBSD/amd64. Peu de personnes constateront cela et nous travaillons à éliminer le plus de situations de ce genre possible.

12.4 - ARM-based appliances

[aucune information pour le moment]

12.5 - HP300

[aucune information pour le moment]

12.6 - HPPA

[aucune information pour le moment]

12.7 - i386

12.7.1 - Cartes réseau ISA

Etant donné qu'OpenBSD fonctionne bien sur le matériel ancien, les utilisateurs finissent souvent par utiliser des cartes réseau NIC avec les systèmes OpenBSD. Le matériel ISA nécessite beaucoup plus de configuration et de compréhension que le matériel PCI. De manière générale, vous ne pouvez pas vous contenter d'insérer votre carte dans la machine et espérer qu'elle fonctionne par magie. Avec plusieurs machines, si votre périphérique ISA n'est pas en mode "Plug 'n' Play" (PNP), vous devez réserver les ressources que la carte utilise au niveau du BIOS.

3Com 3C509B ep(4)

C'est une carte réseau ISA très performante, supportée par le périphérique [ep\(4\)](#). La version 'B' peut être distinguée de la version non-B par le nom figurant sur la carte et par une puce principale plus large sur la carte (approximativement 2.5cm sur un côté pour la version 'B' vs. 2cm sur un côté pour l'ancienne version), et elle fournira de meilleures performances sur un système chargé ou doté de deux cartes réseau. Les 3C509B sont livrées avec une configuration en mode PNP, qui malheureusement n'est pas conforme aux standards, et cause des problèmes dans le support [isapnp\(4\)](#) d'OpenBSD. L'adaptateur est d'abord sélectionné comme périphérique non-PNP, puis une seconde fois lorsque le support PNP devient disponible. Le résultat est l'affichage d'une carte réseau supplémentaire dans le `dmesg`. Le fonctionnement peut alors être correct ou problématique. Il est hautement recommandé de désactiver le mode PNP pour les cartes 3C509B et de configurer manuellement la carte avec des paramètres non conflictuels à l'aide des utilitaires DOS 3Com avant la configuration.

Le pilote `ep(4)` va trouver les cartes en utilisant n'importe quelle combinaison matérielle qui ne cause pas de conflit avec d'autres périphériques dans le système.

Si vous avez plusieurs cartes 3C509 dans votre système, il est recommandé d'inscrire l'adresse MAC sur

la surface des cartes située à l'extérieur du boîtier et d'utiliser `dmesg` pour identifier les cartes.

Il est à noter que les cartes 3C509, 3C905 et 3C590 sont souvent confondues. La carte 3C509 est une carte ISA 10Mbps, les cartes 3C905 et 3C590 sont des cartes PCI.

NE2000

La première carte NE2000 a été développée au milieu des années 1980 par Novell. Depuis, plusieurs constructeurs ont produit des cartes très similaires généralement appelées "compatible NE2000" ou clones. La performance de ces cartes clones varie énormément. Alors que quelques anciennes cartes "compatibles NE2000" ont de très bonnes performances, la plupart des cartes actuellement disponibles ont de faibles performances. Les cartes "compatible NE2000" sont supportées par le pilote [ne\(4\)](#) sous OpenBSD.

OpenBSD va bien gérer quelques cartes "compatible NE2000" capables d'utiliser ISAPNP lorsque le mode ISAPNP est activé. D'autres cartes vont devoir être configurés soit par le biais de cavaliers soit à l'aide d'un utilitaire de configuration sous DOS. Malheureusement, les premières cartes NE2000 n'avaient pas de support pour la configuration logicielle ou ISAPNP, il n'y a aucun standard -- vous avez besoin de l'utilitaire fourni au départ avec votre carte. Ce qui peut souvent être difficile à obtenir.

Le pilote `ne(4)` supporte trois configurations des cartes ISA NE2000 dans le noyau GENERIC OpenBSD :

```
ne0:  port 0x240 irq 9
ne1:  port 0x300 irq 10
ne2:  port 0x280 irq 9
```

Si ces paramètres ne sont pas acceptables, vous pouvez les ajuster en utilisant [User Kernel Configuration \(UKC\)](#) ou en [compilant un noyau personnalisé](#).

Il est à noter que le pilote `ne(4)` est assez "idiot" -- seul le port E/S est sondé, l'IRQ correspondant est *supposée*. [dmesg\(8\)](#) ne va pas refléter l'IRQ réelle de l'adaptateur dans le cas des pilotes ISA `ne(4)`. Si ce n'est l'IRQ réelle utilisée par votre carte, ça ne marchera pas.

Il est à noter qu'il existe des cartes non-ISA qui utilisent le pilote `ne(4)` -- des cartes PCI et PCMCIA existent. Ces notes ne s'appliquent pas à ces cartes qui sont auto-configurés.

12.7.2 - OpenBSD ne fonctionne pas sur mon système 0386/80386SX/80486SX !

80386SX/DX

Le support des processeurs 80386DX et 80386SX a été arrêté depuis OpenBSD 4.2. En plus des limitations du processeur 80386, les systèmes sont simplement trop lents et ont rarement assez de RAM ou un FPU requis pour faire fonctionner OpenBSD.

80486SX

Le processeur 80486SX fût une version "bas de gamme" du 80486. Il lui manque le support matériel de la virgule flottante (comme le 80386) qu'OpenBSD nécessite. Heureusement, des processeurs complets 80486DX sont assez communs et constituent une mise à jour facile sur la plupart des systèmes.

Les processeurs 80486DX et supérieur fonctionnent correctement sous OpenBSD.

12.7.3 - Mon dmesg affiche plusieurs périphériques partageant la même interruption (IRQ) !

C'est entièrement acceptable et, en vérité, même souhaitable pour les périphériques PCI. C'est une caractéristique de la conception du bus PCI. Certaines personnes diront que le partage de requêtes d'interruption est une mauvaise chose, cependant ils confondent la situation avec un bus ISA (où le partage d'IRQs n'est pas permit) ou ils ont une expérience avec du mauvais matériel ou logiciel.

Les périphériques ISA ne peuvent pas partager des IRQs. Si vous trouvez des périphériques ISA partageant des IRQs, vous devez corriger le problème.

12.7.4 - Mon clavier / ma souris n'arrête pas de se bloquer (ou d'avoir un comportement complètement erratique) !

Ces symptômes se produisent souvent lorsqu'on utilise un "switch box" (appelé aussi commutateur KVM) pour connecter plusieurs machines à un seul clavier, un seul écran et une seule souris. Vous pouvez essayer plusieurs "switch box" de différentes marques/conceptions. Cependant, OpenBSD est plus sensible à la commutation de la souris que d'autres systèmes d'exploitation. Le problème vient plus souvent de la commutation de la souris. Si vous n'utilisez pas la souris, la solution est simple : ne connectez pas le câble souris à la machine. Si vous utilisez la souris, une solution de contournement est d'utiliser une souris par machine et de continuer à faire de la commutation clavier et écran. Une autre solution de contournement consiste à utiliser un adaptateur PS/2 vers USB, permettant à OpenBSD de voir une souris de type USB, Si vous voulez uniquement avoir un accès console à la machine, vous devriez peut-être considérer l'utilisation d'une [console série](#).

12.7.5 - Mon Soekris a de faible performance

Les boîtiers Soekris sont jolis, faible consommation, petit et d'un coût faible pour un système PC et

beaucoup d'utilisateurs OpenBSD sont satisfaits. Cependant, il doit être rappelé que les processeurs, NICs et autres périphériques sont tous choisis pour leur faible consommation électrique, pas la performance. Bien qu'ils soient adaptés pour de multiples applications, ce ne sont pas des machines très performantes.

12.7.6 - J'obtiens des erreurs "missing interrupt" sur mon périphérique CF

Avec certains systèmes CF, comme les premiers systèmes Soekris 4801 et certains adaptateurs CF, le support DMA n'est pas correctement câblé, donc vous pouvez voir des messages du genre :

```
pciide0:0:0: bus-master DMA error: missing interrupt,
status=0x20
```

Dans ce cas, vous devez désactiver le DMA dans le driver [wd\(4\)](#) pour éviter ces problèmes. Cela peut être fait en utilisant [ukc](#) pour l'installation, puis [config\(8\)](#) pour le modifier de manière permanente :

```
ukc> change wd
 42 wd* at wdc*|pciide* channel -1 flags 0x0
change [n] y
channel [-1] ?
flags [0] ? 0x0ff0
 42 wd* changed
 42 wd* at wdc*|pciide* channel -1 flags 0xff0
ukc> quit
```

12.7.7 - Est-ce que ce nouveau CPU est supporté ?

La question du support est rarement le CPU lui-même, c'est presque toujours le matériel qui l'entoure et le supporte.

La plupart du temps, oui, le dernier CPU est bien supporté. Le support du matériel qui l'entoure peut ne pas fonctionner si il est différent de manière significative par rapport à des anciennes versions. Si vous avez des problèmes avec un nouveau matériel, soyez sûr d'essayer un [snapshot](#) avant de faire un [rapport de bogue](#).

12.8 - Landisk

[aucune information pour le moment]

12.9 - Luna88k

[aucune information pour le moment]

12.11 - MacPPC

12.11.1 - Pourquoi ma carte réseau bm(4) ne fonctionne pas ?

Le pilote [bm](#), supportant le composant BMAC utilisé sur certains systèmes MacPPC (y compris les premiers iMacs) a des problèmes. Une interface réseau [usb](#) est recommandée pour ces systèmes actuellement.

12.12 - MVME68k

[aucune information pour le moment]

12.13 - MVME88k

[aucune information pour le moment]

12.14 - SGI

[aucune information pour le moment]

12.15 - SPARC

[aucune information pour le moment]

12.16 - UltraSPARC (sparc64)

12.16.1 - Mon système UltraSPARC ne veut pas démarrer à partir de l'image sur disquette

Seul les modèles Ultra 1/1e et Ultra 2 peuvent démarrer *n'importe* quel OS à partir d'une disquette. Utilisez les méthodes d'installation CD-ROM, Miniroot ou démarrage réseau pour effectuer votre installation.

12.13.2 - J'obtiens le message "partition extends past end of unit" dans disklabel

Le disklable BSD ne peut décrire une géométrie disque supérieur à 8Go sous sparc et sparc64, bien que les entrées disklable individuelles peuvent être plus larges.

A chaque fois que vous exécutez disklable(8), ce dernier effectue des vérifications pour s'assurer que les entrées disklable correspondent bien à sa compréhension de la géométrie disque. Seulement, comme il voit une géométrie tronquée, il vous le signale et ne vous permettra pas d'éditer des entrées au delà de cette zone de 8Go sauf si vous lui dites d'utiliser la géométrie réelle. Vous pouvez faire cela à l'aide de la commande 'g' de l'éditeur en ligne de commande de [disklabel\(8\)](#) et dites lui d'utiliser "[d]isk geometry" :

```
# disklabel -E wd0
# Inside MBR partition 3: type A6 start 63 size 17912412
[... ]
Initial label editor (enter '?' for help at any prompt)
> g
[d]isk, [b]ios, or [u]ser geometry: [d] d
> w
> q
```

Vous aurez toujours des messages d'avertissement, mais vous pourrez configurer et utiliser votre disque comme vous le souhaitez. Une meilleure solution nécessiterait d'être compatible avec les systèmes existants déjà utilisés ainsi qu'avec Solaris s'exécutant sur des disques de capacité supérieure à 8Go mais cette solution n'existe pas aujourd'hui.

12.17 - DEC VAX

12.17.1 - Puis-je utiliser le simulateur VAX SIMH ?

Oui !

Le simulateur VAX [SIMH](#) peut être utilisé pour effectivement émuler un vrai VAX. Les instructions sont disponibles [OpenBSD/vax sur SIMH](#).

12.18 - Sharp Zaurus

12.18.1 - Les périphériques USB ne fonctionnent pas correctement

Le Zaurus a une très faible quantité de courant électrique disponible sur son port USB. Par conséquent, plusieurs périphériques USB ne fonctionneront pas lorsqu'ils sont directement branchés à ce port. Vous devez posséder un hub USB alimenté pour pouvoir utiliser ces périphériques.

[\[Index de La FAQ\]](#) [\[Section 11 - Le système X Window\]](#) [\[Section 13 - Multimédia\]](#)



www@openbsd.org

\$OpenBSD: faq12.html,v 1.58 2010/06/12 19:51:10 ajacoutot Exp \$



[\[Index de la FAQ\]](#) [\[Section 12 - Questions Spécifiques Aux Plates-Formes Et Au Matériel\]](#) [\[Section 14 - Configuration des disques\]](#)

13 - Multimedia

Table des matières

- [13.1 - Comment configurer mon périphérique audio ?](#)
 - [13.2 - Jouer différents types de formats audio](#)
 - [13.3 - Comment jouer des CDs dans OpenBSD ?](#)
 - [13.4 - Puis-je utiliser OpenBSD pour enregistrer des séquences audio ?](#)
 - [13.5 - Comment configurer un serveur audio ?](#)
 - [13.6 - Que puis-je faire si j'ai des problèmes audio ?](#)
 - [13.7 - Comment puis-je utiliser mes instruments MIDI ?](#)
 - [13.8 - Parlez moi de l'encodage Ogg Vorbis et MP3 ?](#)
 - [13.9 - Comment lire des vidéos DVDs sous OpenBSD ?](#)
 - [13.10 - Comment graver des CDs et DVDs ?](#)
 - [13.10.1 - Introduction et configuration de base](#)
 - [13.10.2 - Graver des CDs](#)
 - [13.10.3 - Graver des DVDs](#)
 - [13.11 - Je voudrais mes fichiers multimédia au format FOO.](#)
 - [13.12 - Est-il possible de lire des fichiers en streaming sous OpenBSD ?](#)
 - [13.13 - Puis-je utiliser un Java avec mon navigateur ? \(spécifique à i386 & amd64\)](#)
 - [13.14 - Puis-je utiliser Flash avec mon navigateur ? \(spécifique à i386\)](#)
-

13.1 - Comment configurer mon périphérique audio ?

Les périphériques concernant l'audio sous OpenBSD sont : `/dev/audio`, `/dev/sound`, `/dev/audioctl` et `/dev/mixer`. Pour une belle introduction à la couche de support audio, vous pouvez lire la page de manuel [audio\(4\)](#).

Tous les matériels audio supportés sont déjà inclus dans le noyau GENERIC, il n'y a donc pas besoin de configuration supplémentaire ou d'installation de drivers. Pour trouver des informations spécifiques à la puce de votre carte son, vous devrez déterminer quelle puce vous avez. Les puces supportées peuvent être trouvées sur la page des matériels supportés pour votre [plate-forme](#). Si vous avez déjà OpenBSD lancé, cherchez l'apparition d'un driver de carte son dans la sortie de la commande [dmesg\(8\)](#), et lisez le manuel du driver pour des informations plus spécifiques comme les options et autres détails sur celui-ci. Voici un exemple d'une puce audio dans une sortie dmesg :

```

auich0 at pci0 dev 31 function 5 "Intel 82801BA AC97" rev
0x04: irq 10, ICH2 AC97
ac97: codec id 0x41445360 (Analog Devices AD1885)
ac97: codec features headphone, Analog Devices Phat Stereo
audio0 at auich0

```

OpenBSD de base fournit deux outils pour superviser et configurer les périphériques audio. [audioctl\(1\)](#) est utilisé pour les paramètres de traitement audio, comme l'encodage, le taux d'échantillonnage et le nombre de voix, tandis que [mixerctl\(1\)](#) est utilisé pour les paramètres de mixage, comme la voix source, le niveau de gain et la mise en sourdine.

La commande suivante utilise [audioctl\(1\)](#) pour afficher les paramètres audio par défaut d'un périphérique audio.

```

$ audioctl -f /dev/audio
...

```

Il faut remarquer que `-f /dev/audio` est utilisé explicitement. Ouvrir `/dev/audio` fait que le périphérique audio fait un reset aux paramètres par défaut, ce qui est ce que nous voulons voir.

[audioctl\(1\)](#) est aussi très utile pour explorer les capacités d'un périphérique audio. Par exemple, pour savoir si un périphérique supporte quelques taux d'échantillonnage standards, vous pouvez simplement essayer de changer les taux de lecture :

```

$ audioctl play.rate=48000
play.rate: -> 48000
$ audioctl play.rate=44100
play.rate: -> 44100
$ audioctl play.rate=22050
audioctl: set failed: Invalid argument
$ audioctl play.rate=8000
audioctl: set failed: Invalid argument
$

```

Ce périphérique supporte des taux de playback de 48000 et 44100 Hz, mais pas 22050 ou 8000. Il faut remarquer que si le taux d'échantillonnage n'est pas supporté, il n'y a pas obligatoirement un message d'erreur, mais le taux d'échantillonnage renvoyé n'est pas celui désiré.

```
$ audiocctl play.rate=48000
play.rate: -> 48000
$ audiocctl play.rate=44100
play.rate: -> 48000
$ audiocctl play.rate=22050
play.rate: -> 48000
$ audiocctl play.rate=8000
play.rate: -> 48000
$
```

Ce périphérique supporte uniquement la lecture en 48000 Hz.

Un matériel audio est usuellement capable d'au moins quelques capacités de mixage minimale. Exécuter [mixerctl\(1\)](#) sans arguments affichera la liste des contrôles de mixage du périphérique et ses paramètres actuels.

```
$ mixerctl
...
```

Certains périphériques possèdent seulement quelques contrôles, alors que d'autres une centaine ou plus. Il faut remarquer que toutes les options de chaque puce audio n'atteint pas nécessairement le monde extérieur. Il y a donc, par exemple, plus de sorties énumérées que celles qui sont physiquement disponible sur une carte son ou carte mère.

Il y a quelques contrôles qui sont communs à plusieurs périphériques :

- `outputs.master` contrôle le niveau de sortie de lecture
- `inputs.dac` contrôle le niveau du DAC (convertisseur numérique vers analogique), utilisé quand on joue un fichier audio
- `record.source` contrôle les entrées qui sont mixées dans l'ADC (convertisseur analogique vers numérique), utilisé lors d'enregistrements
- `record.volume` ou `record.record` contrôle le niveau d'entrée de l'ADC

Le contrôle d'un périphérique audio peut-être affiché différemment. Par exemple, il pourrait ne pas être `outputs.master` comme ci-dessus, mais `outputs.outputs` qui fait la même chose.

Habituellement, les contrôles ont un label, mais parfois il faut simplement essayer différents réglages pour voir l'effet que chacun a sur le contrôle.

Certains périphériques qui sont connus comme EAPD, qui signifie amplificateur externe d'arrêt. Cependant c'est juste un bouton marche/arrêt. Il fait probablement référence à un "arrêt" car il est souvent utilisé comme économiseur d'énergie, ce qui signifie que ce type de contrôle est plus souvent trouvé sur les portables. Quelquefois il est nécessaire de paramétrer les contrôles avec `eapd` ou `extamp` dans leur nom avec `on` pour obtenir un signal de sortie.

Comme exemple simple d'un usage courant de `mixerctl`, pour paramétrer le volume des voix gauche et droite à 200, vous devrez

```
$ mixerctl outputs.master=200,200
outputs.master: 255,255 -> 207,207
```

Remarquez que la valeur est devenue 207. La raison à cela est que le périphérique de cette carte audio est un codec AC'97, qui utilise seulement 5 bits pour le contrôle, ce qui ne laisse que 32 valeurs possibles. D'autres matériels peuvent avoir une résolution différente.

Pour activer le canal "master", vous devriez faire :

```
$ mixerctl outputs.master.mute=off
outputs.master.mute: on -> off
```

Pour que vos changements soient effectifs à chaque redémarrage vous devrez éditer `/etc/mixerctl.conf`, par exemple :

```
$ cat /etc/mixerctl.conf
outputs.master=200,200
outputs.master.mute=off
outputs.headphones=160,160
outputs.headphones.mute=off
```

13.2 - Jouer différents types de formats audio

Audio digitalisée

Formats audio sans perte (AU, PCM, WAV, FLAC, TTA)

Certains formats audio sans perte peuvent être joués sans qu'un logiciel tierce partie ne soit nécessaire, ils contiennent les enregistrements digitaux non compressés en flots d'octets. Ces formats incluent les fichiers Sun audio (AU), les fichiers raw PCM (sans en-têtes) et les RIFF WAV.

OpenBSD est fourni avec [aucat\(1\)](#), un programme pour enregistrer et jouer de l'audio non compressé.

L'exemple suivant jouera un fichier WAV.

```
$ aukat -i filename.wav
```

[aukat\(1\)](#) supporte autant les écouteurs que les fichiers audio WAV avec l'option -i. aukat joue aussi les fichiers audio Sun ou les données audio sont encodées en 8 kHz monaural mulaw, qui est l'encodage le plus courant pour ce type de fichier audio.

Il est aussi possible de jouer des données audio non compressées en les envoyant directement sur le périphérique audio. Pour jouer un tel fichier, vous devrez connaître ses paramètres principaux : le type d'encodage, le nombre de canaux, la fréquence, et le nombre de bit par morceau. Si vous ignorez ces paramètres, vous pourrez les retrouver en utilisant l'utilitaire : [file\(1\)](#).

```
$ file music.au
```

```
music.au: Sun/NeXT audio data: 16-bit linear PCM, stereo,
44100 Hz
```

```
$ file music.wav
```

```
music.wav: Microsoft RIFF, WAVE audio data, 16 bit, stereo
44100 Hz
```

La seule chose qu'il vous reste à savoir sur ces fichiers d'exemple est qu'ils sont encodés dans le format d'octet "little-endian" utilisant la quantification linéaire signée. Vous pouvez vérifier cela en lisant son en-tête avec [hexdump\(1\)](#). Si vous utilisez un fichier sans en-tête (raw), il n'y a aucun moyen de connaître les paramètres à l'avance. Configurez les paramètres suivants en utilisant [audiocvt\(1\)](#).

```
play.encoding=slinear_le
play.rate=44100
play.channels=2
play.precision=16
```

Ensuite, envoyez le fichier audio au périphérique son :

```
$ cat music.au > /dev/sound
```

Si vous avez appliqué les bons paramètres, vous devriez entendre ce que vous souhaitiez entendre.

Remarque : Utilisez toujours `/dev/sound`, et non `/dev/audio`, si vous souhaitez que les paramètres appliqués restent en place.

Il y a d'autres utilitaires que vous pouvez utiliser, comme `audio/waveplay` dans les paquetages et les

ports. Bien sur, des logiciels populaires comme XMMS sont aussi capables de jouer ces fichiers aussi bien que d'autres formats;

A l'inverse de ce qui a été dit précédemment, il existe des formats audio sans perte de qualité. Free Lossless Audio Codec (FLAC) et TTA en sont des exemples. L'implémentation FLAC a été portée sous OpenBSD et peut être trouvée dans `audio/flac` dans les paquetages et les ports.

Formats audio utilisant la compression par perte (Ogg Vorbis, MP3, WMA, AAC)

La compression par perte de qualité est souvent utilisée pour l'audio ou les autres fichiers multimédia. L'idée est qu'un certain volume de données est supprimé pendant la compression, en utilisant une méthode qui fait que le résultat compressé est toujours très exploitable et dispose d'une bonne qualité pour être joué. L'avantage de ces techniques est qu'ils autorisent de plus gros ratios de compression, ce qui résulte en une plus faible place disque utilisée et de plus besoins en bande passante.

Un bon exemple est le format libre, ouvert et non déposé [Ogg Vorbis](#). Pour jouer des fichiers Ogg Vorbis, vous pouvez utiliser l'utilitaire `ogg123`, qui est inclus dans le paquetage `audio/vorbis-tools`. Par exemple :

```
$ ogg123 music.ogg
```

```
Audio Device: Sun audio driver output
```

```
Playing: music.ogg
```

```
Ogg Vorbis stream: 2 channel, 44100 Hz
```

```
Time: 00:02.95 [02:21.45] of 02:24.40 (133.1 kbps) Output
```

```
Buffer 87.5%
```

Bien sur, les plugins Ogg Vorbis existent pour beaucoup d'autres logiciels audio.

Un autre exemple est le très populaire encodage MPEG-1 Audio Layer 3 (MP3), qui a, cependant, ses problèmes de redistribution et brevets. Beaucoup de programmes peuvent jouer des fichiers MP3, consultez la section `audio` du système de paquetages et de ports et choisissez celui que vous souhaitez.

Que dire du format Windows Media Audio (WMA) ? Les fichiers de ce type peuvent être lus en utilisant `x11/mplayer` qui se sert de l'architecture [FFmpeg](#).

Un bon point de départ pour en savoir plus sur les différents formats de fichiers audio est de lire l'article Wikipedia suivant : [Audio file formats \(en anglais\)](#).

Son synthétisé

MIDI

Le protocole "Musical Instrument Digital Interface" (MIDI) est fourni par [MIDI devices](#). Si vous ne possédez pas de synthétiseur MIDI, mais que vous voulez jouer un fichier MIDI standard (SMF), vous pouvez utiliser un logiciel de rendu de données MIDI générant des fichiers audio. Par défaut, le port `audio/timidity` utilise les fichiers MIDI et les joue sur le périphérique audio :

```
$ timidity file.mid
```

MOD

Un module "Soundtracker" est un format binaire qui mixe les plages audio avec des ordres de séquences, autorisant à jouer de très grands morceaux de musique digitalisée avec un qualité raisonnablement bonne.

Le moyen le plus simple de jouer vos fichiers MOD favoris sur OpenBSD est probablement d'utiliser le logiciel XMMS, disponible via les paquetages et les ports. Vous devriez installer le sous-paquetage `-mikmod` pour XMMS afin de lui permettre d'utiliser la bibliothèque sonore MikMod, qui supporte les modules de formats MOD, S3M, IT et XM.

Vous trouverez aussi liste de "trackers" dans la section `audio` de la collection de ports et de paquetages, comme par exemple `tracker`, `soundtracker`. Avec ces trackers, vous ne pouvez pas seulement jouer mais aussi générer vos propres modules. Remarquez que tous les trackers ne sont pas supportés par les outils de l'arbre des ports. Vous êtes invités à [envoyer un port](#) de votre programme de tracker favori.

13.3 - Comment jouer des CDs dans OpenBSD ?

Il est possible de jouer un CD audio aussi bien en ayant le lecteur de CD jouant le disque et envoyant les données audio analogique à la carte sonore qu'en lisant les données audio en envoyant les échantillons numériques à la carte sonore à travers le bus PCI.

Pour jouer un CD audio, en utilisant la sortie analogique de votre lecteur CD-ROM, vous pouvez

- Utiliser la sortie casque audio, souvent présente sur la face avant de votre lecteur de cdrom.
- Connecter la sortie audio sur le connecteur noir de votre carte audio. Oui, il s'agit d'un câble supplémentaire près de celui des données (SCSI/IDE) et de celui de l'alimentation.

Un utilitaire pratique en ligne de commande appelé [cdio\(1\)](#) a été inclut dans le système de base. Appelé sans paramètres, il entrera en mode interactif. Si vous souhaitez jouer un CD-ROM, entrez :


```
$ cdio play
```

Cela lira depuis le premier lecteur CD-ROM, `cd0`, par défaut. Notez que l'utilisateur qui lance `cdio` doit bénéficier des permissions de lecture sur le périphérique CD-ROM (ex. `/dev/rcd0c`). Etant donné que par défaut ce périphérique n'est accessible en lecture que par `root` et le groupe "operator", pour plus de facilité vous pouvez ajouter l'utilisateur au groupe operator en éditant la ligne correspondante dans `/etc/group`. Une autre solution est de modifier les permissions du périphérique comme vous le souhaitez.

Vous aurez peut-être besoin d'activer l'entrée CD au niveau du mixeur de sons. Au même titre que les sorties, le nom réel de ce canal d'entrée varie selon les systèmes. Mais la commande à utiliser ressemble à la commande suivante :

```
$ mixerctl inputs.cd.mute=off
```

Il est aussi possible qu'il n'y est pas de connexion audio analogique entre votre lecteur CD et le périphérique audio. Dans ce cas vous pouvez utiliser les commandes `cdplay` de `cdio` pour envoyer les données audio du CD vers la carte sonore à travers le bus PCI.

```
$ cdio cdplay
```

Si vous préférez une belle GUI, il y a beaucoup de lecteurs de CDs basés sur X11 dans la collection de paquetages et de ports. Consultez simplement la section `audio`.

13.4 - Puis-je utiliser OpenBSD pour enregistrer des séquences audio ?

Oui. La plupart des périphériques supportent l'enregistrement. [aucat\(1\)](#) est distribué avec OpenBSD et peut-être utilisé pour l'enregistrement.

```
$ aucat -o file.wav
```

La commande précédente commencera l'enregistrement d'un fichier au format WAV. Appuyez sur [CTRL]-C pour terminer l'enregistrement. Le fichier contiendra des échantillons signés 16-bit stéréo, échantillonné à 44.1 kHz. D'autres formats d'échantillonnage, taux d'échantillonnage et nombre de voix peuvent être enregistré. Lisez le manuel pour plus de détail.

Utilisez `aucat` pour rejouer le fichier :

```
$ aucat -i file.wav
```

Si l'enregistrement semble fonctionner mais que la sortie de l'enregistrement est muette ou n'est pas ce que vous souhaitez, le mixer a surement besoin d'être configuré. Soyez sur que vous sélectionnez le bon périphérique pour enregistrer et que la source n'est pas muette. Vous pouvez configurer les paramètres nécessaires en utilisant [mixerctl\(1\)](#). Par exemple :

```
inputs.mic.mute=off
inputs.mic.preamplifier=on
inputs.mic.source=mic0
record.source=mic
record.volume=255,255
record.volume.mute=off
record.mic=255
record.mic.mute=off
```

Il y a des paramètres pour l'enregistrement avec un microphone. La pré-amplification a été activée, car sinon le son enregistré peut être quelque peu silencieux sur certains systèmes.

13.5 - Comment configurer un serveur audio ?

Est-ce que j'ai besoin d'un serveur audio ?

L'utilitaire [aucat\(1\)](#) peut-être utilisé comme serveur audio, qui agit comme une couche entre le pilote audio(4) et l'application audio. Cela vise à :

- Dépasser les incompatibilités entre le matériel et les applications. Par exemple une application ne supporte pas l'encodage et les taux d'échantillonnage du matériel; si c'est le cas, le cas peut faire les conversions nécessaires à la volée.
- Permettre à plusieurs applications d'utiliser le matériel simultanément. Par exemple, pour utiliser un lecteur de musique en tâche de fond pendant que l'on exécute une application jouant des sons, ou pour utiliser simultanément les hauts-parleurs frontaux pour la musique et la prise casque pour la téléphonie est un autre exemple.

Si les applications utilisées sont compatibles avec votre matériel et que vous ne pensez pas utiliser plusieurs applications simultanément, vous n'avez donc pas besoin d'un serveur audio.

Comment configurer aucat(1)

Il n'y a pas de fichier de configuration, et dans la plupart des cas, rien à faire. Tapez :

```
$ aucat -l
```

démarrera le serveur sur le périphérique audio par défaut (celui ou pointe le lien symbolique sur `/dev/audio`) fonctionnant à 44.1KHz et utilisant deux voix (stéréo). Cela signifie que les applications utilisant la stéréo à 44.1KHz fonctionneront de manière optimale, c'est à dire, sans passer par du code de conversion. Si le périphérique ne supporte pas ces paramètres, [aucat\(1\)](#) utilisera automatiquement un autre ensemble de paramètres.

Si vous avez les privilèges nécessaires, vous pouvez augmenter sa priorité, pour diminuer la probabilité d'un problème de cache, par exemple :

```
$ sudo renice -n -20 `pgrep -x aucat`
```

Remarquez de `aucat(1)` démarre en mode full-duplex par défaut, ce qui nécessite que votre matériel supporte le mode full-duplex. Si cela n'est pas le cas, vous devez démarrer `aucat(1)` en mode lecture seule ou enregistrement seul, par exemple :

```
$ aucat -l -m play
```

Comment démarrer `aucat(1)` au démarrage du système ?

Si vous êtes le seul utilisateur local de la machine, vous pouvez démarrer [aucat\(1\)](#) au démarrage du système via [rc.local\(8\)](#), par exemple :

```
nice -n -20 su -m mylogin -c "aucat -l"
```

Le processus `aucat(1)` ne peut pas être partagé entre plusieurs utilisateurs.

Quel temps de latence configurer ?

Le temps de latence est le temps entre la décision pour un programme de jouer un échantillon et le moment où l'utilisateur entend l'échantillon. Comme les données audio sont toujours en cache, ce délai est proportionnel à la taille du cache audio. Les valeurs suivantes sont recommandées :

- Synthétiser temps-réel : 5ms. C'est le temps qui est pris entre l'appuie d'une touche sur le clavier MIDI et entendre la note. Grossièrement, 5ms correspond au temps que mets le son pour se propager sur 1,75m.
- Jeux : 50ms. C'est le temps qui correspond au moment où vous voyez l'événement et entendez le son correspondant.
- Lecteurs de films et assimilés : 500ms et plus. Ces applications sont "connues" pour jouer le son en avance, et envoyer les données audio de tel façon quelles sont jouées simultanément avec l'image correspondante.

Plus le cache audio est petit (pour avoir un temps de latence faible), plus les probabilités sont grandes d'avoir des problèmes de overrun/underrun. Le cache en overruns/underruns donnera au son un "bégaiement".

En mode serveur, `aucat(1)` impose un temps minimum de latence pour toutes les applications audio, et le temps de latence par défaut est autour de 250ms. Si vous prévoyez d'utiliser des applications qui demandent une latence plus courte, utilisez l'option "-b" pour sélectionner la latence désirée (exprimée en nombre d'échantillons). Par exemple à 44100 échantillons/seconde, 50ms de latence correspond à :

$$44100 \text{ échantillons/seconde} \times 0.050 \text{ secondes} = 2205 \text{ échantillons}$$

donc lancez [aucat\(1\)](#) ainsi :

```
$ aucat -b 2205 -l
```

Une latence plus faible améliore-t-elle la synchronisation audio-vidéo ?

Synchroniser l'audio à la vidéo ne nécessite pas une latence faible. Les problèmes de synchronisation sont souvent causés par le logiciel lui-même (mauvaise implémentation, bogues, ...). Forcer l'application à utiliser des cache plus petits (en démarrant `aucat(1)` en mode latence faible) peut cacher les problèmes actuels dans certains cas et donner la sensation que le logiciel fonctionne mieux, mais normalement la bonne chose à faire est de commencer à chercher le bogue correspondant.

13.6 - Que puis-je faire si j'ai des problèmes audios ?

Si vous n'entendez rien quand vous jouez de l'audio, il est possible que le contrôle du mixage soit trop faible ou simplement muet. Voir la section [13.1 - Comment puis-je configurer mon périphérique audio](#) pour configurer le mixage. Merci de rendre non muet **toutes** les entrées et sorties avant de reporter un problème.

Si le son est distordu, il est possible que votre carte audio ne supporte qu'un unique ou nombre limité de taux d'échantillonnage ou d'encodages. Voir la section [13.1 - Comment configurer mon périphérique audio ?](#) par exemple pour déterminer quels paramètres audio votre périphérique supporte.

Si votre périphérique ne supporte seulement que des encodages peu courants ou qu'un ou peu de taux d'échantillonnage, et l'application que vous utilisez ne peut pas faire les conversions audio nécessaires, considérez `aucat(1)` comme serveur audio. Voir le paragraphe [13.5 - Comment configurer un serveur audio ?](#)

Si vous avez toujours des problèmes, voici quelques choses à considérer :

- Quelques vieilles cartes ISA ont des astuces particulières :
 - Certaines ont besoin d'être configurées avec des adresses I/O et IRQ différentes pour éviter les conflits avec d'autres matériels. Vous pouvez essayer facilement différentes combinaisons en utilisant le [User Kernel Configuration \(UKC\)](#).
 - Il est possible qu'un pilote moins optimal soit attaché au périphérique audio et que vous ayez de meilleurs résultats avec un autre pilote. Ce n'est pas la chose la plus simple à observer, mais regardez attentivement votre sortie [dmesg\(8\)](#) et trouvez les lignes où votre pilote audio est attaché. Si vous voyez plus d'un pilote sonore attaché (ou essayant), essayez les un à un en les désactivant via [User Kernel Configuration \(UKC\)](#).
- Trouvez des informations sur votre périphérique sonore. Utilisez la documentation ou utilisez les moteurs de recherche sur Internet pour trouver ses spécifications. Pour les périphériques [ac97\(4\)](#) et [azalia\(4\)](#) regardez la documentation pour le contrôleur et le codec. Cela vous permettra de trouver la source du problème.

Si vous croyez que votre périphérique doit fonctionner, mais que pour une raison inconnue ce n'est pas le cas, il est temps de déboguer. Les étapes suivantes peuvent déterminer si les données sont traitées par le DAC.

```
$ cat > /dev/audio < /dev/zero &
[1] 9926
$ audiocctl play.{seek,samples,errors}
play.seek=48000
play.samples=3312000
play.errors=0
$ audiocctl play.{seek,samples,errors}
play.seek=57600
play.samples=7065600
play.errors=0
$ audiocctl play.{seek,samples,errors}
play.seek=48000
play.samples=9379200
play.errors=0
$ kill %1
$ fg %1
cat > /dev/audio < /dev/zero
Terminated
```

Ici nous observons que le compteur des données traitées `play.samples` augmente chaque fois lors de la vérification, donc les données passent. Nous observons aussi que le périphérique récupère assez de données dans le tampon `play.seek` et que le périphérique n'a pas de problèmes via `play.errors`. C'est une bonne chose.

Il faut remarquer que même si vous avez des enceintes connectées quand vous faites vos tests, vous n'entendrez rien. Le test envoie des zéros au périphérique qui devient silencieux pour tous les encodages par défaut supportés.

Depuis que nous savons que le périphérique peut traiter des données, c'est une bonne idée de vérifier encore les paramètres de mixage. Soyez sûr que toutes les sorties et entrées ne sont pas muettes et à un niveau raisonnable.

Si à ce moment vous continuez d'avoir des problèmes, il est probablement temps [d'envoyer un fichier de rapport de bogue](#). En plus du rapport d'information de bogue comme un `dmesg` complet et une description du problème, merci d'inclure aussi la sortie par défaut de `mixerctl -v` et les sorties des tests précédents du traitement DAC.

13.7 - Comment puis-je utiliser mes instruments MIDI ?

Le protocole "Musical Instrument Digital Interface" (MIDI) fournit une façon standardisée et efficace de représenter les informations musicales comme données électroniques. Une donnée MIDI contient uniquement les informations nécessaires au synthétiseur pour jouer les sons, au lieu des sons eux-mêmes. Plus d'informations : [Tutorial on MIDI and Music Synthesis](#).

Pour jouer des données MIDI, un synthétiseur connecté au port MIDI de la machine est indispensable. De façon similaire, pour enregistrer des données MIDI, un instrument MIDI est indispensable (comme un clavier MIDI). Certaines cartes sonores embarquent des synthétiseurs MIDI qui sont attachés aux ports MIDI. Les instruments MIDI évolués peuvent contenir plusieurs éléments (synthétiseurs, claviers, contrôle de la surface, etc...), ils apparaissent comme plusieurs ports MIDI sous OpenBSD.

Quand OpenBSD est déjà fonctionnel, regardez les ports MIDI dans la sortie de la commande `dmesg(8)`. Un exemple de ports MIDI dans une sortie `dmesg` :

```
midi0 at pcppi0: <PC speaker>
umidi0 at uhub2 port 2 configuration 1 interface 0 "Roland
Roland XV-2020" rev 1.10/1.00 addr 2
midi1 at umidi0: <USB MIDI I/F>
umidi1 at uhub1 port 2 configuration 1 interface 1
"Evolution Electronics Ltd. USB Keystation 61es" rev
1.00/1.25 addr 3
midi2 at umidi1: <USB MIDI I/F>
```

Il montre trois ports MIDI correspondants à :

- `/dev/rmidi0` - le haut parleur du PC

- /dev/rmidi1 - synthétiseur connecté par USB
- /dev/rmidi2 - un clavier MIDI maître

Pour tester votre clavier MIDI vous pouvez utiliser l'utilitaire [hexdump\(1\)](#) qui affiche les données MIDI que vous jouez dessus :

```
$ hexdump -e '1/1 "%02x\n"' </dev/rmidi2
90
3c
71
...
```

La sortie du clavier peut être connectée à l'entrée du synthétiseur comme :

```
$ cat -u /dev/rmidi1 >/dev/rmidi2
```

Maintenant vous pouvez entendre sur le synthétiseur ce que vous jouez sur le clavier MIDI. Les informations que vous jouez sont capturées de /dev/rmidi1 et envoyées sur /dev/rmidi2 en temps réel. Reférez vous aux pages du manuel [midi\(4\)](#) pour plus d'informations.

L'utilitaire principal pour jouer des fichiers MIDI est [midisplay\(1\)](#). Jouer un fichier MIDI standard, dans cet exemple à travers le synthétiseur, est aussi simple que :

```
$ midisplay -d 1 file.mid
```

Il faut remarquer que nous spécifions le numéro de périphérique MID à 1 (/dev/rmidi1) comme paramètre à cause du numéro de périphérique 0 qui est utilisé par défaut.

Pour enregistrer des fichiers MIDI, vous pouvez utiliser l'utilitaire `smfrec` fournit dans le port `audio/midish`, ainsi :

```
$ smfrec -d /dev/rmidi1 -i /dev/rmidi2 file.mid
```

enregistrera ce qui est joué au clavier (/dev/rmidi2) pendant qu'il enverra en temps réel sur le synthétiseur (/dev/rmidi1) ce que vous pourrez entendre en jouant. Une opération plus compliquée comme l'édition, routage, mixage et la transformation de données MIDI peut être réalisée en utilisant l'utilitaire `rmidish` disponible dans le port `audio/midish`.

13.8 - Parlez moi de l'encodage Ogg Vorbis et MP3 ?

Ces formats ont déjà été mentionnés dans [Jouer différents formats audio](#). Dans cette section nous

donnerons une brève introduction à l'encodage de tels fichiers. Si vous êtes intéressé par le fonctionnement de ces codecs de compression audio, de plus amples informations peuvent être trouvées dans les articles Wikipedia concernant [Vorbis](#) et [MP3](#).

Ogg Vorbis

L'encodage de formats audio raw, WAV ou AIFF en [Ogg Vorbis](#) peut être réalisé via l'utilitaire **oggenc**, faisant partie du paquetage `audio/vorbis-tools`, qui est disponible via le système de ports et de paquetages d'OpenBSD.

Disons que vous avez un certain nombre de fichier WAV à encoder, par exemple votre album favori que vous venez juste d'extraire de son CD. Pour encoder tous ces fichiers en utilisant une fréquence d'environ 192 kbps, vous pouvez utiliser une commande comme celle ci :

```
$ oggenc *.wav -b 192
```

Une fois terminé, cela vous fournira un ensemble de fichiers .ogg dans le dossier courant. Des exemples plus précis ainsi que les options d'encodage peuvent être trouvés dans le manuel oggenc.

MPEG-1 Audio Layer 3 (MP3)

Si pour quelque raison que ce soit, vous souhaitez utiliser le format MP3, vous pouvez utiliser "[Lame ain't an MP3 encoder](#)" ([LAME](#)), un outil d'éducation utilisé pour apprendre l'encodage MP3. Lame est inclus dans l'arbre des sources OpenBSD. Notez que pour des raisons de brevets MP3, vous ne trouverez pas ce paquetage dans le [Set de CDs officiels](#).

Ci-dessous ce trouve un exemple simple d'encodage d'un fichier WAV avec une fréquence de 192 kbps :

```
$ lame -b 192 track01.wav track01.mp3
```

Pour toutes les options et les détails, merci de consulter le manuel fournit avec lame.

13.9 - Comment lire des vidéos DVDs sous OpenBSD ?

Pour l'instant, OpenBSD supporte les médias DVD à travers le système de fichiers ISO 9660 qui est aussi utilisé pour les CD-ROMs et, depuis la version 3.8, également à travers le système de fichiers plus récent, [Universal Disk Format \(UDF\)](#), présent sur certains DVDs. Cependant, presque tous les disques DVD-Video et DVD-ROM utilisent le format passerelle UDF, qui est une combinaison des systèmes de fichiers DVD MicroUDF (sous-ensemble de UDF 1.02) et ISO 9660. Ce format est utilisé afin de garantir la compatibilité ascendante.

Puisque la plupart des ordinateurs disposant d'un lecteur DVD-ROM utilisent un décodage logiciel, il est recommandé d'avoir au moins un processeur Pentium II 350-MHz ou équivalent pour disposer d'une bonne qualité d'écoute.

Quelques utilitaires de lecture DVD populaires ont été portés sur OpenBSD. Comme par exemple [ogle](#), [mplayer](#), [xine](#) et [kaffeine](#). Veuillez lire les instructions d'installation livrées avec ces paquetages, car ces fichiers peuvent nécessiter plus de configuration. Avec ces utilitaires il est possible de lire des DVD directement en accédant au périphérique raw. Bien sur, il est possible de monter d'abord le DVD en utilisant [mount_cd9660\(8\)](#), et de jouer les fichiers sur ce système de fichiers ou un autre.

Remarque :

- Quasiment tous les DVDs que vous achetez sont cryptés en utilisant le "Content Scrambling System (CSS)". Pour pouvoir jouer ces DVDs cryptés, vous pouvez utiliser la bibliothèque **libdvd** disponible via les paquetages et les ports.
- Soyez averti qu'un code région doit être présent sur votre disque DVD. Cela ne devrait pas réellement être un problème lorsque les DVDs sont joués sur un ordinateur.

13.10 - Comment graver des CDs et DVDs ?

13.10.1 - Introduction et configuration de base

Vous devriez tout d'abord être sûr que votre graveur CD/DVD a bien été reconnu et configuré par le noyau. La plupart des périphériques SCSI sont reconnus. Les matériels IDE/ATAPI et USB sont supportés au travers de l'émulation SCSI. Vous trouverez rapidement votre périphérique dans la sortie d'un [dmesg\(8\)](#). Cherchez simplement les lignes commençant par "cd", par exemple :

```
cd0 at scsibus0 targ 0 lun 0: <TOSHIBA, CD-ROM XM-5702B,
2826> SCSI0 5/cdrom removable
cd1 at scsibus1 targ 4 lun 0: <PLEXTOR, CD-R PX-R412C,
1.04> SCSI2 5/cdrom removable
```

Mais cdrecord -scanbus ne fonctionne pas!

Oui. OpenBSD utilise un espace de nommage de périphérique différent de celui du système d'exploitation pour lequel cdrecord a été écrit. Tous les périphériques doivent être dans la sortie dmesg, comme mentionné ci-dessus. L'information dont vous avez besoin se trouve ici.

Error: mount_cd9660: /dev/cd2c on /mnt/cdrom: No such file or directory

Par défaut, l'installateur OpenBSD ne crée que deux périphériques `cd cd0` et `cd1`. Pour commencer à utiliser votre périphérique `cd2`, vous devez créer le périphérique nécessaire pour celui-ci. La méthode recommandée pour cela est d'utiliser le script [MAKEDEV\(8\)](#) (Sélectionnez votre plate-forme) :

```
# cd /dev
# ./MAKEDEV cd2
```

Dans ce qui suit, nous accéderons principalement au graveur de CD/DVD à travers le périphérique *raw* et **non** le périphérique *block*.

Vérifier le fonctionnement du graveur de CD/DVD

Il est recommandé de vérifier que votre graveur de CD/DVD fonctionne correctement. Dans cet exemple, j'utilise un graveur DVD USB 2.0 :

```
cd2 at scsibus2 targ 1 lun 0: <LITE-ON, DVDRW LDW-851S,
GS0C> SCSI0 5/cdrom removable
```

Essayez de l'utiliser en y montant un CD/DVD. Si vous le souhaitez, vous pouvez aussi vérifier les taux de transfert que vous obtenez en copiant des fichiers sur votre disque dur. La commande [time\(1\)](#) sera votre meilleure amie.

Si quelque chose se passe mal ici et que vous obtenez des erreurs durant cette phase, il est important de les corriger avant de commencer à écrire un CD/DVD.

Je souhaite maintenant graver un CD ! Pouvons nous commencer ?

Auparavant, il est judicieux de garder quelques conseils en tête :

- Ne lancez pas de tâches utilisant lourdement le disque pendant que vous écrivez un CD/DVD. Cela réduira la vitesse d'écriture vers votre graveur CD/DVD. Si la vitesse descend plus bas que ce que le graveur attend pendant un temps trop long, son buffer sera vide. Ce phénomène est aussi connu sous le nom de "buffer underrun".
- Evitez les chocs pendant la gravure, cela pourrait faire glisser le laser de sa piste, et pourra conduire à des erreurs sur le disque.
- Tous les graveurs de DVD ne supportent pas tous les formats DVD, voyez ci-dessous.

13.10.2 - Graver des CDs

Créer des CD-ROMs de données

Tout d'abord, vous devrez créer un système de fichier ISO 9660 à mettre sur un CD-ROM. Pour cela vous pouvez utiliser l'utilitaire [mkhybrid\(8\)](#) fournit dans le système de base ou mkisofs du [paquetage cdrtools](#) qui est plus performant avec des arborescences importantes. Dans l'exemple suivant, nous utiliserons mkhybrid bien que l'utilisation de mkisofs soit similaire.

Comme exemple d'utilisation, j'ai essayé de sauvegarder les sources d'OpenBSD dans une image ISO 9660 :

```
$ mkhybrid -R -o sys.iso /usr/src/sys

Using ALTQ_RMC.000;1 for /usr/src/sys/altq/
altq_rmclass_debug.h (altq_rmclass.h)
...
Using IEEE8021.00H;1 for /usr/src/sys/net80211/
ieee80211_amrr.c (ieee80211.c)
 10.89% done, estimate finish Sat Nov  3 08:01:23 2007
 21.78% done, estimate finish Sat Nov  3 08:01:28 2007
...
 87.12% done, estimate finish Sat Nov  3 08:01:31 2007
 98.01% done, estimate finish Sat Nov  3 08:01:32 2007
Total rockridge attributes bytes: 896209
Total directory bytes: 2586624
Path table size(bytes): 11886
Max brk space used 0
45919 extents written (89 Mb)
```

L'option `-R` demande à mkhybrid de rajouter des extensions Rock Ridge dans l'image ISO 9660. Le "Rock Ridge Interchange Protocol" a été créé pour supporter le système de fichiers POSIX à l'intérieur du système de fichiers ISO 9660, comme les longs nom de fichiers, les créateurs de fichiers, les permissions, les liens de fichiers, les fichiers de périphériques, les hiérarchies de fichiers profondes (plus de 8 niveaux de sous-dossiers), etc...

Si vous souhaitez que les longs noms de fichiers sur votre CD-ROM soient lisibles sur les systèmes Windows et DOS, vous devriez ajouter le paramètre `-J` pour inclure les extensions Joliet dans l'image ISO 9660.

Après avoir créé les systèmes de fichiers, vous pouvez le vérifier en [montant l'image ISO 9660](#). Si tout se passe bien, vous êtes maintenant prêt à écrire le CD-R(W). La façon la plus aisée pour faire cela consiste à employer l'utilitaire [cdio\(1\)](#).

Si vous utilisez des média réinscriptibles tels que des CD-RW, vous devez tout d'abord procéder à un effacement avant de pouvoir faire une réécriture.

```
# cdio -f cd1c blank
```

Vous êtes maintenant prêt à graver l'image créée dans l'exemple précédant sur un CD-R(W) vierge. Vous pourrez utiliser une commande similaire à :

```
# cdio -f cd1c tao sys.iso#
```

En utilisant les options spécifiées ci-dessus, nous demandons à `cdio` d'utiliser le second périphérique CD-ROM comme graveur de CD.

Pour vérifier que le CD-ROM a bien été écrit, vous pouvez le monter et vérifier que tout ce qui doit y être y est. Pour monter le système de fichier, vous devez utiliser le périphérique *block* pour le lecteur de CD-ROM, qui dans ce cas est toujours le graveur de CD :

```
# mount /dev/cd1c /mnt/cdrom
```

Créer des CDs audio

Pour graver des CDs audio, vous pouvez encore une fois utiliser [cdio\(1\)](#) avec l'option `tao -a`.

Comme exemple, je ferais une copie de sauvegarde d'un de mes CDs de musique. Cela implique deux étapes :

1. Copier les pistes audio depuis votre CD original Par exemple :

```
# cdio -f cd1c cdrip
```

Cette commande va extraire une série de fichiers WAV de votre second lecteur CD-ROM et les écrire sur votre disque.

2. Graver les pistes sur un CD vierge Par exemple :

```
# cdio -f cd1c tao -a *.wav
```

13.10.3 - Graver des DVDs

Il y a quelques détails importants à propos des DVDs que vous devriez savoir avant de créer vos propres DVDs.

Remarques importantes :

- Si vous souhaitez vraiment tout savoir sur les DVDs, je vous suggère de lire la très complète [FAQ DVD](#).
- Cette section a été très peu testée et nous n'avons probablement pas testé toutes les combinaisons de médias et de graveurs possibles. Cependant nous avons eu où reçu des retours d'expériences positifs avec les formats de DVD suivants. Vous êtes invités à [nous informer](#) de vos succès ou échecs.

Différents formats de DVD

Il y a plusieurs formats différents de DVD. Les plus utilisés sont les formats DVD-R, DVD-RW, DVD+R et DVD-RW (R signifie enregistrable une seule fois, RW signifie qu'il peut être réécrit quelques milliers de fois). Il existe d'autres standards plus compétitifs.

Un format différent est le DVD-RAM, qui a principalement été développé comme un lecteur de données et dispose de fonctions d'écriture de paquets avancées, autorisant à l'utiliser comme une sorte de disque dur optique. DVD-RAM n'est pas recommandé pour une utilisation vidéo car les vidéos sont écrites sur le disque dans un format non compatible avec les lecteurs de DVD normaux.

Le point le plus important est d'utiliser le média adapté à votre graveur de DVD. Si vous souhaitez la compatibilité avec d'autres lecteurs de DVD, veuillez lire [cette section](#) de la FAQ DVD.

Vitesse de gravure DVD

Il est important de spécifier que les indications de vitesses des DVDs diffèrent de celles des CD-ROMs. La table suivante en donne un aperçu :

Vitesse de Lecture/Ecriture DVD	Taux de transfert (Mo/s)	Vitesse de Lecture/Ecriture CD-R(W)
1x	1.32	9x
2x	2.64	18x
4x	5.28	36x
8x	10.57	72x

Comme on peut le voir dans le tableau, les taux de transfert sont relativement hauts, et vous devrez vérifier que votre bus (SCSI, (E)IDE/ATAPI, USB) est assez performant pour les supporter. Spécifiquement, les vieilles interfaces USB 1.0 et 1.1 travaillent à des taux plus bas, avec des vitesses de transfert de 1.5 Mbit/s et 12 Mbit/s, respectivement. Cela signifie que l'USB 1.0 a une vitesse maximum de 178.8 kByte/s et USB 1.1 de 1.43 MB/s. USB 2.0 est plus rapide : 480 Mbit/s ou 57.2 MB/s. En général, les vitesses des bus SCSI et (E)IDE/ATAPI sont correctes.

Graver les DVDs

Basiquement, le processus est très similaire à celui d'écrire des CD- R(W)s. Le logiciel, cependant, diffère. A ce moment, la meilleure option est **growisofs** depuis le paquetage `sysutils/dvd+rw-tools`. Cet utilitaire écrit une image ISO 9660 sur le média DVD. Tous les formats DVD sont supportés par `dvd+rw-tools`.

Dans le cas où vous voudriez plus d'informations sur le média présent dans votre graveur de DVD (par exemple si vous avez perdu le texte d'information dans la boîte du cd ou êtes simplement désorganisé comme moi), vous pouvez utiliser l'utilitaire **dvd+rw-mediainfo**. Il y a deux options pour écrire le DVD:

- Pre-masteuriser une image ISO 9660 depuis vos données, stockant l'image sur votre disque dur; puis l'écrire sur le DVD.
- Ecrire une image ISO 9660 de vos données directement sur le DVD.

J'ai créé une image ISO 9660 pré-masteurisée depuis les modules CVS d'OpenBSD (`src`, `xenocara`, `ports` et `www`) présents dans le dossier `/cvs` de mon disque. J'ai utilisé la commande suivante, qui ressemble beaucoup à celle que j'ai utilisé pour créer l'image CD-ROM précédente.

```
$ mkhybrid -r -o cvs.iso /cvs
```

Si vous le souhaitez, testez le système de fichiers ISO 9660 en [montant l'image](#). Pour écrire cette image (environ 2Go) sur un disque DVD, on peut utiliser :

```
# growisofs -dvd-compat -Z /dev/rcd2c=cvs.iso
Executing 'builtin_dd if=cvs.iso of=/dev/rcd2c obs=32k
seek=0'
/dev/rcd2c: pre-formatting blank DVD+RW...
/dev/rcd2c: "Current Write Speed" is 4.1x1385KBps.
23822336/1545832448 ( 1.5%) @3.9x, remaining 5:19
42172416/1545832448 ( 2.7%) @3.9x, remaining 5:20
60522496/1545832448 ( 3.9%) @3.9x, remaining 4:54
...
1504706560/1545832448 (97.3%) @3.9x, remaining 0:07
1523318784/1545832448 (98.5%) @3.9x, remaining 0:04
1541898240/1545832448 (99.7%) @3.9x, remaining 0:00
/dev/rcd2c: flushing cache
/dev/rcd2c: writing lead-out
/dev/rcd2c: reloading tray
```

L'option `-Z` demande à `growisofs` de graver une session initiale sur le périphérique, qui dans ce cas est mon graveur de DVD, attaché à `cd2`. L'option `-dvd-compat` finalise le disque, ce qui signifie

qu'aucune nouvelle session ne peut y être rajoutée. Cela devrait fournir une meilleure compatibilité avec les lecteurs DVD et quelques vieilles unités DVD-ROM.

Remarquez comment growisofs indique la vitesse d'écriture, dans notre cas 3.9x, ce qui est ce que l'on peut attendre en combinant les vitesses du média et du graveur, comme indiqué par le dvd+rw-mediainfo.

Si vous ne disposez pas de suffisamment de place pour stocker une image de ISO 9660 four un DVD, vous pouvez écrire vos données directement sur le DVD. Essayons simplement une simulation de création du système de fichiers.

```
# growisofs -dry-run -Z /dev/rcd2c -R /cvs
```

Si cela fonctionne, retirez simplement l'option -dry-run et commencez l'écriture du DVD.

```
# growisofs -Z /dev/rcd2c -R /cvs
```

Il est aussi possible d'ajouter des données à un DVD existant, en utilisant l'option -M, qui fond une nouvelle session dans une session existante :

```
# growisofs -M /dev/rcd2c -R /mydata
```

Pour plus d'informations à propos de growisofs, consultez sa page de manuel.

Lorsque vous avez terminé d'écrire le DVD, montez le et vérifiez que tout ce que vous souhaitez présent y soit effectivement.

Pourquoi je n'obtiens pas la vitesse que je veux ?

Au lieu de la vitesse d'écriture précédente, vous pourriez voir quelque chose comme ceci :

```
4784128/1545832448 ( 0.3%) @0.7x, remaining 26:50
7929856/1545832448 ( 0.5%) @0.7x, remaining 29:05
14123008/1545832448 ( 0.9%) @0.7x, remaining 27:06
...
```

ce qui est bien plus lent. Cela signifie qu'il ne transite pas assez d'informations sur l'un des bus que votre DVD utilise. Dans l'exemple précédent, le graveur de DVD USB fut installé sur une machine sur laquelle le driver [ehci\(4\)](#), utilisé par les contrôleurs USB 2.0, ne s'est pas initialisé correctement. Comme d'habitude, vous êtes invités à envoyer des patches et résultats de vos tests. Le graveur de DVD fut ralenti par l'interface USB 1.1 plus lente, ce qui a réduit la vitesse de gravure. L'USB 1.1 est limité à 12 Mbit/s,

ce qui correspond à 1.43 MB/s ou 1.08x en terme de vitesse DVD. Le graveur de DVD est descendu à une plus basse vitesse de gravure pour diminuer le risque de buffer overrun (défaut de données à écrire).

13.11 - Je voudrais mes fichiers multimédia au format FOO.

Conversion entre différents formats audios

Admettons que vous souhaitiez convertir le son enregistré avec [FAQ 13 - Enregistrer des séquences audio](#). Cet enregistrement a été enregistré en format brut. Il serait utile de le convertir car le format brut ("raw") ne contient pas d'en-tête et les paramètres devront être spécifiés à chaque utilisation du fichier.

Un bon outil de conversion de format est `audio/sox`, disponible au travers des ports et des paquetages. `sox` supporte les formats AIFF, AU, MP3, Ogg Vorbis, RIFF WAV et raw, ainsi que certains autres plus exotiques. Ci-dessous vous trouverez un exemple de conversion de l'enregistrement vers le format RIFF WAV.

```
$ sox -U -c 1 -r 8000 -b myvoice.raw myvoice.wav
```

Notez que les paramètres spécifiés correspondent aux paramètres d'enregistrement spécifiés avant l'enregistrement. Cela n'est qu'un exemple. De nombreux autres programmes et bibliothèques peuvent être utilisés en conversion audio.

Note : Il n'est pas recommandé de convertir entre différents formats de compression dits destructeurs ("lossy"). Par exemple, les codecs MP3 et Vorbis suppriment certaines informations du fichier audio original. Ainsi, lorsque vous convertissez un fichier MP3 en Ogg Vorbis, le résultat final sonnera certainement moins bon que l'original.

Conversion entre différents formats vidéos

Il est important de faire une distinction entre

- le format du conteneur - par exemple, MP4, OGG, MPEG, MOV, AVI, ASF.
- le codec vidéo - par exemple, MPEG-1, MPEG-2, codecs compatibles MPEG-4 (comme Xvid et DivX), FFmpeg, WMV, ... - lisez [l'article Wikipedia sur les codecs vidéos](#) pour en savoir plus.

Pour le moment, le support des conteneurs MPEG et AVI est le plus abouti.

Il existe deux utilitaires populaires, `multimedia/transcode` et `mencoder` (qui fait partie de `x11/ffmpeg`). Ils utilisent ou peuvent utiliser la bibliothèque `libavcodec` disponible dans le port `graphics/ffmpeg` et qui génère un format de sortie de bonne qualité. Vous pouvez bien sûr utiliser

ffmpeg directement. Il devrait aussi être possible d'utiliser l'encodeur XviD présent dans `multimedia/xvidcore`.

La documentation, qui vient avec ces paquetages sous la forme de pages de manuel ou de documents HTML dans `/usr/local/share/doc`, contient de nombreux exemples, c'est pourquoi il est HAUTEMENT recommandé de lire ces documents.

13.12 - Est-il possible de lire des fichiers en streaming sous OpenBSD ?

Oui, c'est possible. De nombreux streams audio et vidéos fonctionneront sans soucis sur un nombre limité de plates-formes. Certains, quant à eux, ne fonctionneront pas.

Ce document ne représente pas une réponse exhaustive sur la manière de faire fonctionner tous les formats de streaming possibles sur n'importe quelle architecture. Tout d'abord, vous devriez vous renseigner sur la technologie de streaming. Quoiqu'un peu daté, le [chapitre à propos du streaming](#) tiré du livre "Designing Web Audio" aux éditions O'Reilly représente un bon début.

La première chose à savoir est qu'il existe plusieurs protocoles de streaming. Le **protocole** de streaming définit la façon dont les flux seront envoyés à travers le réseau. Ils ont été développés afin de garantir une transmission audio/vidéo efficace en temps réel par internet. En bref, le protocole de streaming est un protocole applicatif (niveau 7) capable d'utiliser les protocoles de transport UDP ou TCP (niveau 4). Le protocole de transport UDP ("User Datagram Protocol") convient parfaitement pour ce type d'application puisqu'il n'effectue aucune retransmission de paquets ou autre type de charge réseau. Un certain nombre de protocoles spécialisés mais propriétaires ont été développés, par exemple Microsoft Media Services (MMS - services média de Microsoft) et le Real Time Streaming Protocol (RTSP - protocole de transmission de flux en temps réel). Comme nous le verrons, HTTP (qui utilise TCP) est également parfois utilisé, même s'il ne permet pas de transmettre des flux à une vitesse régulière comme UDP, RTSP et MMS.

Ensuite, il y a le **format** de streaming qui représente la façon dont les données audio/vidéo ont été organisées et peuvent être lues. Les formats de streaming les plus utilisés sont le MP3, Real Audio (RA, RM) et le Windows Media (ASF) qui sont tous des technologies propriétaires. Parfois, vous rencontrerez certains streams au format Ogg Vorbis.

Pour l'exemple, j'expliquerai, en quelques étapes, comment j'arrive à écouter [Radio 1](#), une des stations radios nationales belges. Sous OpenBSD, les plugins (modules externes) de navigateurs ne sont pas disponibles, ce qui signifie que la mise en place se résume rarement en un "clic et lecture".

- Déterminer le protocole de streaming et le format.
Ceci est généralement indiqué sur le site web sur lequel vous accédez au stream. Dans ce cas, je

clic simplement sur le lien "Listen live" (écoutez en direct) à partir du site principal qui me répond que mon système d'exploitation n'est pas supporté. Ils sont suffisamment polis pour me signifier que je peux également écouter leur stream MP3 sans utiliser leur lecteur Flash qui nécessite un plugin. Ensuite une liste de liens vers les stations radios nationales apparaît, ce qui me permet de passer à la suite. Notez que j'ai dû utiliser un navigateur compatible JavaScript afin d'arriver ici.

- Déterminer l'URL précise.

De nombreux sites internet lient vers un conteneur (du type M3U, ASX, RAM) qui contient la localisation effective du flux. Enregistrez simplement ce fichier et récupérez l'URL à partir de celui-ci. Dans mon exemple, il s'agit de :

```
$ ftp http://internetradio.vrt.be/dab/hoeluisteren/pc/
help/gebruiksvoorwaarden/stream_11.m3U
$ cat stream_11.m3U
http://mp3.streampower.be/radiol-mid.mp3
http://mp3.streampower.be/radiol-low.mp3
http://mp3.streampower.be/radiol-high.mp3
```

Visiblement, je peux même choisir entre différentes qualités de flux ("low, medium et high"). D'autres sites peuvent contenir du code JavaScript pour créer l'URL. Dans ce cas, le meilleur conseil serait : parcourez le source HTML ainsi que les scripts auxquels il se réfère. Il existe une bonne chance que vous soyez en mesure de reconstruire l'URL à partir de ces données.

- Pour lire ces flux, votre meilleure chance est probablement `x11/mplayer`, disponible en tant que paquetage et dans l'arbre des ports. Il supporte la plupart des protocoles et formats de streaming et fonctionne sur les plates-formes amd64, i386, powerpc et sparc64. Mais il existe des alternatives : **ogg123** dans `audio/vorbis-tools` (pour les streams Ogg Vorbis), `audio/mpg123` et `audio/mpg321` (pour les streams MP3), `XMMS` dans `audio/xmms` et le client `Videolan` dans `x11/vlc`. Continuons l'exemple :

```
$ mplayer http://mp3.streampower.be/radiol-mid.mp3
```

- Si vous le souhaitez, vous pouvez faciliter cela en ajoutant un alias dans votre fichier `.profile` :

```
alias radiol='mplayer http://mp3.streampower.be/radiol-
mid.mp3'
```

Les flux Windows Media (ASF) fonctionneront la plupart du temps bien qu'ils puissent contenir des données aux formats supportés uniquement grâce au port `graphics/win32-codecs` qui ne tourne que sous i386 ('`pkg_info win32-codecs`' vous donnera la liste des codecs supportés). Certains flux Real Audio peuvent fonctionner sous i386 en utilisant **mplayer** en conjonction avec les ports `graphics/win32-codecs` et `emulators/fedora/base` (lisez [ce fil](#) de la liste de discussion des ports).

13.13 - Puis-je utiliser un plugin Java avec mon navigateur ? (spécifique à i386 & amd64)

Le plugin Java fait partie du kit de développement Java ("Java Development Toolkit" - JDK). Pour des raisons de licence, OpenBSD ne peut pas distribuer de paquetages binaires du JDK. Cela signifie que vous devrez le compiler à partir des ports. De plus amples informations sur la compilation du JDK sont disponibles dans [FAQ 8 - Langages de programmation](#). Une fois le paquet créé, vous pouvez installer le JDK complet ou simplement l'environnement d'exécution Java ("Java Runtime Environment" - JRE) qui se trouve dans un sous-paquetage et contient le plugin web.

A la fin de l'installation, des instructions seront affichées afin d'utiliser le plugin Java avec Firefox ou un navigateur basé sur Seamonkey. Créez le lien comme expliqué et vous devriez voir le plugin Java lorsque vous entrerez "about:plugins" dans la barre d'adresse.

Pour le navigateur web Konqueror de KDE, le binaire java doit être dans votre PATH, ou son chemin absolu doit être configuré du menu Settings -> Configure Konqueror -> Java & JavaScript. Par défaut le binaire java se trouve dans `/usr/local/jre-version/bin/` ou `/usr/local/jdk-version/bin/`, cela dépend si vous avez installé le JRE ou le JDK.

Remarque : Le support Java a été testé uniquement avec les navigateurs Firefox, Seamonkey et Konqueror. Si il fonctionne bien avec d'autres navigateurs merci de nous le signaler.

13.14 - Puis-je utiliser un plugin Flash avec mon navigateur ? (spécifique à i386)

Le plugin Flash est distribué par Adobe sous forme binaire uniquement. Adobe ne fournit pas de plugin natif pour OpenBSD mais il existe un plugin pour Linux qui peut être utilisé sous émulation. Ce plugin n'est disponible que pour les plates-formes i386.

Avant de poursuivre, vous devriez vous renseigner sur l'émulation Linux dans la page de manuel [compat_linux\(8\)](#) ainsi que [FAQ 9 - Exécution des binaires Linux sous OpenBSD](#).

Si vous avez assimilé ces documents mais n'avez pas encore installé les fichiers nécessaires, ajoutez simplement le paquetage fedora. En admettant que vous ayez correctement défini la variable `PKG_PATH` (voyez [FAQ 15](#)),

```
# pkg_add -i fedora_base
```

La première commande définira automatiquement `kern.emul.linux=1`, mais pas de façon

permanente. Si vous avez besoin de l'émulation Linux en permanence, vous devez le spécifier dans `/etc/sysctl.conf`, comme expliqué dans [FAQ 9 - Exécution des binaires Linux sous OpenBSD](#).

Une autre chose que vous devriez savoir est que les bibliothèques partagées ainsi que les modules Linux ne peuvent pas être utilisés avec des exécutables OpenBSD, ce qui signifie que vous aurez également besoin d'un navigateur Linux.

Un candidat serait le navigateur [Opera](#), disponible dans l'arbre des ports. OpenBSD ne distribue pas de paquetage car la licence d'Opera n'est pas claire à ce sujet. Cependant, l'installation ne devrait pas prendre de temps puisqu'il est distribué sous forme binaire par Opera Software. Après cela vous pouvez facilement installer le plugin Flash en utilisant les ports.

```
# cd /usr/ports/www/opera
# make install
# cd /usr/ports/www/opera-flashplugin
# make install
```

Note : La dernière étape devrait suffire en elle-même car le système des ports devrait installer les dépendances automatiquement. Cependant, dans un souci de clarté nous séparons le processus en plusieurs étapes afin d'en faciliter la compréhension.

Si vous avez suivi ce guide, le plugin Flash devrait être listé lorsque vous tapez "about:plugins" dans la barre d'adresse.

[\[Index de la FAQ\]](#) [\[Section 12 - Questions Spécifiques Aux Plates-Formes Et Au Matériel\]](#) [\[Section 14 - Configuration des disques\]](#)



www@openbsd.org

\$OpenBSD: faq13.html,v 1.53 2010/07/09 10:57:21 ajacoutot Exp \$

14 - Configuration des disques

Table des matières

- [14.1 - Utilisation de disklabel\(8\) sous OpenBSD](#)
 - [14.2 - Utilisation de fdisk\(8\) sous OpenBSD](#)
 - [14.3 - Ajout de nouveaux disques sous OpenBSD](#)
 - [14.4 - Comment gérer l'espace de pagination](#)
 - [14.5 - Soft Updates](#)
 - [14.6 - Comment se déroule le processus de démarrage d'OpenBSD/i386 ?](#)
 - [14.7 - Quels sont les problèmes liés aux disques de grande capacité sous OpenBSD ?](#)
 - [14.8 - Installation des blocs de démarrage \("Bootblocks"\) - spécifique i386/amd64](#)
 - [14.9 - Se préparer au désastre : faire une sauvegarde vers une bande et effectuer une restauration](#)
 - [14.10 - Montage des images disque sous OpenBSD](#)
 - [14.11 - A l'aide ! J'ai des erreurs avec IDE DMA !](#)
 - [14.13 - Options RAID avec OpenBSD](#)
 - [14.14 - Pourquoi df \(1 \) me dit que j'ai plus de 100% d'espace disque utilisé ?](#)
 - [14.15 - Récupération de partitions après une suppression du disklabel](#)
 - [14.16 - Est-il possible d'accéder aux données présentes sur des systèmes de fichiers autres que FFS ?](#)
 - [14.16.1 - Les partitions n'apparaissent pas dans mon disklabel ! Que dois-je faire ?](#)
 - [14.17 - Est-il possible d'utiliser un périphérique de masse \('flash memory device'\) sous OpenBSD ?](#)
 - [14.17.1 - Mémoire Flash comme périphérique de stockage portable](#)
 - [14.17.2 - Mémoires Flash comme disque de démarrage](#)
 - [14.17.3 - Comment puis-je créer un périphérique de démarrage USB "Live" ?](#)
 - [14.18 - Optimiser les performances des disques durs](#)
 - [14.19 - Pourquoi nous n'utilisons pas de Montage Asynchrone ?](#)
-

14.1 - Utilisation de disklabel(8) sous OpenBSD

Qu'est-ce que disklabel(8) ?

Pour commencer, lisez le manuel de [disklabel\(8\)](#).

La façon de configurer les disques sous OpenBSD diffère légèrement selon les plates-formes. Sous [i386](#), [amd64](#), [macppc](#), [zaurus](#) et [armish](#), la configuration se déroule en deux étapes. Tout d'abord, la tranche contenant OpenBSD sur le disque est définie en utilisant `fdisk(8)` et ensuite divisée en partitions OpenBSD avec `disklabel(8)`.

Toutes les autres plates-formes utilisent `disklabel(8)` pour la gestion complète des partitions OpenBSD. Les plates-formes qui utilisent aussi `fdisk(8)` placent toutes les partitions `disklabel(8)` dans une tranche `fdisk`.

Les labels contiennent certaines informations sur votre disque, comme sa géométrie ou les systèmes de fichiers présents sur celui-ci. Ils contiennent également des données sur le disque lui-même comme la vitesse de rotation, l'ordonnancement etc, présentent pour des raisons historiques et souvent incorrectes. Il est inutile de s'en inquiéter. Le `disklabel` est utilisé par le programme de bootstrap pour accéder aux disques et connaître le type de systèmes de fichiers présents. Vous pouvez accéder à de plus amples informations sur `disklabel` en lisant le manuel [disklabel\(5\)](#).

Sur certaines plates-formes, l'utilisation de `disklabel` permet de passer outre les limitations de partitionnement liées aux types d'architectures. Par exemple, sur `i386`, vous pouvez avoir 4 partitions primaires mais avec [disklabel\(8\)](#), vous utilisez une de ces partitions 'primaires' pour stocker *toutes* vos partitions OpenBSD (par exemple, 'swap', '/', '/usr', '/var', etc...) et il vous reste encore 3 partitions disponibles pour d'autres systèmes d'exploitation !

disklabel(8) à l'installation d'OpenBSD

Une partie importante de l'installation d'OpenBSD est la création initiale des labels. A l'installation, vous utiliserez `disklabel(8)` afin de créer les différentes partitions. Vous pourrez définir vos points de montage à partir de `disklabel(8)` et le changer plus tard pendant l'installation ou même après celle-ci.

Il n'existe pas de "bonne" façon de créer un label disque mais il en existe beaucoup de mauvaises. Avant de créer un label sur votre disque, lisez [ce chapitre](#) sur le partitionnement.

Pour un exemple d'utilisation de `disklabel(8)` pendant l'installation, référez-vous à [la configuration des disques](#) du [guide d'installation](#).

Utilisations de disklabel(8) après l'installation

Après l'installation, un des usages les plus courants de `disklabel(8)` est de vérifier la disposition de votre disque. La commande suivante affichera le `disklabel` courant sans le modifier :

```
# disklabel wd0 <-- Selon le disque que vous souhaitez
vérifier
# Inside MBR partition 3: type A6 start 63 size 29880837

# /dev/rwd0c:
type: ESDI
disk: ESDI/IDE disk
```

```

label: Maxtor 51536H2
flags:
bytes/sector: 512
sectors/track: 63
tracks/cylinder: 16
sectors/cylinder: 1008
cylinders: 16383
total sectors: 29888820
rpm: 3600
interleave: 1
trackskew: 0
cylinderskew: 0
headswitch: 0          # microseconds
track-to-track seek: 0 # microseconds
drivedata: 0

```

16 partitions:

#	size	offset	fstype	[fsize	bsize	cpg]	
a:	614817	63	4.2BSD	2048	16384	328	#
Cyl	0*-	609					
b:	409248	614880	swap				# Cyl
610 -	1015						
c:	29888820	0	unused	0	0		#
Cyl	0 -	29651*					
d:	6291936	1024128	4.2BSD	2048	16384	328	# Cyl
1016 -	7257						
e:	409248	7316064	4.2BSD	2048	16384	328	# Cyl
7258 -	7663						
f:	1024128	9822960	4.2BSD	2048	16384	328	# Cyl
9745 -	10760						
h:	2097648	7725312	4.2BSD	2048	16384	328	# Cyl
7664 -	9744						

Notez que, pour le moment, ce disque n'a qu'une partie de son espace disponible allouée. Disklabel offre deux modes d'édition différents, un mode de commandes (utiliser à l'installation d'OpenBSD) et un éditeur complet tel que [vi\(1\)](#). Le mode de commandes peut paraître plus simple puisqu'il vous guide à travers les différentes étapes et propose une aide intégrée, mais le mode éditeur possède également une véritable utilité.

Ajoutons une partition au système précédent.

Attention : chaque fois que vous touchez à votre disklabel, vous mettez en danger toutes les données présentes sur le disque. Soyez sûr que vos données soient sauvegardées avant d'éditer un disklabel existant !

Nous utiliserons le mode de commandes intégré qui est lancé avec l'option "-E" de disklabel(8)

```

# disklabel -E wd0
...
> a k
offset: [10847088]
size: [19033812] 2g
Rounding to nearest cylinder: 4194288
FS type: [4.2BSD]
> p m
device: /dev/rwd0c
type: ESDI
disk: ESDI/IDE disk
label: Maxtor 51536H2
bytes/sector: 512
sectors/track: 63
tracks/cylinder: 16
sectors/cylinder: 1008
cylinders: 16383
total bytes: 14594.2M
free bytes: 7245.9M
rpm: 3600

16 partitions:
#          size          offset  fstype  [fsize  bsize  cpg]
  a:         300.2M         0.0M   4.2BSD   2048 16384  328 #
Cyl      0*-    609
  b:         199.8M        300.2M    swap                    # Cyl
610 - 1015
  c:        14594.2M         0.0M  unused           0     0     #
Cyl      0 - 29651*
  d:        3072.2M        500.1M   4.2BSD   2048 16384  328 # Cyl
1016 - 7257
  e:         199.8M       3572.3M   4.2BSD   2048 16384  328 # Cyl
7258 - 7663
  f:         500.1M       4796.4M   4.2BSD   2048 16384  328 # Cyl
9745 - 10760
  h:         1024.2M       3772.1M   4.2BSD   2048 16384  328 # Cyl
7664 - 9744
  k:         2048.0M       5296.4M   4.2BSD   2048 16384   16 # Cyl
10761 - 14921
> q
Write new label?: [y]

```

Dans ce cas, `disklabel(8)` a été capable de calculer un bon point de départ pour la partition. La plupart du temps ce sera le cas mais si vous avez des "trous" dans votre `disklabel` (ex. vous avez supprimé votre partition ou vous aimez simplement vous compliquer la vie) il vous faudra prendre le temps de calculer un offset correct. Notez que même si `disklabel(8)` effectue certaines vérifications il est très possible de faire de grosses erreurs. Faites attention et comprenez bien les valeurs que vous utilisez.

Sur la plupart des plates-formes OpenBSD, il y a 16 partitions disklabel disponibles, notées de "a" à "p" (certains systèmes "spécialisés" n'en ont que huit). Chaque disklabel doit avoir une partition 'c' avec un "fstype" défini à "unused" qui couvre l'ensemble du disque physique. Si ce n'est pas le cas de votre disklabel, il doit être fixé, l'option "D" (plus bas) peut être utilisée. Ne tentez jamais d'utiliser la partition "c" pour autre chose que l'accès en mode raw des secteurs du disque et n'essayez pas de créer un système de fichiers sur "c". Sur le périphérique de démarrage et seulement sur celui-ci, "a" est réservé pour la partition racine et "b" pour le swap. Tous les autres périphériques peuvent utiliser l'ensemble des quinze partitions en dehors de 'c' pour leur système de fichiers.

Trucs et astuces sur disklabel

- **Obtenir de l'aide** : en mode commandes, taper "?" affichera une liste des commandes disponibles. "M" affichera la page manuel de disklabel(8).
- **Valeurs par défaut** : dans certains cas, vous pourriez avoir besoin de commencer avec disklabel par défaut. La commande "D" écrasera l'ancien disklabel comme s'il n'avait jamais existé pour le remplacer par un nouveau.
- **Dupliquer un disklabel** : dans certains cas, vous pourriez avoir besoin de dupliquer le partitionnement d'un disque sur un autre mais par précisément (par exemple, vous voudriez avoir les mêmes partitions mais sur des disques de taille différente). Utilisez l'option '-e' (mode éditeur en plein écran) de disklabel(8) pour enregistrer les partitions du disque "source", copiez-les sur le nouveau disque, supprimez la partition 'c' du disque "source", enregistrez et vous aurez copié la disposition des partitions sur le sur le nouveau disque sans altérer ses paramètres de base.
- (sparc/sparc64) **ne placez pas la partition swap au début du disque.**
- (i386, amd64) **laissez la première piste ("track") libre** : sur certaines plates-formes, vous devriez laisser la première piste logique inutilisée dans disklabel(8) et fdisk(8). Cette recommandation est parfois injustement précisée par "démarez les partitions au secteur 63", ce qui n'est SEULEMENT valable que si elle correspond bien à la taille d'une piste sur votre matériel. N'assumez en rien que cela soit vrai, disklabel vous dira quel est le nombre de secteurs par piste. De nombreuses autres plates-formes s'attendent à voir les partitions OpenBSD démarrer au secteur 0.
- **Disques sans disklabel** : si un périphérique ne possède pas de disklabel OpenBSD mais a déjà un autre système de fichiers (par exemple, un disque avec un système de fichiers FAT32 déjà existant), OpenBSD va en "créer" un en mémoire qui posera les bases d'un disklabel à enregistrer sur le disque. Cependant, si un disklabel OpenBSD est créé et enregistré sur le disque et qu'un système de fichiers non OpenBSD est ajouté par la suite, le disklabel ne sera pas automatiquement mis à jour. Vous devrez l'éditer vous-même si vous souhaitez que OpenBSD puisse y accéder. Plus d'informations sont disponibles [plus bas](#).
- **"q" ou "x"** : Pour des raisons historiques, en mode de commandes, "q" enregistre les changements et quitte le programme alors que "x" quitte sans sauvegarder. C'est l'opposé de ce que beaucoup de gens peuvent voir sous d'autres environnements. disklabel(8) demandera confirmation avant d'enregistrer les changements mais n'affichera aucune alerte si "x" est utilisé.

14.2 - Utilisation de fdisk(8) sous OpenBSD

Avant de commencer, lisez le manuel de [fdisk\(8\)](#).

fdisk(8) est utilisé sur certaines plates-formes (i386, amd64, macppc, zaurus et armich) pour créer une partition reconnue par la ROM de démarrage du système dans laquelle les partitions disklabel d'OpenBSD peuvent être

inscrites. Les autres plates-formes n'ont pas besoin d'utiliser fdisk(8). fdisk(8) peut également être utilisé pour la manipulation du MBR ("Master boot record") pouvant avoir un impact sur tous les systèmes présents sur la machine. A l'inverse de certains programmes dont les fonctionnalités sont proches de celles de fdisk, sous OpenBSD cette commande assume que vous savez ce que vous voulez faire et en règle générale n'interviendra pas dans vos décisions, ce qui en fait un outil très puissant. Il vous laissera également faire des choses que vous n'aviez pas forcément prévues et doit donc être utilisé avec précaution.

Normalement, seule une partition fdisk OpenBSD sera inscrite sur le disque. Cette partition sera divisée par [disklabel](#) en différentes partitions systèmes de fichiers OpenBSD.

Pour simplement voir votre table de partition avec fdisk, utilisez :

```
# fdisk sd0
```

Ce qui vous donnera une sortie similaire à celle-ci :

```
Disk: sd0          geometry: 553/255/63 [8883945 Sectors]
Offset: 0         Signature: 0xAA55
  #: id   Starting      Ending      LBA Info:
      C   H   S -     C   H   S [   start:      size   ]
-----
*0: A6    3    0   1 -   552 254 63 [   48195:      8835750 ]
OpenBSD
 1: 12    0    1   1 -     2 254 63 [     63:         48132 ]
Compaq Diag.
 2: 00    0    0   0 -     0   0   0 [     0:           0 ]
unused
 3: 00    0    0   0 -     0   0   0 [     0:           0 ]
unused
```

Dans cet exemple, nous voyons la sortie de fdisk concernant le premier lecteur SCSI. Nous pouvons voir les partitions OpenBSD (A6). Le * indique que la partition OpenBSD est amorçable.

Dans l'exemple précédent, nous avons simplement accédé à une information. Qu'en est-il si nous souhaitons éditer notre table de partition ? Et bien, pour ce faire, nous devons utiliser l'option -e. Ceci fera apparaître une invite de commandes pour interagir avec fdisk.

```
# fdisk -e wd0
Enter 'help' for information
fdisk: 1> help
      help          Command help list
      manual        Show entire OpenBSD man page for fdisk
      reinit        Re-initialize loaded MBR (to defaults)
      setpid        Set the identifier of a given table entry
      disk          Edit current drive stats
```

```

edit          Edit given table entry
flag         Flag given table entry as bootable
update       Update machine code in loaded MBR
select       Select extended partition table entry MBR
swap         Swap two partition entries
print        Print loaded MBR partition table
write        Write loaded MBR to disk
exit         Exit edit of current MBR, without saving
changes
quit         Quit edit of current MBR, saving current
changes
abort        Abort program without saving current
changes
fdisk: 1>

```

Voici un aperçu des commandes disponibles lorsque vous utilisez l'option **-e**.

- **help** Affiche la liste des commandes utilisables par fdisk en mode d'édition interactif.
- **reinit** Initialise la copie en mémoire du bloc de démarrage actuellement sélectionné. Pratique pour initialiser une partition OpenBSD sur tout le disque, mettre à jour le code de démarrage et, de manière générale, préparer le système pour OpenBSD (et OpenBSD seulement).
- **disk** Affiche la géométrie actuelle du disque détectée par fdisk. Vous pouvez l'éditer selon vos désirs.
- **setpid** Change l'identifiant de partition de l'entrée sélectionnée de la table de partition.
- **edit** Edite l'entrée sélectionnée de la table dans la copie mémoire du bloc de démarrage actuel. Vous pouvez entrer les valeurs en géométrie BIOS ou en secteurs, offsets et tailles.
- **flag** Rend amorçable la partition sélectionnée. Seule une entrée peut être marquée comme amorçable. Si vous souhaitez démarrer à partir d'une partition étendue, vous devez la marquer comme amorçable. (OpenBSD ne peut démarrer qu'à partir de partitions primaires, mais vous pouvez marquer n'importe quelle partition comme étant amorçable).
- **update** Met à jour le code machine dans la copie mémoire du bloc de démarrage actuellement sélectionné.
- **select** Sélectionne et charge en mémoire le bloc de démarrage désigné par la table de partition étendue dans le bloc de démarrage actuel.
- **swap** Echange deux entrées MBR afin que vous puissiez le réordonner.
- **print** Affiche la copie en mémoire actuelle du bloc de démarrage et de son MBR sur le terminal.
- **write** Ecrit la copie en mémoire actuelle du bloc de démarrage sur le disque. Vous serez amené à confirmer cette action.
- **exit** Quitte le niveau actuel de fdisk, revenant à la copie précédente du bloc de démarrage ou sortant du programme si une telle copie n'existe pas.
- **quit** Quitte le niveau actuel de fdisk, revenant à la copie précédente du bloc de démarrage ou sortant du programme si une telle copie n'existe pas. A l'inverse d'"exit", le bloc de démarrage modifié est inscrit sur le disque.
- **abort** Quitte le programme sans sauvegarder les changements.

trucs et astuces fdisk

- fdisk(8) permet d'éditer les partitions soit en mode secteurs purs, soit en utilisant les Cylindres/Têtes/

Secteurs. Ces deux options sont disponibles pour une raison ; certaines tâches sont plus simples à effectuer d'une façon ou d'une autre. Ne vous limitez pas à l'utilisation d'une seule de ces deux options.

- Un disque vierge aura besoin qu'on y inscrive le code de démarrage dans le MBR avant qu'il ne puisse booter. Pour ce faire, vous pouvez utiliser les options "reinit" ou "update". Si vous ne le faites pas, vous écrirez une table de partitions valide avec fdisk mais vous ne pourrez pas démarrer. Si vous ne savez pas d'où vient le code de démarrage présent sur votre disque, il est possible que vous souhaitiez le mettre à jour de toutes façons.
- Si votre système possède une partition de "maintenance" ou de "diagnostique", il est recommandé de la laisser en place ou de l'inscrire avant d'installer OpenBSD.
- Pour des raisons historiques, "q" enregistre les changements et quitte le programme alors que "x" quitte sans sauvegarder. C'est l'opposé de ce que beaucoup de gens peuvent voir sous d'autres environnements. fdisk(8) ne demandera aucune confirmation avant d'enregistrer les changements alors utilisez cette commande avec précaution

14.3 - Ajout de nouveaux disques sous OpenBSD

Une fois votre disque **CORRECTEMENT** installé, vous devez utiliser [fdisk\(8\)](#) (*i386 seulement*) et [disklabel\(8\)](#) afin de le configurer pour OpenBSD.

Pour les utilisateurs i386, commencez avec fdisk. Les autres architectures peuvent ignorer cette étape. Dans l'exemple suivant, nous ajouterons un troisième disque SCSI au système.

```
# fdisk -i sd2
```

Cette commande va initialiser la "véritable" table de partition du disque pour un usage exclusif par OpenBSD. Ensuite, vous devez créer un disklabel. Ceci risque de sembler confus.

```
# disklabel -e sd2
```

```
(screen goes blank, your $EDITOR comes up)
```

```
type: SCSI
```

```
...bla...
```

```
sectors/track: 63
```

```
total sectors: 6185088
```

```
...bla...
```

```
16 partitions:
```

#	size	offset	fstype	[fsize	bsize	cpg]	
c:	6185088	0	unused	0	0		# (Cyl.
0 - 6135)							
d:	1405080	63	4.2BSD	1024	8192	16	# (Cyl.
0*- 1393*)							
e:	4779945	1405143	4.2BSD	1024	8192	16	# (Cyl.
1393*- 6135)							

Tout d'abord, ignorez la partition 'c', celle-ci est toujours présente et est nécessaire au fonctionnement de programmes comme disklabel ! Le type de système de fichiers ("fstype") pour OpenBSD est 4.2BSD. Le nombre

total de secteurs ("total sectors") représente la taille du disque. Admettons qu'il s'agit d'un disque de 3 gigaoctets. 3 gigaoctets en terme de constructeur de disques représente 3000 megaoctets. Divisons 6185088 par 3000 (on utilise [bc\(1\)](#)). Vous obtenez 2061. Ainsi, pour décider des tailles des partitions a, d, e, f, g, ... multipliez simplement X par 2061 afin d'avoir X megaoctets d'espace sur cette partition. L'offset de votre première partition doit être égal à "sectors/track" comme rapporté par disklabel. Pour nous, cette valeur est égale à 63. L'offset de chaque partition suivante résultera de la combinaison de la taille et de l'offset de chacune d'elle (sauf la partition 'c' qui n'intervient pas dans cette équation).

Ou, si vous souhaitez simplement une partition couvrant tout le disque, afin d'offrir un espace de stockage web, un répertoire d'utilisateurs ("home") ou autre chose, ôtez simplement la valeur "sectors/track" à la taille totale du disque. $6185088 - 63 = 6185025$. Votre partition est :

```
d: 6185025      63      4.2BSD      1024  8192      16
```

Si tout ceci vous semble inutilement complexe, vous pouvez simplement utiliser disklabel -E pour entrer dans le même mode de partitionnement dont vous disposez à l'installation ! Dans celui-ci, vous pouvez utiliser "96M" pour spécifier "96 megaoctets" ou, 96G pour 96 gigaoctets.

Voilà qui était dense. Mais tout n'est pas terminé. Vous devez à présent créer un système de fichiers sur ce disque en utilisant [newfs\(8\)](#).

```
# newfs sd2d
```

Ou autre, selon le nom donné à votre disque par le système de nommage d'OpenBSD (regardez la sortie de [dmesg\(8\)](#) pour voir comment OpenBSD a nommé votre disque).

Maintenant, il vous faut décider où vous souhaitez monter cette partition nouvellement créée. Disons que vous vouliez la monter sur /u. Tout d'abord, créez le répertoire /u. Puis montez-le.

```
# mount /dev/sd2d /u
```

Enfin, ajoutez-le à [/etc/fstab\(5\)](#).

```
/dev/sd2d /u ffs rw 1 1
```

Et si vous souhaitiez migrer un répertoire existant comme /usr/local ? Vous devriez monter le nouveau disque sur /mnt et utiliser `cpio - pdum` pour copier /usr/local vers le répertoire /mnt. Editez le fichier [/etc/fstab\(5\)](#) afin de préciser que votre partition /usr/local est maintenant /dev/sda2d (votre nouvelle partition formatée). Exemple :

```
/dev/sd2d /usr/local ffs rw 1 1
```

Redémarrez en mode single user avec `boot -s`, déplacez le répertoire /usr/local vers /usr/local-backup (ou si vous vous sentez en veine, supprimez-le) et créez un répertoire /usr/local vide. Enfin, redémarrez le système et voilà, les fichiers sont présents !

14.4 - Comment gérer l'espace de pagination ?

14.4.1 - A propos de la pagination

Historiquement, toutes sortes de règles ont été expliquées pour guider l'administrateur sur la quantité d'espace de pagination à configurer sur leurs machines. Le problème, bien sur, c'est qu'il y a peu d'applications "normales".

Une non-utilisation manifeste de l'espace de pagination est d'être un lieu où le noyau peut générer une copie du système de base en cas de mode panique pour une analyse ultérieure. Pour que cela fonctionne vous devez avoir une partition de pagination (pas un fichier de pagination) au moins aussi grande que la taille de votre RAM. Par défaut, le système sauvera une copie de ce vidage mémoire dans `/var/crash` au redémarrage, donc si vous désirez faire cela automatiquement, vous devez avoir assez d'espace *libre* dans `/var`. Cependant, vous pouvez aussi démarrer le système en mode mono-utilisateur et utiliser [savecore\(8\)](#) pour faire le vidage ailleurs.

Plusieurs types de systèmes peuvent être configurés de façon appropriée sans aucun espace de pagination. Par exemple, les pare-feu ne doivent pas paginer en fonctionnement normal. Les machines avec un stockage flash ne doivent généralement pas paginer. Si votre pare-feu est basé sur une flash, vous bénéficierez (légèrement) par la non-allocation d'une partition d'espace de pagination, mais dans la plupart des cas, une partition d'espace de pagination ne fera de mal à personne; la plupart des disques possèdent assez d'espace pour en allouer un petit peu à l'espace de pagination.

Il y a toutes sortes d'astuces pour optimiser l'espace de pagination (ou sur le disque, disques séparés, etc...), mais si vous vous trouvez vous-même dans une situation où l'espace de pagination est une solution, vous avez probablement besoin de plus de RAM. En général, la meilleure optimisation pour l'espace de pagination est de ne pas en avoir besoin.

Sur OpenBSD, l'espace de pagination est géré par le programme [swapctl\(8\)](#) qui ajoute, supprime, affiche et donne des priorités sur les fichiers et périphériques d'espace de pagination.

14.4.2 - Paginer sur une partition

Sur OpenBSD, la partition 'b' du disque de démarrage est utilisée par défaut et automatiquement pour l'espace de pagination. Pas de configuration est nécessaire pour mettre cela en place. Si vous ne désirez pas utiliser d'espace de pagination sur le disque de démarrage, ne définissez pas une partition "b". Si vous désirez utiliser l'espace de pagination sur une autre partition ou sur un autre disque, vous devez définir ces partitions dans `/etc/fstab` avec des lignes ressemblant à cela :

```
/dev/sd3b none swap sw 0 0
/dev/sd3d none swap sw 0 0
```

14.4.3 - Paginer dans un fichier

(Note : si vous souhaitez ajouter un espace de pagination dans un fichier parce que vous recevez des erreurs du type "virtual memory exhausted", vous devriez d'abord essayer d'augmenter la limite des processus avec [unlimit](#)

(1) pour csh ou [ulimit\(1\)](#) pour sh)

Quelquefois, votre envie initiale sur la quantité de swap nécessaire est fautive, et vous devez ajouter plus d'espace de pagination, occasionnellement dans l'urgence (comme dans "Geez, at the rate it is burning swap, we'll be wedged in five minutes"). Si vous vous retrouvez dans cette situation, ajouter de l'espace de pagination dans un fichier sur un système de fichier existant peut-être une solution rapide.

Ce fichier ne doit pas résider sur un système de fichiers dont l'option "SoftUpdates" est activée (cette option est désactivée par défaut). Pour commencer, vous pouvez voir la taille totale et utilisée de votre swap actuel en utilisant l'utilitaire [swapctl\(8\)](#). Pour ce faire, lancez la commande :

```
$ swapctl -l
Device      512-blocks      Used      Avail Capacity  Priority
swap_device      65520          8      65512      0%      0
```

Cela vous affiche les différents dispositifs utilisés pour la pagination ainsi que leurs statistiques actuelles. Dans l'exemple précédent, il n'y a qu'un seul dispositif nommé "swap_device". Il s'agit de l'espace disque prédéfini utilisé pour paginer (partition b visible sous disklabel). Comme vous pouvez le constater dans l'exemple précédent, cet espace n'est pas beaucoup utilisé pour le moment. Mais pour les besoins de ce document, nous imaginerons que 32Mo supplémentaires sont nécessaires.

La première chose à faire pour configurer un fichier en tant que dispositif de pagination est de créer ce fichier. Le mieux est d'utiliser la commande [dd\(1\)](#). Voici un exemple montrant la création d'un fichier `/var/swap` d'une taille de 32Mo.

```
$ sudo dd if=/dev/zero of=/var/swap bs=1k count=32768
32768+0 records in
32768+0 records out
33554432 bytes transferred in 20 secs (1677721 bytes/sec)
```

Une fois que cela est fait, nous pouvons activer la pagination vers ce fichier. Pour ce faire, utilisez la commande suivante :

```
$ sudo chmod 600 /var/swap
$ sudo swapctl -a /var/swap
```

A présent, vérifions que ce fichier a bien été ajouté à la liste des dispositifs de pagination disponibles.

```
$ swapctl -l
Device      512-blocks      Used      Avail Capacity  Priority
swap_device      65520          8      65512      0%      0
/var/swap      65536          0      65536      0%      0
Total          131056          8     131048      0%
```

Maintenant que le fichier de pagination est activé, vous devez ajouter une ligne à votre fichier `/etc/fstab` afin que

les changements soient pris en compte au prochain redémarrage. Si cette ligne n'est pas ajoutée, ce dispositif ne sera plus activé au prochain reboot.

```
$ cat /etc/fstab
/dev/wd0a / ffs rw 1 1
/var/swap /var/swap swap sw 0 0
```

14.5 - Soft Updates

Les "Soft Updates" sont basés sur une idée de [Greg Ganger et Yale Patt](#) et ont été développés pour FreeBSD par [Kirk McKusick](#). Les Soft Updates imposent une réorganisation partielle des opérations sur le tampon permettant la suppression du code FFS, des écritures synchrones des entrées. Une augmentation des performances est ainsi réalisée lors des écritures sur disque.

L'activation des Soft Updates est effectuée par une option de montage. Lors du montage d'une partition avec l'utilitaire [mount\(8\)](#), vous pouvez activer les Soft Updates sur celle-ci. Voici l'exemple d'une entrée dans le fichier [/etc/fstab\(5\)](#) permettant de monter la partition *sd0a* avec les Soft Updates activés.

```
/dev/sd0a / ffs rw,softdep 1 1
```

Note aux utilisateur sparc : n'activez pas Soft Updates sur les machines de type sun4 ou sun4c. Ces architectures ne supportent qu'un montant très faible de mémoire noyau et ne peuvent utiliser cette fonctionnalité. En revanche, les machines de type sun4m ne posent pas de problème.

14.6 - Comment se déroule le processus de démarrage d'OpenBSD/i386 ?

Le processus de démarrage d'un système OpenBSD/i386 est loin d'être simple et une bonne compréhension de celui-ci peut être utile afin de diagnostiquer un problème lorsque les choses ne fonctionnent pas. Ce processus comporte quatre étapes clefs :

1. **Master Boot Record (MBR)** : Le "Master Boot Record" (enregistrement de démarrage principal) est le premier secteur (512 octets) du disque. Il contient la table de partition primaire et un petit programme permettant de charger le "Partition Boot Record" (PBR - enregistrement de démarrage des partitions). Notez que dans certains environnements, le terme "MBR" se réfère uniquement à la portion de code comprise dans le premier bloc du disque et non pas au bloc entier (incluant la table de partition). Il est primordial de comprendre le sens d'"initialiser le MBR" ; la terminologie sous OpenBSD sous-entend la réécriture complète du secteur MBR et pas seulement son code, comme cela peut être le cas avec certains systèmes. Vous n'en aurez l'utilité que très rarement. Pour ce faire, utilisez l'option "-u" de la commande `fdisk(8)` ("`fdisk -u wd0`").

Bien qu'un MBR soit inclut avec OpenBSD, vous n'êtes pas obligé de l'utiliser puisque virtuellement, n'importe quel MBR est capable de le faire démarrer. Le MBR se manipule avec l'aide du programme `fdisk(8)` qui est utilisé pour éditer la table de partition mais également pour installer le code MBR sur le disque.

Le MBR d'OpenBSD s'annonce avec le message suivant :

```
Using drive 0, partition 3.
```

montrant le disque ainsi que la partition d'où il s'apprête à charger le PBR. On peut également voir un point (".") en fin de ligne, ce qui signifie que la machine est capable d'utiliser la conversion LBA ("Logical Block Addressing") au démarrage. Si la machine avait été incapable d'une telle conversion, le point aurait été remplacé par un point-virgule (";"), indiquant une conversion CHS ("Cylinder-Head-Sector") :

```
Using drive 0, partition 3;
```

Notez que le point ou le point-virgule en fin de ligne peut servir d'indicateur du "nouveau" MBR OpenBSD introduit à partir de la version 3.5.

2. **Partition Boot Record (PBR)** : Le "Partition Boot Record" (enregistrement de démarrage de partition), aussi appelé PBR ou [biosboot\(8\)](#) (d'après le nom du fichier contenant le code) représente le premier secteur physique de la partition OpenBSD du disque. Le PBR est le "first-stage boot loader" (chargeur de démarrage de niveau un) d'OpenBSD. Il est exécuté par le code MBR et se charge de lancer le "second-stage boot loader" (chargeur de démarrage de niveau deux), [boot\(8\)](#). Comme le MBR, le PBR est constitué d'une toute petite quantité de code et de données d'une taille totale de 512 octets. Ce n'est pas suffisant pour avoir une application reconnaissant un système de fichiers, donc, plutôt que de demander au PBR de localiser `/boot` sur le disque, la localisation de `/boot` accessible par le BIOS est codée physiquement dans le PBR lors de l'installation.

Le PBR est installé par [installboot\(8\)](#), décrit dans [plus en détails dans la suite de ce document](#). Le PBR s'annonce avec le message suivant :

```
Loading...
```

affichant un point pour chaque bloc qu'il essaye de charger. Comme précédemment, le PBR montre s'il utilise une conversion LBA ou CHS pour s'exécuter. Dans le cas de CHS, il affichera un message suivi d'un point- virgule :

```
Loading;...
```

L'ancien `biosboot(8)` (avant la version 3.5) affichait le message `reading boot...`.

3. **Second Stage Boot Loader, /boot** : `/boot` est chargé par le PBR et se charge d'accéder au système de fichiers OpenBSD par le BIOS de la machine afin de localiser et de lancer le noyau. `boot(8)` permet également de passer plusieurs options et informations au noyau.

`boot(8)` est un programme interactif. Après son exécution, il tente de localiser et lire `/etc/boot.conf` si celui-ci existe (ce qui n'est pas le cas dans une installation par défaut) puis lance les commandes qui y sont indiquées. A moins que le fichier `/etc/boot.conf` ne l'en empêche, `boot(8)` offre à l'utilisateur une invite de commandes similaire à celle-ci :

```

probing: pc0 com0 com1 apm mem[636k 190M a20=on]
disk: fd0 hd0+
>> OpenBSD/i386 BOOT 3.02
boot>

```

Par défaut, l'utilisateur a cinq secondes pour entrer une commande avant le démarrage du noyau, `bsd`, à partir de la partition racine du premier disque dur, ce qui représente le comportement par défaut. Le "second- stage boot loader" détecte (et examine) votre matériel grâce au BIOS (puisque le noyau OpenBSD n'est pas encore chargé). Précédemment on peut voir qu'il a détecté :

- o **pc0** - Clavier et affichage vidéo d'un système i386 standard.
- o **com0, com1** - Deux ports série.
- o **apm** - BIOS compatible APM ("Advanced Power Management" - gestion de courant évoluée).
- o **636k 190M** - Le montant de mémoire conventionnelle (sous 1Mo) et étendue (au-dessus d'1Mo) trouvée.
- o **fd0 hd0+** - Les disques accessibles par le BIOS, dans ce cas, un lecteur de disquettes et un disque dur.

Le signe '+' après "hd0" signifie que le BIOS a indiqué à `/boot` que ce disque peut être accessible par LBA. Lors d'une première installation, vous verrez souvent un '*' après un disque dur ; ceci signifie que ce disque ne possède pas de label OpenBSD valide.

4. **Noyau** : `/bsd`: il s'agit de la finalité du processus de démarrage, avoir le noyau OpenBSD chargé en mémoire RAM et fonctionnant correctement. Une fois le noyau exécuté, OpenBSD peut accéder directement au matériel sans passer par le BIOS.

Ainsi, le tout début du processus de démarrage pourrait ressembler à ceci :

```

Using drive 0, partition 3.                <- MBR
Loading....                               <- PBR
probing: pc0 com0 com1 apm mem[636k 190M a20=on] <- /boot
disk: fd0 hd0+
>> OpenBSD/i386 BOOT 3.02
boot>
booting hd0a:/bsd 4464500+838332 [58+204240+181750]=0x56cfd0
entry point at 0x100120

[ using 386464 bytes of bsd ELF symbol table ]
Copyright (c) 1982, 1986, 1989, 1991, 1993      <- Noyau
    The Regents of the University of California. All rights
reserved.
Copyright (c) 1995-2008 OpenBSD. All rights reserved. http://www.
OpenBSD.org

OpenBSD 4.6 (GENERIC.MP) #89: Thu Jul  9 21:32:39 MDT 2009
  deraadt@i386.openbsd.org:/usr/src/sys/arch/i386/compile/GENERIC.
MP
...

```

Ce qui peut mal se passer

- **Bad/invalid/incompatible MBR** : habituellement, un disque déjà utilisé possède un code MBR, mais s'il est neuf ou déplacé à partir d'une plate-forme différente ET que vous ne répondez pas "w" à la question "Use (W)hole disk or (E)dit the MBR?" lors de l'[installation](#), vous pourriez vous retrouver avec un disque sans MBR valide, le rendant ainsi non-bootable même si sa table de partition est correcte.

Vous pouvez installer le MBR OpenBSD sur votre disque dur à l'aide du programme fdisk. Démarrez à l'aide du médium d'installation; choisissez "Shell" pour vous retrouver en ligne de commandes :

```
# fdisk -u wd0
```

Vous pouvez également installer un MBR particulier avec fdisk :

```
# fdisk -u -f /usr/mdec/mbr wd0
```

ce qui installera le fichier /usr/mdec/mbren tant que MBR de votre système. Dans une installation standard d'OpenBSD, ce fichier représente le MBR par défaut codé dans fdisk, mais n'importe quel autre MBR pourrait ici être spécifié.

- **Invalid /boot location installed in PBR** : lorsqu'installboot(8) installe le PBR, il inscrit l'offset et le numéro du bloc de l'inode de /boot dans le celui-ci. Ainsi, supprimer puis remplacer /boot sans relancer [installboot\(8\)](#) empêchera votre système de redémarrer puisque le PBR chargera ce qui se trouve à l'ancien inode spécifié, ce qui ne sera certainement plus le "second-stage boot loader" désiré ! Puisque que /boot est accédé au travers d'appels BIOS, les anciennes versions de PBR étaient dépendantes de la conversion BIOS des disques. Si vous avez altéré la géométrie (à savoir, pris un disque d'un ordinateur utilisant la conversion CHS pour le mettre dans une machine utilisant LBA, ou simplement changé l'option de conversion dans votre BIOS), il *apparaîtra au niveau du BIOS* comme étant déplacé vers une autre location (un bloc numérique différent doit être accédé afin de récupérer les mêmes informations du disque) et vous devrez donc relancer installboot(8) avant que le système ne puisse être redémarré. Le nouveau PBR (OpenBSD 3.5 et plus) est beaucoup plus tolérant dans les changements de conversion.

Le PBR étant très petit, son éventail de messages d'erreur est limité et quelque peu complexe. La plupart des messages ressemblent à :

- **ERR R** - Le BIOS a retourné une erreur en tentant de lire un bloc à partir du disque. Habituellement, ceci est assez explicite : votre disque n'était pas lisible.
- **ERR M** - Un nombre [magic\(5\)](#) invalide a été lu dans l'entête du "second-stage bootloader". Ceci signifie généralement que ce qui a été lu n'était PAS /boot, à savoir qu'installboot(8) n'a pas été correctement exécuté, que le fichier /boot a été altéré ou encore que vous avez excédé la capacité de votre BIOS à lire les [disques de grande capacité](#).

Les autres messages d'erreur sont expliqués en détail dans le manuel de [biosboot\(8\)](#).

Pour plus d'informations sur le processus de démarrage pour i386, référez-vous à :

- [boot_i386\(8\)](#)
- <http://www.ata-atapi.com/hiw.html> - Documents "How it Works" (comment cela fonctionne) de Hale Landis.

14.7 - Quels sont les problèmes liés aux disques de grande capacité sous OpenBSD ?

OpenBSD supporte les systèmes de fichiers FFS et FFS2 (aussi connu comme UFS et UFS2). FFS est le système de fichiers historique de OpenBSD, FFS2 est nouveau depuis la 4.3. Avant de regarder les limites de chaque système, nous avons besoin de regarder à des limites système plus générale.

Bien entendu, les capacités du système de fichiers et celles d'un matériel défini sont deux choses bien différentes. Un nouveau disque dur IDE de 250Go pourrait provoquer des problèmes sur des interfaces trop anciennes (d'une capacité standard de 137Go maximum, mais il fonctionne très bien), certains adaptateurs SCSI très anciens peuvent avoir des problèmes avec des lecteurs modernes et des BIOS un peu trop vieux peuvent se figer lorsqu'ils tombent sur un disque dur de trop grande capacité. De fait, vous devez respecter les possibilités de votre matériel.

Taille des partitions et limitations au niveau de la localisation

Malheureusement, les fonctionnalités du système d'exploitation ne sont accessibles qu'une fois celui-ci chargé en mémoire. Le processus de démarrage devra se contenter de la ROM de lancement du système (bien plus limitée).

Pour cette raison, le fichier /bsd (le noyau) doit se trouver dans l'espace disque adressable par la ROM. Cela signifie que sur certains systèmes i386, la partition racine doit se situer dans les premiers 504Mo, mais les ordinateurs plus récents peuvent avoir des limites de l'ordre de 2Go, 8Go, 32Go, 128Go ou plus. Il est important de noter que de nombreux ordinateurs récents supportant un adressage au delà de 128Go sont en fait limités par le BIOS et ne peuvent démarrer que sur les 128 premiers gigaoctets. Vous pouvez utiliser ces systèmes avec de gros disques durs, mais la partition racine devra être placée dans l'espace supporté par la ROM de démarrage.

Notez qu'il est toujours possible d'installer un disque dur de 40Go sur un vieux 486 et d'y placer OpenBSD sur une seule et énorme partition ; vous pourriez penser avoir réussi à violer la règle précédente. Cependant, un beau jour, celle-ci risque de revenir vous hanter de manière peu plaisante :

- Vous installez tout sur une partition / de 40Go. Cela fonctionne car le système de base et ses fichiers (/bsd inclut) se situent dans les premiers 504Mo.
- Vous utilisez le système et finissez par avoir plus de 504Mo de données dessus.
- Vous mettez à jour, compilez un nouveau noyau et vous copiez votre nouveau /bsd sur l'ancien.
- Vous redémarrez.
- C'est alors que vous recevez un message tel que "ERR M" ou rencontrez des problèmes lors du démarrage.

Pourquoi ? Simplement parce que vous avez "écrasé" l'ancien /bsd par le nouveau, ce fichier n'a pas été écrit au-dessus de l'autre, mais a été relocalisé vers un autre endroit sur le disque, probablement au-delà des 504 premiers Mo supportés par le bios. Le chargeur de démarrage se voit alors dans l'impossibilité d'atteindre /bsd et le système se fige.

Afin de permettre le lancement d'OpenBSD, les chargeurs de démarrage (biosboot(8) et /boot dans le cas d'un i386/amd64) ainsi que le noyau (/bsd) doivent se situer à l'intérieur de l'espace d'adressage supporté par la ROM de boot et par leurs propres capacités. Afin de ne jamais rencontrer de problèmes, la règle est simple :

la partition racine dans son ensemble doit se situer dans l'espace d'adressage supporté par le BIOS (ou la ROM de démarrage).

Certains utilisateurs n'utilisant pas d'i386 peuvent penser qu'ils sont immunisés contre ce problème, cependant, de nombreuses plates-formes possèdent une certaine limite fixée par la ROM de démarrage concernant la taille du disque. Trouver cette limite peut être compliquée.

Et c'est une autre raison pour [partitionner votre disque dur](#), plutôt que d'utiliser une seule grosse partition.

fsck(8) : impératifs de durée et de mémoire

Les autres points à prendre en considération avec les gros systèmes de fichiers sont le temps et la mémoire nécessaires à l'utilisation de [fsck\(8\)](#) après un crash ou une coupure de courant. On ne peut pas créer un système de fichiers de 120Go sur un disque et espérer utiliser fsck(1) avec succès après un crash. Une règle simple à appliquer est de considérer que le système doit posséder au moins 1Mo de mémoire libre pour chaque 1Go d'espace disque afin d'utiliser fsck sans problème. La partition de swap peut être mise à contribution mais avec un coût tellement élevé au niveau des performances que cela en devient généralement inacceptable excepté dans certains cas.

Le temps nécessaire à l'utilisation de fsck peut devenir un problème au fur et à mesure que le système de fichiers grossit, mais vous n'avez besoin d'utiliser fsck que sur les systèmes de fichiers montés. C'est une autre raison de NE pas allouer tout l'espace disque Juste Parce Ce Qu'il Est La. Garder le système de fichier monté RO ou pas monté aide à les garder du besoin d'être fsck(8) après l'arrachage du câble d'alimentation.

N'oubliez pas que si vous avez plusieurs disques sur le système, ils pourraient se retrouver tous à devoir subir un fsck(8) en même temps après un crash et donc demander plus de RAM qu'un disque seul.

Avec le temps avoir un disque avec un système de fichiers plus grand que 1To avec la taille de blocs et fragmentation par défaut, demande à fsck 1Go de RAM pour fonctionner, qui sont les limites d'une application sous OpenBSD. Des fragments et/ou blocs plus grands réduira le nombre d'inodes et permettra des systèmes de fichiers plus grand.

FFS vs. FFS2

Utilisant FFS, OpenBSD supporte un système de fichiers individuel avec $2^{31}-1$, ou 2,147,483,647 secteurs et chaque secteur possède une taille de 512 octets, ce qui est un petit peu moins que 1To. FFS2 est capable de supporter des disques beaucoup plus grands, donc il faudra longtemps avant d'atteindre les nouvelles limites du système de fichiers.

Les noyaux de démarrage/installation *supportent seulement FFS*, pas FFS2, donc les partitions systèmes clés (/ , /usr , /var , /tmp) ne doivent pas être en FFS2 ou de sérieux problèmes de maintenance arriveront (il

n'y a aucune raison pour avoir ces partitions aussi large de toute façon). Pour cette raison, les partitions très grandes doivent être utilisées uniquement pour des partitions "non-système", par exemple `/home`, `/var/www/`, `/bigarray`, etc.

Avant de mettre à jour, vous devez marquer toutes les partitions FFS2 comme "noauto" pour qu'elles ne soient (pas) prises en compte par le noyau d'installation (qui ne supporte pas les partitions FFS2).

Il faut remarquer que tous les contrôleurs et pilotes ne supportent pas les grands disques. Par exemple, [ami\(4\)](#) a une limite de 2To par volume logique. Beaucoup n'ont pas encore été testés, par exemple, à la date de rédaction de ce document, il n'y a pas de disque IDE ou SATA >1To disponibles pour des tests, donc nous ne pouvons pas affirmer que tout fonctionne correctement.

14.8 - Installation des blocs de démarrage ("Bootblocks") - spécifique i386/amd64

Les versions récentes d'OpenBSD (3.5 et plus) possèdent un chargeur de démarrage très robuste qui est beaucoup moins sensible à la géométrie des disques que l'ancien. Cependant, il reste sensible à l'emplacement du fichier [/boot](#) sur le disque. Si vous effectuez une manipulation ayant pour cause de déplacer `boot(8)` vers un autre endroit sur le disque (en fait un nouvel inode), vous "casserez" votre système l'empêchant de démarrer normalement. Pour réparer votre bloc de démarrage afin que vous puissiez booter normalement, insérez une disquette dans votre lecteur (ou utilisez un CD-ROM bootable) et à l'invite de commandes, tapez "b hd0a:/bsd" afin de le forcer à démarrer sur le premier disque (et non sur la disquette). Votre machine devrait se lancer normalement. Vous devez à présent réinstaller le chargeur de premier niveau ([biosboot\(8\)](#)) en se basant sur le nouvel emplacement du fichier `/boot` en utilisant le programme [installboot\(8\)](#).

Notre exemple assume que votre disque de démarrage est `sd0` (mais pour de l'IDE, ce pourrait être `wd0`, etc...) :

```
# cd /usr/mdec; ./installboot /boot biosboot sd0
```

Si une nouvelle version des blocs de démarrage est nécessaire, vous devrez la compiler vous même. Pour ce faire, tapez simplement :

```
# cd /sys/arch/i386/stand/
# make && make install
# cd /usr/mdec; cp ./boot /boot
# ./installboot /boot biosboot sd0 (ou quelque soit le périphérique
désignant votre disque dur)
```

14.9 - Se préparer au désastre : faire une sauvegarde vers une bande et effectuer une restauration

Introduction :

Si vous envisagez d'administrer ce que l'on pourrait appeler un serveur de production, il est préférable d'avoir une sauvegarde au cas où l'un de vos disques durs tomberait en panne.

Cette documentation vous assistera dans l'utilisation des utilitaires standards [dump\(8\)](#)/[restore\(8\)](#) fournis avec OpenBSD. Un autre utilitaire de sauvegarde plus avancé nommé "[Amanda](#)" est également disponible via les [ports](#) afin de sauvegarder plusieurs serveurs vers un unique lecteur de bandes. Dans la plupart des environnements, [dump\(8\)](#)/[restore\(8\)](#) suffit. Cependant, si vous avez besoin de sauvegarder plusieurs machines, il peut être utile de se documenter sur Amanda.

Les périphériques utilisés en exemple dans ce document se réfèrent à une configuration utilisant des disques et lecteurs de bandes SCSI. En environnement de production, les disques SCSI sont recommandés en place des disques IDE pour la façon dont ils gèrent les blocs défectueux. Ceci ne signifie pas pour autant que ce document est sans intérêt pour les possesseurs de disques IDE ou d'autres types de lecteurs de bandes, les noms de vos périphériques changeront légèrement. Par exemple, sd0a deviendra wd0a dans un système à base d'IDE.

Sauvegarder vers une bande :

Sauvegarder sur bande demande de savoir exactement où vos systèmes de fichiers sont montés. Vous pouvez le déterminer en utilisant la commande [mount\(8\)](#) dans un terminal. Vous devriez obtenir une sortie similaire à celle-ci :

```
# mount
/dev/sd0a on / type ffs (local)
/dev/sd0h on /usr type ffs (local)
```

Dans cet exemple, le système de fichiers racine (/) réside physiquement sur sd0a, ce qui désigne la partition a du disque SCSI fixe 0. Le système de fichiers /usr réside sur sd0h, ma partition h du disque SCSI fixe 0.

Un autre exemple de ce qu'une table de montage plus avancée pourrait être :

```
# mount
/dev/sd0a on / type ffs (local)
/dev/sd0d on /var type ffs (local)
/dev/sd0e on /home type ffs (local)
/dev/sd0h on /usr type ffs (local)
```

Dans cet exemple plus compliqué, le système de fichiers racine (/) réside physiquement sur sd0a ; /var sur sd0d ; /home sur sd0e et enfin /usr sur sd0h.

Afin de sauvegarder votre machine, vous aurez besoin de renseigner dump avec le nom exact de chaque partition fixe. Voici un exemple des commandes nécessaires pour sauvegarder la table de montage simple vue plus haut :

```
# /sbin/dump -0au -f /dev/nrst0 /dev/rsd0a
# /sbin/dump -0au -f /dev/nrst0 /dev/rsd0h
# mt -f /dev/rst0 rewind
```

Pour la table de montage plus avancée, vous utiliserez des commandes similaires à :

```
# /sbin/dump -0au -f /dev/nrst0 /dev/rsd0a
# /sbin/dump -0au -f /dev/nrst0 /dev/rsd0d
# /sbin/dump -0au -f /dev/nrst0 /dev/rsd0e
# /sbin/dump -0au -f /dev/nrst0 /dev/rsd0h
# mt -f /dev/rst0 rewind
```

Vous pouvez vous référer à la page de manuel de [dump\(8\)](#) afin de connaître les fonctionnalités de chaque option.

Voici une brève description des paramètres utilisés :

- **0** - Effectue un dump de niveau 0, sauvegarde tout.
- **a** - Tente de détecter automatiquement la taille de la bande.
- **u** - Met à jour le fichier /etc/dumpdates afin de savoir quand la dernière sauvegarde a été effectuée.
- **f** - Lecteur de bande à utiliser (ici /dev/nrst0).

Et finalement, la partition à sauvegarder (/dev/rsd0a, etc...).

La commande [mt\(1\)](#) est utilisée à la fin pour rembobiner la bande. Référez-vous au manuel de mt afin de connaître les options disponibles (comme eject).

Si vous n'êtes pas sûr du nom de périphérique du lecteur, utilisez dmesg pour le localiser. Un lecteur de bande peut, par exemple, apparaître ainsi dans dmesg :

```
st0 at scsibus0 targ 5 lun 0: <ARCHIVE, Python 28388-XXX, 5.28>
```

Vous avez peut-être noté que, lors d'une sauvegarde, le lecteur est accédé par son nom de périphérique "nrst0" au lieu de "st0" qui apparaît dans dmesg. Lorsque vous accédez st0 à la place de nrst0, vous accédez au même lecteur bande mais en mode "raw" et en lui indiquant de ne pas rembobiner la bande à la fin du travail. Pour sauvegarder plusieurs systèmes de fichiers sur une même bande, soyez certains d'utiliser le périphérique de non-rembobinage ; dans le cas contraire (si vous utilisez rst0), la sauvegarde du système de fichiers précédent sera écrasée à la prochaine écriture sur la bande. Vous pouvez trouver une description plus conséquente de plusieurs lecteurs de bande dans la page de manuel de dump.

Si vous souhaitez écrire un petit script appelé "backup", il pourrait ressembler à celui-ci :

```
echo " Starting Full Backup..."
/sbin/dump -0au -f /dev/nrst0 /dev/rsd0a
/sbin/dump -0au -f /dev/nrst0 /dev/rsd0d
/sbin/dump -0au -f /dev/nrst0 /dev/rsd0e
/sbin/dump -0au -f /dev/nrst0 /dev/rsd0h
echo
echo -n " Rewinding Drive, Please wait..."
mt -f /dev/rst0 rewind
echo "Done."
```


echo

Si des sauvegardes nocturnes planifiées sont nécessaires, [cron\(8\)](#) pourra être utilisé pour lancer votre utilitaire de sauvegarde automatiquement.

Il serait également utile de préciser (sur un bout de papier) la taille que devrait avoir chaque système de fichiers. Vous pouvez utiliser "df -h" afin de connaître l'espace actuellement utilisé par chaque partition. En effet, ceci pourra vous aider lorsque votre disque dur tombera en panne et que vous devrez recréer la table de partition sur le nouveau disque.

Restaurer vos données aidera aussi à réduire la fragmentation. Afin d'être sûr de récupérer tous les fichiers, le meilleur moyen de sauvegarder est de redémarrer votre système en mode "single user". Les systèmes de fichiers n'ont pas besoin d'être montés pour être sauvegardés. N'oubliez pas de monter la racine (/) en mode r/w (lecture/écriture) après avoir redémarré en mode "single user" ou dump retournera une erreur lorsqu'il essayera d'écrire les "dumpdates". Au démarrage, entrez "bsd -s" à l'invite de commandes boot> pour lancer le système en mode "single user".

Voir le contenu d'une sauvegarde sur bande :

Après avoir sauvegardé votre système de fichiers pour la première fois, il est bon de vérifier rapidement votre bande afin d'être certain que les données sont bien présentes.

Vous pouvez utiliser l'exemple suivant afin de voir le catalogue de fichiers présents sur une bande :

```
# /sbin/restore -tvs 1 -f /dev/rst0
```

Cette commande affichera la liste des fichiers présents sur la première partition de la bande. Si l'on se réfère à l'exemple précédent, 1 serait votre système de fichiers racine (/).

Pour afficher ce qui est présent sur la deuxième partition vers un fichier, vous utiliserez une commande similaire à :

```
# /sbin/restore -tvs 2 -f /dev/rst0 > /home/me/list.txt
```

Pour une table de partition assez simple, 2 serait /usr ; si la vôtre est plus compliquée, 2 pourrait être /var/ ou un autre système de fichiers. L'ordre des séquences est similaire à celui utilisé pour sauvegarder les systèmes de fichiers sur la bande.

Restaurer à partir d'une bande :

Le scénario de l'exemple suivant pourrait être utile si votre disque tombait complètement en panne. Dans le cas où vous ne souhaiteriez récupérer qu'un seul fichier à partir de la bande, référez-vous à la page de manuel de la commande restore et soyez attentifs aux instructions concernant le mode interactif.

Si vous avez bien tout préparé, remplacer un disque et restaurer vos données à partir d'une bande peut être une

procédure très rapide. La disquette standard install/boot sous OpenBSD contient déjà l'utilitaire restore ainsi que les exécutables nécessaires au partitionnement de votre nouveau disque et permettant de le rendre bootable. Dans la plupart des cas, cette disquette et votre bande de sauvegarde la plus récente sont tout ce dont vous avez besoin pour remettre votre machine en production.

Après avoir remplacé le disque en panne, suivez les étapes suivantes afin de restaurer vos données :

- Démarrez à partir de la disquette install/boot d'OpenBSD. Au menu, choisissez Shell. Insérez la bande de votre sauvegarde la plus récente bande, en lecture seule, dans votre lecteur.
- En vous aidant de la commande [fdisk\(8\)](#), créez une partition primaire OpenBSD sur le nouveau disque. Par exemple :

```
# fdisk -e sd0
```

Référez-vous à l'[Utilisation de fdisk\(8\) sous OpenBSD](#) pour de plus amples informations.

- Avec la commande disklabel, recréez votre table de partition OpenBSD à l'intérieur de la partition primaire que vous venez de créer avec fdisk. Par exemple :

```
# disklabel -E sd0
```

(N'oubliez pas la partition d'échange, swap, référez-vous à [Utilisation de disklabel](#) pour plus d'informations)

- Utilisez la commande newfs pour créer un système de fichier vierge sur chaque partition que vous venez de créer. Par exemple :

```
# newfs /dev/rsd0a
# newfs /dev/rsd0h
```

- Monter votre nouvelle partition racine (/) sur /mnt. Par exemple :

```
# mount /dev/sd0a /mnt
```

- Allez dans le nouveau répertoire racine et démarrez la procédure de restauration. Par exemple :

```
# cd /mnt
# restore -rs 1 -f /dev/rst0
```

- Vous aurez besoin de rendre ce disque bootable, inspirez-vous la commande suivante pour y inscrire un nouveau MBR. Par exemple :

```
# fdisk -i sd0
```

- En plus d'inscrire un nouveau MBR, vous aurez besoin d'installer les blocs de démarrage afin de pouvoir lancer la machine. Voici un bref exemple :

```
# cp /usr/mdec/boot /mnt/boot
# /usr/mdec/installboot -v /mnt/boot /usr/mdec/biosboot sd0
```

- Votre nouveau système de fichiers racine devrait être suffisamment prêt pour que vous puissiez démarrer dessus et continuer de restaurer le reste de vos systèmes de fichiers. Puisque votre système d'exploitation n'est pas encore complet, soyez sûrs de redémarrer en mode single-user. A l'invite de commandes, effectuez la procédure suivante afin de démonter et redémarrer le système :

```
# umount /mnt
# halt
```

- Retirez la disquette install/boot du lecteur et redémarrez votre machine. A l'invite de commandes de boot d'OpenBSD, lancez les commandes suivantes :

```
boot> bsd -s
```

La commande "bsd -s" ordonnera au noyau de démarrer en mode single-user qui ne nécessite que le système de fichiers racine (/).

- En admettant que vous ayez effectué correctement les étapes précédentes et que tout se soit bien passé, vous devriez vous retrouver à une invite de commandes vous demandant le chemin d'accès à un shell ou d'appuyer sur entrée. Appuyez sur entrée pour utiliser sh. Ensuite, vous devrez remonter la racine en mode r/w plutôt qu'en lecture seule. Lancez la commande suivante :

```
# mount -u -w /
```

- Une fois la racine en mode r/w, vous pourrez continuer à restaurer vos autres systèmes de fichiers. Par exemple :

(table de montage simple)

```
# mount /dev/sd0h /usr; cd /usr; restore -rs 2 -f /dev/rst0
```

(table de montage avancée)

```
# mount /dev/sd0d /var; cd /var; restore -rs 2 -f /dev/rst0
```

```
# mount /dev/sd0e /home; cd /home; restore -rs 3 -f /dev/rst0
```

```
# mount /dev/sd0h /usr; cd /usr; restore -rs 4 -f /dev/rst0
```

A la place, vous pourriez utiliser "**restore rvsf**" ou juste rsf pour voir le nom des fichiers extraits de la sauvegarde.

- Pour terminer, après avoir restauré tous vos systèmes de fichiers sur le disque, redémarrez en mode multi-utilisateurs. Si tout s'est déroulé comme prévu, votre système devrait se retrouver dans l'état dans lequel il se trouvait à la date de votre sauvegarde la plus récente et être totalement fonctionnel.

14.10 - Montage des images disque sous OpenBSD

Pour monter une image disque (images ISO, images disques créées avec dd, etc...) sous OpenBSD, vous devez

configurer un périphérique [vnd\(4\)](#). Par exemple, si vous avez une image ISO localisée dans `/tmp/ISO.image`, voici les étapes nécessaires pour la monter :

```
# vnconfig svnd0 /tmp/ISO.image
# mount -t cd9660 /dev/svnd0c /mnt
```

Remarquez que, puisqu'il s'agit d'une image ISO-9660 utilisée sur les CDs et DVDs, vous devez spécifier le type `cd9660` lors du montage. Ceci est vrai peu importe le type, par exemple, vous devez utiliser `ext2fs` lors du montage d'images disque Linux.

Pour démonter une image, utilisez les commandes suivantes.

```
# umount /mnt
# vnconfig -u svnd0
```

Pour plus d'information, référez-vous à la page de manuel de [vnconfig\(8\)](#).

14.11 - A l'aide ! J'ai des erreurs avec IDE DMA !

Les transferts IDE DMA supportés par [pciide\(4\)](#) ne sont pas fiables avec plusieurs combinaisons de matériel. Jusqu'à récemment, la plupart des "principaux" systèmes d'exploitation prétendant supporter les transferts DMA sur périphériques IDE n'activaient pas cette fonctionnalité par défaut à cause de certains matériels peu fiables. A présent, beaucoup de ces machines sont utilisées sous OpenBSD.

OpenBSD est agressif et tente d'utiliser le mode DMA le plus haut qu'il puisse configurer. Ceci pourrait engendrer une corruption dans le transfert des données sur certaines configurations à cause de certaines puces ("chipsets") de cartes mères, de disques de mauvaise qualité et/ou de la friture sur les câbles. Heureusement, les modes Ultra-DMA protègent les transferts de données avec CRC afin de détecter une éventuelle corruption. Si le CRC de l'Ultra-DMA échoue, OpenBSD affichera un message d'erreur et recommencera l'opération de transfert.

```
wd2a:  aborted command, interface CRC error reading fsbn 64 of 64-
79
(wd2 bn 127; cn 0 tn 2 sn 1), retrying
```

Après plusieurs échecs, OpenBSD fera descendre le mode Ultra-DMA (à priori plus fiable). Lorsqu'un mode Ultra-DMA atteint 0, le mode PIO prendra le relais.

Les erreurs UDMA sont souvent causées par des câbles endommagés ou de mauvaise qualité. C'est pourquoi les câbles devraient être les premiers suspectés si vous avez de nombreuses erreurs de DMA ou des performances dégradées. De plus, il n'est pas conseillé de placer un lecteur de CD-ROM sur la même nappe qu'un disque dur.

Si le remplacement des câbles ne résout pas le problème et qu'OpenBSD n'arrive pas à faire descendre le niveau UDMA, que votre machine se fige ou que vous obtenez trop de messages d'erreur sur la console et dans les logs, vous pouvez forcer le système à utiliser un mode DMA ou UDMA inférieur par défaut. Ceci peut être effectué avec [UKC](#) ou [config\(8\)](#) afin de changer les options du périphérique [wd\(4\)](#).

14.13 - Options RAID avec OpenBSD

Un ensemble redondant de disques durs indépendants (RAID - "Redundant Array of Inexpensive Disks") vous offre la possibilité d'utiliser plusieurs disques afin d'améliorer les performances, la capacité et/ou la redondance qu'un disque seul ne serait pas capable d'apporter. Bien qu'une description des avantages ou des risques apportés par une configuration RAID ne soit pas le propos de cet article, il est important de noter les points suivants :

- Un RAID ne constitue pas un système de sauvegarde.
- Seul, un RAID n'éliminera pas le délais d'indisponibilité.

Si ces informations sont nouvelles pour vous, ce document n'est pas un bon point de départ pour débiter avec un RAID.

Options logicielles

OpenBSD inclut [softraid\(4\)](#), un framework supportant plusieurs types de transformations I/O, incluant le RAID et différents chiffrements. Softraid(4) est configuré en utilisant [bioctl\(8\)](#). OpenBSD inclut aussi RAIDFRAME ([raid\(4\)](#), nécessite un noyau personnalisé), et [ccd\(4\)](#) une implémentation historique de RAID, mais actuellement OpenBSD ne vous le suggère pas pour implémenter une solution RAID pour une nouvelle installation ou réinstallation.

Options matérielles

De nombreuses [plates-formes](#) OpenBSD supportent plusieurs contrôleurs RAID matériels. Ceux-ci varient selon les plates-formes, référez-vous à la page des plates-formes supportées (listées [ici](#)).

Remarquez que plusieurs contrôleurs RAID peuvent être configuré via le pilote [bio\(4\)](#) et l'application [bioctl\(8\)](#). Cela permet de surveiller et contrôler le système RAID d'une manière standard dans OpenBSD.

Une autre option accessible à plusieurs plates-formes est l'un des nombreux produits qui font apparaître plusieurs disques comme un seul disque IDE ou SCSI et sont ensuite branchés sur un contrôleur IDE ou SCSI standard. Ces périphériques peuvent virtuellement fonctionner sur n'importe quelle plate-forme supportant IDE ou SCSI.

Quelques fabricants de ce type de produits :

- [Arco](#)
- [Accusys](#)
- [Maxtronic](#)
- [Infortrend](#)

(Note : ce sont des produits que certains utilisateurs d'OpenBSD utilisent ; il ne s'agit en aucun cas d'une liste exhaustive ni de matériels certifiés).

Options non valides

Une question récurrente sur les [listes de diffusion](#) consiste à savoir si "les contrôleurs RAID IDE ou SATA peu coûteux (tels que les contrôleurs HighPoint, Promise ou Adaptec HostRAID) sont supportés ?". La réponse est "Non". Ces cartes et puces ne sont pas de véritables contrôleurs RAID matériels mais plutôt des RAID logiciels assistés par un BIOS. Puisqu'OpenBSD supporte déjà le RAID logiciel indépendamment du type de matériel, il n'y a pas de volonté parmi les développeurs OpenBSD d'implémenter un support spécifique à ces cartes.

Presque tous ces contrôleurs "RAID" SATA ou IDE embarqués sont de type logiciel et fonctionneront sans problème comme contrôleur SATA ou IDE en utilisant le pilote IDE standard ([pciide\(4\)](#)), mais ne fonctionneront pas comme RAID matériel sous OpenBSD.

14.14 - Pourquoi `df(1)` me dit que j'ai plus de 100% d'espace disque utilisé ?

Certaines personnes sont parfois surprises de voir qu'elles ont un espace disque disponible *négatif* ou que plus de 100% de l'espace d'un système de fichiers est utilisé, comme on peut le voir avec [df\(1\)](#).

Lorsqu'un système de fichiers est créé avec [newfs\(8\)](#), un certain montant d'espace disque inaccessible aux utilisateurs est réservé. Cela permet d'avoir une marge d'erreur en cas de remplissage accidentel du disque et de garder la fragmentation au minimum. Par défaut, l'espace réservé correspond à 5% de la capacité du disque ce qui signifie que si root a imprudemment rempli le disque, vous pouvez vous retrouver avec 105% de capacité disponible utilisée.

Si une valeur de 5% ne vous semble pas appropriée, vous pouvez la changer avec la commande [tunefs\(8\)](#).

14.15 - Récupération de partitions après une suppression du disklabel

Si vous avez une table de partitions endommagée, vous disposez de plusieurs moyens afin d'essayer de la récupérer.

Premièrement, paniquez. Vous le faites relativement souvent, et ceci devrait donc bien se passer. Cependant, ne faites rien de stupide. Paniquez à l'écart de votre machine. Ensuite, relaxez vous, et voyez si les étapes suivantes ne pourraient pas vous aider.

Une copie de votre disklabel est sauvegardée pour chaque disque dans `/var/backups` via la maintenance système journalière. A condition d'avoir encore votre partition var, vous pouvez simplement lire la sortie, et la réintégrer dans disklabel.

Dans le cas où vous ne pouvez plus voir cette partition, il y a deux options. Corriger une assez grande partie du disque afin de la voir, ou fixer une partie suffisamment grande du disque afin de pouvoir récupérer vos données. Selon ce qui s'est passé, l'une ou l'autre de ces deux solutions serait préférable (avec des disques mourants vous

voudrez en premier les données, avec des doigts mouillés vous ne pourrez avoir que le label).

Le premier bon outil dont vous avez besoin est [scan_ffs\(8\)](#) (remarquez l'underscore, il ne s'appelle pas "scanffs"). `scan_ffs(8)` regardera au travers du disque, et essaiera de trouver les partitions, en vous avertissant des informations qu'il trouve sur celles-ci). Vous pouvez utiliser ces informations pour recréer le `disklabel`. Si vous voulez uniquement restaurer `/var`, vous pouvez recréer la partition pour `/var`, et ensuite récupérer le label et ajouter le reste.

[disklabel\(8\)](#) mettra à jour à la fois la compréhension du `disklabel` par le noyau, et essaiera également d'écrire le label sur le disque. Ainsi, même si la partie du disque qui contient le `disklabel` est illisible, vous serez en mesure de la monter avec [mount\(8\)](#) jusqu'au prochain redémarrage.

14.16 - Est-il possible d'accéder aux données présentes sur des systèmes de fichiers autres que FFS ?

Oui. Les systèmes de fichiers supportés sont : ext2 (Linux), ISO9660 et UDF (CD-ROM, DVD), FAT (MS-DOS et Windows), NFS, NTFS (Windows) et AmigaDOS. Certains n'ont qu'un support limité, comme par exemple l'accès en lecture uniquement.

Nous allons donner un aperçu sur la façon d'utiliser un de ces systèmes de fichiers sous OpenBSD. Afin d'utiliser un système de fichiers, celui-ci doit être monté. Pour plus de détails sur les options de montage, référez-vous au manuel de [mount\(8\)](#) ainsi qu'à celui correspondant au système de fichiers que vous souhaitez monter. Par exemple, `mount_msdos`, `mount_ext2fs`, ...

Tout d'abord, vous devez connaître le périphérique sur lequel votre système de fichiers est présent. Cela peut être tout simplement votre disque dur, `wd0` ou `sd0`, mais ça n'est pas toujours évident. Tous les périphériques reconnus et configurés par votre système apparaissent dans la sortie de la commande [dmesg\(1\)](#) : un nom de périphérique, suivi par une ligne descriptive. Ainsi, mon premier CD-ROM est reconnu de la façon suivante :

```
cd0 at scsibus0 targ 0 lun 0: <COMPAQ, DVD-ROM LTD163, GQH3> SCSI0
5/cdrom removable
```

Afin d'obtenir une liste beaucoup plus courte des disques disponibles, vous pouvez utiliser [sysctl\(8\)](#). La commande

```
# sysctl hw.disknames
```

affichera tous les disques reconnus par votre système, par exemple :

```
hw.disknames=cd0 , cd1 , wd0 , fd0 , cd2
```

A présent, il est temps de déterminer quelles partitions sont présentes sur le périphérique et sur quelle partition le système de fichiers qui nous intéresse est présent. Nous examinerons le périphérique en utilisant [disklabel\(8\)](#). Le 'disklabel' comporte une liste de partitions, 16 au maximum. La partition `c` désigne toujours le périphérique dans

son ensemble. Les partitions a-b et d-p sont utilisées par OpenBSD. Les partitions i-p peuvent être allouées automatiquement aux systèmes de fichiers d'autres systèmes d'exploitation. Dans notre cas, j'analyserai le 'disklabel' de mon disque dur qui contient un certain nombre de systèmes de fichiers.

REMARQUE : OpenBSD a été installé après les autres systèmes d'exploitation sur cette machine et pendant l'installation, le disklabel contenant les partitions pour systèmes de fichiers natifs et étrangers a été installé. Cependant, si vous ajoutez des systèmes de fichiers étrangers après l'installation du disklabel d'OpenBSD vous devrez probablement les ajouter ou les modifier ultérieurement. Cette procédure est expliquée [ici](#).

```
# disklabel wd0
```

```
# using MBR partition 2: type A6 off 20338290 (0x1365672) size
29318625 (0x1bf5de1)
# /dev/rwd0c:
type: ESDI
disk: ESDI/IDE disk
label: ST340016A
flags:
bytes/sector: 512
sectors/track: 63
tracks/cylinder: 16
sectors/cylinder: 1008
cylinders: 16383
total sectors: 78165360
rpm: 3600
interleave: 1
trackskew: 0
cylinderskew: 0
headswitch: 0          # microseconds
track-to-track seek: 0 # microseconds
drivedata: 0
```

```
16 partitions:
```

#	size	offset	fstype	[fsize	bsize	cpg]
a:	408366	20338290	4.2BSD	2048	16384	16 # Cyl
20176*- 20581						
b:	1638000	20746656	swap			# Cyl
20582 - 22206						
c:	78165360	0	unused	0	0	#
Cyl 0 - 77544						
d:	4194288	22384656	4.2BSD	2048	16384	16 # Cyl
22207 - 26367						
e:	409248	26578944	4.2BSD	2048	16384	16 # Cyl
26368 - 26773						
f:	10486224	26988192	4.2BSD	2048	16384	16 # Cyl
26774 - 37176						
g:	12182499	37474416	4.2BSD	2048	16384	16 # Cyl


```

37177 - 49262*
  i:          64197          63 unknown          #
Cyl      0*-    63*
  j:      20274030          64260 unknown          #
Cyl      63*- 20176*
  k:      1975932          49656978  MSDOS          # Cyl
49262*- 51223*
  l:      3919797          51632973 unknown          # Cyl
51223*- 55111*
  m:      2939832          55552833  ext2fs          # Cyl
55111*- 58028*
  n:      5879727          58492728  ext2fs          # Cyl
58028*- 63861*
  o:      13783707          64372518  ext2fs          # Cyl
63861*- 77535*

```

Comme on peut le constater, les partitions OpenBSD sont affichées en premier. Puis l'on peut voir un certain nombre de partitions ext2, une partition MSDOS ainsi que plusieurs partitions inconnues ('unknown'). Sous i386 ou amd64, vous pouvez généralement en savoir plus sur celles-ci en utilisant l'utilitaire [fdisk\(8\)](#). Pour le lecteur curieux, la partition i est une partition de maintenance créée par le revendeur, la partition j est une partition NTFS et la partition l est une partition d'échange Linux ('Linux swap').

Après avoir déterminée la partition que vous souhaitez utiliser, vous pouvez enfin monter le système de fichiers qu'elle contient. La plupart des systèmes de fichiers sont supportés par le noyau GENERIC : référez-vous à son fichier de configuration situé dans le répertoire `/usr/src/sys/arch/<arch>/conf`. Cependant, certains dont le support est expérimental (comme NTFS) ne sont pas inclus dans le noyau GENERIC. Si vous souhaitez utiliser un de ces systèmes de fichiers, vous devrez [recompiler votre noyau](#).

Une fois les informations nécessaires entre vos mains, il est temps de monter le système de fichiers. Imaginons que le répertoire `/mnt/otherfs` existe, nous l'utiliserons pour monter le système de fichiers souhaité. Dans l'exemple suivant, nous monterons un système de fichiers ext2 dans la partition m :

```
# mount -t ext2fs /dev/wd0m /mnt/otherfs
```

Si vous souhaitez utiliser ce système de fichiers régulièrement, vous gagnerez du temps en insérant la ligne suivante dans votre fichier `/etc/fstab` :

```
/dev/wd0m /mnt/otherfs ext2fs rw,noauto,nodev,nosuid 0 0
```

Notez les valeurs 0 dans les cinquièmes et sixièmes champs. Cela indique que nous ne souhaitons pas que le système de fichiers soit sauvegardé (avec dump) ni vérifié avec fsck. Généralement, il est préférable de laisser le système d'exploitation associé à ces systèmes de fichiers s'occuper de cela.

14.16.1 - Les partitions n'apparaissent pas dans mon disklabel ! Que dois-je faire ?

Si vous installez des systèmes de fichiers étrangers sur votre machine (souvent suite à l'ajout d'un système d'exploitation) après l'installation d'OpenBSD, un disklabel sera présent mais ne sera pas mis à jour automatiquement pour prendre en compte les nouveaux systèmes de fichiers. Si vous souhaitez pouvoir y accéder, vous aurez besoin d'ajouter ou de modifier ces partitions manuellement en utilisant [disklabel\(8\)](#).

Pour prendre un exemple, j'ai modifié une de mes partitions ext2 existantes : en utilisant le programme fdisk de Linux, j'ai réduit la taille de la partition 'o' (voir la sortie de disklabel plus haut) à 1Go. Il sera facile de la reconnaître grâce à sa position de départ (offset: 64372518) et sa taille (13783707) sur le disque. Notez que ces valeurs sont exprimées en numéros de secteurs (et pas en megaoctets ou autre mesure) et qu'utiliser de tels nombres reste le moyen le plus sûr et le plus précis d'obtenir ces informations.

Avant notre changement, la partition ressemblait à ceci (en utilisant le [fdisk\(8\)](#) d'OpenBSD et en ne gardant que ce qui nous intéresse) :

```
# fdisk wd0
. . .
Offset: 64372455      Signature: 0xAA55
      Starting      Ending      LBA Info:
#: id   C   H   S -   C   H   S [      start:      size   ]
-----
0: 83 4007   1   1 - 4864 254 63 [ 64372518: 13783707 ]
Linux files*
. . .
```

Comme vous pouvez le voir, la position de départ et la taille sont exactement les mêmes que celles rapportées précédemment par disklabel(8) (ne vous méprenez pas sur par la valeur indiquée par "Offset" : il s'agit de la position de départ de la partition étendue à laquelle la partition ext2 fait partie).

Après avoir changé la taille de la partition sous Linux, elle ressemble à ceci :

```
# fdisk wd0
. . .
Offset: 64372455      Signature: 0xAA55
      Starting      Ending      LBA Info:
#: id   C   H   S -   C   H   S [      start:      size   ]
-----
0: 83 4007   1   1 - 4137 254 63 [ 64372518: 2104452 ]
Linux files*
. . .
```

Ceci doit être changé par disklabel(8). Vous pouvez, par exemple, utiliser `disklabel -e wd0` qui lancera l'éditeur correspondant à la variable d'environnement EDITOR (par défaut il s'agit de vi). Une fois dans l'éditeur, changez la dernière ligne du disklabel afin qu'elle corresponde à la nouvelle taille :

```
o:      2104452      64372518  ext2fs
```

Pour terminer, enregistrez le disklabel sur le disque. A présent que le disklabel est à jour, vous devriez pouvoir monter vos partitions comme indiqué précédemment.

Vous pouvez suivre une procédure similaire afin d'ajouter de nouvelles partitions.

14.17 - Est-il possible d'utiliser un périphérique de masse ('flash memory device') sous OpenBSD ?

14.17.1 - mémoire flash comme périphérique de stockage portable

Théoriquement, un périphérique mémoire devrait être reconnu dès son insertion dans la machine. Peut après l'avoir branché, un certain nombre de messages du noyau devrait apparaître sur la console. Ainsi, lorsque je branche mon périphérique USB, voici ce qui apparaît sur la console :

```
umass0 at uhub1 port 1 configuration 1 interface 0
umass0: LEXR PLUG DRIVE LEXR PLUG DRIVE, rev 1.10/0.01, addr 2
umass0: using SCSI over Bulk-Only
scsibus2 at umass0: 2 targets
sd0 at scsibus2 targ 1 lun 0: <LEXAR, DIGITAL FILM, /W1.> SCSI2 0/
direct removable
sd0: 123MB, 123 cyl, 64 head, 32 sec, 512 bytes/sec, 251904 sec
total
```

Ces lignes indiquent que le pilote [umass\(4\)](#) (périphérique de masse USB -'USB mass storage') à été rattaché au périphérique mémoire et qu'il utilise le système SCSI. Les deux dernières lignes sont les plus importantes : elles indiquent sur quel fichier de périphérique le matériel a été rattaché ainsi que le montant total d'espace de stockage. Si vous avez manqué ces lignes, vous pouvez vous y référer plus tard à l'aide de la commande [dmesg\(1\)](#). La géométrie CHS est fictive puisque le périphérique USB est traité comme un disque SCSI classique.

Nous allons voir deux scénarios possibles.

Le périphérique est nouveau/vide et vous souhaitez l'utiliser exclusivement avec OpenBSD.

Vous allez devoir initialiser un 'disklabel' sur le périphérique et créer au minimum une partition. Pour plus de détails, lisez [Utilisation de disklabel\(8\) sous OpenBSD](#) ainsi que sa [page de manuel](#).

Dans cet exemple, je ne vais créer qu'une seule partition *a* dans laquelle je placerai un système de fichiers FFS :

```
# newfs sd0a
Warning: inode blocks/cyl group (125) >= data blocks (62) in last
cylinder group. This implies 1984 sector(s) cannot be
allocated.
/dev/rsd0a:      249856 sectors in 122 cylinders of 64 tracks, 32
sectors
```

```

    122.0MB in 1 cyl groups (122 c/g, 122.00MB/g, 15488 i/g)
super-block backups (for fsck -b #) at:
 32,

```

Montons le système de fichiers créé dans la partition *a* sur `/mnt/flashmem`. Si le point de montage n'existe pas, créez-le.

```

# mkdir /mnt/flashmem
# mount /dev/sd0a /mnt/flashmem

```

Vous avez reçu le périphérique de quelqu'un avec qui vous souhaitez échanger des données.

Il y a une grande chance que la personne en question n'utilise pas OpenBSD, ce qui signifie qu'un système de fichiers étranger doit être présent sur le périphérique. Nous allons donc devoir découvrir quelles partitions sont présentes, comme décrit dans la section précédente : [Est-il possible d'accéder aux données présentes sur des systèmes de fichiers autres que FFS ?](#).

```

# disklabel sd0

# /dev/rsd0c:
type: SCSI
disk: SCSI disk
label: DIGITAL FILM
flags:
bytes/sector: 512
sectors/track: 32
tracks/cylinder: 64
sectors/cylinder: 2048
cylinders: 123
total sectors: 251904
rpm: 3600
interleave: 1
trackskew: 0
cylinderskew: 0
headswitch: 0          # microseconds
track-to-track seek: 0 # microseconds
drivedata: 0

16 partitions:
#          size          offset  fstype  [fsize  bsize   cpgh]
#   c:         251904           0  unused         0     0     #
Cyl   0 -   122
#   i:         250592          32   MSDOS                #
Cyl   0*-   122*

```

Le 'disklabel' nous apprend qu'il n'y a qu'une seule partition *i*, contenant un système de fichiers FAT créé sur une

machine Windows. Comme d'habitude, la partition *c* représente l'ensemble du disque.

A présent, nous allons monter le système de fichiers présent dans la partition *i* vers `/mnt/flashmem`.

```
# mount -t msdos /dev/sd0i /mnt/flashmem
```

Vous pouvez désormais l'utiliser comme n'importe quel autre disque.

ATTENTION : Vous devriez toujours **démonter** le système de fichiers **avant de débrancher** le périphérique de masse. Si vous ne le faites pas, le système de fichiers pourrait être laissé dans un état inconsistant ce qui aurait pour résultat d'avoir des données corrompues.

Après avoir débranché le périphérique de votre machine, vous verrez à nouveau des messages du noyau sur la console :

```
umass0: at uhub1 port 1 (addr 2) disconnected
sd0 detached
scsibus2 detached
umass0 detached
```

14.17.2 - Mémoires Flash comme disque de démarrage

On peut utiliser les mémoires flash de différentes façons comme disque de démarrage avec OpenBSD. Cela peut-être fait avec des périphériques USB (en s'assurant que votre ordinateur peut démarrer sur un périphérique flash USB, ce qui n'est pas le cas pour tous) ou non USB (par exemple CF) avec des adaptateurs IDE ou SATA. (Les périphériques non USB attachés à un adaptateur USB sont traités comme un périphérique USB). Dans certains cas, vous pouvez actuellement utiliser un périphérique des deux façons (chargez le média dans l'adaptateur USB, pour utilisez le dans un adaptateur IDE)

Un périphérique flash attaché à un port USB sera visible comme un périphérique SCSI [sd\(4\)](#). Attaché à une adaptateur IDE, il sera visible comme un périphérique [wd\(4\)](#).

Dans le cas d'un média flash dans un adaptateur IDE, il pourra être démarré de tout système qui peut démarrer un disque dur IDE. D'un certain sens, le système voit le média flash comme un disque IDE. Configurez simplement le matériel de manière appropriée, et installez OpenBSD sur le disque flash comme d'habitude.

Dans le cas de démarrage d'un périphérique USB, votre système doit être capable de démarrer d'un périphérique USB sans être distrait par un autre périphérique sur le système. Il faut remarquer que si votre intention est de faire un environnement portable de démarrage sur un périphérique USB, le périphérique USB doit être vu comme un disque SCSI, habituellement `sd0`. Cependant, si vous insérez ce périphérique dans un système qui possède déjà quelques disques SCSI, il sera sûrement pris en compte avec un identifiant différent, ce qui compliquera l'utilisation de ce périphérique flash de système en système, vous devrez donc mettre à jour `/etc/fstab`.

Quelques remarques :

- **Vitesse** : En general, les périphériques flash sont beaucoup plus lents que les disques durs, spécialement en cas d'écriture. Utiliser [soft updates](#) aidera considérablement, en utilisant aussi l'option de montage "noatime".
- **"fatigue d'écriture"** : Beaucoup ont écrit sur le nombre de fois qu'une cellule flash individuelle peut-être réécrite avant incident. En pratique cependant, il y a plusieurs façons pour qu'un périphérique flash disfonctionne, la fatigue d'écriture et juste l'une d'elle. Les périphériques flash modernes vérifient l'écriture, et en cas de problème, réallouent le secteur defectueux avec un autre disponible. La plupart des utilisateurs de périphériques flash ne seront jamais ennuyés avec la "fatigue d'écriture". Vous ferez probablement l'expérience de moins d'arrêt de service du à des erreurs d'astuces "saines" utilisées pour éviter d'écrire sur le disque flash plutôt que d'utiliser les disques comme média en lecture-écriture.
- **Fiabilité** : Le fait que les médias flash ne possèdent pas de partie mobile a fait croire à beaucoup de personnes que les médias flash sont plus fiables que les disques durs. Il n'est probablement pas prouvé que l'utilisation de flash vous évitera les pertes de données ou les disques defectueux. Les gens ont rapporté de considérables variations dans la qualité des médias flash, il est plus probable de considérer le stockage flash comme silencieux et une alternative faible consommation au disque plutôt qu'un stockage sans erreurs.
- **Créer un disque flash USB de démarrage** : Bien qu'un périphérique USB ne peut seulement démarrer que sur une machine supportant les disques USB, il peut être crée sur n'importe quelle machine qui supporte le matériel USB. Vous pourrez, bien sur, être incapable de tester votre travail tant que vous ne posséderez pas un système démarrant sur USB.
- **Aller d'une interface IDE vers USB** : Depuis que le média flash peut-être lu et écrit aussi bien d'un adaptateur USB et IDE, vous pouvez créer un média pour démarrer avec un de ces types d'adaptateurs mais le maintenir ou l'utiliser avec un autre type d'adaptateur.
- **Partager OpenBSD et d'autres partitions sur un périphérique** : OpenBSD considère les disques flash comme tous les autres disques donc vous pouvez utiliser [fdisk\(8\)](#) pour partitionner le périphérique flash, comme sur un disque dur. Vous pouvez avoir un système de fichiers OpenBSD sur une partition et utiliser une autre partition pour d'autres systèmes de fichiers, par exemple FAT32. Cependant tous les systèmes de fichiers ne sont pas traités en USB de façon "égale". Windows, en particulier, ne voudra pas utiliser ou créer une partition qui ne commencera pas au début du périphérique, tout comme les outils de partitionnement Windows ne vous permettront pas de partitionner le disque, ils respecteront les partitions existantes. Donc, si vous souhaitez créer un disque flash USB qui puisse démarrer avec OpenBSD, mais aussi fonctionne comme un périphérique FAT32 sur d'autres OS, vous voudrez faire quelque chose comme cela :
 1. Partitionnez le média avec le fdisk d'OpenBSD, créer une partition du type désiré pour Windows qui utilise le début du disque et une partition OpenBSD à la fin du disque.
 2. Installez OpenBSD comme d'habitude sur la partition fdisk OpenBSD, n'oubliez pas de marquer la partition OpenBSD comme "Active" pour démarrer.
 3. Formatez les autres partitions? Cela peut-être (et doit l'être) fait sur l'OS "cible" (Windows dans notre cas).

Il faut remarquer que si le type de l'autre partition est choisi de manière appropriée, il est possible que OpenBSD accède aux deux partitions sur le périphérique. Donc, un utilisateur Windows peut mettre sur la partition FAT32 des fichiers MP3 qui pourront être joués quand ils seront démarrés de la partition OpenBSD.

14.17.3 - Comment puis-je créer un périphérique de démarrage USB "Live" ?

Il est très facile de créer un disque flash USB de démarrage (ou autre !) qui peut-être utilisé comme un système OpenBSD "live" sans installer OpenBSD sur le disque dur local de la machine. Evidemment, la machine cible doit pouvoir démarrer d'un périphérique USB, mais le chargement initial peut-être fait de n'importe quelle machine avec une interface USB.

Des raisons pour lesquelles vous voudriez le faire :

- Une "machine" sécurisée, portable que vous pouvez emporter dans votre poche.
- Résoudre des problèmes dans OpenBSD et d'autres installations avec des outils qui ne sont pas disponibles dans [bsd.rd](#)
- Evaluer le matériel pour une compatibilité avec OpenBSD au moment de l'achat.
- Installer les machines qui manquent de toute autre forme de média de démarrage.
- Collecter des sorties [dmesg](#) des ordinateurs de vos amis ! (Les dmesg d'OpenBSD sont utiles pour identifier le matériel, cela peut-être un bonne façon de préparer l'installation pour d'AUTRES OS sur une machine -- soyez sûr d'avoir tous les bons pilotes nécessaires avec de démarrer).

Créer un tel "disque OpenBSD live" est simple.

- Montez le disque USB sur la machine sur laquelle vous allez l'installer.
- Démarrer votre média d'installation favoris d'OpenBSD.
- [Installez](#) normalement, soyez attentionné en sélectionnant votre disque flash comme cible d'installation.
- Démarrez votre nouveau périphérique USB crée.

Voilà !

Il y a certaines choses que vous voudrez faire après l'installation pour améliorer vos résultats :

- Installez tous les paquetages et utilitaires que vous désirez disponibles.
- Des machines cibles différentes auront des NICs différents. Vous pouvez créer plusieurs fichiers [hostname.if\(5\)](#) dans `/etc`, chacun contenant uniquement `dhcp`, pour tous les NICs que vous rencontrerez (`fxp0`, `re0`, `rl0`, `bge0`, `bnx0`, `em0`, etc...) sur les machines démarrant en USB, plus peut-être des exemples de fichiers de configuration sans-fil bien évidemment. OpenBSD ignorera tous les fichiers `hostname.if(5)` pour les périphériques non présent au démarrage.
- Vous pouvez avoir envie de posséder une copie des fichiers d'installation et peut-être aussi de fichiers paquetage `.tgz` comme cela vous pourrez utiliser le disque comme média d'installation (démarez `bsd.rd` à la place de le laisser démarrer normalement).
- Sur la plupart des machines, `X` "fonctionnera" sans fichier de configuration, mais vous pouvez avoir envie de collecter les fichiers de configuration `X` pour les systèmes qui en ont besoin.
- [Soft updates](#) sera quelque chose que vous voudrez utiliser.
- Pour un maximum de flexibilité, vous voudrez plutôt utiliser [i386](#) plutôt que [amd64](#). Cependant, si vous voulez l'utiliser comme média d'installation, vous pouvez avoir un `amd64 bsd.rd` et les fichiers d'installation en plus de `i386`.
- Si votre machine cible possède une interface SCSI ou [ahci\(4\)](#), vous verrez surement l'identifiant de votre disque USB changer. Posséder plusieurs versions de votre fichier `/etc/fstab` permettra de "corriger"

cela plus facilement (en mode mono-utilisateur).

- Vous pouvez désirer avoir une partition FAT sur le disque USB, créez la comme détaillé [ci-dessus](#).
- Vous pouvez avoir envie d'utiliser [softraid\(4\)](#) pour chiffrer une partition de données.

14.18 - Optimiser les performances des disques durs

La performance des disques est un facteur significatif de la vitesse globale de votre machine. Ce facteur devient de plus en plus important quand votre machine héberge un environnement multi utilisateur (des utilisateurs de toutes les catégories tels que les utilisateurs qui se connectent de manière interactive et les utilisateurs qui voient votre machine comme un serveur de fichiers ou un serveur Web). Le stockage de données demande une attention constante et particulièrement quand vos partitions ne contiennent plus d'espace libre ou quand vos disques ne fonctionnent plus. OpenBSD possède peu d'options pour augmenter la vitesse des opérations sur disque.

- [Soft Updates](#)
- [Taille du cache namei\(\)](#)

14.18.1 - Soft Updates

Un outil important qui peut être utilisé pour accélérer la vitesse de votre système est softupdates. La mise à jour des informations meta ou metainfo (qui a lieu quand vous créez ou supprimez des fichiers et des répertoires entre autres) est une des opérations les plus lentes du système de fichiers BSD traditionnel. Softupdates tente de mettre à jour les metainfo dans la RAM au lieu d'écrire chaque mise à jour de metainfo sur le disque. Une autre conséquence est que les metainfo sur le disque devraient être toujours complètes, mais pas forcément à jour. Pour en savoir plus, consultez [La FAQ Softupdates](#).

14.18.2 - Taille du cache namei()

Le cache de traduction nom-vers-inode (alias name-to-inode ou namei()) contrôle la vitesse de la traduction chemin vers [inode\(5\)](#). Une valeur raisonnable de fixer une valeur pour le cache, si on venait à remarquer à l'aide d'un outil comme [systat\(1\)](#), des erreurs d'allocation au niveau du cache, est d'examiner la valeur courante générée avec [sysctl\(8\)](#), (qui appelle ce paramètre "kern.maxvnodes") et d'augmenter cette valeur soit jusqu'à ce que les le taux de réponse à partir du cache namei s'améliore soit jusqu'à ce qu'on détermine que le système ne bénéficie plus substantiellement de l'augmentation de la taille du cache namei. Une fois que la valeur est déterminée, vous pouvez la fixer au démarrage du système à l'aide de [sysctl.conf\(5\)](#).

11.3 - Pourquoi nous n'utilisons pas de montage asynchrone ("async mount") ?

Question : "Je fais simplement "mount -u -o async /" ce qui rend un paquetage dont j'ai besoin (qui touche à une centaine de chose de temps à autre) utilisable. Pourquoi le montage asynchrone n'est pas vu d'un bon œil et n'est pas activé par défaut (comme c'est le cas sur d'autres versions d'Unix) ? C'est un mécanisme sûrement plus simple et plus sûr d'améliorer les performances de certaines applications."

Réponse : " les montages asynchrones sont en effet plus rapides que des montages synchrones, mais ils sont aussi moins sûrs. Qu'arrive-t-il dans le cas d'une panne de courant ? Ou un problème matériel ? la quête de la vitesse ne doit pas sacrifier la fiabilité et la stabilité du système. Reportez-vous à la page du manuel de [mount\(8\)](#)."

```

    async    All I/O to the file system should be done
asynchronously.

    This is a dangerous flag to set since it does not
guaran-
tee to keep a consistent file system structure on the
disk.  You should not use this flag unless you are pre-
pared to recreate the file system should your system
crash.  The most common use of this flag is to speed up
restore(8) where it can give a factor of two speed in-
crease.

```

D'un autre côté, quand vous travaillez avec des données temporaires que vous pouvez recréer après un plantage, vous pouvez gagner en vitesse en utilisant une partition à part montée en asynchrone, utilisée uniquement pour ce type de données. Encore une fois, n'effectuez cette opération *que si* vous ne voyez pas d'inconvénient à perdre toutes les données de cette partition si quelque chose va mal. Pour cette raison, les partitions [mfs\(8\)](#) sont montées en mode asynchrone vu que de toute façon, elles vont être écrasées et recréées après un redémarrage.

[\[Index de la FAQ\]](#) [\[Section 13 - Multimédia\]](#) [\[Section 15 - Paquetages et Ports\]](#)



www@openbsd.org

\$OpenBSD: faq14.html,v 1.71 2010/05/16 20:33:43 ajacoutot Exp \$

OpenBSD

[\[Index de la FAQ\]](#) [\[Section 14 - Configuration des disques\]](#)

15 - Système de Ports et de Paquetages OpenBSD

Table des matières

- [15.1 - Introduction](#)
- [15.2 - Gestion des paquetages](#)
 - [15.2.1 - Comment cela fonctionne-t-il ?](#)
 - [15.2.2 - Faire les choses simplement](#)
 - [15.2.3 - Trouver des paquetages](#)
 - [15.2.4 - Installer de nouveaux paquetages](#)
 - [15.2.5 - Lister les paquetages installés](#)
 - [15.2.6 - Mettre à jour les paquetages installés](#)
 - [15.2.7 - Supprimer les paquetages installés](#)
 - [15.2.8 - Suppression ou installation incomplète de paquetage](#)
- [15.3 - Utilisation des ports](#)
 - [15.3.1 - Comment cela fonctionne-t-il ?](#)
 - [15.3.2 - Récupérer l'arbre des ports](#)
 - [15.3.3 - Configuration du système des ports](#)
 - [15.3.4 - Rechercher dans l'arbre des ports](#)
 - [15.3.5 - Installation normale : un exemple simple](#)
 - [15.3.6 - Nettoyer après une compilation](#)
 - [15.3.7 - Désinstaller un paquetage de port](#)
 - [15.3.8 - Utiliser les saveurs \("flavors"\) et les sous-paquetages \("subpackages"\)](#)
- [15.4 - FAQ](#)
 - [15.4.1 - J'obtiens toutes sortes d'erreurs folles. Je n'arrive pas du tout à faire fonctionner ces ports.](#)
 - [15.4.2 - Pourquoi la dernière version de mon logiciel favori/préfééré n'est pas disponible ?](#)
 - [15.4.3 - Pourquoi n'y a-t-il pas de paquetage de mon logiciel favori/préfééré ?](#)
 - [15.4.4 - Pourquoi n'y a-t-il pas de port de mon logiciel favori/préfééré ?](#)
 - [15.4.5 - Pourquoi mon logiciel favori/préfééré ne fait pas partie du système de base ?](#)
 - [15.4.6 - Qu'est ce que je dois utiliser : paquetages ou ports ?](#)

- [15.4.7 - Comment puis-je optimiser ces ports pour obtenir le maximum de performance ?](#)
 - [15.4.8 - J'ai soumis un nouveau port/une mise à jour depuis plusieurs semaines. Pourquoi cela n'a pas été intégré \("committed"\) ?](#)
 - [15.5 - Comment signaler un problème](#)
 - [15.6 - Comment nous aider](#)
-

15.1 - Introduction

Il y a quantité d'application tierces disponibles dont une que l'on souhaiterait utiliser sur un système OpenBSD. Pour faire que ce logiciel soit plus simple à installer et à administrer, et aussi pour l'aider à ce qu'il suive la politique et les objectifs d'OpenBSD, ce logiciel tiers est *porté* sur OpenBSD. Cet effort de portage peut impliquer plusieurs choses différentes. Par exemples : faire que le logiciel utilise les répertoires standards de OpenBSD (exemple : les fichiers de configuration vont dans `/etc`), se conformer aux spécifications des bibliothèques partagées d'OpenBSD, faire que le logiciel soit plus sécurisé autant que possible, etc...

Le résultat final de l'effort de portage sont des paquetages binaires prêts à être installés. Le but du système de paquetages est de garder une trace des logiciels installés, de sorte qu'a n'importe quel moment on puisse le mettre à jour ou le supprimer très simplement. De cette façon, il ne reste aucune trace de fichiers inutiles et les utilisateurs peuvent garder leur système propre. Le système de paquetage s'assure également que rien n'est supprimé par accident, arrêtant ainsi le logiciel de fonctionner correctement. Un autre avantage est que **les utilisateurs ont rarement le besoin de compiler un logiciel à partir des sources**, alors que les paquetages sont déjà compilés, disponibles et prêt à être utilisé sur un système OpenBSD. En quelques minutes, un grand nombre de paquetages peuvent être récupérés et installés avec tout à la bonne place.

Les paquetages et la collection des ports NE passent PAS par le même audit complet de sécurité qui est réalisé sur le système de base d'OpenBSD. Bien que nous tâchions de maintenir la qualité de la collection des paquetages élevée, nous n'avons juste pas assez de ressources humaines pour assurer le même niveau de robustesse et de sécurité.

15.2 - Gestion des paquetages

15.2.1 - Comment cela fonctionne-t-il ?

Les paquetages sont les binaires pré-compilés d'une partie des logiciels tiers les plus utilisés. Les paquetages peuvent-être gérés facilement avec l'aide de plusieurs utilitaires, également désignés comme les outils `pkg*` :

- [pkg_add\(1\)](#) - un utilitaire pour installer et mettre à jour les paquetages logiciels.
- [pkg_delete\(1\)](#) - un utilitaire pour effacer les paquetages logiciels précédemment installés.
- [pkg_info\(1\)](#) - un utilitaire pour afficher des informations sur les paquetages logiciels.
- [pkg_create\(1\)](#) - un utilitaire pour créer des paquetages logiciels.

Afin de fonctionner correctement, une application X peut exiger de que d'autres applications Y et Z soient installées. L'application X serait dépendante de ces autres applications, c'est pourquoi Y et Z s'appellent les *dépendances* de X. Alternativement, Y peut exiger d'autres applications P et Q, et Z peut exiger l'application R pour fonctionner correctement. De cette façon, un *arbre entier de dépendance* est formé.

Les paquetages ressemblent à de simples archives `.tgz`. Fondamentalement ils sont juste cela, mais il y a une différence cruciale : ils contiennent des *informations supplémentaires sur le paquetage*. Ces informations sont utilisées par [pkg_add\(1\)](#) pour plusieurs buts :

- Différents contrôles : le paquetage a-t-il déjà été installé ou est-il en conflit avec d'autres paquetages installés ou noms de fichier ?
- Les dépendances qui ne sont pas encore présentes sur le système sont automatiquement récupérées et installées, avant de procéder à l'installation du paquetage.
- L'information sur le(s) paquetage(s) sont enregistrées dans un répertoire centralisé, par défaut il se trouve dans `/var/db/pkg`. Ceci, entre autres, empêchera les dépendances d'un paquetage d'être supprimée avant que le paquetage lui-même ait été supprimé. Ceci aide à s'assurer qu'une application ne peut pas être accidentellement cassée par un utilisateur négligent.

15.2.2 - Faire les choses simplement: PKG_PATH

Vous pouvez faire les choses vraiment très simplement en utilisant la variable d'environnement `PKG_PATH`. Faites la juste pointer à votre endroit favoris, et `pkg_add(1)` ira automatiquement regarder les paquetages que vous spécifierez, **et** aussi récupérera et installera automatiquement les dépendances nécessaires à ce paquetage.

Une liste des endroits possibles pour récupérer des paquetages est donnée dans la [section suivante](#).

Exemple 1: récupérer de votre [CDROM](#), en assurant qu'il est bien monter sur `/mnt/cdrom`

```
$ export PKG_PATH=/mnt/cdrom/4.7/packages/~machine -a`/
```

Exemple 2: récupérer d'un [miroir FTP](#) proche

```
$ export PKG_PATH=ftp://your.ftp.mirror/pub/OpenBSD/4.7/
```

```
packages/~machine -a`/`
```

C'est habituellement une bonne idée d'ajouter une ligne similaire aux exemples ci-dessus dans votre `~/.profile`. Comme avec la variable classique `PATH`, vous pouvez spécifier plusieurs endroits séparés par deux points. **Avant OpenBSD 4.4, tous les chemins de la variable `PKG_PATH` DOIVENT se terminer par un slash (/)**. De cette façon, `pkg_add(1)` peut découper le chemin correctement même si il possède des URL contenant des deux points. Si la première entrée dans `PKG_PATH` échoue, la suivante est essayée, ainsi de suite, jusqu'à ce que le paquetage soit trouvé. Si toutes les entrées échouent, une erreur est générée.

Il est à noter l'utilisation de [machine\(1\)](#) dans les lignes de commandes précédentes. Cela va automatiquement substituer votre "architecture d'application" OpenBSD installée, qui est normalement, mais pas toujours, le nom de votre plate-forme. Bien sur, si vous utilisez les snapshots, vous devrez remplacer "4.7" par "snapshots".

15.2.3 - Trouver des paquetages

Une large collection de paquetages pré-compilés est disponible pour les architectures les plus courantes. Regardez juste pour vos paquetages à l'un de ces endroits :

- L'un des trois [CD-ROMs](#), dépendant de votre architecture. Les CD-ROMs embarquent seulement les plus courants utilisés, paquetages librement distribuables pour les plates-formes les plus couramment utilisées.
- Sur les [serveurs FTP miroirs](#). Les paquetages se trouvent dans le répertoire `/pub/OpenBSD/4.7/packages`. De là, les paquetages sont découpés en fonction de leur architecture.

Si vous avez l'arbre des ports sur votre système, vous pouvez rapidement trouver le paquetage que vous recherchez comme expliqué dans [Rechercher dans l'arbre des ports](#).

Vous pouvez remarquer que certains paquetages sont disponibles dans différentes variétés, formellement appelées **saveurs** ("**flavors**"). D'autres sont des morceaux de la même application qui peuvent être installés séparément. Ils sont appelés **sous-paquetages** ("**subpackages**"). Cela sera détaillé plus loin dans [Utiliser les saveurs \("flavors"\) et les sous-paquetages \("subpackages"\)](#) mais cela signifie simplement qu'un saveur est un paquetage installé avec un groupe d'options différentes. Actuellement, plusieurs paquetages possèdent des saveurs, par exemple : le support de base de données, le support de système sans X ou l'ajout de réseaux comme SSL et IPv6. Chaque saveur d'un paquetage aura un suffixe différent dans le nom de son paquetage. Pour des informations plus détaillées sur le nom des paquetages, veuillez vous reporter à [packages-specs\(7\)](#).

Note : Tous les paquetages possibles ne sont pas nécessairement disponibles sur les serveurs FTP !

Certaines applications ne fonctionnent simplement pas sur toutes les architectures. Certaines applications ne peuvent pas être distribuées par FTP (ou CDROM) pour des raisons de licences. Il y a aussi de multiples combinaisons de saveurs d'un port, et le projet OpenBSD ne possède pas les ressources pour les construire toutes. Si vous avez besoin d'une combinaison qui n'est pas disponible, vous devrez fabriquer le port à partir des sources. Pour plus d'informations pour savoir comment faire cela, lire [Utiliser les saveurs \("flavors"\) et les sous-paquetages \("subpackages"\)](#) dans la section Ports de ce document.

15.2.4 - Installer de nouveaux paquetages

Pour installer de nouveaux paquetages, l'utilitaire `pkg_add(1)` est utilisé. Si vous avez [fait les choses simplement](#) pour vous-même en configurant la variable d'environnement `PKG_PATH`, vous pouvez juste appeler `pkg_add(1)` avec le nom du paquetage, comme dans l'exemple simple suivant.

```
$ sudo pkg_add -v screen-4.0.3p1
parsing screen-4.0.3p1
installed /etc/screenrc from /usr/local/share/examples/
screen/screenrc | 71%
screen-4.0.3p1: complete
```

Dans cet exemple l'option `-v` a été utilisé pour obtenir une sortie plus bavarde. Cet option n'est pas nécessaire mais elle est utile pour déboguer et a été utilisée ici pour donner un peu plus de perspicacité dans ce que `pkg_add(1)` est entrain de faire. Notez le message mentionnant `/etc/screenrc`. Spécifier plusieurs fois l'option `-v` produira une sortie encore plus bavarde.

Utiliser `pkg_add(1)` en mode interactif

Depuis OpenBSD 3.9, `pkg_add(1)` possède un mode interactif qui peut être activé en l'invoquant avec l'option `-i`, et cela aura pour conséquence de vous posez des questions quand il ne pourra pas prendre des décisions par lui-même. Par exemple, si vous ne connaissez pas le numéro de version d'un paquetage à l'avance, vous pouvez essayer quelque chose comme :

```
$ sudo pkg_add -i screen
Ambiguous: screen could be screen-4.0.3p1 screen-4.0.3p1-
shm screen-4.0.3p1-static
Choose one package
    0: <None>
    1: screen-4.0.3p1
    2: screen-4.0.3p1-shm
    3: screen-4.0.3p1-static
Your choice: 1
screen-4.0.3p1: complete
```

Pour certains paquetages, des informations additionnelles importantes vous seront données par rapport à la configuration ou l'utilisation de l'application sur un système OpenBSD. Comme c'est important, elles seront affichées que vous utilisiez ou non l'option **-v**. Considérez l'exemple suivant:

```
$ sudo pkg_add ghostscript-fonts-8.11
ghostscript-fonts-8.11: complete
You may wish to update your font path for /usr/local/share/
ghostscript/fonts
--- ghostscript-fonts-8.11 -----
To install these fonts for X11, just make sure that the
fontpath
lists the 75dpi or 100dpi bitmap fonts before the
ghostscript fonts,
and make sure you have the string ":unscaled" appended to
the bitmap
font's fontpath. This way, the bitmap fonts will be used if
they
match, and the Type 1 versions will be used if the font
needs to be
scaled. Below is the relevant section from a typical xorg.
conf file.
```

```
FontPath    "/usr/X11R6/lib/X11/fonts/misc/"
FontPath    "/usr/X11R6/lib/X11/fonts/75dpi:unscaled"
FontPath    "/usr/X11R6/lib/X11/fonts/100dpi:unscaled"
FontPath    "/usr/local/lib/X11/fonts/ghostscript/"
FontPath    "/usr/X11R6/lib/X11/fonts/Type1/"
```

Continuons maintenant avec l'exemple d'un paquetage qui possède des dépendances :

```
$ sudo pkg_add -v tin-1.8.2p0
parsing tin-1.8.2p0
Dependencies for tin-1.6.2 resolve to: gettext-0.14.6,
libutf8-0.8, pcre-6.4p1, libiconv-1.9.2p3 (todo: libiconv-
1.9.2p3,gettext-0.14.6,pcre-6.4p1,libutf8-0.8)
tin-1.8.2p0:parsing libiconv-1.9.2p3
tin-1.8.2p0:libiconv-1.9.2p3: complete
tin-1.8.2p0:parsing gettext-0.14.6
Dependencies for gettext-0.14.6 resolve to: expat-2.0.0,
libiconv-1.9.2p3 (todo: expat-2.0.0)
tin-1.8.2p0:parsing expat-2.0.0
tin-1.8.2p0:expat-2.0.0: complete
```

```
tin-1.8.2p0:gettext-0.14.6: complete
tin-1.8.2p0:parsing pcre-6.4p1
tin-1.8.2p0:pcre-6.4p1: complete
tin-1.8.2p0:parsing libutf8-0.8
tin-1.8.2p0:libutf8-0.8: complete
tin-1.8.2p0: complete
```

Encore une fois nous avons ajouté l'option **-v** pour en voir plus de ce qui se passe. Après des investigations dans les informations du paquetage, des dépendances ont été trouvées et celles-ci ont été installées en premier. Quelque part au milieu vous pouvez voir le paquetage `gettext` qui a été installé, qui dépendait de `libiconv`. Avant d'installer `gettext`, les informations sur le paquetage ont été examinées et il a été vérifié que `libiconv` a bien été installé.

Il est possible de spécifier plusieurs noms de paquetages sur une ligne, ils seront tous installés en une fois avec les dépendances possibles.

Si pour une raison vous décidez de ne pas utiliser `PKG_PATH`, il est aussi possible de spécifier la position absolue d'un paquetage via la ligne de commande. Cette position absolue doit être un chemin local ou une URL se référant à une position FTP, HTTP ou SCP. Considérons l'installation via FTP dans l'exemple suivant:

```
$ sudo pkg_add ftp://ftp.openbsd.org/pub/OpenBSD/4.7/
packages/~machine -a~/screen-4.0.3p1.tgz
screen-4.0.3p1: complete
```

Dans cet exemple l'option **-v** n'a pas été utilisée donc seulement les messages nécessaires ont été affichés. Vous pouvez noter que le nom de fichier complet a été tapé en ajoutant le suffixe **.tgz**. Vous pouvez aussi ne pas mettre le suffixe comme dans les exemples précédents car il est auto-complété par `pkg_add(1)`.

Note : Toutes les architectures n'ont pas les mêmes paquetages disponibles. Certains ports ne fonctionnent pas sur certaines architectures, et les performances limitent le nombre de paquetages qui peuvent être construits.

Par sécurité, si vous installez un paquetage que vous avez déjà installé et supprimé précédemment (ou une ancienne version de celui-ci), `pkg_add(1)` ne re-écrit **pas** les fichiers de configuration qui ont été modifiés. A la place, il vous en informera comme dans l'exemple suivant (seulement en utilisant l'option **-v**, cependant !):

```
$ sudo pkg_add -v screen-4.0.3p1
parsing screen-4.0.3p1
The existing file /etc/screenrc has NOT been
```



```

changed** | 71%
It does NOT match the sample file /usr/local/share/examples/
screen/screenrc
You may wish to update it manually
screen-4.0.3p1: complete

```

Quelquefois vous pouvez rencontrer une erreur comme dans l'exemple suivant :

```

$ sudo pkg_add xv-3.10ap4
xv-3.10ap4:jpeg-6bp3: complete
xv-3.10ap4:png-1.2.14p0:
complete
xv-3.10ap4:tiff-3.8.2p0:
complete
Can't install xv-3.10ap4: lib not found X11.9.0
Even by looking in the dependency tree:
    tiff-3.8.2p0, jpeg-6bp3, png-1.2.14p0
Maybe it's in a dependent package, but not tagged with
@lib ?
(check with pkg_info -K -L)
If you are still running 3.6 packages, update them.

```

Il y a `pkg_add(1)` installant bien des dépendances, quand soudain l'installation de `xv` est avortée. C'est une autre mesure de sécurité qui est disponible depuis OpenBSD 3.7. Les informations sur le packaging embarquées dans le packaging lui-même inclus des informations sur les bibliothèques partagées que le packaging espère être déjà installées, les bibliothèques systèmes comme les bibliothèques tierces. Si une des bibliothèques nécessaires ne peut pas être trouvée, le packaging ne sera pas installé car il ne fonctionnerait pas de toute façon.

Pour résoudre ce type de conflit, vous devez découvrir quoi installer dans l'ordre afin d'obtenir les bibliothèques nécessaires sur votre système. Il y a plusieurs choses à vérifier :

- Vous pouvez avoir de vieux paquetages installés : une vieille version de la bibliothèque requise est présente. Dans ce cas, mettez à jour ces paquetages.
- Votre système est peut-être incomplet : vous n'avez pas installé l'un des paquetages qui contenait la bibliothèque requise. Il faut juste [ajouter un paquetage après l'installation](#).
- Votre système est peut-être dépassé : vous avez une ancienne version de la bibliothèque requise. Démarrez une image d'installation (comme détaillé dans [FAQ 4](#)), et choisissez (U)pgrade de votre système complet.

15.2.5 - Lister les paquetages installés

Vous pouvez voir une liste des paquetages installés en utilisant l'utilitaire [pkg_info\(1\)](#).

```
$ pkg_info
aterm-0.4.2p1      color vt102 terminal emulator with
transparency support
bzip2-1.0.4       block-sorting file compressor,
unencumbered
expat-2.0.0       XML 1.0 parser written in C
fluxbox-0.9.15.1p0 window manager based on the original
Blackbox code
gettext-0.14.6    GNU gettext
imlib2-1.3.0      image manipulation library
jpeg-6bp3         IJG's JPEG compression utilities
libiconv-1.9.2p3  character set conversion library
libltdl-1.5.22p1  GNU libtool system independent dlopen
wrapper
libungif-4.1.4p0  tools and library routines for working
with GIF images
libutf8-0.8       provides UTF-8 locale support
mutt-1.4.2.2i     tty-based e-mail client
pcre-6.4p1        perl-compatible regular expression
library
png-1.2.14p0      library for manipulating PNG images
screen-4.0.3p1    multi-screen window manager
tcsh-6.14.00p1    extended C-shell with many useful
features
tiff-3.8.2p0      tools and library routines for working
with TIFF images
tin-1.8.2p0       threaded NNTP and spool based UseNet
newsreader
```

En donnant le nom d'un paquetage installé (ou l'endroit où se trouve un paquetage qui doit être installé), `pkg_info(1)` donnera plus d'informations détaillées sur ce paquetage spécifique.

15.2.6 - Mettre à jour les paquetages installés

Disons que vous avez une vieille version de `unzip` installée avant de mettre à jour cette machine de OpenBSD 4.6 à 4.7. Maintenant vous pouvez simplement mettre à jour vers le nouveau paquetage 4.7 comme suit :

```
$ sudo pkg_add -u unzip
unzip-5.52p0 (extracting): complete
```

```
unzip-5.52 (deleting): complete
unzip-5.52p0 (installing): complete
Clean shared items: complete
```

Invoquer `pkg_add(1)` avec l'option **-u** et sans nom de paquetage va tenter de mettre à jour tous les paquetages installés. examiner tous les paquetages installés pour des versions mises à jour. Quand un paquetage possède des dépendances, elles seront aussi examinées pour mises à jour.

Note : Le mécanisme **-u** est relié à la variable d'environnement `PKG_PATH`. Si elle n'est pas définie, `pkg_add(1)` ne sera pas capable de trouver les mises à jour.

A partir d'OpenBSD 4.2, le fait d'avoir plusieurs entrées dans `PKG_PATH` ne veut plus dire que toutes les entrées seront vérifiées à la recherche de mises à jour. Au lieu de cela, `pkg_add(1)` s'arrêtera à la première entrée qui contient des candidats convenables.

Si vous aviez un fichier de configuration appartenant à la vieille version que vous avez modifié, il sera laissé intacte par défaut. Vous pouvez, cependant, le remplacer par le fichier de configuration de la nouvelle version en appelant `pkg_add(1)` avec l'option **-c**.

15.2.7 - Supprimer les paquetages installés

Pour supprimer un paquetage, il suffit de prendre simplement le nom propre du paquetage comme vu avec `pkg_info(1)` (regardez [Lister les paquetages installés](#) ci-dessus) et utilisez [pkg_delete\(1\)](#) pour supprimer le paquetage. Dans l'exemple suivant le paquetage `screen` sera supprimé. Notez que dans certaines occasions il y a des instructions sur des éléments supplémentaires qui ont besoin d'être supprimés que `pkg_delete(1)` ne supprimera pas pour vous. Comme avec l'utilitaire `pkg_add(1)`, vous pouvez utiliser l'option **-v** pour obtenir une sortie plus bavarde.

```
$ sudo pkg_delete screen
screen-4.0.3p1: complete
Clean shared items: complete
```

Note : Souvent, il n'est pas nécessaire de spécifier le numéro de version et les saveurs avec le nom du paquetage, puisque `pkg_delete(1)` sera capable de trouver le nom du paquetage complet de lui-même. Vous devez spécifier le nom complet du paquetage (dans l'exemple c'est `screen-4.0.3p1`) seulement si une ambiguïté est possible due à l'installation multiple de paquetages avec le nom spécifié. Dans ce cas `pkg_delete(1)` ne peut pas savoir quel paquetage supprimer.

Par précaution, `pkg_delete(1)` ne supprimera pas des fichiers de configuration si ils ont été modifiés. Au lieu de cela il vous en informera comme suit :

```
$ sudo pkg_delete screen
```

```
screen-4.0.3p1: complete
Clean shared items: complete
--- screen-4.0.3p1 -----
You should also remove /etc/screenrc (which was modified)
```

Si vous préférez avoir ces fichiers de configuration supprimés automatiquement, vous pouvez le faire en utilisant l'option **-c**.

15.2.8 - Suppression ou installation incomplète de paquetage

Dans quelques rares cas, vous pouvez avoir un paquetage qui n'a pas été installé ou supprimé complètement à cause de conflits avec d'autres fichiers. L'installation incomplète est normalement marquée avec un préfixe "partial-" au nom du paquetage. Cela peut, dans certains cas, intervenir quand vous pressez accidentellement CTRL+C durant l'installation :

```
$ sudo pkg_add screen-4.0.3p1
screen-4.0.3p1:
complete
7%
Adjusting md5 for /usr/local/info/screen.info-3 from
49fb3fe1cc3a3b0057518459811b6dac to
3b9c7811244fb9f8d83bb27d3a0f60d8
/usr/sbin/pkg_add: Installation of screen-4.0.3p1 failed ,
partial installation recorded as partial-screen-4.0.3p1
```

C'est toujours une bonne idée de supprimer un paquetage partiel de son système et de corriger les problèmes potentiels qui ont conduit à cette erreur. C'est souvent l'indication que vous n'avez pas un système propre avec tout d'installé à partir des paquetages, mais sûrement des paquetages mélangés avec d'autres logiciels installés à partir des sources.

15.3 - Utilisation des ports

Comme mentionné dans l'introduction, les paquetages sont compilés à partir de l'arbre des ports. Dans cette section nous allons vous expliquer comment l'arbre des ports fonctionne, quand vous devez l'utiliser et comment vous pouvez l'utiliser.

NOTE IMPORTANTE : L'arbre de ports est destiné aux utilisateurs avancés. **Tout le monde est encouragé à utiliser les [paquetages binaires pre-compilés](#)**. NE posez pas de questions de débutant sur les listes de diffusion comme "Comment est-ce que je peux faire fonctionner l'arbre des ports ?". Si vous avez des questions sur l'arbre des ports, il est considéré que vous avez lu les pages du manuel, cette FAQ et que vous êtes capable de travailler avec.

15.3.1 - Comment cela fonctionne-t-il ?

L'arbre des ports, un concept à l'origine emprunté à [FreeBSD](#), est un ensemble de Makefiles, un pour chaque application tierce, pour contrôler

- où et comment récupérer les sources du logiciel,
- de quel autre logiciel il dépend,
- comment modifier les sources (si nécessaire),
- comment le configurer et le compiler,
- comment le tester (optionnel),
- comment l'installer.

En dehors du Makefile, chaque port contient aussi au moins ce qui suit :

- un fichier PLIST ou liste de paquetage, qui contient les instructions pour la création du paquetage dès que l'application a été compilé,
- un fichier DESCR ou description de l'application,
- un fichier distfile contenant les tailles et sommes de contrôle ("checksums") des fichiers fournis.

Toutes ces informations sont contenues dans une hiérarchie de répertoires sous `/usr/ports`. Cette hiérarchie contient trois sous-répertoires spéciaux :

- `distfiles/` - où le système des ports stocke l'ensemble de la distribution des logiciels après téléchargement.
- `infrastructure/` - le répertoire principal de l'infrastructure des ports, contenant tous les scripts nécessaires et makefiles.
- `packages/` - contenant tous les paquetages binaires construits par le système des ports.

Tous les autres sous-répertoires forment les différentes catégories d'applications, qui contiennent les sous-répertoires des ports réels. Les ports complexes doivent être organisés à un niveau encore plus profond, par exemple si ils ont un noyau principal et un ensemble d'extensions, ou une version stable et une snapshot de l'application. Tous les répertoires de port doivent contenir un sous-répertoire `pkg/` qui contient une liste(s) de paquetage et un fichier de description(s). Ils peuvent aussi contenir un sous-répertoire `patches/` et `files/`, pour les patchs source et les fichiers additionnels, respectivement.

Quand un utilisateur exécute [make\(1\)](#) dans un sous-répertoire d'un port spécifique, le système va récursivement traverser toutes les dépendances de l'arbre, vérifier si les dépendances exigées sont installées, compiler et installer toute dépendance manquante, et enfin continuer la construction du port désiré. Toute la construction se fait dans le *répertoire de travail* que le port crée. Normalement c'est sous `${WRKOBJDIR}`, qui est par défaut `/usr/ports/pobj`, mais vous pouvez le surcharger (voir [Configuration du système des ports](#)). Si `WRKOBJDIR` a été explicitement désactivé, un sous-répertoire dans le

répertoire principal du port (nom du paquetage prefixé par "w-") sera utilisé à la place.

Note : Les ports ne sont jamais directement installés sur votre système ! Ils utilisent un *faux répertoire d'installation*. Tous ce qui est installé ici est empaqueté ensemble dans un paquetage (qui est stocké dans le sous-répertoire `packages/` de l'arbre des ports comme mentionné précédemment). Installer un port signifie réellement : création d'un paquetage, et enfin installation de ce paquetage !

Plus d'informations sur le système des ports peut-être trouvé dans les pages de manuel suivantes :

- [ports\(7\)](#) - décrit les différents stades (création de cibles) de l'installation des ports, l'utilisation des saveurs et des sous-paquetages et quelques autres options.
- [bsd.port.mk\(5\)](#) - des informations détaillées sur la création de cibles, les variables, le faux cadre de travail (répertoire d'installation), etc...

15.3.2 - Récupérer l'arbre des ports

Avant de continuer, vous devez lire la section sur comment **NE PAS [mélanger votre système OpenBSD et l'arbre des ports](#)**. Dès que vous avez décidé quel saveur de l'arbre des ports vous désirez, vous pouvez récupérer l'arbre des ports de différentes sources. La table ci-dessous donne une vue d'ensemble d'où vous pouvez trouver les différentes saveurs, et sous quelle forme. Un 'x' marque la disponibilité et un '-' qu'il n'est pas disponible en utilisant les sources spécifiées.

Source	Form	Flavor			
		-release	-stable	snapshots	-current
CD-ROM	.tar.gz	x	-	-	-
FTP mirrors	.tar.gz	x	-	x	-
AnonCVS	cvs checkout	x	x	-	x

Sur le CD-ROM et les miroirs FTP, recherchez un fichier dénommé `ports.tar.gz`. Vous pouvez désarchiver ("untar") ce fichier dans le répertoire `/usr`, ce qui va créer `/usr/ports`, et tous les répertoire en dessous. Par exemple:

```
$ cd /tmp
$ ftp ftp://ftp.openbsd.org/pub/OpenBSD/4.7/ports.tar.gz
$ cd /usr
$ sudo tar xzf /tmp/ports.tar.gz
```

Les snapshots disponibles sur les FTP miroirs sont générés quotidiennement de l'arbre des ports -current. Vous pouvez trouver les snapshots dans le répertoire `pub/OpenBSD/snapshots/` Si vous utilisez

un snapshot de l'arbre des ports, vous devez avoir installé un snapshot correspondant d'OpenBSD. Soyez sûr d'avoir votre arbre des ports et votre système OpenBSD synchrone !

Pour plus d'informations sur la façon d'obtenir l'arbre des ports via AnonCVS, veuillez lire la [page AnonCVS](#) qui contient une liste des serveurs disponibles et un nombre d'exemples.

15.3.3 - Configuration du système des ports

NOTE : Cette section introduit des paramètres de configuration globale additionnels pour construire des applications des ports. Vous pouvez sauter cette section, mais alors vous devrez exécuter plusieurs fois `make(1)` dans les exemples comme `root`.

Parce que le projet OpenBSD ne possède pas les ressources pour auditer tout le code source de toutes les applications dans l'arbre des ports, vous pouvez configurer le système de port pour prendre quelques mesures de sécurité. L'infrastructure des ports est capable d'exécuter toute la construction comme un utilisateur normal, et d'exécuter seulement les étapes qui exigent des privilèges de super-utilisateur comme `root`. Les exemples sont les fausses créations et installations de cibles. Cependant, parce que les privilèges `root` sont toujours requis à un certain point, le système de port ne vous sauvera pas quand vous déciderez de construire une application malveillante.

- Vous pouvez installer [sudo\(8\)](#) et avoir un système de port qui l'utilise pour les tâches qui requiert les droits en super-utilisateur. Ajoutez juste la ligne suivante à `/etc/mk.conf`

```
SUDO=/usr/bin/sudo
```

- Vous pouvez modifier les droits de propriété de l'arbre des ports pour que vous puissiez y écrire comme un utilisateur normal. Dans ce cas, l'utilisateur normal devra être ajouté au groupe `wsrc`, et les répertoires dépendants devront avoir le groupe en écriture.

```
# chgrp -R wsrc /usr/ports
# find /usr/ports -type d -exec chmod g+w {} \;
```

- Vous pouvez avoir le système des ports utiliser [systrace\(1\)](#) en ajoutant la ligne suivante à `/etc/mk.conf`

```
USE_SYSTRACE=Yes
```

Ceci impose la procédure de construction de rester à l'intérieur des répertoires permis, et interdit d'écrire dans des endroits illégaux, réduisant de ce fait considérablement le risque d'un système endommagé. Il est à noter que l'utilisation de `systrace(1)` ajoute environ 20% de temps supplémentaire à la construction.

Il est possible d'utiliser un arbre des ports en lecture seulement en séparant les répertoires qui sont écrits durant la construction du port :

- Le répertoire de travail des ports. Il est contrôlé par la variable `WRKOBJDIR`, qui spécifie le répertoire qui doit contenir les répertoires de travail.
- Le répertoire contenant les fichiers distribués. Il est contrôlé par la variable `DISTDIR`.
- Le répertoire contenant les nouveaux paquetages binaires créés. Il est contrôlé par la variable `PACKAGE_REPOSITORY`.

Par exemple, vous pouvez ajouter les lignes suivantes dans `/etc/mk.conf`

```
WRKOBJDIR=/usr/obj/ports
DISTDIR=/usr/distfiles
PACKAGE_REPOSITORY=/usr/packages
```

Si désiré, vous pouvez aussi changer les droits de propriété de ces répertoires par votre nom d'utilisateur et de groupe local, de sorte que le système des ports puisse créer les répertoires de travail correspondants comme un utilisateur normal.

15.3.4 - Rechercher dans l'arbre des ports

Des que vous avez l'arbre des ports en place sur votre système, il devient très simple de rechercher un logiciel. Il faut juste utiliser `make search key="searchkey"`, comme montré dans l'exemple suivant.

```
$ cd /usr/ports
$ make search key=rsnapshot
Port:    rsnapshot-1.3.1p0
Path:    net/rsnapshot
Info:    remote filesystem snapshot utility
Maint:   Antoine Jacoutot <ajacoutot@openbsd.org>
Index:   net sysutils
L-deps:
B-deps:  :net/rsync
R-deps:  :devel/p5-Lchown :net/rsync
Archs:   any
```

Le résultat de la recherche donne une sympathique vue d'ensemble de chaque application qui a été trouvée : le nom du port, le chemin vers le port, une ligne de description, le mainteneur du port, les mots clés relatifs au port, les dépendances de bibliothèques/compilation/exécutables, et l'architecture sur laquelle le port est connu pour fonctionner.

Ce mécanisme, cependant, est très basique, il exécute juste `awk(1)` sur le fichier d'index des ports. Depuis OpenBSD 4.0, un nouveau port appelé "sqlports" a été créé, permettant de faire des recherches en utilisant SQL de manière très précise. C'est une base de données SQLite, mais fondamentalement n'importe quel format de base de données peut-être créé pour être utilisé dans l'infrastructure des ports. Le port de sqlports inclut le script utilisé pour générer la base de données, qui pourrait être utilisé comme base pour générer des bases de données dans des formats différents.

Il faut juste `pkg_add(1)` le paquetage de sqlports, et dans ce cas, le paquetage de sqlite3 sera démarré. Une session d'exemple peut ressembler à :

```
$ sqlite3 /usr/local/share/sqlports
SQLite version 3.3.12
Enter ".help" for instructions
sqlite> SELECT FULLPKGNAME,COMMENT FROM Ports WHERE COMMENT
LIKE '%statistics%';
Guppi-0.40.3p1|GNOME-based plot program with statistics
capabilities
mailgraph-1.12|a RRDtool frontend for Postfix statistics
R-2.4.1|clone of S, a powerful math/statistics/graphics
language
py-probstat-0.912p0|probability and statistics utilities
for Python
darkstat-3.0.540p1|network statistics gatherer with graphs
pfstat-2.2p0|packet filter statistics visualization
tcpstat-1.4|report network interface statistics
wmwave-0.4p2|Window Maker dockapp to display wavelan
statistics
diffstat-1.43p0|accumulates and displays statistics from a
diff file
sqlite>
```

L'exemple ci-dessus reste une recherche très simple. Avec SQL, à peu près tout peut être recherché avec, incluant les dépendances, les options de configuration, bibliothèques partagées, etc...

15.3.5 - Installation normale : un exemple simple

Pour clarifier les choses, considérons un exemple simple : `rsnapshot`. Cette application possède une dépendance : `rsync`.

```
$ cd /usr/ports/net/rsnapshot
$ make install
===> Checking files for rsnapshot-1.2.9
```

```

>> rsnapshot-1.2.9.tar.gz doesn't seem to exist on this
system.
>> Fetch http://www.rsnapshot.org/downloads/rsnapshot-1.2.9.
tar.gz.
100% |*****|
173 KB    00:02
>> Size matches for /usr/ports/distfiles/rsnapshot-1.2.9.
tar.gz
>> Checksum OK for rsnapshot-1.2.9.tar.gz. (sha1)
==> rsnapshot-1.2.9 depends on: rsync-2.6.9 - not found
==> Verifying install for rsync-2.6.9 in net/rsync
==> Checking files for rsync-2.6.9
>> rsync-2.6.9.tar.gz doesn't seem to exist on this system.
>> Fetch ftp://ftp.samba.org/pub/rsync/old-versions/rsync-
2.6.9.tar.gz.
100% |*****|
792 KB 00:31
>> Size matches for /usr/ports/distfiles/rsync-2.6.9.tar.gz
>> Checksum OK for rsync-2.6.9.tar.gz. (sha1)
==> Verifying specs:  c
==> found c.40.3
==> Extracting for rsync-2.6.9
==> Patching for rsync-2.6.9
==> Configuring for rsync-2.6.9
    [...snip...]
==> Building for rsync-2.6.9
    [...snip...]
==> Faking installation for rsync-2.6.9
    [...snip...]
==> Building package for rsync-2.6.9
Link to /usr/ports/packages/i386/ftp/rsync-2.6.9.tgz
Link to /usr/ports/packages/i386/cdrom/rsync-2.6.9.tgz
==> Installing rsync-2.6.9 from /usr/ports/packages/i386/
all/rsync-2.6.9.tgz
rsync-2.6.9: complete
==> Returning to build of rsnapshot-1.2.9
==> rsnapshot-1.2.9 depends on: rsync-2.6.9 - found
==> Extracting for rsnapshot-1.2.9
==> Patching for rsnapshot-1.2.9
==> Configuring for rsnapshot-1.2.9
    [...snip...]
==> Building for rsnapshot-1.2.9
    [...snip...]

```

```

===> Faking installation for rsnapshot-1.2.9
[...snip...]
===> Building package for rsnapshot-1.2.9
Link to /usr/ports/packages/i386/ftp/rsnapshot-1.2.9.tgz
Link to /usr/ports/packages/i386/cdrom/rsnapshot-1.2.9.tgz
===> rsnapshot-1.2.9 depends on: rsync-2.6.9 - found
===> Installing rsnapshot-1.2.9 from /usr/ports/packages/
i386/all/rsnapshot-1.2.9.tgz
rsnapshot-1.2.9: complete

```

Comme vous pouvez le voir, le système des ports fait beaucoup de choses automatiquement. Il récupère, extrait, modifie le code source, configure et construit (compile) le source, installe les fichiers dans un faux répertoire, crée un paquetage (correspondant à la liste de paquetage) et installe ce paquetage sur votre système (usuellement sous `/usr/local/`). Et il fait cela récursivement **pour toutes les dépendances** du port. Il faut juste voir les lignes "===> Verifying install for ..." et "===> Returning to build of ..." dans la sortie ci-dessus, indiquant le chemin à travers l'arbre des dépendances.

Si une version précédente de l'application que vous voulez installer a déjà été installée sur votre système, vous pouvez utiliser **make update** à la place de **make install**. Cela appellera `pkg_add(1)` avec l'option **-r**.

Note : Les grosses applications nécessitent une quantité importante de ressources système pour être construite. De bons exemples sont les compilateurs comme GCC 4.0 ou le Java 2 Software Development Kit. Si vous obtenez des erreurs du type "out of memory" quand vous construisez ce type de port, ceci a habituellement une de ces deux causes :

- Les limites de vos ressources sont trop restrictives. Ajustez les avec le `ulimit` de `ksh` ou le `limit` de `csh`. Si cela n'aide pas, devenez juste `root` avant de démarrer la construction, ou utilisez [sudo\(8\)](#) avec l'option `-c` pour exécuter la construction avec les ressources limitées par la classe de login (se référer à [login.conf\(5\)](#) pour des détails sur les classes de login).
- Vous n'avez simplement pas assez de RAM sur votre machine.

15.3.6 - Nettoyer après une compilation

Vous voulez probablement nettoyer le répertoire de travail par défaut du port après avoir construit et installé le paquetage.

```

$ make clean
===> Cleaning for rsnapshot-1.2.9

```

En outre, vous pouvez aussi nettoyer les répertoires de travail de toutes les dépendances du port avec

cette cible de make :

```
$ make clean=depends
===> Cleaning for rsync-2.6.9
===> Cleaning for rsnapshot-1.2.9
```

Si vous désirez supprimer l'ensemble des sources de la distribution du port, vous pouvez utiliser

```
$ make clean=dist
===> Cleaning for rsnapshot-1.2.9
===> Dist cleaning for rsnapshot-1.2.9
```

Dans le cas où vous avez compilé de multiples saveurs du même port, vous pouvez nettoyer les répertoires de travail de toutes ces saveurs en une fois en utilisant

```
$ make clean=flavors
```

Vous pouvez aussi effectuer un nettoyage au fur et à mesure que la compilation a lieu en positionnant une variable spéciale. Les répertoires de compilation seront automatiquement nettoyés une fois les paquetages créés :

```
$ make package BULK=Yes
```

15.3.7 - Désinstaller un paquetage de port

Il est très simple de désinstaller un port :

```
$ make uninstall
===> Deinstalling for rsnapshot-1.2.9
rsnapshot-1.2.9: complete
Clean shared items: complete
```

Cela appellera `pkg_delete(1)` pour avoir le paquetage correspondant supprimé de votre système. Si vous le désirez, vous pouvez aussi désinstaller et re-installer un paquetage de port en utilisant

```
$ make reinstall
===> Cleaning for rsnapshot-1.2.9
/usr/sbin/pkg_delete rsnapshot-1.2.9
rsnapshot-1.2.9: complete
Clean shared items: complete
===> Installing rsnapshot-1.2.9 from /usr/ports/packages/
```

```
i386/all/rsnapshot-1.2.9.tgz
rsnapshot-1.2.9: complete
```

Si vous voudriez vous débarrasser des paquetages que vous avez juste construit, vous pouvez faire comme suit :

```
$ make clean=packages
===> Cleaning for rsnapshot-1.2.9
rm -f /usr/ports/packages/i386/all/rsnapshot-1.2.9.tgz
```

15.3.8 - Utiliser les saveurs ("flavors") et les sous-paquetages ("subpackages")

Veillez lire la page du manuel [ports\(7\)](#) qui donne une bonne vue d'ensemble du sujet. Il y a deux mécanismes pour contrôler le paquetage d'un logiciel en fonction des différents besoins.

Le premier mécanisme est appelé **saveurs ("flavors")**. Une saveur indique habituellement un certain ensemble d'options de compilation. Par exemple, quelques applications ont une saveur "no_x11" qui peut être employée sur des systèmes sans X. Certains shells possèdent une saveur "static", qui construira une version avec les bibliothèques liées statiquement. Il y a des ports qui ont différentes saveurs pour les construire avec différents toolkits graphiques. D'autres exemples incluent : le support pour différents formats de bases de données, différentes options réseaux (SSL, IPv6, ...), différentes taille de papiers, etc...

Résumé : Très probablement vous emploierez des saveurs quand un paquet n'a pas été rendu disponible pour la saveur que vous recherchez. Dans ce cas-ci vous indiquerez la saveur désirée et construirez le port vous-même.

La plupart des saveurs de ports auront leur propre répertoire de travail durant leur construction et chaque saveur sera packagée avec un nom de paquetage correspondant pour éviter toute confusion. Pour voir les différentes saveurs d'un port, vous devrez aller dans son sous-répertoire et taper

```
$ make show=FLAVORS
```

Vous devriez aussi regarder les fichiers DESCR d'un port étant donné qu'ils sont supposés décrire les saveurs disponibles.

Le second mécanisme est appelé **sous-paquetages ("subpackages")**. Un porteur de paquetage peut décider de créer des sous-paquetages pour différentes parties de la même application, si elles peuvent être logiquement séparées. Vous verrez souvent des sous-paquetages pour la partie cliente et pour la partie serveur d'un programme. Quelquefois une documentation importante est livrée dans un sous-

paquetage séparé parce qu'elle prend trop d'espace disque. Toute fonctionnalité additionnelle qui a beaucoup de dépendances sera souvent packagée séparément. La personne effectuant le portage devra aussi décider quel sous-paquetage est considéré comme principal, à installer par défaut. D'autres exemples incluent : des suites de tests importants qui sont fournis avec le logiciel, des modules séparés pour supporter différentes choses, etc...

Résumé : Certains ports sont répartis en plusieurs paquetages. `make install` va uniquement installer le sous-paquetage principal.

Pour afficher les différents paquetages émanant d'un port donné, utilisez :

```
$ make show=PKG_NAMES
```

`make install` installera uniquement le premier sous-paquetage. Pour les installer tous, utilisez :

```
$ make install-all
```

Pour afficher les différents sous-paquetages disponibles pour un port, utilisez

```
$ make show=MULTI_PACKAGES
```

Il est possible de sélectionner quel(s) sous-paquetage(s) à installer à partir de l'arbre des ports. Après quelques tests, cette procédure appellera juste [pkg_add\(1\)](#) pour installer le(s) sous-paquetage(s) désiré(s).

```
$ env SUBPACKAGE="-server" make install
```

Note : Le mécanisme de sous-paquetage manipule seulement des paquetages, il ne modifie aucune option de configuration avant de construire le port. Dans ce but vous devez employer les saveurs.

15.4 - FAQ

15.4.1 - J'obtiens toutes sortes d'erreurs folles. Je n'arrive pas du tout à faire fonctionner ces ports.

Il est très probable que vous employiez un système et un arbre des ports qui n'est pas synchrone.

Pardon ?

- Lire TOUT sur [Saveurs \("Flavors"\) OpenBSD](#) : `-release`, `-stable` et `-current`. Un court résumé suit, mais veuillez lire le document mentionné ci-dessus pour avoir une idée du sujet si vous voulez

l'employer.

- [Release](#): Ce qui est sur le CD.
- [Stable](#): Release, avec de la sécurité et plus de fiabilité.
- [Current](#): La version de développement d'OpenBSD.
- Ne PAS récupérer un arbre des ports -current et espérer qu'il fonctionne sur un système en -release ou -stable. C'est l'une des erreurs les plus communes et vous irriterez des personnes quand vous demandez de l'aide en disant "rien ne semble fonctionner !" **Si vous suivez -current, vous aurez besoin d'un système en -current et d'un arbre des ports en -current.** Oui, cela signifie vraiment qu'un nouveau port merveilleux ne fonctionnera pas sur votre "vieux" système -- même si ce système était -current juste quelques semaines auparavant. Gardez en mémoire que si vous utilisez X11 comme une partie de votre système, il doit aussi suivre la branche correspondante !
- Parce que des changements non intrusifs sont faits dans -stable, il est possible d'utiliser des paquetages et ports -release sur un système -stable et vice-versa. Il n'est pas nécessaire de mettre à jour tous les paquetages installés après avoir appliqué quelques patches correctifs d'erreurs sur votre système.

Un autre erreur courante est l'installation oubliée de X11. Même si le port que vous essayez de construire n'a pas de dépendance directe avec X11, un sous-paquetage ou ces dépendances peuvent avoir besoin des entetes X11 ou des bibliothèques. Construire des ports sur un système sans X11 n'est pas supporté, même si vous insistez pour le faire ainsi, vous serez seul pour le gérer. Pour plusieurs ports, il y a cependant, la saveur de paquetage "no_x11" disponible, avec laquelle vous pouvez installer sans avoir besoin de X11 sur votre système.

15.4.2 - Pourquoi la dernière version de mon logiciel favori/préfééré n'est pas disponible ?

Si vous utilisez une version révision ou stable d'OpenBSD, vous ne trouverez pas de mise à jour de paquetage avant la prochaine révision, ou alors si un problème de sécurité intervient et justifie une mise à jour du port dans la branche -stable et du paquetage correspondant.

ATTENTION : NE PAS mélanger les versions des Ports et d'OpenBSD :

En faisant ainsi tôt ou tard (probablement très tôt en fait) cela va vous causer des maux de tête en essayant de résoudre [toutes sortes d'erreurs](#) !

Mais hé, je suis en -current ici !

La collection des ports est un projet de volontaires. Quelquefois le projet n'a simplement pas les ressources en développeur pour avoir tout de mis à jour en permanence. Les développeurs assez souvent reprennent ce qui les intéressent et ce qu'ils peuvent tester dans leur environnement. Vos [donations](#) peuvent faire la différence pour le test des ports sur plus de plates-formes.

Quelques ports isolés peuvent traîner derrière les versions traditionnelles pour cette raison. La collection des ports peut avoir une version de retard d'un programme de Janvier pendant qu'une nouvelle version du programme a été livrée par ces développeurs en mai trois mois plus tard. Souvent c'est une décision réfléchie; la nouvelle version peut avoir des problèmes sur OpenBSD que le mainteneur essaye de corriger, ou alors ils ont fait une version plus mauvaise que la version précédente : OpenBSD peut avoir différents [buts](#) que les autres principaux développeurs dans d'autres projets, qui quelquefois résulte dans une fonctionnalité et un désign ou un choix d'implémentation qui est indésirable du point de vue des développeurs de OpenBSD. La mise à jour peut également être remise à plus tard parce que la nouvelle version n'est pas considérée une mise à jour cruciale.

Si vous avez réellement besoin d'une nouvelle version du port, vous pouvez contacter le mainteneur du port pour qu'il le mette à jour (voir [ci-dessous](#) comment trouver qui est le mainteneur). Si vous pouvez nous [aider](#) avec ceci, tout ira pour le mieux.

15.4.3 - Pourquoi n'y a-t-il pas de paquetage de mon logiciel favori/préfér  ?

Il y a plusieurs raisons possibles   cela :

- Sur les [CD-ROMs](#) d'OpenBSD, il n'y a pas assez d'espace pour inclure tous les paquetages possibles pour toutes les plates-formes possibles. Par cons quent seulement les paquetages les plus utilis s sont inclus sur les CDs. En plus, certains logiciels peuvent seulement  tre redistribu  gratuitement, ceci signifie qu'ils ne peuvent pas  tre inclus sur le CDs. Si vous ne trouvez pas un paquetage sur les CDs, essayez une autre source, comme un miroir FTP.
- Certains logiciels ne peuvent simplement pas  tre redistribu  sous la forme d'un paquetage binaire en suivant leurs licences. D'autres logiciels sont encombr s par des brevets et ne peuvent donc pas  tre redistribu . Si votre logiciel favori/pr f r  tombe dans ces cat gories, vous aurez besoin d'utiliser le port et de le compiler des sources.
- Evidemment, mais parfois oubli  : il n'y a [pas de port de votre logiciel favori/pr f r ](#). Vous pouvez le v rifier en [recherchant dans l'arbre des ports](#). S'il n'y a en effet aucun port de votre logiciel favori/pr f r , vous  tes le bienvenue pour [aider](#).

15.4.4 - Pourquoi n'y a-t-il pas de port de mon logiciel favori/pr f r  ?

La collection des ports est un projet de volontaires. Le d veloppement de port actif est r alis  par un nombre limit  de personnes,   leur temps perdu. Ces personnes r alise souvent de nouveaux ports seulement sur les logiciels qu'ils utilisent directement ou qui les int ressent.

Vous pouvez [aider](#). Voyez pour cr er votre propre port. Il y a de la documentation disponible dans : [OpenBSD Porter's Handbook](#). Lisez la et lisez la encore. Sp cialement la partie sur *maintenir* votre port. Donc essayez de construire un nouveau port, et testez-le soigneusement point par point. Si finalement il

fonctionne parfaitement pour vous, soumettez le à la liste de diffusion des ports à ports@openbsd.org. Vous avez beaucoup de chances de recevoir un retour et des tests d'autres personnes. Si l'essai est réussi, votre port sera considéré comme pris dans l'arbre des ports.

15.4.5 - Pourquoi mon logiciel favori/préfééré ne fait pas partie du système de base ?

Puisque OpenBSD est supposé être un petit système d'exploitation Unix-like autonome, nous devons tracer une ligne quand à quoi inclure. Généralement, pour une application qui doit être incluse dans le système de base :

- Elle doit suivre les grands standards de qualité, suivre les mêmes [buts](#) que le projet OpenBSD.
- Sa licence ne doit pas être trop restrictive et doit être compatible avec la licence BSD.
- Elle ne doit pas être trop volumineuse, afin de maintenir la taille du système de base acceptable.

Plus de réponses à cette question peuvent aussi être trouvées dans la [FAQ 1](#).

15.4.6 - Qu'est ce que je dois utiliser : paquetages ou ports ?

En général, il est **fortement conseillé** d'utiliser les paquetages au lieu de construire une application des ports. L'équipe des ports de OpenBSD considère que les paquetages sont le but de leur effort de portage, pas les ports en eux-mêmes.

Construire un application complexe des sources n'est pas trivial. Non seulement l'application doit être compilée, mais les outils utilisés pour la construire doivent être compilés aussi. Malheureusement, OpenBSD, les outils, et les applications sont en évolution, et souvent, obtenir que tous les morceaux fonctionnent ensemble est un défi. Des que tout fonctionne, une révision d'une des pièces le jour suivant peut tout casser. Tous les [six mois](#), comme une [nouvelle révision](#) d'OpenBSD est réalisée, un effort est fait pour tester la construction de chaque port sur toutes les plates-formes, mais durant le cycle de développement il est courant que certains ports soient cassés.

En plus d'avoir toutes les pièces fonctionnant ensemble, il y a juste la matière en temps et ressource requise pour compiler quelques applications des sources. Un exemple courant est [CVSup](#), un outil communément utilisé pour [suivre l'arbre des ports de OpenBSD](#). Pour installer CVSup sur un système modérément rapide avec une bonne connexion Internet cela prend environ dix secondes -- le temps requis pour télécharger et désarchiver un unique fichier de paquetage de 779Ko. En contraste, construire CVSup sur la même machine des sources est une tache énorme, qui requiert beaucoup d'outils et de "bootstrapper" un compilateur, prenant au moins un heure et demi sur la même machine. D'autres applications comme [Mozilla](#) ou [KDE](#) prennent des heures et un espace disque énorme en plus de beaucoup de RAM/swap pour les construire. Pourquoi aller vers autant de temps et d'efforts quand les programmes sont déjà compilés et attendent sur votre [CD-ROM](#) ou [miroir FTP](#) d'être utilisés ?

Bien évidemment, il y a quelques bonnes raisons d'utiliser les ports au lieu des paquetages dans certains cas :

- Les lois de distributions interdisent OpenBSD de distribuer un paquetage.
- Vous désirez modifier ou déboguer l'application ou étudier son code source.
- Vous avez besoin d'une saveur du port qui n'est pas construite par l'équipe des ports d'OpenBSD.
- Vous désirez modifier la structure des répertoires (par exemple modifier PREFIX ou SYSCONFDIR).

Cependant, pour la plupart des gens et des applications, utiliser les paquetages est plus simple, et définitivement la méthode recommandée pour ajouter des applications à un système OpenBSD.

15.4.7 - Comment puis-je optimiser ces ports pour obtenir le maximum de performance ?

OpenBSD est orienté stabilité et sécurité. Comme le noyau GENERIC qui est par défaut et le seul noyau supporté, l'équipe des ports s'assure que les ports fonctionnent et sont stables. Si vous voulez modifier plusieurs options de compilation, c'est de votre propre chef. S'il vous plaît ne posez pas de questions sur les listes de diffusion sur le pourquoi cela ne fonctionne plus quand vous essayez quelques options pour que cela s'exécute plus rapidement. En général, toutes ces optimisations ne sont pas nécessaires pour plus de 99% des utilisateurs, et c'est souvent une perte de temps complète pour vous, les utilisateurs, aussi bien que pour les développeurs qui lisent vos "problèmes" alors qu'en réalité il n'y en a aucun.

15.4.8 - J'ai soumis un nouveau port/une mise à jour depuis plusieurs semaines. Pourquoi cela n'a pas été intégré ("committed") ?

L'équipe des ports possède des ressources limitées et aucun "committer" n'est capable de regarder à votre port/mise à jour dans les temps. C'est très frustrant mais ignorez ce fait. Prenez soin de votre port, envoyez des mises à jour et éventuellement quelqu'un s'occupera de lui. Il est déjà arrivé dans le passé que des personnes ont soudain du temps de libre pour intégrer des ports ou leur intérêt à changé vers de nouveaux domaines et soudainement votre ancien envoi devient intéressant, si l'on s'en souvient.

15.5 - Comment signaler un problème

Si vous avez des soucis avec un port existant, veuillez envoyer un e-mail au mainteneur du port. Pour voir qui est le mainteneur du port, tapez par exemple :

```
$ cd /usr/ports/archivers/unzip
$ make show=MAINTAINER
```

Alternativement, si il n'y a pas de mainteneur, ou si vous ne pouvez pas le/la contacter, envoyez un e-mail à la liste de diffusion des ports d'OpenBSD ports@openbsd.org. Veuillez NE PAS utiliser la liste de diffusion misc@openbsd.org pour des questions sur les ports.

Dans tous les cas veuillez fournir :

- Votre version d'OpenBSD incluant tous les patches que vous avez appliqué. Le numéro de version du noyau est donné par : `sysctl -n kern.version`
- La version de votre arbre des ports : si le fichier `/usr/ports/CVS/Tag` existe, fournissez en le contenu. Si le fichier est absent, vous devez utiliser l'arbre des ports `-current`.
- Une description complète du problème. Ne soyez pas effrayé de fournir des détails Mentionnez toutes les étapes que vous avez suivi avant que le problème survienne. Est-ce que ce problème est reproductible ? Plus d'informations vous fournirez, plus d'aide vous recevrez.

Pour les ports qui ne se construisent pas correctement, un rapport complet de construction est souvent toujours requis. Vous pouvez utiliser le script `portslogger`, trouvé dans `/usr/ports/infrastructure/build`, pour cela. Un exemple d'exécution de `portslogger` ressemble à :

```
$ mkdir ~/portslogs
$ cd /usr/ports/archivers/unzip
$ make clean install 2>&1 | /usr/ports/infrastructure/build/
portslogger \
    ~/portslogs
```

Après cela, vous devez avoir un fichier de log de la construction dans votre répertoire `~/portslogs` que vous pouvez envoyer au mainteneur du port. Toutefois, soyez sur que vous n'avez pas utilisé des options spéciales lors de la construction, par exemple dans `/etc/mk.conf`.

Alternativement, vous pouvez

- Utiliser [script\(1\)](#) pour créer un rapport complet de construction. Ne supprimez pas les informations de configuration.
- Attachez la sortie à [pkg_info\(1\)](#) si cela semble approprié
- Les erreurs internes du compilateur [gcc\(1\)](#) vous demande de reporter les bogues à la liste de diffusion `gcc`. Cela économise du temps si vous suivez ces directives, et fournissez au moins les divers fichiers produits par `gcc -save-temps`.

15.6 - Comment nous aider

Il y a plusieurs façons de nous aider. Elle sont listée ci-dessous, par ordre croissant de difficulté.

- [Signaler les problèmes](#) que vous rencontrez.
- Vous pouvez systématiquement tester les ports et reporter les cassures ("breakages"), ou suggérer des améliorations. Jetez un oeil au [Guide de Tests des Ports](#).
- Testez les mises à jour des ports qui sont postées dans la liste de diffusion des ports.
- Envoyez des mises à jour ou des patches ou mainteneur du port, ou sur la liste de diffusion des ports si il n'y a pas de mainteneur du port. Généralement cela est très apprécié, sauf si vos patches causent aux développeurs une perte de temps au lieu d'un gain de celui-ci.
- Créer de nouveaux ports. Si vous êtes très affamé et voulez savoir tout sur le portage d'applications sur OpenBSD, un bon point de démarrage est [OpenBSD Porter's Handbook](#).

Note : Pour toute création de nouveaux ports et des tests conséquents, ou pour tester des mises à jour de port, vous **devez avoir un système -current !** En général, ce n'est pas un environnement désirable à cause de ces changements permanents par nature, donc procédez seulement si vous êtes sure de pouvoir mettre à jour vous-même le système -current.

[\[FAQ Index\]](#) [\[Section 14 - Configuration des disques\]](#)



www@openbsd.org

\$OpenBSD: faq15.html,v 1.25 2010/06/12 19:51:10 ajacoutot Exp \$