Le système d’exploitation NetBSD

guide rapide

Federico Lupi

Le système d’exploitation NetBSD: guide rapide
par Federico Lupi

Traduit et revu par Guillain Seuillot

Table des matières

Liste des illustrations

Liste des exemples

But de ce guide

Ce guide décrit l'installation et la configuration du système d'exploitation NetBSD. Il s'adresse principalement aux personnes venant d'autres systèmes d'exploitation en espérant apporter la solution aux petits problèmes qui ne manquent pas de survenir en utilisant de nouveaux outils.

Ce guide ne permet pas d'apprendre Unix : une connaissance de base de quelques concepts et outils est conseillée pour le comprendre. Vous devez, par exemple, savoir ce qu'est un fichier et un répertoire ainsi que comment utiliser un éditeur. Il existe de nombreux ouvrages sur le sujet, donc, si vous ne savez pas de quoi il s'agit, je vous conseille de vous en procurer un. Je pense qu'il est préférable de choisir un livre général et d'éviter les titres du genre "Apprendre Unix-XYZ version 1.2.3.4 en 10 jours" mais c'est un avis personnel. Si vous installez un système BSD, vous serez, tôt ou tard, confronté à l'éditeur vi: sans documentation, cet obstacle peut devenir insurmontable. Lorsque le système sera complètement installé, vous serez libre de choisir les éditeurs et les programmes que vous voudrez.

Il reste encore du travail avant d'avoir fini cette courte introduction à NetBSD: quelques chapitres ne sont pas encore finis (certains pas encore commencés) et certains sujets sont encore en test (oui, un guide doit aussi être testé). J'essaierai de travailler dessus et de l'améliorer dans mes moments libres mais si vous voulez m'aider, vous êtes les bienvenus: vous pouvez écrire de nouveaux chapitres (ou des parties) ou m'envoyer des corrections à apporter.

`flupi@mclink.it`

Chapitre 1. Licence

Copyright (c) 1999, 2000, 2001 Federico Lupi. Tout droits réservés.

La redistribution et l'utilisations des formes source et binaire, avec ou sans modification, sont permises à condition que les points suivants soient respectés:

1. La redistribution du code source doit contenir le copyright ci-dessus, les liste des conditions et les réserves suivantes.
2. La redistribution sous forme binaire doit contenir le copyright ci-dessus, les liste des conditions et les réserves suivantes dans la documentation et/ou tout matériel fourni avec la distribution.
3. Toute publicité mentionnant les caratéristiques ou l'utilisation de ce logiciel doit comporter la menstion suivante: Ce produit contient du logiciel développé par Federico Lupi pour le projet NetBSD.
4. Le nom de l'auteur ne peut pas être utilisé pour crédibiliser ou promouvoir des produits dérivés de ce logiciel sans permission spécifiquement écrite.

CE LOGICIEL EST FOURNIT "TEL QUEL" PAR L'AUTEUR ET IL N'EST DONNÉ AUCUNE GARANTIE, IMPLICITE OU EXPLICITE, QUANT À SON UTILISATION COMMERCIALE, PROFESSIONNELLE OU AUTRE. L'AUTEUR NE PEUT EN AUCUN CAS ÊTRE TENU POUR RESPONSABLES DE QUELQUE DOMMAGE OU PRÉJUDICE DIRECT, INDIRECT, SECONDAIRE OU ACCESSOIRE (Y COMPRIS LES PERTES FINANCIÈRES DUES AU MANQUE À GAGNER, À L'INTERRUPTION D'ACTIVITÉS OU LA PERTE D'INFORMATIONS ET AUTRES) DÉCOULANT DE L'UTILISATION DE CE LOGICIEL MÊME EN CONNAISSANCE DE POSSIBILITÉ DE TELS DOMMAGES.

Chapitre 2. Qu'est-ce que NetBSD ?

NetBSD est un système d'exploitation libre de type Unix, hautement portable et existant sur la plupart des plateformes, de l'AlphaServers 64 bits aux machines portables. Sa conception soignée et ses fonctions avancées en font un excellent outil à la fois pour la recherche et pour la production. Il est approuvé par les utilisateurs grâce à ses sources complètement disponibles. De nombreuses applications sont facilement accessibles.

2.1. L'Histoire de NetBSD

La première version de NetBSD (0.8) date de 1993 et sort, d'une part, du système d'exploitation 4.3BSD Lite, une version d'Unix développée par Berkeley, l'université de Californie, (BSD = Berkeley Software Distribution: Distribution Logicielle de Berkeley) et, d'autre part, du système 386BSD, le premier portage de BSD sur processeur Intel 386. Pendant les années qui suivirent, les modifications apportées à 4.4BSD Lite (la dernière version du groupe de Berkeley) ont été intégrées au système. La branche BSD d'Unix a eu une énorme influence dans l'histoire de ce système auquel il a apporté de nombreux outils, idées et améliorations (l'éditeur vi, l'interpréteur C, le contrôle des tâches, le système de fichiers rapide de Berkeley, les signaux fiables, la mémoire virtuelle, l'implémentation de TCP/IP, pour en nommer quelques uns) qui sont désormais des standards dans tous les environnements Unix. Cette tradition de recherche et de développement survie aujourd'hui dans les sytèmes BSD, libres et commerciaux, et, en particulier, dans NetBSD.

2.2. Fonctions de NetBSD

NetBSD fonctionne sur un grand nombre de plateforme et est largement portable, probablement le système d'exploitation le plus portable du monde. Les sources complètes du noyau et des programmes sont disponibles pour toutes les plateformes supportées. Voyez, pour plus de détails, le site officiel du Projet NetBSD (http://www.netbsd.org/).

Une liste détaillée des fonctions de NetBSD se trouve ici: http://www.netbsd.org/Misc/features.html.

Les fonctions de base de NetBSD sont:

- Portabilité (plus de vingt plateformes supportées)
- Précision et qualité du code
- Adhésion aux standards
- Recherche et innovation

Les caractéristiques ci-dessus apportent aussi des avantages indirects. Par exemple, si vous travaillez sur une seule plateforme, vous pouvez penser que vous n'êtes pas intéressé par la portabilité. Pourtant, celle-ci est étroitement liée à la qualité du code: sans un code de base bien écrit et bien organisé, il serait impossible de supporter tant de plateformes. La qualité du code est la base d'un système bon et solide bien que, étonnament, peu de personnes ne semblent s'en rendre compte. L'attention portée à l'architecture et à la qualité des versions est récompensée par l'énorme potentiel du code de NetBSD et de la qualité de ses pilotes.

Une des caractéristiques propre à NetBSD n'est pas d'être satisfait par des implémentations partielles. Certains systèmes semblent penser que • si ça marche, c'est bon• . NetBSD pourrait plutôt penser que • ça ne marche pas tant que ce n'est pas bon• . Pensez comment certains programmes sur-enrichis s'effondrent sous leur propre poids et • fonctionnalités• et vous comprendrez pourquoi NetBSD veut à tout prix éviter cette situation.

2.3. Plateformes supportées

NetBSD 1.4.1 supporte les plateformes suivantes (vous pouvez trouver des détails techniques sur le site de NetBSD):

- Digital Alpha (64bit)
- Commodore Amiga, MacroSystem DraCo
- Acorn RiscPC/A7000, CATS, Digital Shark, EBSA-285, VLSI RC7500
- Atari TT030, Falcon, Hades
- Hewlett-Packard 9000/300 et 400
- PC IBM de la famille i386 et clones
- Apple Macintosh
- Apple Power Macintosh
- Motorola MVME 68k SBCs
- Matériel NeXT 68k 'black'
- Le PC532
- Digital MIPS-based DECstations et DECsystems
- Sun SPARC
- Sun 3 et Sun3x
- Digital VAX
- Sharp X680x0

2.4. Utilisateurs visés par NetBSD

Le site de NetBSD précise que: • Le projet NetBSD fournit un système disponible librement que les professionnels, les pationnés et les chercheurs peuvent utiliser de la manière qu'ils veulent.• . Je voudrais ajouter que NetBSD est le système idéal si vous voulez apprendre Unix, principalement grâce à son adhésion aux standards (un des buts du projet) et puisqu'il peut tourner sur du matériel considéré comme obsolète par la plupart des autres systèmes. Nous pouvons dire: • pour apprendre Unix, pas besoin d'acheter du matériel dispendieux. Vous pouvez réutiliser le vieux Mac ou PC qui traine dans votre grenier• . Aussi, si vous avez besoin d'un Unix qui tourne correctement sur divers plateformes, NetBSD est probablement le meilleur (et seul) choix.

2.5. Applications pour NetBSD

Lorsque vous installez NetBSD, vous avez un large choix de programmes et d'applications que vous pouvez ajouter au système. En plus de tous les standards d'Unix comme les outils de productivités, éditeurs, formatteurs, compilateurs et débogueurs C/C++... il existe un grand nombre (sans cesse grandissant, je crois qu'il y en a déja plus de 1000) de paquetages à installer à partir des sources ou des versions pré-compilées. Tout ceux que vous vous attendez à trouver sur un système bien configuré sont disponibles librement pour NetBSD et il existe aussi des applications commerciales. Vous pouvez aussi trouver les émulateurs pour faire tourner des binaires issus d'autres systèmes *nix. L'émulation de Linux est sans doutes l'exemple le plus important, de gros efforts y étant apportés et la plupart des utilisateurs de NetBSD l'utilisant. Vous pouvez faire tourner les versions Linux de

- Netscape
- Acrobat Reader
- Doom, Quake
- Adobe FrameMaker
- de nombreux autre programmes

NetBSD est aussi capable d'émuler FreeBSD, BSDI et d'autres systèmes.

2.6. La philosophie de NetBSD

Contrairement à beaucoup d'autres systèmes contemporains, l'installation de NetBSD n'est pas gourmande en place car elle s'attache à produire un système stable et complet sans être redondant. Après l'installation, vous avez un système complet en état de marche dont il manque encore de nombreuses applications comme, par exemple, un navigateur web (NetBSD, contrairement à d'autres systèmes, ne considère pas le navigateur comme une partie du système). Vous êtes entièrement libre de choisir les programmes à installer sur votre machine et celle-ci est facilitée par la collection de paquetages.

Un autre avantage de cette approche est que le système de base fonctionne sans paquetages supplémentaire. Si vous décidez de mettre à jour votre version de Perl, vous n'avez pas peur de casser quelque chose dans votre système. Lorsque vous installez NetBSD, vous ne vous retrouvez pas avec une grande quantité d'applications: vous pouvez penser que c'est un inconvénient mais quand vous commencez à comprendre la philosophie sous-tenante, vous trouvez que c'est la liberté. Lorsque vous installez une collection de logiciels (ce que quelqu'un d'autre a décidé pour vous) vous remplissez votre disque dur de tonnes de programmes dont la majortité reste inutilisés (et inconnus) et ne fait que gacher la place (et peut rendre le système instable). C'est ce que l'utilisateur typique de BSD ne veut pas.

Même lorsque vous commencez à connaitre NetBSD, il y a toujour quelque chose qui continue à vous étonner. L'extrème concision et la logique du système ainsi que l'attention donnée aux détails: rien ne résulte de la chance et tout est finement pensé. Oui, voilà ce qu'est la qualité et, à mon avis, voilà la caractéristique qui distingue NetBSD.

Nous pourrions passer des jours à argumenter sur les mérites relatifs des systèmes d'exploitation (et certaines personnes aiment ça) mais si vous n'essayez pas sérieusement, vous ne pourrez pas juger. Je suis convaincu, parce que je l'ai vu de nombreuses fois sur les listes de diffusion, que si vous essayez NetBSD vous serez conquis par l'équilibre parfait entre la complexité et l'efficacité. Tout problème comporte plusieurs solutions: NetBSD n'est pas satisfait par • une• solution mais essaie toujours de trouver la plus facile et la plus élégante. Dans cet esprit, c'est l'outil optimal. C'est comme utiliser un stylo: vous travaillez dur pour savoir l'utiliser mais, une fois que vous savez, vous pouvez écrire ou dessiner en oubliant compètement le stylo.

2.7. Obtenir NetBSD

Il n'existe aucun • fournisseur• officiel de cédéroms NetBSD mais de nombreux revendeurs. Vous trouverrez la liste la plus à jour sur la page (http://www.netbsd.org/Sites/cdroms.html) appropriée du site de NetBSD. Bien entendu, vous pouvez aussi télécharger NetBSD sur internet depuis l'un des miroirs.

Chapitre 3. Installation

3.1. Documentation

La documentation de NetBSD est en grande partie sous forme de pages de manuel constituant une excellente référence technique sur le système. Je ne nie pas qu'elles n'ont rien d'un guide (sans parler du fait qu'il faut avoir installé NetBSD pour pouvoir les lire) et c'est la raison d'être de ce document.

> **Note :** Vous pouvez aussi lire les pages de manuel par le web mais je ne pense pas que ce soit la façon la plus pratique d'apprendre le système...

Après l'installation, vous trouverez des guides BSD dans le répertoire `/usr/share/doc`. Ils sont divisés en trois sections principales: *psd* (UNIX Programmer's Supplementary Documents: Documents supplémentaires pour le programmeur Unix), *smm* (UNIX System Manager's Manual: Manuel de l'administrateur système) et *usd* (UNIX User's Supplementary Documents: Doccuments supplémentaires pour l'utilisateur Unix). Vous pouvez les lire sur un terminal avec, par exemple:

```
$ cd /usr/share/doc/smm/09.sendmail
$ nroff -me 09.sendmail/intro.me | more
```

ou bien générer un fichier PostScript à l'aide des fichiers makefile.

Il existe un réel manque de guides HOWTO, il vous faut donc faire le maximum avec ceux qui existent. NetBSD contient des documents au format texte et le site web de NetBSD procure d'autres information et FAQ.

> **Documentation originale :** Le site NetBSD contient de nombreuses pages de documentations et HOWTO à la fois génériques et spécifiques à une plateforme. Cette information est correctement écrite et souvent facile à comprendre. Vous pouvez, par exemple, trouver:
>
> - Comment accéder à une partition DOS/Windows depuis NetBSD
> - Comment démarrer NetBSD avec le démarreur de Windows NT
> - ...

Toutes les versions de NetBSD contiennent les fichiers suivants:

INSTALL

Notes d'installation. C'est le document le plus important et vous devez le lire (et relire) attentivement. Il contient la description du système NetBSD, une liste du matériel supporté et, notamment, les instruction d'installation.

README.first

autre fichier à lire.

release.man

décrit la structure de la version de NetBSD que vous êtes en train d'installer. C'est un fichier texte au format *man* placé dans le répertoire des pages de manuel. Il est préformatté et peut être lu avec n'importe quel éditeur de texte.

Sur le site web de NetBSD, en plus des autres, vous trouverez les guides suivants:

NetBSD FAQ

informations générales et liens vers d'autres FAQ.

NetBSD/i386 FAQ

FAQ spécifique à NetBSD/i386..

Basic NetBSD Networking

Guide du réseau et configuration de PPP.

3.2. Organisation de l'installation de NetBSD

L'organisation des fichiers d'une installation de NetBSD est décrite dans le fichier désormais appelé`INSTALL`. Par exemple, pour la plateforme i386, les binaires sont dans le répertoire `i386/binary/sets` et les sources dans `source/sets`. Le répertoire `source/patches` contient les correctifs à apporter à la version de base pour faire disparaitre des bugs ou des problèmes liés à la sécurité trouvés après la publication.

3.3. Installation

La première chose à faire avant d'installer NetBSD est de lire les informations de version et les notes d'installation du fichier `INSTALL`. C'est la description officielle de la procédure d'installation. Ensuite, vous décidez du support à partir duquel se fera l'installation. Vous avez le choix entre:

- ftp
- nfs
- CDROM
- disquette
- système de fichier non monté
- répertoire local

3.3.1. Clavier

sysinst ne vous permet pas de changer la configuration du clavier pendant l'installation: si vous avez un clavier US, pas de problèmes mais pour le reste du monde, l'ennui est mineur. Si vous installez depuis un cédérom, vous n’avez besoin que des touches alphanumériques (qui sont à quelques exceptions près aux mêmes endroits dans tous les pays) et parfois d’autres touches. J’espère que les version suivantes permettront de changer cette configuration, pour le moment, vous pouvez utiliser le tableau de correspondances suivant.

US	**IT**	**DE**	**FR**
-	’	ß	)
/	-	-	! (= sur Mac)
=	ì	’	= (- sur Mac)
:	ç	Ö	M
;	ò	ö	m
#	£	§	3
"	°	Ä	%
*	(	(	8
(	)	)	9
)	=	=	0
’	à	ä	ù
‘	\	^	vide (@ sur Mac)
\	ù	#	* (‘ sur Mac)
Q			A
W			Z
A			Q
Z			W
M			?
m			,

Si vous utilisez un clavier non-US, la première chose à faire après l’installation est de modifier les paramètrages du clavier. En attendant, soyez patient.

3.3.2. Géométries

Le programme d’installation fait référence à deux types de géométries du disque dur. Vous devez savoir de quoi il s’agit:

- vraie géométrie (real geometry)
- géométrie BIOS (BIOS geometry)

La *vraie géométrie* est celle détectée par le système. La *géométrie BIOS* est celle utilisée par le BIOS et elle peut être différente de la précédente (par exemple, le BIOS peut réorganiser le disque avec LBA).

Le disque utilisé pour l’exemple d’installation qui suit est un IDE avec la géométrie suivante:

```
vraie: 6232 cyl,   16 têtes,  63 sec
```

```
BIOS :  779 cyl,  128 têtes,  63 sec   (LBA)
```

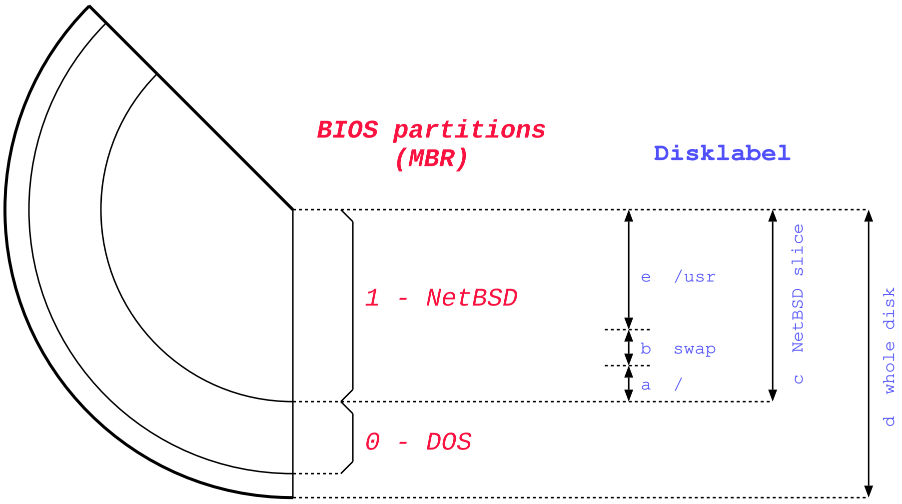
Comme vous le constatez, le BIOS réorganise le disque à l'aide de LBA. Il réduit effectivement le nombre de cylindres et augmente le nombre pistes (mais le résultat reste le même: 6232 * 16 = 779 * 128 = 99712). Un secteur contient 512 octets, ce qui veut dire que la taille du disque est de 6232 * 16 * 63 * 512 = 3 Go. NetBSD n'a pas besoin de * réorganiser la géométrie (et, en fait, ne le fait pas). Pendant l'installation, il est possible de manuellement modifier ces informations si sysinst les a mal reconnues.

3.3.3. Partitions

La terminologie des partitions utilisée par NetBSD est différente de celle de DOS/Windows. En fait, il y a deux sortes de patitionnement. NetBSD s'installe sur l'une des quatre partitions BIOS primaires (celles définies dans la table des partitions du disque).

Dans sa partition BIOS (aussi appellée *slice*), NetBSD définit son partitionnement BSD à l'aide d'un *disklabel*: ces sous-partitions ne peuvent être vues que par NetBSD et sont identifiées par des lettres minuscules (en commençant par • a•). Par exemple, wd0a fait référence à la partition • a• du premier disque IDE (wd0) et sd0a à la partition • a• du premier disque SCSI. Dans Figure 3-1, il existe deux partitions BIOS primaires, l'une utilisée par DOS et l'autre par NetBSD. NetBSD organise le disque grâce au disklabel.

Figure 3-1. Partitions



Note : La signification des partition s"c" et • d• est typique du portage i386. Les autres portages peuvent utiliser des conventions différentes (par exemple, • c• pour le disque complet).

Note : Si NetBSD partage le disque dur avec un autre système d'exploitation (comme dans l'exemple précédent), vous aurez sans doutes besoin d'installer un *gestionnaire de démarrage* (boot manager), c'est à dire un programme qui vous permet de choisir le système à lancer au démarrage. sysinst peut automatiquement installer et configurer un gestionnaire simple mais efficace.

Si WindowsNT est installé sur le même disque, vous pouvez utiliser le gesionnaire NT pour démarrer NetBSD. La façon de faire est décrite sur le site web de NetBSD.

3.3.4. Besoins en espace disque

La place requise par NetBSD dépend de ce que vous envisagez de faire (c'est à dire serveur ou station de travail). Par exemple, considérons un ordinateur de bureau muni d'un disque dur de 420 Mo (assez petit comparé aux standards actuels) avec X, les sources du noyau et quelques applications (Netscape...). La partition de mémoire virtuelle (swap) est de 32 Mo. **df** indique:

```
Filesystem  1K-blocks     Used    Avail Capacity  Mounted on
/dev/wd1a       31887    16848    13444    56%    /
/dev/wd1e      363507   173202   172129    50%    /usr
```

Comme vous pouvez le voir, il reste encore 180 Mo sur le disque.

3.3.5. Ré-essayer

Lorque l'on installe un système pour la première fois, c'est rarement avec succès et NetBSD ne fait pas exception. Même si tout se passe correctement, dès que vous commencez à utiliser le système vous réalisez, par exemple, que vous auriez pu choisir une meilleure organisation des partitions. C'est important de ne pas abandonner. Si vous ré-essayez, vous verrez que ce qui était difficile à comprendre devient peu à peu plus clair grâce à l'accumulation de l'expérience et aux relectures du document `INSTALL`.

Pour la première installation, il est plus indiqué d'accepter par défaut les suggestions de sysinst et d'éviter, par exemple, de changer le disklabel. La seule chose que vous devez décider est la place disque pour la partition fdisk de NetBSD.

3.4. Exemple d'installation

Le reste de ce chapitre décrit une réelle installation pour la plupart des cas: l'installation à partir d'un cédérom. Les concepts sont les mêmes pour les autres types d'installation (ftp...), la seule différence étant la façon qu'a sysinst de trouver les binaires. Notez que certains détails diffèrent suivant la version de NetBSD. Cet exemple est tiré de la version 1.5.

Pour les besoins de l'enseignement, dans l'exemple suivant, l'option la plus • difficile• sera systématiquement choisie:

- Table des partitions BIOS complète: une ou plusieurs partitions seront détruites pour faire de la place pour NetBSD.
- fdisk avec les secteurs au lieu des Mo.
- modification manuelle du disklabel créé par sysinst, aussi avec les secteurs.

- installation personnalisée (c'est à dire que vous choisissez quels sont les binaires que vous installez).

Ces choix donnent l'impression que l'installation est très compliquée et demande beaucoup de travail: souvenez-vous que si vous prenez les choix par défaut, tout est plus simple. D'un autre côté, un guide qui n'explique que les parties • faciles• n'est pas très utile (à part du point de vue commercial...)

3.4.1. Préparer l'installation

Avant de commencer l'installation, ce serait une bonne idée de détailler les étapes que vous aurez à passer. Tout d'abord, lisez le fichier `INSTALL` (promis, c'est la dernière fois que je vous le dis) pour connaitre le détail de l'installation et pour vérifier les compatibilités matérielles. Ensuite, s'il y a déja quelque chose sur votre disque dur, demandez-vous comment libérer de la place pour NetBSD. S'il doit partager le disque avec un autre système, vous devrez probablement créer une nouvelle partition (avec sysinst) voire modifier la taille d'une existante. sysisnt ne vous permet pas de modifier une taille mais certains produits commerciaux (comme Partition Magic) ou gratuits (FIPS, pfdisk) peuvent le faire.

L'installation se divise en trois étapes logiques. Dans la première, vous créez une partition pour NetBSD et vous établissez le disklabel correspondant. Dans la seconde, vous décidez quels sont les binaires que vous voulez installer et vous les extrayez dans la partition nouvellement créée. La troisième est indépendante de la méthode d'installation (cédérom, ftp, NFS...). A la fin de la première partie, rien n'a été encore été écrit sur le disque et l'on vous demande de confirmer l'installation. Si vous confirmez, l'installation commence sinon, vous retournez au menu principal et le disque dur reste intact.

3.4.2. Créer une disquette d'installation

> **Note :** Si votre cédérom NetBSD est démarrable, vous n'avez pas besoin de créer de disquette d'installation: activer l'option "boot from CD-ROM" de votre BIOS, insérez le cédérom et redémarrez la machine. Cette option n'est probablement pas présente sur les anciens ordinateurs.

Avant d'installer, vous avez besoin de créer une disquette de démarrage, c'est à dire copier l'image disquette depuis le cédérom vers la disquette. Pour effectuer cette opération depuis DOS, vous pouvez utiliser le programme rawrite du répertoire `i386/installation/misc`. Le fichier image est `i386/installation/floppy/boot.fs`.

> vérifiez vos disquettes avant de copier l'image disque: cette simple opération est souvent négligée alors qu'elle peut vous épargner de nombreux problèmes.

1. Formattez la disquette.
2. Allez dans le répertoire `I386\INSTALLATION\FLOPPY` du cédérom.
3. Lancez le programme **..\MISC\RAWRITE**. Le "fichier source" est `BOOT.FS` et le lecteur de destination est A:

Si vous établissez la disquette depuis un environnement Unix, vous pouvez utiliser la commande **dd**. Par exemple:

```
# cd i386/installation/floppy
# dd if=boot.fs of=/dev/fd0a bs=36b
```

dd copie des blocs de 512 octets: l'option `bs=36b` copie 36 blocs à la fois rendant l'opération plus rapide.

> **Note :** une disquette 1440K contient 1474560 octets organisés de 80 cylindres, 2 pistes, 18 secteurs de 512 octets chacun, c'est à dire 80 * 2 * 18 = 2880 blocs. Ainsi, `bs=36b` copie un cylindre (18*2 blocs) en une fois et répète cette opération 80 fois au lieu de 2880.

3.4.3. Dernières étapes préparatoires

Tout est maintenant prêt pour l'installation mais, avant de continuer, récupérez quelques informations matérielles au sujet de votre ordinateur.

La plus importante est de connaitre le type de disque dur (IDE, SCSI) et sa géométrie. Vous pouvez trouver ces informations dans le manuel du disque ou à l'aide d'un programme de diagnostiques. Certains disques durs possèdent une étiquette où sont écrites ces données. Vous pouvez aussi aller les chercher sur le site web du constructeur.

Si vous effectuez votre installation par ftp ou NFS vérifiez les paramètrages de votre carte réseau: si le noyau d'installation pense que la carte a un certain IRQ alors qu'elle en a un autre, vous serez dans l'incapacité de poursuivre l'installation. Par exemple, le noyau d'installation peut reconnaitre une carte compatible NE2000 avec un des deux paramètrages:

```
ne0     at isa? port 0x280 irq 9        # NE[12]000 ethernet cards
ne1     at isa? port 0x300 irq 10
```

Si votre carte réseau NE2000 est paramètrée différemment, elle ne sera pas détectée (après l'installation, vous pourrez compiler votre propre noyau avec vos propres paramètrages).

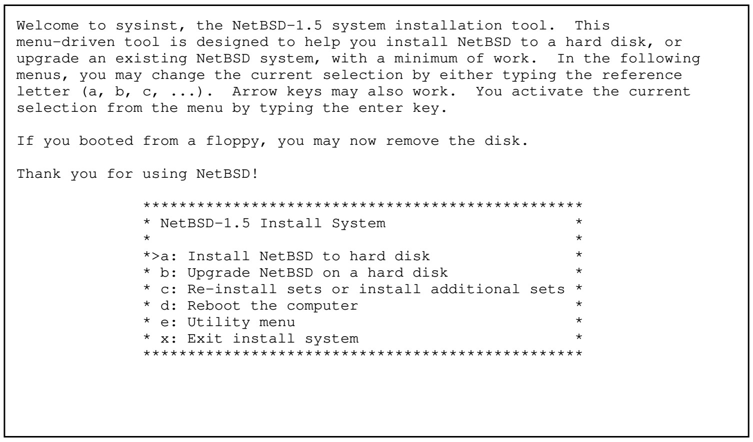
Récueillez encore d'autres information matérielles comme, par exemple, le nombre de ports série, parallèle... Ce n'est pas obligatoire pour l'installation mais cela peut être utile par la suite. Vérifiez une fois encore les paramètrages (IRQ, ports E/S...) décrits dans le fichier `INSTALL`.

> **Note :** vous pouvez effectuer l'installation même si vous n'avez aucune information sur votre disque dur. Dans ce cas, il vous faudra faire confiance à sysinst qui détermine (généralement) très bien les géométries.

3.4.4. Commencer l'installation

Insérez la disquette nouvellement créée dans le lecteur A: et redémarrez l'ordinateur (ou bien démarrez depuis le cédérom). Le noyau démarre et affiche de nombreux messages à l'écran, la plupart désignant des périfériques non trouvés ou non paramètrés. C'est normal: le noyau placé sur la disquette tente de détecter à pau près tous le matériel géré par NetBSD. Vous n'avez sans doutes pas tous ces périphériques sur votre machine !

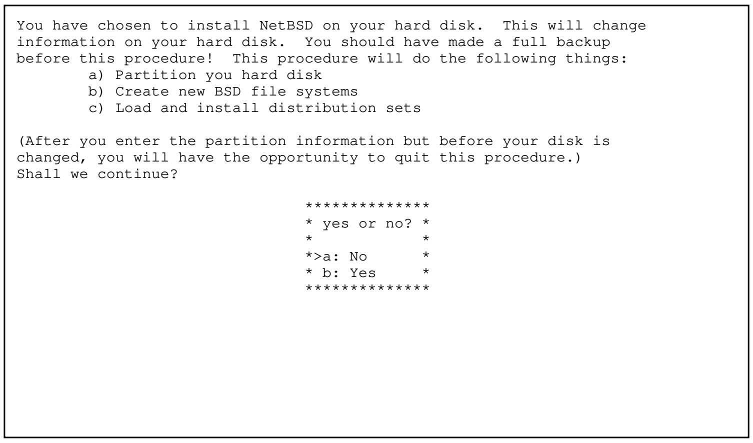
Figure 3-2. Commencer l'installation



```
Welcome to sysinst, the NetBSD-1.5 system installation tool.  This
menu-driven tool is designed to help you install NetBSD to a hard disk, or
upgrade an existing NetBSD system, with a minimum of work.  In the following
menus, you may change the current selection by either typing the reference
letter (a, b, c, ...).  Arrow keys may also work.  You activate the current
selection from the menu by typing the enter key.

If you booted from a floppy, you may now remove the disk.

Thank you for using NetBSD!

              *************************************************
              * NetBSD-1.5 Install System                     *
              *                                               *
              *>a: Install NetBSD to hard disk                *
              * b: Upgrade NetBSD on a hard disk              *
              * c: Re-install sets or install additional sets *
              * d: Reboot the computer                        *
              * e: Utility menu                               *
              * x: Exit install system                        *
              *************************************************
```

Lorsque le démarrage est terminé, vous vous trouvez dans la page principale du programme d'installation, comme le montre la Figure 3-2. Ne soyez pas déçu par l'aspect spartiate de sysinst: c'est un outil puissant et flexible. A partir de cet écran, vous pouvez suivre les instructions en prenant le fichier `INSTALL` comme référence. tous les écrans de sysints ont plus ou moins le même aspect: la partie haute présente une courte description de l'opération en cours ou une aide rapide, la partie centrale indique les paramètres détectés par NetBSD et la partie basse affiche la liste des différents choix disponibles. Si vous choisissez l'option • a• , vous arrivez à l'écran suivant (Figure 3-3) qui vous demande confirmation.

Figure 3-3. Confirmer l'installation

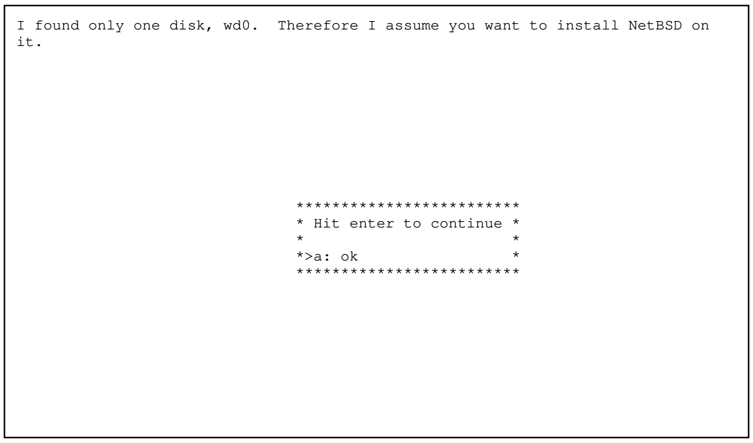


```
You have chosen to install NetBSD on your hard disk.  This will change
information on your hard disk.  You should have made a full backup
before this procedure!  This procedure will do the following things:
        a) Partition you hard disk
        b) Create new BSD file systems
        c) Load and install distribution sets

(After you enter the partition information but before your disk is
changed, you will have the opportunity to quit this procedure.)
Shall we continue?

                                ************** 
                                * yes or no? *
                                *            *
                                *>a: No      *
                                * b: Yes     *
                                **************
```

Une fois que vous avez choisi de continer avec • b• , vous avez à indiquer le disque sur lequel installer NetBSD. Si vous en avez plusieurs, sysinst affiche la liste de ceux que vous pouvez choisir. Dans notre exemple, seul un disque est présent donc le programme d'installation se contente d'afficher le message informatif montré Figure 3-4.

Note : les information de cet écran diffèrent suivant le type et le nombre de disques durs installés dans votre ordinateur.

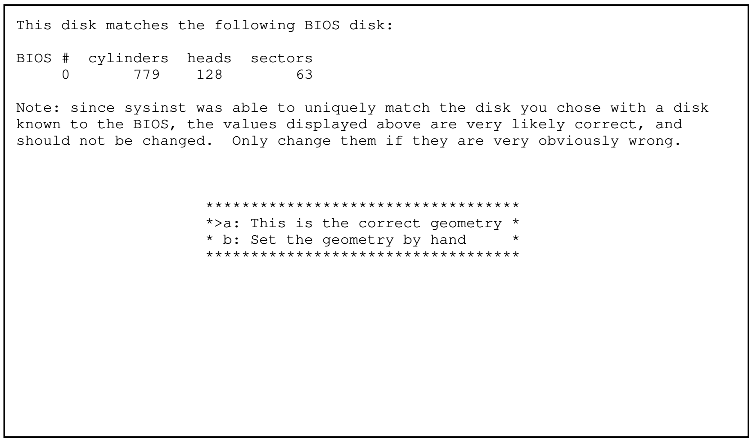
Figure 3-4. Choisir le disque dur



```
I found only one disk, wd0.  Therefore I assume you want to install NetBSD on
it.

                               *************************
                               * Hit enter to continue *
                               *                       *
                               *>a: ok                 *
                               *************************
```

Ensuite, (Figure 3-5), sysinst affiche la géométrie BIOS du disque choisi. Vous pouvez confirmer si elle est correcte ou la modifier à la main si elle est erronée.

Figure 3-5. Géométrie BIOS



```
This disk matches the following BIOS disk:

BIOS #  cylinders  heads  sectors
     0       779    128        63

Note: since sysinst was able to uniquely match the disk you chose with a disk
known to the BIOS, the values displayed above are very likely correct, and
should not be changed.  Only change them if they are very obviously wrong.


                     ***********************************
                     *>a: This is the correct geometry *
                     * b: Set the geometry by hand     *
                     ***********************************
```

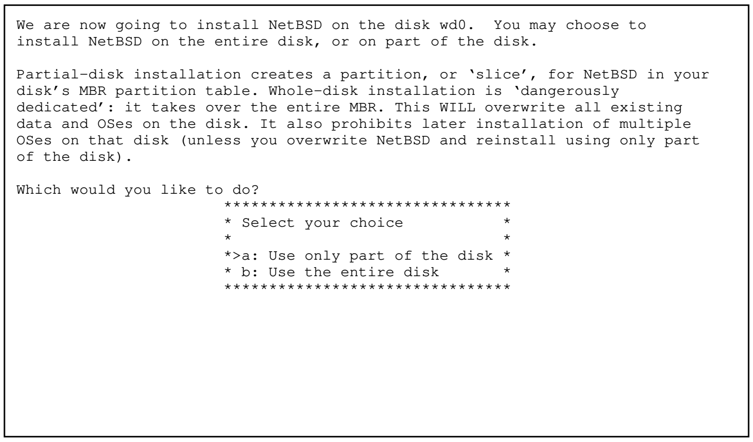
3.4.5. Partitions

Voici la plus importante partie de l'installation: le partitionnement du disque dur. Vous devez tout d'abord décider si NetBSD va utiliser une partition (choix conseillé) ou le disque complet (choix • dangereux•). Dans le premier cas, il est toujours possible de créer une partition qui utilise le disque entier (Figure 3-6) donc je vous recommande de choisir cette option qui, si je comprends bien, permet à la table des partitions BIOS de conserver la compatibilité avec les autres systèmes d’exploitation.

Dans notre exemple, nous allons utiliser un disque avec la géométrie "réelle" suivante, correspondant à la géométrie BIOS de la Figure 3-5.

```
6232 cyl, 16 têtes, 63 sec  (6232 x 16 x 63 = 6281856 secteurs en tout)
1 secteur = 512 octets
1 piste = 63 secteurs = 63 * 512 octets = 32 Ko
1 cylindre = 16 * 63 * 512 octets = 504 Ko
```

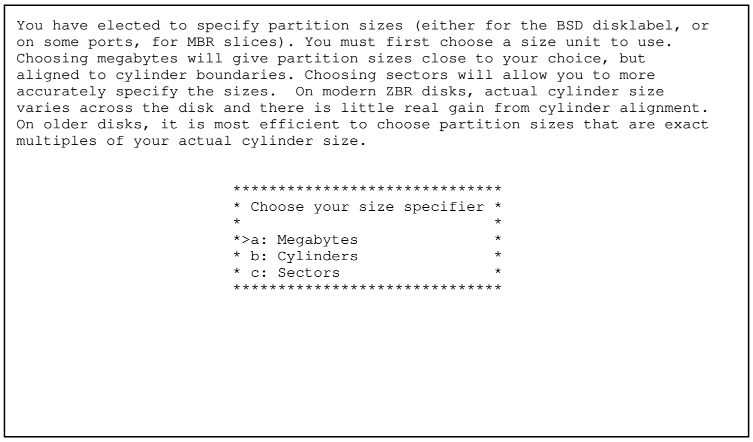
Figure 3-6. Choisir l’organisation des partitions



```
We are now going to install NetBSD on the disk wd0.  You may choose to
install NetBSD on the entire disk, or on part of the disk.

Partial-disk installation creates a partition, or ‘slice’, for NetBSD in your
disk’s MBR partition table. Whole-disk installation is ‘dangerously
dedicated’: it takes over the entire MBR. This WILL overwrite all existing
data and OSes on the disk. It also prohibits later installation of multiple
OSes on that disk (unless you overwrite NetBSD and reinstall using only part
of the disk).

Which would you like to do?
                       ********************************
                       * Select your choice           *
                       *                              *
                       *>a: Use only part of the disk *
                       * b: Use the entire disk       *
                       ********************************
```

L’étape suivante, montrée dans la Figure 3-7, est le choix de l’unité de mesure à utiliser pour le partitionnement du disque: • sectors• donne le plus de flexibilité et de précision (notez qu’il est préférable d’alligner une partition sur l’ensemble des cylindres pour de meilleures performances, au moins sur les vieux disques durs). • Megabytes• est plus facile à utiliser car ce choix ne nécessite aucun calcul et est plus • intuitif• .

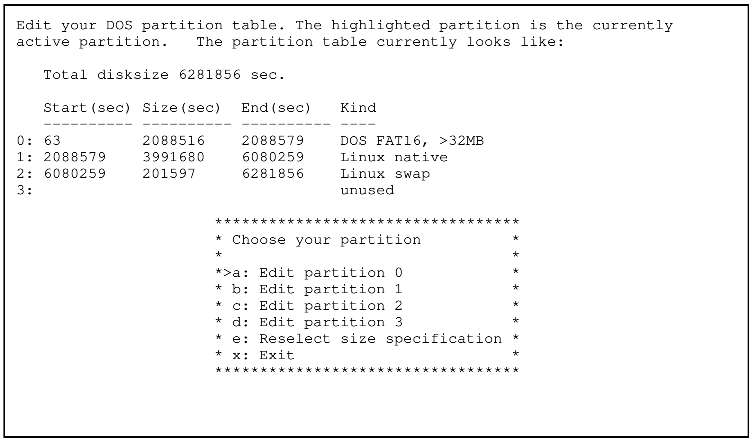
Figure 3-7. Choisir l’unité de mesure



```
You have elected to specify partition sizes (either for the BSD disklabel, or
on some ports, for MBR slices). You must first choose a size unit to use.
Choosing megabytes will give partition sizes close to your choice, but
aligned to cylinder boundaries. Choosing sectors will allow you to more
accurately specify the sizes.  On modern ZBR disks, actual cylinder size
varies across the disk and there is little real gain from cylinder alignment.
On older disks, it is most efficient to choose partition sizes that are exact
multiples of your actual cylinder size.


                        ******************************
                        * Choose your size specifier *
                        *                            *
                        *>a: Megabytes               *
                        * b: Cylinders               *
                        * c: Sectors                 *
                        ******************************
```

Ce guide va utiliser les secteurs car c'est le plus utile pour les besoins de l'apprentissage. Choisir loption • c• nous mène à l'interface du programme fdisk.

Figure 3-8. fdisk



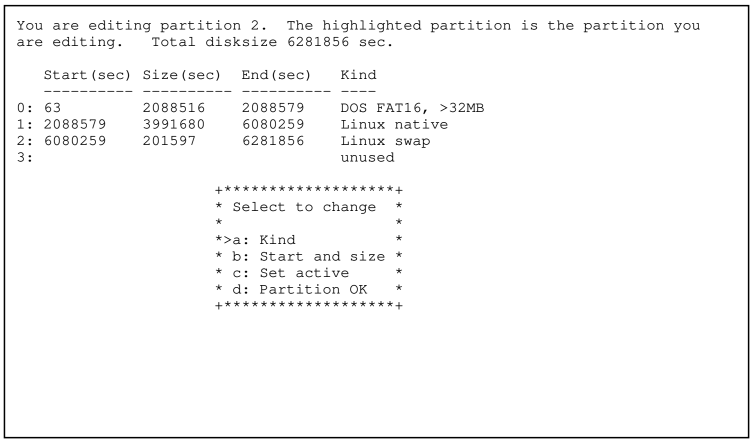
```
Edit your DOS partition table. The highlighted partition is the currently
active partition.   The partition table currently looks like:

   Total disksize 6281856 sec.

   Start(sec) Size(sec)  End(sec)   Kind
   ---------- ---------- ---------- ----
0: 63         2088516    2088579    DOS FAT16, >32MB
1: 2088579    3991680    6080259    Linux native
2: 6080259    201597     6281856    Linux swap
3:                                  unused

                      *********************************
                      * Choose your partition         *
                      *                               *
                      *>a: Edit partition 0           *
                      * b: Edit partition 1           *
                      * c: Edit partition 2           *
                      * d: Edit partition 3           *
                      * e: Reselect size specification *
                      * x: Exit                       *
                      *********************************
```

La Figure 3-8 présente la situation du disque dur avant l'installation de NetBSD. Il y a quatre partitions primaires : une pour DOS/Windows, deux pour GNU/Linux et une inutilisée. Il n'y a pas d'espace libre: la colonne End(sec) à la partition 3 montre que les 6281856 secteurs sont occupés.

> **Note :** Dans cette écran, fdisk utilise la formule:
>
> `Start(sec) + Size(sec) = End(sec)`
>
> C'est à dire secteur de départ + taille = secteur final. Cela signifie que la valeur End(sec) d'une partition est égale à la valeur Start(sec) de la partition suivante. Cette façon de faire n'est pas très intuitive puisque le dernier secteur d'une partition appartient à la suivante. Disklabel utilise une autre convention plus logique.

Afin de faire de la place, les deux partitions GNU/Linux seront sacrifiées, en commençant par la dernière. Sysinst affiche un écran qui permet de modifier les données présentes et la Figure 3-9 montre les données actuelles pour la partition 2.

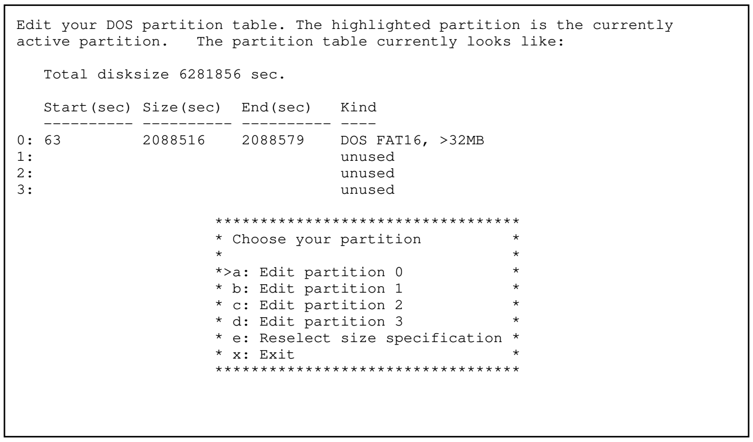
Figure 3-9. Effacer une partition



```
You are editing partition 2.  The highlighted partition is the partition you
are editing.   Total disksize 6281856 sec.

   Start(sec) Size(sec)  End(sec)   Kind
   ---------- ---------- ---------- ----
0: 63         2088516    2088579    DOS FAT16, >32MB
1: 2088579    3991680    6080259    Linux native
2: 6080259    201597     6281856    Linux swap
3:                                  unused

                      +*******************+
                      * Select to change  *
                      *                   *
                      *>a: Kind           *
                      * b: Start and size *
                      * c: Set active     *
                      * d: Partition OK   *
                      +*******************+
```

Pour effacer une partition, choisissez le type *unused* de l'option • a• , puis tapez l'option • b• pour vider les champs • Start• et • Size• (appuyez sur Entrée pour laisser ces champs blancs). Pour terminer, confirmez en tapant • d• . Vous retournez alors à l'écran principal de fdisk où la partition 3 est désormais vide. Agissez de même pour les partitions 2 et 1, ne laissant que la partition 0 sur le disque (Figure 3-10).

Figure 3-10. Partition effacée



```
Edit your DOS partition table. The highlighted partition is the currently
active partition.   The partition table currently looks like:

   Total disksize 6281856 sec.

   Start(sec) Size(sec)  End(sec)   Kind
   ---------- ---------- ---------- ----
0: 63         2088516    2088579    DOS FAT16, >32MB
1:                                  unused
2:                                  unused
3:                                  unused

                      **********************************
                      * Choose your partition          *
                      *                                *
                      *>a: Edit partition 0            *
                      * b: Edit partition 1            *
                      * c: Edit partition 2            *
                      * d: Edit partition 3            *
                      * e: Reselect size specification *
                      * x: Exit                        *
                      **********************************
```

Seule reste la partition DOS/Windows, utilisant 2088516 secteurs, c'est à dire 1029 Mo (à peu près 1 Go). L'espace libre est calculé comme la différence entre nombre total déja calculé de secteurs et le dernier secteur de la partition DOS (celui de la colonne End(sec)).

```
6281856 - 2088579 = 4193277 secteurs = 2047 Mo libres sur le disque
```

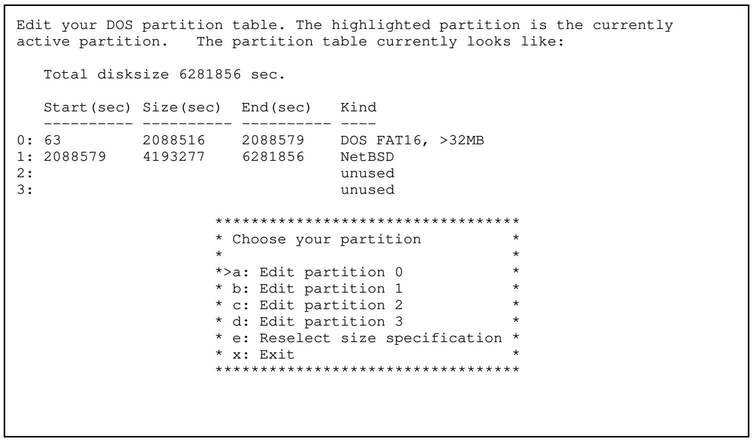
> **Note :** La partition DOS commence au secteur 63 et non au 0 comme on pourrait s'y attendre. Ce n'est pas inhabituel: la première piste (63 secteurs) est réservée. Au cylindre 0, piste 0, secteur 1 se trouve le *Master Boot Record* (MBR. Lorsque le système démarre, le BIOS charge le MBR en mémoire, détecte les partitions actives et charge en mémoire le secteur de démarrage de cette partition, auquel il cède le contrôle. Ensuite, le secteur de démarrage charge le système d'exploitation de sa partition.

Maintenant, avec l’option "b", une nouvelle partition NetBSD est créée, à partir de la fin de la partition DOS. Pour créer une nouvelle partition, les données suivantes doivent être fournies:

- le type de la nouvelle partition
- le premier secteur de la nouvelle partition
- la taille (en secteurs) de la nouvelle partition

Choisissez le type "NetBSD" (avec l’option "a: kind") et entrez, avec l’option • b• les données que nous avons calculées: début (start) =2088579 et taille (size) = 4193277. Vérifiez que tout est correct et confirmez la création avec l’option • d• qui vous ramène à l’écran pricipal de fdisk. Le résultat est montré dans la Figure 3-11 qui affiche l’organisation finale de la table des partitions. Choisissez l’option • x• pour passer au menu suivant.

Figure 3-11. Partitionnement terminé



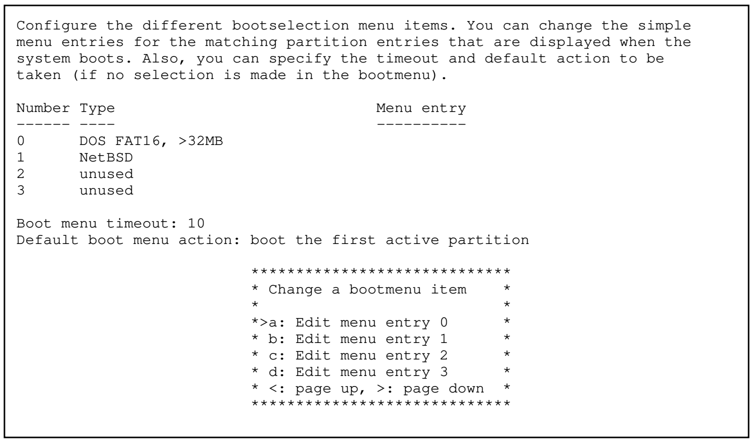
```
Edit your DOS partition table. The highlighted partition is the currently
active partition.   The partition table currently looks like:

   Total disksize 6281856 sec.

   Start(sec) Size(sec)  End(sec)   Kind
   ---------- ---------- ---------- ----
0: 63         2088516    2088579    DOS FAT16, >32MB
1: 2088579    4193277    6281856    NetBSD
2:                                  unused
3:                                  unused

                      **********************************
                      * Choose your partition          *
                      *                                *
                      *>a: Edit partition 0            *
                      * b: Edit partition 1            *
                      * c: Edit partition 2            *
                      * d: Edit partition 3            *
                      * e: Reselect size specification *
                      * x: Exit                        *
                      **********************************
```

Si vous avez fait une erreur dans le partitionnement (par exemple une partition qui déborde du disque), sysinst affichera un message et vous conseillera de retourner dans l’écran de fdisk (vous pouvez aussi continuer). Si les données sont correctes mais que la partition NetBSD se trouve au-delà des secteurs démarrables par le BIOS, sysinst vous en avertira et vous demandera si vous voulez quand même continuer. Cela peut alors poser des problèmes sur les anciens ordinateurs: dans notre exemple, nous aurons l’avertissement mais la machine démarrera correctement. Il est impossible de donner une règle générale (cela dépend du BIOS). Si votre PC n’est pas trop vieux, je vous conseille d’ignorer l’avertissement et de continuer.

> **Note :** cette limitation ne provient pas de NetBSD. Certains vieux BIOS ne peuvent démarrer de partition résidant au-delà des 1024 premiers cylindres. Pour bien comprendre le problème, étudiez les différents BIOS ainsi que les nombreux shémas d’adressage (CHS physique, CHS logique, LBA...). Ces sujets ne sont pas abordés dans ce guide.
>
> Avec les BIOS les plus récents, supportant les extensions int13, il est possible d’installer NetBSD sur des partitions se trouvant au-delà des premiers 8Go du disque dur à condition que le gestionnaire de démarrage de NetBSD soit installé.

Si les données sont correctes et que sysinst détecte que vous avez plus d'un système d'exploitation sur votre disque, il vous proposera d'ajouter un gestionnaire de démarrage. A l'aide du programme d'installation, vous pouvez à la fois installer le gestionnaire et le configurer. Vous pouvez définir le nom des systèmes d'exploitations à afficher au démarrage, la partition à démarrer par défaut et le délais laissé à l'utilisateur pour choisir. Cet écran est montré Figure 3-12.

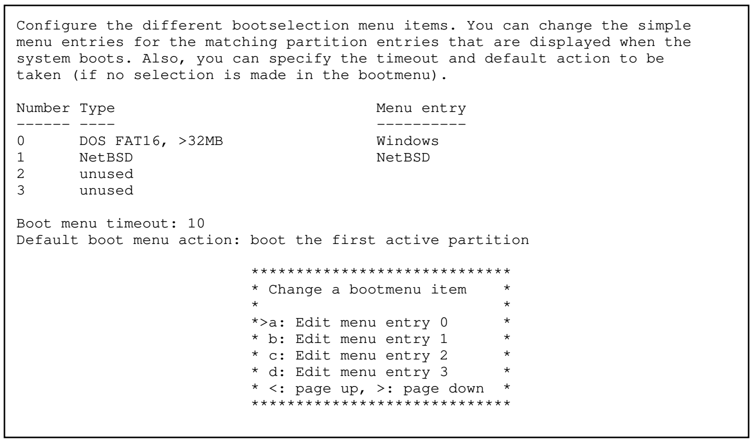
Figure 3-12. Configurer le gestionnaire de démarrage



```
Configure the different bootselection menu items. You can change the simple
menu entries for the matching partition entries that are displayed when the
system boots. Also, you can specify the timeout and default action to be
taken (if no selection is made in the bootmenu).

Number Type                            Menu entry
------ ----                            ----------
0      DOS FAT16, >32MB
1      NetBSD
2      unused
3      unused

Boot menu timeout: 10
Default boot menu action: boot the first active partition

                          *****************************
                          * Change a bootmenu item    *
                          *                           *
                          *>a: Edit menu entry 0      *
                          * b: Edit menu entry 1      *
                          * c: Edit menu entry 2      *
                          * d: Edit menu entry 3      *
                          * <: page up, >: page down  *
                          *****************************
```

Choisissez les partitions qui apparaitront dans le menu du gestionnaire et définissez le nom de chacune en utilisant les options de • a• à • d• . Dans la colonne "Menu entry", vous devez voir apparaitre quelque chose pour chaque partition démarrable, comme le montre la Figure 3-13.

Figure 3-13. Configuration du gestionnaire de démarrage



```
Configure the different bootselection menu items. You can change the simple
menu entries for the matching partition entries that are displayed when the
system boots. Also, you can specify the timeout and default action to be
taken (if no selection is made in the bootmenu).

Number Type                            Menu entry
------ ----                            ----------
0      DOS FAT16, >32MB                Windows
1      NetBSD                          NetBSD
2      unused
3      unused

Boot menu timeout: 10
Default boot menu action: boot the first active partition

                          *****************************
                          * Change a bootmenu item    *
                          *                           *
                          *>a: Edit menu entry 0      *
                          * b: Edit menu entry 1      *
                          * c: Edit menu entry 2      *
                          * d: Edit menu entry 3      *
                          * <: page up, >: page down  *
                          *****************************
```

L'option "e" vous permet de choisir un délais pour le menu de démarrage: une fois dépassé, sans choix d'utilisateur, c'est la partition par défaut (choisie avec l'option • f•) qui est démarrée. Vous pouvez choisir une des options par défaut suivantes:

- une partition
- un autre disque dur

- la première partition active

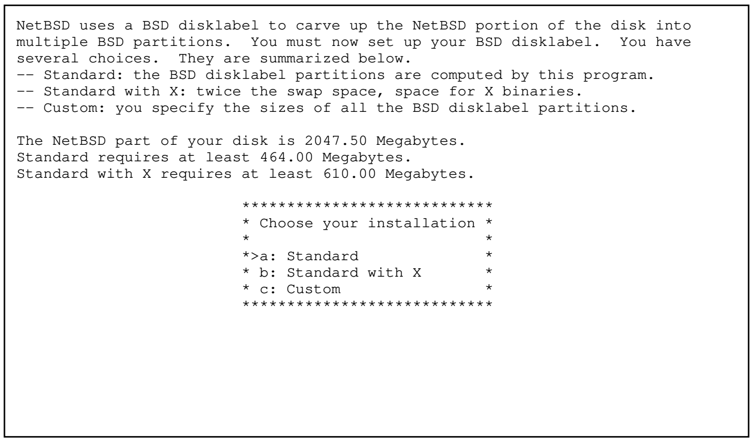
Une fois terminée la configuration du gestionnaire, la première partie de l'installation, appelée partitionnement du disque, est terminée.

Les partitions BIOS, appelées aussi *slices* par BSD, sont créées. Il y en a maintenant deux: une DOS et une NetBSD. Il est temps de définir les partitions BSD.

3.4.6. Disklabel

There are three alternatives for the creation of the BSD partitions, as shown by Figure 3-14.

Figure 3-14. Disklabel



```
NetBSD uses a BSD disklabel to carve up the NetBSD portion of the disk into
multiple BSD partitions.  You must now set up your BSD disklabel.  You have
several choices.  They are summarized below.
-- Standard: the BSD disklabel partitions are computed by this program.
-- Standard with X: twice the swap space, space for X binaries.
-- Custom: you specify the sizes of all the BSD disklabel partitions.

The NetBSD part of your disk is 2047.50 Megabytes.
Standard requires at least 464.00 Megabytes.
Standard with X requires at least 610.00 Megabytes.

                         ****************************
                         * Choose your installation *
                         *                          *
                         *>a: Standard              *
                         * b: Standard with X       *
                         * c: Custom                *
                         ****************************
```

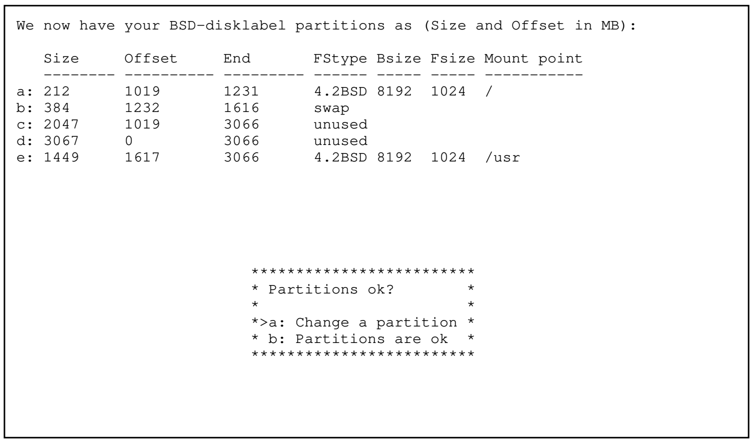
Si c'est la première fois que vous effectez une installation, je vous suggère de choisir l'option • a• ou • b• , laissant à sysinst la décision du partitionnement. Dans notre exemple, nous allons nous compliquer la vie en modifiant manuellement le disklabel (uniquement pour les besoins de l'apprentissage, bien entendu).

> **Note :** même si vous laissez le système choisir pour vous, il est préférable de bien examiner le disklabel généré. Si l'espace disque est insuffisant, sysinstest suffisamment intelligent pour s'en rendre compte et vous prévenir. Les versions précédentes ne le faisaient pas et créaient discrètement des partititons invalides.

3.4.7. Créer un disklabel

Tout d'abord, laissez le programme créer automatiquement un disklabel. En choisissant l'option • b• de la Figure 3-14, nous passons à la Figure 3-15.

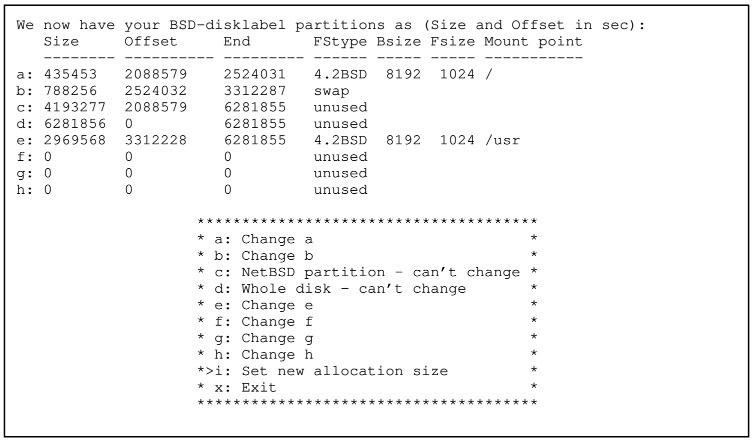
Figure 3-15. Disklabel standard



```
We now have your BSD-disklabel partitions as (Size and Offset in MB):

   Size     Offset     End       FStype Bsize Fsize Mount point
   -------- ---------- --------- ------ ----- ----- -----------
a: 212      1019       1231      4.2BSD 8192  1024  /
b: 384      1232       1616      swap
c: 2047     1019       3066      unused
d: 3067     0          3066      unused
e: 1449     1617       3066      4.2BSD 8192  1024  /usr





                          ************************
                          * Partitions ok?       *
                          *                      *
                          *>a: Change a partition *
                          * b: Partitions are ok  *
                          ************************
```

Cela fait, il vous reste à tout confirmer (option • b•) pour achever le travail. Au lieu de ça, voyons comment faire pour réduire la taille de la partition de mémoire virtuelle (swap) afin de laisser plus de place à la partition `/usr`. Pour changer la taille de la partition swap, choisissez l'option • a• : ce nouvel écran nous permet de modifier l'unité de mesure pour utiliser les secteurs. Le résultat est montré par la Figure 3-16.

Figure 3-16. Modifier le disklabel (sec)



```
We now have your BSD-disklabel partitions as (Size and Offset in sec):
   Size     Offset     End       FStype Bsize Fsize Mount point
   -------- ---------- --------- ------ ----- ----- -----------
a: 435453   2088579    2524031   4.2BSD  8192  1024 /
b: 788256   2524032    3312287   swap
c: 4193277  2088579    6281855   unused
d: 6281856  0          6281855   unused
e: 2969568  3312228    6281855   4.2BSD  8192  1024 /usr
f: 0        0          0         unused
g: 0        0          0         unused
h: 0        0          0         unused
                    *************************************
                    * a: Change a                       *
                    * b: Change b                       *
                    * c: NetBSD partition - can't change *
                    * d: Whole disk - can't change      *
                    * e: Change e                       *
                    * f: Change f                       *
                    * g: Change g                       *
                    * h: Change h                       *
                    *>i: Set new allocation size        *
                    * x: Exit                           *
                    *************************************
```

Cette suite d'identifiants de partition est standard: certaines lettres sont réservées à des utilisations prédéfinies.

- *a* est habituellement la partition racine (*root*). partition.
- *b* est la partition de mémoire virtuelle (*swap*).
- *c* couvre toute la partition NetBSD.
- *d* couvre tout le disque dur: étendu au delà de la partition (slice) NetBSD. Avec une méthode analogue, vous pourriez faire une partition DOS ou Linux visible par NetBSD en créant une partition BSD se trouvant en dehors du • slice• NetBSD.

- *e* est la première partition libre. Habituellement, /usr est montée en • e• .

> **Note :** La signification des identifiants de partitions peut changer d'un portage à l'autre. La présente explication s'applique au portage i386.

Vous n'avez normalement pas à changer les partitions *b* et *c*. Vous êtes, en revanche, libre de modifier la taille et le point de montage des autres et d'en créer de nouvelles (8 au maximum, à l'aide des lettres *e* à *h*).

Pour modifier la partition de mémoire virtuelle (swap), vous devez le faire avec la lettre "*b*". Vous pouvez aussi vouloir changer la partition • e• afin qu'elle commence juste après la • b• . Les • c• et • d• peuvent rester telles-quelles.

Nous allons maintenant créer une partition de mémoire virtuelle de 150 Mo (307200 secteurs). Cela signifie que la partition • b• débutera au secteur 2524032 pour finir au 2831231 (2524032+307200-1).

```
id:      Size    Offset       End FStype Bsize Fsize Mount point
---      ----    ------       --- ------ ----- ----- -----------
 a:    435453   2088579   2524031 4.2BSD  8192  1024 /
 b:    307200   2524032   2831231   swap
 ...
```

L'espace nouvellement libéré sera assigné à la partition • e• avec les caractéristiques: départ = 2831232, taille = 3450624 et fin = 6281855. Ces valeurs sont calculées de la façon suivante: "départ" est le secteur suivant imméditement le dernier de la partition • b• ; • fin• est le dernier secteur de la partittion NetBSD; • taille• est égal à "fin - départ -1".

```
id:      Size    Offset       End FStype Bsize Fsize Mount point
---      ----    ------       --- ------ ----- ----- -----------
 a:    435453   2088579   2524031 4.2BSD  8192  1024 /
 b:    307200   2524032   2831231   swap
 ...
 e:   3450624   2831232   6281855 4.2BSD  8192  1024 /usr
```

L'exemple précédent montre le disklabel que nous voulons. Avec les option • b• et • e• , vous pouvez entrer les valeurs que nous venons de calculer. Le résultat est montré à la Figure 3-17.

Figure 3-17. Modifier une partition BSD

```
You should set the file system (FS) kind first.  Then set the other values.

The current values for partition b are:

    Size      Offset    End       FStype Bsize Fsize Mount point
    --------- --------- --------- ------ ----- ----- -----------
 b: 788256    2524032   3312287   swap






                              ******************
                              * Change what?   *
                              *                *
                              *>a: FS kind     *
                              * b: Offset/size *
                              * c: Bsize/Fsize *
                              * d: Mount point *
                              * x: Exit        *
                              ******************
```

La Figure 3-18 montre le disklabel modifié.

Figure 3-18. Disklabel Modifié

```
We now have your BSD-disklabel partitions as (Size and Offset in sec):
   Size     Offset     End       FStype Bsize Fsize Mount point
   -------- ---------- --------- ------ ----- ----- -----------
a: 435453   2088579    2524031   4.2BSD  8192  1024 /
b: 307200   2524032    2831231   swap
c: 4193277  2088579    6281855   unused
d: 6281856  0          6281855   unused
e: 3450624  2831232    6281855   4.2BSD  8192  1024 /usr
f: 0        0          0         unused
g: 0        0          0         unused
h: 0        0          0         unused

                    **************************************
                    * a: Change a                        *
                    * b: Change b                        *
                    * c: NetBSD partition - can’t change *
                    * d: Whole disk - can’t change       *
                    * e: Change e                        *
                    * f: Change f                        *
                    * g: Change g                        *
                    * h: Change h                        *
                    *>i: Set new allocation size         *
                    * x: Exit                            *
                    **************************************
```

> **Taille des partitions :** Il est très difficile de donner une règle générale pour le nombre et la taille des partitions: cela dépend de l’utilisation que vous aurez de l’ordinateur (serveur, porte de traivail, serveur de courriel...). C’est pourquoi je vous conseille, si c’est votre première installation, de vous en tenir aux paramètres générés parsysinst. Un serveur sophistiqué aura sans doute besoin d’un partitionnement plus complexe mais ceux qui ont ce genre de problèmes connaissent souvent la solution.

Lorsque vous êtes satisfaits du résultat, choisissez l’option • x• pour sauvegarder et quitter. Vous retournez alors à la Figure 3-15 où vous choisirez l’option • b• .

3.4.8. Dernières opérations

La partie la plus difficile (création des partitions BIOS et BSD) est terminée. La suite des opérations est beaucoup plus simple. Il vous reste à choisir le nom de votre disque dur (par défaut, il se nomme *mydisk*) et à confirmer les opérations.

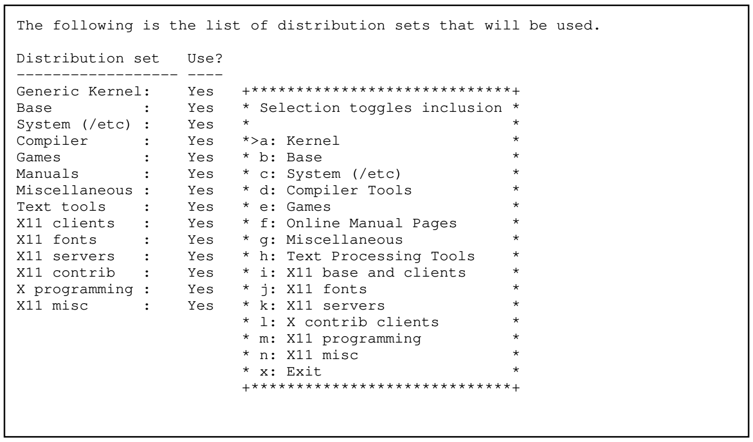
> **Note :** rien de ce qui a été défini pour le moment n'a été écrit sur le disque: il est encore possible de changer d'avis et de retourner au menu principal de sysinst sans toucher à rien.

sysinst va maintenant créer les partitions et les systèmes de fichiers à l'aide de **fdisk**, **newfs**, **fsck** et **installboot** puis installera les composantes de NetBSD.

3.4.9. Choisir le support d'installation

Nous en avons fini avec la première et plus difficile partie de l'installation. Dans l'étape suivante, vous devez choisir le type d'installation: *full* pour tout installer ou *custom* pour choisir les éléments. Si vous n'êtes pas trop limités par la place sur votre disque, je vous conseille la première solution. Pour les besoins de l'exemple, nous allons choisir la deuxième pour voir à quoi elle ressemble. Nous allons donc vers la Figure 3-19.

Figure 3-19. Choisir les éléments



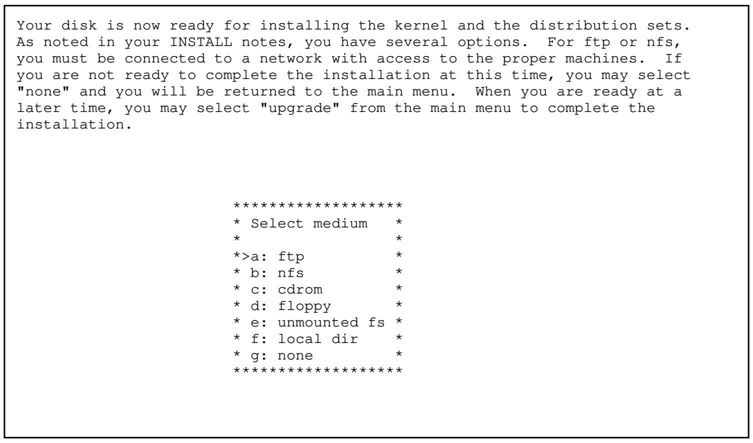
```
The following is the list of distribution sets that will be used.

Distribution set   Use?
------------------ ----
Generic Kernel:    Yes   +*****************************+
Base          :    Yes   * Selection toggles inclusion *
System (/etc) :    Yes   *                             *
Compiler      :    Yes   *>a: Kernel                   *
Games         :    Yes   * b: Base                     *
Manuals       :    Yes   * c: System (/etc)            *
Miscellaneous :    Yes   * d: Compiler Tools           *
Text tools    :    Yes   * e: Games                    *
X11 clients   :    Yes   * f: Online Manual Pages      *
X11 fonts     :    Yes   * g: Miscellaneous            *
X11 servers   :    Yes   * h: Text Processing Tools    *
X11 contrib   :    Yes   * i: X11 base and clients     *
X programming :    Yes   * j: X11 fonts                *
X11 misc      :    Yes   * k: X11 servers              *
                         * l: X contrib clients        *
                         * m: X11 programming          *
                         * n: X11 misc                 *
                         * x: Exit                     *
                         +*****************************+
```

Les trois premiers éléments sont obligatoires : sans eux, le système ne peut fonctionner. Vous pouvez cocher ou non les éléments qui suivent à l'aide des options du menu. Au départ, tous sont cochés, ce qui revient au même que de choisir l'installation "*full*". Laissez tout en l'état et passez à l'étape suivante avec l'option "x: Exit".

sysinst vous demande alors si vous voulez visualiser le nom des fichiers durant leur l'installation.

Ensuite, sysinst a besoin de savoir où trouver les éléments NetBSD à installer, c'est à dire les fichiers `.tgz`, et c'est à vous de lui dire. Le menu vous offre plusieurs choix:

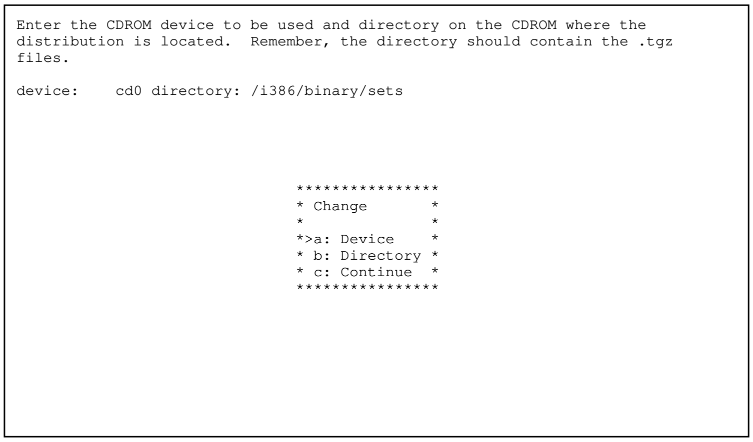
Figure 3-20. Support d’installation



```
Your disk is now ready for installing the kernel and the distribution sets.
As noted in your INSTALL notes, you have several options.  For ftp or nfs,
you must be connected to a network with access to the proper machines.  If
you are not ready to complete the installation at this time, you may select
"none" and you will be returned to the main menu.  When you are ready at a
later time, you may select "upgrade" from the main menu to complete the
installation.



                        *******************
                        * Select medium   *
                        *                 *
                        *>a: ftp          *
                        * b: nfs          *
                        * c: cdrom        *
                        * d: floppy       *
                        * e: unmounted fs *
                        * f: local dir    *
                        * g: none         *
                        *******************
```

Ces options sont expliquées en détail dans le fichier `INSTALL`. Il est possible d’installer à partir d’un système de fichier non monté (à condition qu’il soit reconnu par le noyau installé): cela veut dire, par exemple, qu’il est possible de copier l’ensemble des fichiers NetBSD sur une partition DOS et de procéder à l’installation à partir de là.

Figure 3-21. Installation depuis un cédérom



```
Enter the CDROM device to be used and directory on the CDROM where the
distribution is located.  Remember, the directory should contain the .tgz
files.

device:    cd0 directory: /i386/binary/sets




                               ****************
                               * Change       *
                               *              *
                               *>a: Device    *
                               * b: Directory *
                               * c: Continue  *
                               ****************
```

Choisissez "cdrom", sysinst vous demandera le nom du périférique (par exemple `cd0`) pour le monter automatiquement. Vous devez aussi lui indiquer le chemin d’accès aux éléments d’installation du cédérom, s’il est différent de la valeur par défaut. Si, par exemple, la distribution NetBSD est dans le répertoire `NetBSD-1.5`, utilisez l’option • b• pour modifier le chemin d’accès, comme ceci:

```
/NetBSD-1.5/i386/binary/sets
```

> **Note :** Si vous utilisez un clavier non-US, faites bien attention au caractère "/". Voyez la Section 3.3.1.

Nom du lecteur de cédéroms : Si vous ne connaissez pas son nom, vous pouvez le trouver de cette façon:

1. Appuyez sur ctrl-Z pour mettre sysinst en pause et passer à l'interpréteur de commandes (très pratique !)
2. Tapez:

   ```
   # cat /kern/msgbuf
   ```

   pour afficher les messages de démarrage du noyau, comprenant le nom du lecteur de cédéroms (par exemple *cd0*).
3. Si l'affichage est trop rapide, utilisez l'éditeur **ed**:

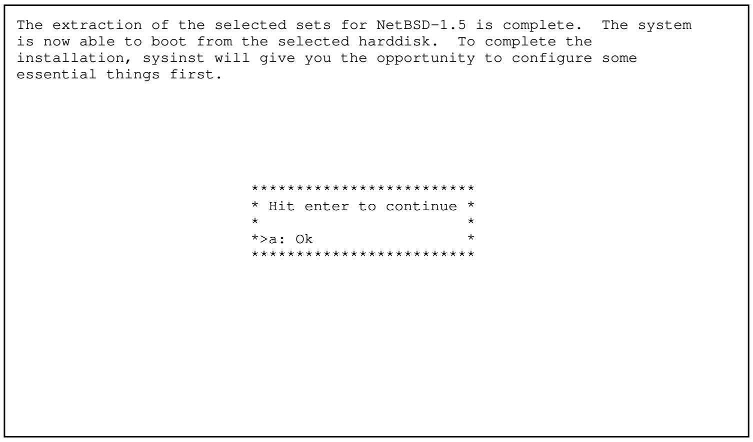
   ```
   # ed /kern/msgbuf
   ```

4. Returnez au programme d'installation avec la commande:

   ```
   # fg
   ```

A la fin de l'installation, sysinst vous annonce que tout s'est bien passé. En choisissant l'option "a: ok", les fichiers de périfériques seront créés.

Figure 3-22. Félicitations



```
The extraction of the selected sets for NetBSD-1.5 is complete.  The system
is now able to boot from the selected harddisk.  To complete the
installation, sysinst will give you the opportunity to configure some
essential things first.




                          *************************
                          * Hit enter to continue *
                          *                       *
                          *>a: Ok                 *
                          *************************
```

L'installation est terminée. Sysinst peut maintenant effectuer quelques paramètrages sur le système avant de redémarrer. Configurez votre fuseau horaire puis, à l'écran suivant, choisissez un mot de passe passe pour root. Vous pouvez alors redémarrer. Choisissez "a: ok" pour retourner au menu principal, enlevez la disquette du lecteur puis choisissez l'option "d: Reboot the computer" pour redémarrer l'ordinateur.

Chapitre 4. Premier démarrage

Après l'installation, l'ordinateur redémarre sur le disque dur. Si tout se passe bien, vous verrez apparaitre, au bout de quelques secondes ou minutes suivant votre matériel, l'invite de commandes (ou prompt). Le système n'est pas configuré mais ce n'est pas grave car le paramètrage est très facile, d'approche très commode et premet une grande flexibilité. Nous allons voir comment tout configurer rapidement et, par la même occasion, vous pourrez en apprendre plus sur le fonctionnement du système. A l'avenir, en cas de problèmes, vous saurez où chercher.

4.1. Si quelque chose ne se passe pas correctement

Si le système ne démarre pas, cela peut venir d'une mauvaise installation du gestionnaire de démarrage ou bien d'un problème sur le MBR (Master Boot Record). Redémarrez à partir de la disquette et à l'invite:

```
booting fd0a:netbsd - starting in ...
```

appuyez sur la barre d'espace pendant les cinq secondes du compte à rebours. Le démarrage s'interrompt et l'invite de commandes s'affiche. Une aide minimum peut être obtenue avec "?" ou la commande • help• .

```
type "?" or "help" for help.
> ?
commands are:
boot [xdNx:][filename] [-adrs]
     (ex. "sd0a:netbsd.old -s")
ls [path]
dev xd[N[x]]:
help|?
quit
> boot wd0a:netbsd
```

Le système devrait alors démarrer sur le disque dur à la place de la disquette. Si NetBSD démarre correctement, il y a probablement un problème avec le MBR: installez le gestionnaire de démarrage ou bien modifiez sa configuration avec la commande **fdisk -B**. Reportez-vous à la Section 18.4 pour plus de détails.

4.2. Se connecter

Pour la première fois, vous utiliserez le super-utilisateur *root*, le seul déja défini à la fin de l'installation. Lorsque le système vous demande le mot de passe root, tapez celui que vous avez défini lors de l'installation. Si vous n'en avez pas défini, appuyez simplement sur Entrée.

```
NetBSD/i386 (Amnesiac) (ttyE0)
login: root
password
```

```
...
We recommend creating a non-root account and using su(1) for root access.
#
```

4.3. Changer le clavier

Le clavier est toujours dans sa configuration US. Si vous avez un clavier différent, changez sa définition maintenant avant de poursuivre la configuration du système. Par exemple, pour utiliser le clavier français, utilisez la commande:

```
# wsconsctl -k -w encoding=fr
encoding -> fr
```

La liste complète des définitions de claviers se trouve dans `/sys/dev/wscons/wsksymdef.h` mais les plus courantes sont:

- de
- dk
- fr
- it
- jp
- sv
- uk
- us

Ce paramètrage restera jusqu'au démarrage suivant. Pour le rendre permanent, ajoutez la ligne précédente au fichier `/etc/rc.local` elle sera exécutée à chaque démarrage.

```
# echo "wsconsctl -k -w encoding=fr" >> /etc/rc.local
```

> **Note :** attention, prenez garde de bien taper deux ">" : si vous n'en tapez qu'un, le fichier sera effacé au lieu de se voir ajouter une ligne.

Une approche encore meilleure pour règler le problème du clavier: vous pouvez compiler un noyau qui utilise le votre par défaut. Cela est décrit au Chapitre 7.

4.4. La commande man

Si vous n'avez jamais utilisé de système (de type) Unix, votre meilleure amie est désormais la commande **man** qui affiche des pages de manuels: celles de NetBSD sont certainement les meilleures et les plus détaillées que vous pourrez trouver. Elles sont par ailleurs très techniques.

man nom affiche la page de manuel de la commande • nom• et **man -k nom** affiche une liste de commandes en rapport avec • nom• (vous pouvez aussi taper **apropos nom**).

Pour les principes de base de la commande **man**, tapez:

```
# man man
```

Le manuel est divisé en neuf sections contenant, non seulement les informations sur les commandes, mais aussi la description de certaines caractéristiques et structures de NetBSD. Par exemple, regardez la page de manuel hier(7) qui décrit en détail la hiérarchie des répertoires utilisée par NetBSD.

```
# man hier
```

D'autres pages similaires sont release(7) et packages(7). Chaque section du manuel possède une description de contenu dans la page *intro*. Par exemple, essayez:

```
# man 8 intro
```

Exemple 4-1. Les sections du manuel

1. commandes générales (outils et utilitaires)
2. appels système et numéros d'erreur
3. bibliothèques C
4. fichiers spéciaux et matériel supporté
5. formats de fichiers
6. jeux
7. diverses informations
8. maintenance et commandes d'utilisation du système
9. interne au noyau

Un sujet peut apparaitre dans plusieurs sections à la fois. Pour voir une page spécifique, ajoutez le numéro de section dans la ligne de commandes. Par exemple, *time* est dans la section 1 (commande utilisateur), section 3 (les fonctions de la bibliothèque C) et section 9 (les variales horaires du système). Pour visualiser la page des bibliothèques C, tapez:

```
# man 3 time
```

Pour voir toutes les pages disponibles:

```
# man −a time
```

4.5. Changer le mot de passe root

Si vous n’avez pas défini de mot de passe root lors de l’installation (ce n’était pas possible de le faire avant la version 1.5), il est alors temps d’utiliser la commande **passwd**.

```
# passwd
Changing local password for root.
New password:
Retype new password:
```

Le mot de passe n’est pas affiché à l’écran lorsque vous le tapez. Plus loin, nous verrons comment ajouter d’autres comptes utilisateur au système.

4.6. Changer d’interpréteur de commandes

L’interpréteur par défaut est **csh**. Si cela ne vous dit pas grand chose, vous pouvez commencer à vous familiariser avec lui avec **man csh**. Il s’agit d’un bon interpréteur interractif bien qu’il ne garde pas les commandes en mémoire (voyez tcsh, bash ou même NetBSD `/bin/sh` pour cela). Si vous voulez changer d’interpréteur de commandes, utilisez la commande **chsh**. Les interpréteur présent sur NetBSD après l’installation sont:

- csh
- sh
- ksh

Le nouvel interpréteur prendra ses fonction la prochaine fois que vous vous connecterez. En attendant, vous pouvez taper la commande suivante:

```
# set filec
```

qui complète automatiquement la ligne de commande (avec la touche ESC; utilisez ctrl+D pour avoir la liste des possibilités).

Vous pouvez ausi installer d’autres interpréteur sur le système. tcsh, bash, zsh et d’autres encore se trouvent dans la collection de paquetages (que nous étudierons plus tard).

C’est l’occasion de créer les fichiers d’initialisation de l’interpréteur (.chsrc, .login, ...)

4.7. Heure système

NetBSD, comme tout système Unix, utilise une horloge basée sur l’heure de Greewich (UTC), c’est pourquoi vous devez aussi règler votre heure système. Si vous voulez que votre horloge système soit à l’heure locale (parce que, par exemple, vous avez un deuxième système comme Windows installé), vous devez en avertir NetBSD en modifiant la variable système `rtc_offset`. Vous pouvez soit corriger le fichier de configuration du noyau et recompiler soit • patcher• directement le noyau existant (la nouvelle heure ne prendra effet qu’après redémarrage). C’est plus facile qu’on ne peut le penser. Par exemple:

```
# gdb --write /netbsd
GNU gdb 4.17
Copyright 1998 Free Software Foundation, Inc.
...
This GDB was configured as "i386--netbsd"...(no debugging symbols found)...
(gdb) set rtc_offset=-60
(gdb) quit
```

La valeur donnée (-60) est le nombre de minutes à l'ouest de Greenwich.

Pour la valeur actuelle de `rtc_offset`:

```
# sysctl kern.rtc_offset
kern.rtc_offset = -60
```

Désormais, le noyau sait comment convertir l'heure interne en heure UTC mais devez quand même configurer votre fuseau horaire (que vous trouverez dans `/usr/share/zoneinfo/`). Si vous avez déjà effectué ce paramètrage lors de l'installation, vous pouvez sauter cette étape (bien qu'il soit prudent de vérifier que les valeurs soient correctes). Par exemple, pour l'Italie:

```
# rm -f /etc/localtime
# ln -s /usr/share/zoneinfo/Europe/Rome /etc/localtime
```

Une fois que tout est paramètré correctement, vous pouvez changer l'heure système avec la commande suivante:

```
# date [[[[[cc]yy]mm]dd]hh]mm
```

4.8. Configuration de base `/etc/rc.conf`

NetBSD utilise le fichier `/etc/rc.conf` pour se configurer au démarrage: il détermine ce qui doit être exécuté. Il est très important de comprendre ce fichier.

Avec la version 1.5 de NetBSD, l'administration de rc.conf a changé. Avant, toutes les valeurs par défaut se trouvaient dans ce fichier et l'utilisateur était sencé les modifier directement. La version 1.5 introduit le fichier `/etc/defaults/rc.conf` pour les valeurs par défaut. Pour donner ses propres valeurs, l'utilisateur doit les écrire dans le fichier `/etc/rc.conf`, remplaçant les valeurs par défaut de `/etc/defaults/rc.conf` (qui, lui, reste inchangé).

Il est très important de comprendre ce fichier. La page de manuel contient des descriptions détaillées de toutes les options.

```
# man rc.conf
```

Les premières modifications sont:

- "rc_configured=YES" (cela doit avoir déjà été fait par le logiciel d'installation).
- "wscons=YES" pour activer les consoles virtuelles.

- "lpd=YES" pour activer le gestionnaire d'impression.
- Choisissez un *nom d'hôte* pour votre machine (un nom qualifié). Si votre machine est autonome, vous pouvez utiliser le nom que vous voulez (par exemple "woody.jouets.net"). Si votre machine est connecté à un réseau, vous devrez donner un nom de réseau valide.

  Plutôt que d'inscrire le nom de votre ordinateur dans le fichier de configuration, vous pouvez l'écrire dans le fichier `/etc/myname` : le résultat est identique.

4.9. Activer les consoles virtuelles

Les paramètrages de cette section ne sont plus nécessaires à partir de la version 1.5, les consoles virtuelles étant activées par défaut.

Corrigez le fichier `/etc/ttys` et vérifiez que ttyE0 à ttyE3 sont sur • on• . ttyE4 peut rester à • off• (pour être utiliser avec X). Par exemple:

```
console "/usr/libexec/getty Pc"         pc3     off secure
ttyE0   "/usr/libexec/getty Pc"         vt220   on secure
ttyE1   "/usr/libexec/getty Pc"         vt220   on secure
ttyE2   "/usr/libexec/getty Pc"         vt220   on secure
ttyE3   "/usr/libexec/getty Pc"         vt220   on secure
ttyE4   "/usr/libexec/getty Pc"         vt220   off secure
```

> **Pour les impatients :** une fois le fichier `/etc/ttys` modifié, vous pouvez activer les consoles virtuelles sans avoir à redémarrer. Tapez:
>
> 1. **sh /etc/rc.wscons**
> 2. **kill -1 1**

Une fois ctivées, vous pouvez passer d'une console à l'autre avec la combinaisons de touches Ctrl-Alt-Fn (Fn est une touche de fonction: F1 pour la console 1, F2 pour la 2...) Pour plus de détails, reportez-vous au Chapitre 12.

> **Note :** Utilisez les touches Contrôle et Alt, à gauche de la barre d'espace, en combinaison avec les touches de fonction.

4.10. Redémarrer le système

Dans cette première session, vous avez

- Configuré le clavier

- Changé le mot de passe root
- Changé l'interpréteur de commandes de root (optionel)
- Changé l'heure système et le fuseau horaire
- Défini l'heure locale
- Configuré `/etc/rc.conf`
- Activé les consoles virtuelles

Il est alors temps de redémarrer le système avec la commande:

```
# reboot
```

Chapitre 5. Deuxième démarrage

Lors du premier démarrage, vous avez effectué les paramètrages de base du système. Ce chapitre décrit quelques commandes et opérations courantes.

5.1. dmesg

Lorsqu'il démarre, le noyau affiche de nombreux messages à l'écran: ils donnent des informations sur l'état du noyau (par exemple mémoire disponible...) et sur les périfériques détectés. Ceci est très important pour le diagnostic matériel, l'origine de problèmes et la détermination du nom d'un périférique (par exemple, vous pouvez vérifier si votre carte réseau a été détectée comme `ne0` ou `ne1`). D'une manière générale, ces messages défilent trop rapidement pour être utiles mais vous pouvez utiliser la commande **dmesg** pour les revoir.

```
# dmesg | more
```

Si quelque chose a l'air de ne pas fonctionner correctement sur votre système et que vous demandez de l'aide à l'une des listes de diffusion de NetBSD, n'oubliez pas d'inclure les informations appropriées issues de dmesg: cela aidera les autre personnes à diagnostiquer votre problème.

> Depuis NetBSD 1.4.2, le processus de démarrage écrit une copie des messages de dmesg dans le fichier `/var/run/dmesg.out`. Cela est utile car le système efface les vieux messages au fur et à mesure.

5.2. Monter un cédérom

Les nouveaux utilisateurs sont souvent surpris par le fait que, bien que le programme d'installation ait parfaitement reconnu et monté leur cédérom, le système installé, lui, semble avoir "oublié" comment faire. Il n'y a rien de magique, vous pouvez le monter comme tout système de fichiers. Tout ce que vous avez à connaitre est le nom du périférique et quelques options de la commande **mount**. Le nom peut se trouver l'aide de la commande **dmesg**. Par exemple, si dmesg affiche:

```
# dmesg | grep ^cd
cd0 at atapibus0 drive 1: <ASUS CD-S400/A, , V2.1H> type 5 cdrom removable
```

le nom du périférique est `cd0` et vous pouvez monter le cédérom avec les commades:

```
# mkdir /mnt/cdrom
# mount -t cd9660 -o ro /dev/cd0a /mnt/cdrom
```

Pour que ce soit plus facile, ajoutez ligne dans le fichier `/etc/fstab`:

```
/dev/cd0a /mnt/cdrom cd9660 ro,noauto 0 0
```

Sans avoir besoin de redémarrer, vous pouvez désormais monter le cédérom avec:

```
# mount /mnt/cdrom
```

Lorsque le cédérom est monté, vous ne pouvez pas l'éjecter à la main, il vous faut d'abord le démonter:

```
# umount /mnt/cdrom
```

Il existe aussi une commande logicielle qui démonte le cédérom et l'éjecte:

```
# eject /dev/cd0a
```

5.3. Monter une disquette

Pour monter une disquette, il vous faut connaitre le nom du lecteur de disquettes ainsi que le système de fichiers qu'elle contient. Par exemple, pour lire et écrire une disquette MS-DOS, utilisez la commande suivante:

```
# mount -t msdos /dev/fd0a /mnt
```

A la place de `/mnt`, vous pouvez utiliser le répertoire de votre choix. Vous pourriez, par exemple, créer un répertoire `/disquette`, comme pour le cédérom. Si vous êtes amené à souvent utiliser des disquettes MS-DOS, regardez le paquetage • mtools• qui permet l'accès aux disquettes MS-DOS (ou partition de disque dur) sans avoir besoin de les monter. Cela est vraiment pratique pour rapidement copier des disquettes.

5.4. Accéder à une partition DOS/Windows

Si NetBSD partage le disque dur avec MS-DOS ou Windows, il est possible de modifier le disklabel et de rendre visible la partition DOS. Tout d'abord, vous devez déterminer la géométrie de la partition DOS en utilisant, par exemple, **fdisk**.

```
# fdisk wd0
NetBSD disklabel disk geometry:
cylinders: 6232 heads: 16 sectors/track: 63 (1008 sectors/cylinder)
...
Partition table:
0: sysid 6 (Primary 'big' DOS, 16-bit FAT (> 32MB))
    start 63, size 2088516 (1019 MB), flag 0x80
        beg: cylinder    0, head   1, sector  1
        end: cylinder  259, head   0, sector  4
1: sysid 169 (NetBSD)
    start 2088579, size 4193277 (2047 MB), flag 0x0
        beg: cylinder  259, head   0, sector  4
        end: cylinder  779, head   0, sector  1
2: <UNUSED>
3: <UNUSED>
```

> **Note :** cet exemple utilise le disque `wd0`: remplacez-le par le nom de votre propre disque.

La commande fdisk nous montre que la partition DOS débute au secteur 63 et a une taille de 2088516 secteurs. La partition NetBSD, elle, débute au secteur 2088579 (2088579 = 2088516 + 63). Vous utiliserez ces données pour modifier le disklabel BSD: tout ce que vous avez à faire est d'ajouter une ligne qui définit cette partition DOS en choisissant une des lettres de partitions inutilisées. Utilisez la commande **disklabel** pour le faire. Si vous donnez l'option `-e`, cela lancera votre éditeur préféré ($EDITOR). Par exemple:

```
# disklabel -e wd0
...
#        size   offset     fstype  [fsize bsize  cpg]
...
  e:  3450624  2831232    4.2BSD    1024  8192    16   # (Cyl.  2808* - 6231)
  f:  2088516       63    MSDOS
```

Les partitions de "a" à "e" ont déja été utilisées par NetBSD, la première disponible est • f• . Les champs • size• (taille) et • offset• (position) ont été remplies avec les valeurs précédament calculées. Le point de montage doit ensuite être créé. Par exemple:

```
# mkdir /msdos
```

Enfin, ajoutez une ligne dans le fichier `/etc/fstab`.

```
/dev/wd0f /msdos msdos rw,noauto 1 3
```

Désormais, la partition MS-DOS sera montée avec la commande:

```
# mount /msdos
```

Avec cette méthode, vous pouvez monter des partitions FAT ou FAT32. Si vous voulez qu'elles montent automatiquement au démarrage, enlevez l'option • noauto• de `/etc/fstab`.

```
/dev/wd0f /msdos msdos rw 1 3
```

5.5. Ajouter des utilisateurs

Il est temps d'ajouter de nouveaux utilisateurs au système puisque vous n'allez pas utiliser le compte • root• pour votre travail quotidien. NetBSD n'a pas de programme pour le faire tout seul, il vous faut lire la page de manuel *adduser*.

```
# man adduser
```

En suivant les instructions données, vous allez aussi commencer à utiliser **vipw** qui est l'outil de base pour l'admistration de comptes sous NetBSD.

> **Note :** NetBSD 1.4.2 possède un ensemble d'outils d'administration, une commande **useradd** et d'autres encore. Par exemple, pour créer un nouvel utilisateur:

```
# useradd -m joe
```

Les valeurs par défaut de **useradd** peuvent être modifiées. Reportez-vous à la page de manuel de useradd(8).

Si vous avez une version récente de NetBSD et que vous ne voulez pas ajouter de comptes à la main, vous pouvez installer le paquetage addnerd de la collection. Je vous conseille cependant de lire la page de manuel et d'ajouter au moins un compte à la main.

5.6. Mots de passe Shadow

Les mots de passe shadow sont activés par défaut sur NetBSD et ne peuvent pas être désactivés: tous les mots de passe du fichier `/etc/passwd` contiennent un "*", les vrais mots de passe, criptés, sont dans le fichier `/etc/master.passwd` qui ne peut être lu que par • root• . Lorsque vous démarrez **vipw** pour éditer le fichier de mots de passe, le programme ouvre une copie de `/etc/master.passwd`. Lorsque vous quittez, **vipw** vérifie la validité de la copie, crée un nouveau `/etc/passwd` puis installe un nouveau `/etc/master.passwd`. Enfin, **vipw** lance **pwd_mkdb**qui établie les fichiers `/etc/pwd.db` et `/etc/spwd.db`, deux bases de données équivalentes à `/etc/passwd` et `/etc/master.passwd` mais plus rapides à consulter.

Comme vous pouvez le voir, les mots de passe sont automatiquement gérés par NetBSD. Si vous utilisez **vipw** pour modifier un mot de passe, vous n'avez aucune procédure administrative supplémentaire à effectuer.

Il est très important de *toujours* utiliser **vipw** et les autres outils (chfn, chsh, chpass, passwd) pour administrer les comptes et de ne *jamais* modifier directement le fichier `/etc/master.passwd`.

5.7. Arrêter et redémarrer le système

La commande utilisée pour arrêter ou redémarrer le système est **shutdown**.

```
# shutdown -h now
# shutdown -r now
```

Deux autres commandes font la même chose:

```
# halt
# reboot
```

halt/reboot et **shutdown** ne sont pas des synonymes. Cette dernière commande est plus sophistiquée. En environnement multi-utilisateurs vous devez absolument utiliser shutdown: vous pouvez programmer l'extinction, prévenir les utilisateurs... Pour plus de détails, reportez-vous aux pages shutdown(8), halt(8) et reboot(8).

Chapitre 6. Imprimer

Ce chapitre décrit une configuration d'impression simplifiée qui utilise une imprimante HP Deskjet 690C connectée au premier port parallèle comme exemple. Tout d'abord, le système sera configuré pour imprimer des documents texte puis la configuration sera étendue vers l'impression de fichiers PostScript à l'aide du programme Ghostscript.

6.1. Activer le démon d'impression

Une fois l'installation achevée, il n'est pas encore possible d'imprimer car le démon d'impression, **lpd**, n'est pas activé. Pour l'activer, il faut changer une ligne du fichier `/etc/rc.conf`. Passer de:

```
lpd=NO
```

à

```
lpd=YES
```

La modification sera effective au prochain démarrage mais vous pouvez quand même lancer le démon à la main:

```
# lpd -s
```

Pour vérifier que **lpd** est bien actif, tapez:

```
# ps ax | grep lpd
  179 ??  Is     0:00.01 lpd
```

Si vous n'obtenez rien en réponse, c'est que le démon n'est pas actif.

Avant de configurer `/etc/printcap`, mieux vaut effectuer un test d'imprimante pour vérifier que les connexions sont correctes. Par exemple:

```
# lptest 20 10 > /dev/lpt0
```

Pour voir ce qui est envoyé à l'imprimante, relancez la même commande mais sans la redirection:

```
# lptest 20 10
```

Le problème le plus fréquent est que les colonnes imprimées ne sont pas bien allignées mais forment des escaliers. C'est parce qu'il arrive que l'imprimante soit configurée pour passer à la ligne chaque fois qu'elle reçoit les caractères • CR• (retour chariot, ASCII 13) et • LF• (saut de ligne, ASCII 10). NetBSD n'envoie que le caractère • LF• . Vous pouvez corriger le problème:

- en changeant la configuration de l'imprimante
- en utilisant un filtre d'impression (décrit plus loin)

> **Note :** Dans l'exemple précédent, le gestionnaire lpd n'est pas invoqué puisque l'affichage du programme est directement envoyé sur l'imprimante (`/dev/lpt0`) sans passer par le disque dur.

6.2. Configurer `/etc/printcap`

Cette section explique comment configurer l'imprimante exemple pour imprimer des documents texte.

L'imprimante doir avoir une entrée dans le fichier `/etc/printcap` qui doit contenir son identifiant (son nom) et sa description. L'identifiant *lp* est le nom par défaut utilisé par de nombreux programmes.

Exemple 6-1. `/etc/printcap`

```
lp|local printer|HP DeskJet 690C:\
        :lp=/dev/lpa0:sd=/var/spool/lpd/lp:lf=/var/log/lpd-errs:\
        :sh:pl#66:pw#80:if=/usr/local/libexec/lpfilter:
```

La forme et les options du fichiers sont décrites dans la page de manuel printcap(5). Veuillez noter qu'un filtre (*input filter*) a été mis en place (avec l'option *if*) pour prendre en compte le problème d'escaliers.

```
if=/usr/local/libexec/lpfilter
```

> **Pilotes d'impression et imprimantes HP :** l'Exemple 6-1 utilise le périférique *lpa#* (pilote • polled•) au lieu de *lpd#* (pilote standard à interruptions). L'utilisation des interruptions peut causer des problèmes de communication avec certaines imprimantes, dont la HP Deskjet 690C: l'impression est très lente, une page Postscript peut mettre des heures à sortir. Le problème se résoud en utilisant le pilot *lpa*. Il est aussi possible de compiler un noyau dans lequel lpd est • polled• .

L'entrée printcap de l'imprimante précise aussi le répertoire d'enregistrement (spool) (qui doit être créé). Celui-ci sera utilisé par le démon lpd pour accumuler les données à imprimer.

```
# cd /var/spool/lpd
# mkdir lp
# chown daemon:daemon lp
# chmod 770 lp
```

Il nous manque encore une information: le filtre `lpfilter` qui doit être écrit. La seule chose que celui-ci doit faire est de configurer l'imprimante pour éliminer le problème d'escaliers avant d'envoyer le texte à imprimer. L'imprimante utilisée dans notre exemple utilise la chaine d'initialisation "Esc &k2G".

Exemple 6-2. `/usr/local/libexec/lpfilter`

```
#!/bin/sh
# Treat LF as CR+LF
printf "\033&k2G" && cat && exit 0
exit 2
```

```
# cd /usr/local/libexec
# chmod 755 lpfilter*
```

> **Note :** Un autre filtre peut être utilisé:
>
> `:if=/usr/libexec/lpr/lpf:`
>
> Ce filtre est plus complexe que le précédent. Il a été écrit pour traiter les sorties de **nroff** afin de gérer les soulignés et les sur-impressions, étendre les caractères tabulation et convertir les LF en CR+LF. Les sources de ce programme de filtrage se trouvent dans `/usr/src/usr.sbin/lpr/filters/lpf.c`.

La commande **lptest** peut à nouveau être lancée puis il sera temps d'utiliser la gestion d'enregistrements de **lpd**.

```
# lptest 20 10 | lpr -h
```

Le programme **lpr** imprime du texte à l'aide du gestionnaire. L'option `-h` désactive l'impression de la page de garde (pas vraiment nécéssaire à cause de l'option *sh* dans `/etc/printcap`).

Il existe de nombreuses méthodes et outils pour résoudre le problème d'escaliers, par exemple un programme en C. La solution présentée a l'avantage d'être vraiment simple.

6.3. Configurer Ghostscript

Maintenant que les fonctions de base de l'imprimante fonctionnent, nous pouvons ajouter les fichiers gérant le PostScript. L'imprimante individuelle de notre exemple ne comprend pas, nativement, le PostScript. Un programme doit être utilisé pour convertir ce type de fichiers en une suite de commandes compréhensibles par l'imprimante. Le programme Ghostscript, issu de la collection de paquetages, en est capable (voir le Chapitre 8). Cette section explique comment configurer Ghostscript pour pouvoir imprimer des fichiers PostScript sur une imprimante HP Deskjet 690C.

Un second identifiant d'imprimante doit être mis en place dans `/etc/printcap`: il utilisera un filtre différent qui fera appel à Ghostscript pour effectuer l'impression. De cette manière, les documents texte seront imprimés avec l'imprimante *lp* et les documents PostScript avec *ps*: les deux noms utilisent la même imprimante physique mais avec des filtres différents.

Le même résultat peut être obtenu en utilisant des configurations différentes. Par exemple une unique entrée qui fait appel à un filtre capable de déterminer automatiquement le format du fichier et de lancer le programme correspondant. Cette approche est simple mais nécéssite un filtre complexe. Si cela vous plait, vous pouvez vous intéresser au programme magicfilter de la collection de paquetages: il fait cela et bien d'autres choses automatiquement.

Le nouveau fichier `/etc/printcap` a cette allure:

Exemple 6-3. `/etc/printcap`

```
lp|local printer|HP DeskJet 690C:\
        :lp=/dev/lpa0:sd=/var/spool/lpd/lp:lf=/var/log/lpd-errs:\
        :sh:pl#66:pw#80:if=/usr/local/libexec/lpfilter:
```

```
ps|Ghostscript driver:\
        :lp=/dev/lpa0:sd=/var/spool/lpd/ps:lf=/var/log/lpd-errs:\
        :mx#0:sh:if=/usr/local/libexec/lpfilter-ps:
```

L’option `mx#0` est très importante dans le cas de fichiers PostScript puisqu’elle élimine les restrictions de taille des fichiers à imprimer. Les fichiers PS ont tendance à être assez gros. L’option `if` pointe sur un nouveau filtre et il y a un autre répertoire d’enregistrement.

La dernière étape consiste à créer le répertoire d’enregistrement et le programme de filtrage.

```
# cd /var/spool/lpd
# mkdir ps
# chown daemon:daemon ps
# chmod 770 ps
```

Le programme de filtrage PostScript est beaucoup plus complexe que celui pour le texte. Le fichier à imprimer est envoyé à l’interpréteur qui, à son tour, envoie une série de commandes dans le langage de l’imprimante. Nous avons transformé une imprimante couleur bon marché en un périférique capable de traiter du PostScript, grâce au système d’exploitation NetBSD et à de puissants paquetages libres. Les options utilisées pour configurer Ghostscript sont décrites dans la documentation qui l’accompagne: `cdj550` est le périférique utilisé pour piloter les imprimantes HP.

Exemple 6-4. `/usr/local/libexec/lpfilter-ps`

```
#!/bin/sh
# Treat LF as CR+LF
printf "\033&k2G" || exit 2
# Print the postscript file
/usr/pkg/bin/gs -dSAFER -dBATCH -dQUIET -dNOPAUSE -q -sDEVICE=cdj550 \
-sOutputFile=- -sPAPERSIZE=a4 - && exit 0
exit 2
```

En résumé: deux noms d’imprimantes ont été créés sur le système. Ils gèrent la même imprimante physique mais utilisent des options, des filtres et des répertoires d’enregistrement différents. Les fichiers texte et les fichiers PostScript peuvent être imprimés. Pour ces derniers, le paquetage Ghostscript doit être installé.

6.4. Commandes de gestion d’imprimantes

Cette section fait la liste des commandes BSD les plus utiles pour gérer les imprimantes et les impressions. En plus des commandes **lpr** et **lpd** déjà mentionnées, nous avons:

lpq

examine la file d’attente d’impression.

lprm

efface des impressions de la file d’attente.

lpc

vérifie le système d’impression, active ou désactive une imprimante ou une fonction d’imprimante.

6.5. Imprimer en réseau

Il est possible de configurer le système d’impression afin qu’il puisse communiquer avec une imprimante connectée à un autre ordinateur. Disons, par exemple, que nous travaillons sur *wotan* et que nous voulons imprimer sur l’imprimante connectée à *loge*. Le fichier `/etc/printcap` de loge est celui de l’Exemple 6-3. Depuis wotan, il est possible d’envoyer des fichiers postscript puisque loge possède Ghostscript.

La première étape est de demander à loge d’accepter les impressions venant de wotan. Cela se fait en insérant une ligne contenant wotan dans le fichier `/etc/hosts.lpd` de loge. Le format de ce fichier est très simple: chaque ligne contient le nom de l’hôte à autoriser.

Ensuite, le fichier `/etc/printcap` de wotan doit être configuré pour envoyer les impressions sur loge. Par exemple:

```
lp|line printer on loge:\
 :lp=:sd=/var/spool/lpd/lp:lf=/var/log/lp-errs:\
 :rm=loge:rp=lp

ps|Ghostscript driver on loge:\
 :lp=:sd=/var/spool/lpd/lp:lf=/var/log/lp-errs:\
 :mx#0:\
 :rm=loge:rp=ps
```

Il existe quatre différences entre cette configuration et celle de l’Exemple 6-3.

1. La définition de "lp" est vide.
2. L’entrée "rm" définit le nom de l’hôte sur lequel est connectée l’imprimante.
3. L’entrée "rp" définit le nom de l’imprimante connectée.
4. Il n’est pas nécessaire de définir de filtres puisque ceux de loge seront utilisés.

Maintenant, les impression "lp" et "ps" de wotan seront automatiquement envoyées sur l’imprimante connectée à loge.

Chapitre 7. Compiler le noyau

Tôt ou tard, les utilisateurs de NetBSD voudront compiler un noyau à leur mesure. Cela apporte plusieurs avantages:

- vous pouvez largement réduire la taille du noyau et donc, sa place en mémoire. Avec la version 1.5 de NetBSD, un noyau personnalisé peut faire passer la taille de 4,7 à 1,9 Mo.
- vous pouvez améliorer les performances.
- vous pouvez règler le système.
- vous pouvez résoudre des problèmes de détections ou de conflits de périfériques.
- vous pouvez personnaliser certaines options (par exemple le clavier, le décalage avec l'horloge BIOS...
- vous pouvez acquérir une meilleure connaissance du système.

7.1. Installer les sources du noyau

> Vous pouvez obtenir les sources du noyau à partir du même endroit où vous avez trouvé le système (par exemple ftp://ftp.netbsd.org.) Prenez soin de télécharger la même version des sources que celle du système installé.

Sysinst, le programme d'installation, ne copie pas les sources sur le disque dur, celles-ci doivent être extraites à la main. L'archive se trouve dans le répertoire `source/sets` et s'appelle `syssrc.tgz`.

```
# gzip -dc syssrc.tgz | (cd / ; tar xvf -)
```

Soyez patient. Cette opération dure plusieurs minutes car l'archive contient des centaines de fichiers. Ceux-ci sont extraits dans le répertoire `/usr/src/sys`et le lien symbolique `sys` pointe dessus. Les deux commandes suivantes ont donc le même effet:

```
# cd /usr/src/sys
# cd /sys
```

Une fois extraites les sources du noyau, vous pouvez gagner de l'espace disque en effaçant les répertoires des architectures qui ne vous concernent pas. Allez dans le répertoire `/sys/arch` puis effacez les répertoires dont vous n'avez pas besoin. Pour le portage i386, par exemple, ne conservez que le répertoire i386. Pour les autres architectures, il peut être nécessaire de conserver plusieurs répertoires source.

Lorsque les sources sont installées, vous pouvez créer votre noyau personnalisé: ce n'est pas aussi difficile que vous le pensez. En fait, un nouveau noyau peut être établi en quatre étapes décrites dans les sections suivantes.

7.2. Clavier italien

Avant de compiler le noyau, les utilisateurs italiens peuvent vouloir modifier la correspondance des touches pour l'adapter à leur clavier. Celle-ci est définie dans le fichier `/sys/dev/pckbc/wskbdmap_mfii.c`. Dans la configuration par défaut, quelques caractères utiles sont absents (comme les accolades ou le tilde). Voici une autre configuration:

```
static const keysym_t pckbd_keydesc_it[] = {
...
KC(8),   KS_7,          KS_slash,       KS_braceleft,
KC(9),   KS_8,          KS_parenleft,   KS_bracketleft,
KC(10),  KS_9,          KS_parenright,  KS_bracketright,
KC(11),  KS_0,          KS_equal,       KS_braceright,
KC(12),  KS_apostrophe, KS_question,    KS_grave,
KC(13),  KS_igrave,     KS_asciicircum, KS_asciitilde,
KC(26),  KS_egrave,     KS_eacute,      KS_bracketleft, KS_braceleft,
KC(27),  KS_plus,       KS_asterisk,    KS_bracketright,KS_braceright,
...
```

La définition ci-dessus définit la correspondance suivante:

Touches	Caractères
Alt Gr + 7	{
Alt Gr + 8	[
Alt Gr + 9	]
Alt Gr + 0	}
Alt Gr + '	‘
Alt Gr + ì	~
Alt Gr + é	[
Alt Gr + +	]
Shift + Alt Gr + è	{
Shift + Alt Gr + +	}

Pilote de console : à partir de la version 1.4, NetBSD utilise le pilote multiplateforme wscons pour gérer l'écran, le clavier et la souris. Les précédentes versions utilisent pccons ou pcvt. Pour plus de détails, reportez-vous au Chapitre 12.

7.3. Recompiler le noyau

Pour recompiler le noyau, vous devez avoir installé les compilateurs (`comp.tgz`).

Etapes de base pour compiler le noyau

1. Créer ou modifier le fichier de configuration

2. Configurer le noyau
3. Générer les dépendnces
4. Construire ("Make") le noyau
5. Installer le noyau

7.4. Créer le fichier de configuration du noyau

> **Note :** Les répertoires de cette section sont spécifiques de i386. Les utilisateurs des autres architectures devront substituer les répertoires appropriés (généralement les sous-dossiers de `arch`).

Le fichier de configuration du noyau définit le type, le nombre et les caractéristiques des périphériques supportés ainsi que divers autres options. Ces fichiers se trouvent dans le répertoire `/sys/arch/i386/conf`. Le plus simple est de dupliquer un fichier existant et de le modifier. La plupart du temps, le meilleur choix sur la plupart des plateformes est la configuration GENERIC. Dans ce fichier, des commentaires décrivent les options. Une description plus complète se trouve dans la page de manuel options(4).

```
# cd /sys/arch/i386/conf/
# cp GENERIC MON-NOYAU
# vi MON-NOYAU
```

> **Noms de noyaux :** historiquement, ceux-ci s'écrivent tout en majuscule.

La modification d'un fichier de configuration du noyau se divise en trois opérations basiques:

1. gestion de périfériques ajoutée ou retirée du noyau (par exemple, le SCSI peut être enlevé s'il n'est pas utile).
2. activation ou désactivation de services (par exemple, activer la gestion du client NFS, la compatibilité Linux...)
3. règlage des paramètres du noyau.

Les lignes commençant par un "#" sont des commentaires. Les lignes sont activées en enlevant le "#" ou désactivée en l'ajoutant. De cette manière, il est toujours possible de revenir sur une modification.

Les indications données par la commande **dmesg** peuvent être utilisées pour savoir quelles lignes peuvent être désactivées. Pour tout ligne de type:

```
<XXX> at <YYY>
```

à la fois XXX et YYY doivent être actifs dans le fichier de configuration. Vous devrez sans doutes expérimenter un peu avant de trouver la configuration minimale mais sur les machines démunies de SCSI et de PCMCIA, vous pouvez diviser la taille du noyau par deux.

Vous pouvez aussi examiner les options du fichier et désactiver celles dont vous n'avez pas besoin. Chacune est accompagnée d'une brève description qui est souvent suffisante pour comprendre de quoi il s'agit. Un grand nombre d'entre elles possède une description plus détaillée dans la page de manuel options(4). Une fois terminé, vous devez paramètrer correctement la gestion du clavier national ainsi que l'heure locale de l'horloge CMOS. Par exemple, pour l'Italie:

```
options RTC_OFFSET=-60
...
options PCKBD_LAYOUT="KB_IT"
```

Le script perl `adjustkernel`, qui se trouve sur http://www.feyrer.de/Misc/adjustkernel, analyse les indications de la commande **dmesg** et genère automatiquement un fichier de configuration minimum. L'installation de nouveaux logiciels est décrite au Chapitre 8. Si vous voulez installer Perl dès maintenant, téléchargez le paquetage pré-compilé `perl-5.00404.tgz` et tapez la commande suivante:

```
# pkg_add perl-5.00404.tgz
```

Perl est désormais installé, configuré et prêt à fonctionner. Pas moyen de faire plus simple...

Lancez le script avec:

```
# cd /sys/arch/i386/conf
# perl adjustkernel GENERIC > MYKERNEL
```

J'ai essayé ce script, il marche très bien, m'évitant une longue édition à la main. Attention, le script ne configure que les périfériques présents. Vous devez configurer le reste des options à la main (comme l'émulation Linux...)

7.5. Configurer le noyau

Lorsque vous avez fini de modifier le fichier de configuration (que nous avons appelé `MON-NOYAU`), tapez la commande suivante:

```
# config MON-NOYAU
```

Si `MON-NOYAU` ne contient pas d'erreurs, le programme **config** crée les fichiers nécessaires à la compilation sinon, il vous faudra les corriger avant de relancer le programme.

7.6. Générer les dépendances et recompiler

Cela se fait en tapant les commandes suivantes:

```
# cd ../compile/MON-NOYAU
# make depend
```

```
# make
```

Il peut arriver que la compilation s’arrête suite à des erreurs. Celles-ci peuvent avoir plusieurs origines mais les plus courantes proviennent du fichier de configuration et que **config** n’a pas détecté. Parfois le problème vient du matériel (barette mémoire défectueuse): la compilation sollicite le système beacoup plus que le font les applications. Une autre erreur typique provient du fait qu’une option B est active et a besoin d’une option A qui est inactive.

Une compilation complète peut prendre de quelques minutes à plusieurs heures suivant le matériel. Voici quelques exemples:

CPU	RAM (MB)	Durée approx.
486 DX2 50	20	1 heure
P166	96	15 minutes
PIII	128	5 minutes
68030/25	8	4 heures

La commande "make" établie le fichier `netbsd` du répertoire de compilation: celui-ci doit être copié dans le répertoire racine après avoir sauvegardé la précédente version:

```
# mv /netbsd /netbsd.old
# mv netbsd /
```

La personnalisation peut considérablement réduire la taille du noyau. Dans l’exemple suivant, `netbsd.old` est le noyau installé et `netbsd` le nouveau noyau.

```
-rwxr-xr-x  1 root  wheel  1342567 Nov 13 16:47 /netbsd
-rwxr-xr-x  1 root  wheel  3111739 Sep 27 01:20 /netbsd.old
```

Le nouveau noyau est mis en service après avoir redémarré:

```
# reboot
```

7.7. Si quelque chose se passe mal

Lorsque l’ordinateur est redémarré, il peut arriver que le nouveau noyau ne se comporte pas de la façon voulue ou bien qu’il ne démarre pas du tout. Pas de panique, si cela arrive, redémarrez avec l’ancien noyau sauvegardé et enlevez le nouveau (il vaut mieux redémarrer en "mono-utilisateur").

- Redémarrez la machine
- Appuyez sur la barre d’espace pendant le compte à rebours de 5 secondes.

  ```
  boot:
  ```

- Tapez

```
> boot netbsd.old -s
```

- Tapez enfin les commandes suivantes afin de restaurer la précédente version du noyau:

```
fsck /
mount /
mv netbsd.old netbsd
exit
```

Chapitre 8. La collection de paquetages

La collection de paquetages de NetBSD est un outil qui permet de largement simplifier la compilation et l'installation d'un grand nombre de logiciels libres pour les systèmes Unix. Une ou deux commandes seulement sont nécessaires pour installer un paquetage parfaitement configuré et en état de marche.

Le premier contact avec les paquetages de NetBSD peut entrainer de petites confusions: il existe des commandes différentes qui font apparemment la même chose. La question est en fait toute simple: il existe *deux façons* d'installer un programme. Vous pouvez:

- compiler un paquetage à partir de ses sources. Cela se fait à partir de la collection de paquetages qui peut automatiquement les télécharger sur internet, les compiler, installer et configurer le programme et sa documentation. Tous ça en deux commandes. La collection de paquetages est constituée d'un ensemble de fichiers de compilation (makefile) et de configuration qui utilisent les outils standard Unix installés avec le système de base. Une autre caractéristique intéressante est que le système de paquetages peut retrouver les paquetages dépendants, les télécharger et les installer. La collection de paquetages n'est pas installée automatiquement avec le système de base car elle subit de fréquentes mises à jour: les sections suivantes expliquent comment la télécharger et l'installer sur votre ordinateur. Le site de NetBSD contient des détails techniques approfondis sur la question.
- installer une version pré-compilée et pré-configurée du programme. Cela se fait à l'aide des utilitaires *pkgtools* qui existent avec le système de base. Cette méthode est plus rapide mais moins souple que la précédente (par exemple, vous n'avez pas accès aux options de compilation). Les utilitaires pkgtools sont aussi utiles pour gérer les programmes installés (à la fois pré-compilés et à partir des sources) et répertoriés dans une base de données. Par exemple lister les paquetages installés, en enlever un...

  Si vous voulez seulement installer un programme pré-compilé, vous n'avez pas besoin de télécharger la collection de paquetages.

Ces deux méthodes nécessitent que quelqu'un ait "empaqueté le programme", c'est à dire que le programme soit porté et configuré pour NetBSD. Bien que la collection comprenne plus de 1000 programmes, il est possible que celui que vous voulez n'y soit pas. Dans ce cas, vous pouvez toujours le compiler sans avoir recours au système de paquetages et, une fois que vous l'aurez fait fonctionner, "l'empaqueter" et l'ajouter à la collection: d'autres utilisateurs bénéficieront de votre travail.

8.1. Installer la collection de paquetages

Avant d'installer un programme depuis ses sources, vous devez télécharger et installer la collection de paquetages depuis le site de NetBSD ou d'un de ses miroirs. Voici comment faire:

1. Téléchargez la dernière version des sources du système de paquetages. Elle comprend tous les fichiers de compilation et de configuration nécéssaires. Vous la trouverez sur `ftp://ftp.netbsd.org/pub/NetBSD/NetBSD-current/tar_files/`. Le fichier s'appelle `pkgsrc.tar.gz`.
2. Le cas échéant, enlevez la collection existant sur votre disque à l'aide des commandes:

   ```
   # cd /usr
   ```

```
# rm -rf pkgsrc
```

3. Installez la collection que vous venez de télécharger:

   ```
   # tar -xzvpf pkgsrc.tar.gz -C /usr
   ```

   L'exécution de ces commandes peut prendre plusieurs minutes car il y a un grand nombre de (petits) fichiers à extraire. Une fois terminé, l'environnement nécessaire à l'installation de nouveaux programmes est prêt.

   Note : il est clair que nous n'avons, jusqu'à maintenant, fait que mettre en place les fichiers de configuration nécessaires à l'installation automatique de programmes. Nous n'avons pas encore installé de programmes ! En fait, le système possède désormais la liste des paquetages disponibles ainsi que les instructions pour aller les chercher, les compiler et les installer.

Une fois la collection de paquetages installée, vous pouvez la parcourir avec lynx ou netscape ainsi que lire les détails et les descriptions des catégories et des paquetages. Par exemple:

```
$ cd /usr/pkgsrc
$ lynx README.html
```

Ne perdez pas les distfiles

Lorsque vous effacez une collection de paquetages pour la remplacer par une nouvelle version, prenez garde de bien faire une sauvegarde du répertoire `/usr/pkgsrc/distfiles` avant la destruction: il contient toutes les archives que vous avez téléchargées. Si vous ne voulez pas prendre de risques, paramètrez le système pour qu'il utilise un autre répertoire, qui ne soit pas dans `/usr/pkgsrc`. Par exemple, créez un nouveau répertoire:

```
# mkdir /usr/pkgsrc_distfiles
```

ajoutez la ligne suivante au fichier `/etc/mk.conf`:

```
DISTDIR=/usr/pkgsrc_distfiles
```

Bien entendu, vous pouvez utiliser le répertoire de votre choix à la place de `/usr/pkgsrc_distfiles`.

Vous pouvez configurer de nombreux aspects du système de paquetages avec `/etc/mk.conf`. Vous trouverez des exemples détaillés dans le fichier `/usr/pkgsrc/mk/mk.conf.example`.

8.2. Mettre à jour la collection de paquetages

La collection de paquetages est fréquemment mise à jour: vous pouvez trouver une nouvelle version, sur le site ftp, à peu près chaque semaine. Pour mettre à jour votre collection, suivez les mêmes étapes que pour sa première installation.

Lors de cette mise à jour, il peut parfois aussi être nécessaire de mettre à jour les utilitaires • pkgtools• . Pour le savoir, la marche est simple: lorsque vous essayez d’installer un programme, le système vous prévient que vos outils • pkgtools• sont dépassés.

```
# make
===> Validating dependencies for gqmpeg-0.6.3
Your package tools need to be updated to 2000/02/02 versions.
The installed package tools were last updated on 1999/01/01.
Please make and install the pkgsrc/pkgtools/pkg_install package.
*** Error code 1
```

La plus simple façon de les mettre à jour:

```
# cd /usr/pkgsrc/pkgtools/pkg_install
# make install
```

Une fois fait, vous pouvez reprendre l’installation du paquetage qui vous a causé l’erreur.

> **Note :** vous pouvez déterminer la version requise en regardant dans le fichier `pkgsrc/mk/bsd.pkg.mk`: recherchez la ligne indiquant:
>
> ```
> PKGTOOLS_REQD = 20000202
> ```
>
> (la date 20000202 n’est là que pour l’exemple). Cela veut dire que la version des programmes qu’il vous faut se trouve dans le fichier `pkg_install-20000202.tar.gz`. Il se trouve sur le site ftp de NetBSD: `packages/distfiles/LOCAL_PORTS`. `pkg_install` s’installe comme tout autre programme.

8.3. Exemple: installer un programme depuis ses sources

Cette section décrit, pour l’exemple, l’installation du programme addnerd, qui simplifie la création de nouveaux comptes sur le système. Tout d’abord, allez dans le répertoire `/usr/pkgsrc/sysutils/addnerd`.

8.3.1. Télécharger les sources

Si vous êtes connecté à internet, le fichier Makefile ira chercher les sources requises et vous n’avez pas besoin de lire cette section.

Télécharger avec une autre machine

Le scénario le plus courant est que vous téléchargez les sources du paquetage depuis un ordinateur possédant un accès rapide à internet (par exemple au bureau) puis que vous les copiez sur votre système NetBSD (par exemple chez vous).

Dans les autres cas, il vous faut récupérer les archives vous-même. Vous devez alors en connaitre le nom. Pour cela, regardez dans le fichier `Makefile` à la ligne

```
DISTNAME = addnerd-1.6
```

Le nom complet du paquetage est donc `addnerd-1.6.tar.gz`.

Vous arriverez au même résultat mais plus simplement avec les commandes suivantes:

```
# cd /usr/pkgsrc/sysutils/addnerd
# make fetch-list
```

qui montre aussi la liste des sites sur lesquels peut être téléchargé le paquetage.

8.3.2. Compiler et installer

Pour compiler le paquetage, tapez:

```
# cd /usr/pkgsrc/sysutils/addnerd
# make
```

Cette commande va chercher l'archive contenant les sources (si elle ne se trouve pas dans le répertoire des distfiles), les extrait, applique les correctifs éventuels pour les adapter à NetBSD puis compile le paquetage.

Pour l'installer:

```
# make install
```

L'installation du nouveau programme est enregistrée dans votre système. Vous pouvez vérifier avec la commande **pkg_info -a**.

Le paquetage *addnerd* est prêt à être utilisé. Vous pouvez faire de la place en effaçant les fichiers intermédiaires créés par le compilateur:

```
# make clean
# make clean-depends
```

La deuxième commande n'est utile que si des paquetages dépendants ont été installés, ce qui n'est pas le cas de addnerd. Le même résultat peut être obtenu avec:

```
# make clean CLEANDEPENDS=1
```

8.4. installer un paquetage binaire

Comme je l'ai expliqué dans la première partie de ce chapitre, le système de paquetages peut installer un programme à partir de ses sources mais peut aussi en installer une version binaire, préparée par quelqu'un d'autre pour NetBSD. Cette seconde méthode est plus rapide puisqu'elle ne nécessite pas de compilation et l'archive est généralement plus petite et plus rapide à quérir. Pour installer un paquetage binaire, la collection de paquetages n'est pas nécessaire: seuls les utilitaires • pkgtools• sont utiles.

L'archive du programme binaire porte généralement l'extension `.tgz` alors que celle des sources finit par `.tar.gz`.

> **Note :** toutes les archives de sources ne finissent pas par `.tar.gz`. Le système de paquetages peut gérer d'autres types tels que `.zip`, `.bz2`...

Il n'est pas strictement nécessaire de télécharger les paquetages binaires avant de les installer. Vous pourvez aussi utiliser les URL ftp://-. Par exemple:

```
ftp://ftp.netbsd.org/pub/NetBSD/packages/1.4.2/i386/All/tcsh-6.09.00.tgz
```

Si vous ne savez pas quelle version du paquetage est disponible sur le site, vous pouvez même ne rien indiquer et **pkg_add** recherchera la dernière présente. Par exemple:

```
# pkg_add ftp://ftp.netbsd.org/pub/NetBSD/packages/1.4.2/i386/All/tcsh
```

Il est aussi possible de donner une liste (dont le séparateur est un ";") de chemins d'accès et d'URL comme valeur à PKG_PATH. Vous pouvez alors omettre les chemins d'accès:

```
# PKG_PATH="/cdrom;/usr/pkgsrc/packages/All;ftp://ftp.netbsd.org/pub/NetBSD/packages/1.4.2/i
export PKG_PATH
# pkg_add tcsh
```

Cette dernière commande installe le premier paquetage tcsh trouvé par pkg_add.

Par exemple, installons le programme texinfo dans sa forme pré-compilée.

1. Copiez `gtexinfo-3.12.tgz` dans un répertoire temporaire.
2. Tapez la commande suivante:
   ```
   # pkg_add -v gtexinfo-3.12.tgz
   ```
3. Vérifiez que le paquetage a été installé:
   ```
   # pkg_info
   ```
4. Effacez le fichier `gtexinfo-3.12.tgz` du répertoire temporaire.

Les paquetages pré-compilés sont très pratiques car ils demandent un minimum d'effort et de temps pour être installés. Mais les paquetages de sources permettent plus de contrôle puisqu'il est possible d'avoir accès aux options de compilation. C'est alors beaucoup plus long à installer, à cause de la compilation, ce qui peut être critique sur certaines (anciennes) plateformes.

Avant d'installer un paquetage avec **pkg_add**, il est préférable de l'examiner avec **pkg_info**. Par exemple:

```
# pkg_info -f jpeg-6b.tgz
```

Il peut aussi être intéressant d'examiner la première commande CWD pour voir où se trouve le répertoire de base du paquetage. Les répertoires les plus courants sont `/usr/pkg` et `/usr/X11R6`. Si celui-ci n'est pas celui que vous voulez, vous pouvez le changer avec l'option `-p` de la commande **pkg_add**. Par exemple, le paquetage `jpeg-6b.tgz` se trouve, par défaut, dans `/usr/pkg` mais vous pouvez l'installer dans `/usr/X11R6` si vous effectuez l'extraction avec la commande suivante:

```
# pkg_add -p /usr/X11R6 -v jpeg-6b.tgz
```

8.5. Commandes de gestion de paquetages

Les commandes les plus importantes pour gérer les paquetages sont:

pkg_add

ajoute un paquetage pré-compilé.

pkg_delete

enlève un paquetage installé. Les noms peuvent être donnés avec ou sans la version. Si aucune version n'est donnée, pkg_delete trouvera celle qui est installée. Les caractères d'équivalences peuvent être utilisés à condition d'être détachés de l'interpréteur de commandes. Par exemple:

```
# pkg_delete "*emacs*"
```

L'option `-r` est très puissante: elle enlève tous les paquetages requis par celui en question puis celui en question lui-même. Par exemple:

```
# pkg_delete -r jpeg
```

enlève jpeg et tous les paquetages dont il a besoin. Cela permet de le mettre à jour.

pkg_info

affiche des informations sur les paquetages installés ou non-installés.

pkg_create

crée un nouveau paquetage pour la collection. Ce programme sert pour les versions pré-compilées. Il est appelé par le compilateur et n'a donc pas besoin d'être tapé à la main.

pkg_admin

exécute diverses fonctions administratives sur le système de paquetages.

8.6. Guide rapide pour empaqueter

Cette section (Guide rapide pour empaqueter) est une contribution de Jason R. Fink

Cette section décrit une méthode rapide pour construire de relativement petits paquetages pour NetBSD. Pour plus de détails sur la complexité du système de paquetages de NetBSD, reportez-vous au fichier `Packages.txt` dans l'arborescence de pkgsrc.

8.6.1. Outils

Il existe trois outils fondamentaux pour rapidement construire un petit paquetage supplémentaire pour NetBSD:

url2pkg
un paquetage de modèles
pkglint

8.6.1.1. url2pkg

L'utilitaire url2pkg peut être installé depuis l'arborescence de pkgsrc. Il aide le constructeur de paquetages à rapidement remplir et tester les aspects rudimentaires de la construction.

8.6.1.2. Paquetage de modèles

Il sert à celui qui compile. `Packages.txt` montre un exemple de compilation à la section 11 "A simple example of a package: bison". La plupart des paquetages de pkgsrc ne sont pas aussi complexes.

8.6.1.3. pkglint

L'utilitaire pkglint peut être installé depuis l'arborescence de pkgsrc. Il vérifie l'intégrité du paquetage.

8.6.2. Première étape

Démarrer le processus est assez simple. Il est important que le constructeur (par exemple vous) ait déjà testé la compilation sur un système NetBSD. Autrement, établir un paquetage peut devenir problématique si la compilation échoue. Notez que la plupart du temps, un correctif doit être écrit pour les sources et être inclus dans le paquetage. Mais cela n'entre pas dans le cadre de ce guide rapide (voyez Packages.txt pour plus de détails).

8.6.2.1. Utiliser url2pkg

L'étape suivante utilise url2pkg.

Voici les étapes à suivre afin que url2pkg crée les premiers fichiers utiles au nouveau paquetage:

1. Créez le répertoire du nouveau paquetage dans le répertoire approprié de pkgsrc. Ne mettez encore rien dedans.
2. Allez (cd) dans ce répertoire.
3. Tapez

   ```
   $ url2pkg
   ```

4. On vous demande une url, donnez-là et appuyez sur • Retour• .
5. Un session vi s'ouvre.

   Note : cela utilise la localisation par défaut, nvi, de vi sur NetBSD. Si vous avez l'habitude d'utiliser un clone tel que vim, vous pourrez avoir des erreurs .exrc.

   La session vi sert à établir le fichier Makefile du nouveau paquetage. Il vous faut entrer le nom du paquetage, l'adresse électronique de son développeur et sa catégorie.

6. Sauvegardez et quittez.
7. url2pkg va automatiquement chercher le paquetage et le place dans le sous-répertoire de travail.
8. url2pkg génère ensuite les fichiers md5.

Nous en avons terminé avec url2pkg. Veuillez noter qu'il ne remplit aucun fichier hormis Makefile. Il génère un fichier PLIST vide.

8.6.3. Etapes suivantes

Maintenant que le fichier Makefile est généré, le reste des fichiers doit être créé. A l'aide de votre paquetage de modèles, copiez les fichiers suivant depuis le sous-répertoire pkg:

DESCR

Une description du logiciel sur plusieurs lignes. Elle doit aussi contenir les crédits.

COMMENTS

Description du logiciel sur une ligne. Pas besoin de préciser le nom du paquetage, il sera automatiquement ajouté par les outils pkg_* au moment voulu.

PLIST

Ce fichier contient la localisation des fichiers à installer sur le système. Pour un petit paquetage (par exemple, un binaire et une ou deux pages de manuel), jeter un oeil au Makefile de la distribution, au script d'installation... peut facilement donner une idée de l'endroit où les placer.

8.6.4. Vérification avec pkglint

Une fois tous les fichiers prêts, il reste à les vérifier avec l'outil pkglint. La plupart du temps, il faut, dans le fichier Makefile, déplacer une section, la modifier ou en ajouter une. Cependant, pour la première fois, il est utile de lancer pkglint pour savoir exactement ce que vous avez à changer. Voici un extrait issu du fichier Packages.txt d'une session pkglint:

```
$ pkglint
OK: checking pkg/COMMENT.
OK: checking pkg/DESCR.
OK: checking Makefile.
OK: checking files/md5.
OK: checking patches/patch-aa.
looks fine.
```

Si une erreur survient, elle est normalement facile à résoudre. Voici un exemple d'erreur que j'ai obtenue lorsque de la construction d'un paquetage:

```
extract suffix not required
```

Je n'avais pas besoin de définir de suffixe d'extraction dans le fichier Makefile.

8.6.5. Lancer et vérifier la compilation et l'installation

A ce stade, si pkglint est passé correctement, je lance, normalement, une vérification complète du processus: aller chercher, compiler et installer. Pour le faire proprement, il faut effacer le répertoire de travail ainsi que les distfiles de `/usr/pkgsrc/distfiles`. De cette manière, je suis sûr que le test sera effectué entièrement.

8.6.6. Soumettre un paquetage avec send-pr

Tout d'abord, faites une archive de l'arborescence du paquetage (y compris le sous-répertoire pkg/work) comme ceci:

```
$ tar -czf packagename.tgz package_dir
```

Ensuite, déposez cette archive à un endoit accessible au gestionnaires des paquetages de NetBSD. Si vous ne le pouvez pas, contactez NetBSD pour savoir s'il existe d'autres méthodes pour lui donner.

La meilleure méthode pour prévenir le gestionnaire de paquetages de NetBSD est d'utiliser l'utilitaire send-pr avec la catégorie • pkg• , un synopsis indiquant le nom et la version, une courte description et l'URL du fichier tar.

Vous pouvez utiliser l'utilitaire **send-pr** soit depuis votre NetBSD soit en ligne à l'adresse http://www.NetBSD.org/cgi-bin/sendpr.cgi?gndb=netbsd au cas où l'utilitaire soit indisponible.

8.6.7. Dernière note

Encore une fois, ce petit guide n’est utile que pour les petits paquetages ne contenant que quelques fichiers à installer sur NetBSD. Il part du principe que ce paquetage ne nécessite aucun correctif et que la compilation s’effectue sans dépendances.

Pour une utilisation plus complexe, reportez-vous au fichier Packages.txt.

Chapitre 9. Réseau

Ce chapitre introduit quelques notions de base sur le réseau telles que la connexion à internet ou à un réseau local (LAN). La connexion entre deux ordinateurs par l'intermédiaire d'un câble est aussi décrite.

Pour que deux ordinateurs puissent échanger des données, certains types de connexions doivent exister entre eux. Par exemple, les deux machines doivent avoir une carte réseau (NIC) reliées entre elles par un câble réseau, éventuellement par l'intermédiaire de concentrateurs (Hub) ou de commutateurs. Ce type de connexion est généralement appelée *réseau local* (ou LAN: Local Area Network).

L'internet, en revanche, est un *réseau général* (ou WAN: Wide Area Network): lorsque vous êtes connectés à internet, vous pouvez accéder à des ordinateurs vraiment très éloignés sans avoir besoin de connaitre le détail des connexions intermédiaires.

9.1. Fichiers de configuration du réseau

La liste qui suit présente les fichiers utilisés pour configurer le réseau. Leur utilisation, certains ayant déjà été rencontrés dans les chapitres précédents, seradécrite dans les sections suivantes.

`/etc/hosts`

Base de données des hôtes locaux. Chaque ligne contient des informations sur un hôte connu comme son adresse, son nom et ses alias. Les petits réseaux peuvent être configurés avec ce fichier sans avoir recours à un *serveur de noms*.

`/etc/resolv.conf`

Ce fichier indique de quelle manière les routines qui permettent l'accès au Système de Noms de Domaines doivent opérer. D'une manière générale, il contient la liste des serveurs de noms.

`/etc/ifconfig.xxx`

Ce fichier permet de configurer automatiquement les cartes réseau au démarrage.

`/etc/mygate`

Contient l'adresse IP de la passerelle (gateway).

`/etc/nsswitch.conf`

Fichier de configuration du sélecteur de noms de domaines. Il permet de définir l'ordre dans lequel un programme recherche dans les diverses bases de données un nom, un utilisateur, un groupe... Par exemple, la ligne:

```
hosts:    files dns
```

indique que la base de données des hôtes provient de deux sources: *files* (le fichier `/etc/hosts` local) et *DNS* (Serveur de Noms de Domaines). La recherche s'effectue d'abord dans le fichier puis avec le DNS.

Il n'est généralement pas nécessaire de modifier ce fichier.

9.2. Connexion à Internet par PPP

Il existe de nombreux types de connexions à internet: cette section explique comment se connecter par l'intermédiaire d'un fournisseur d'accès (FAI) avec un modem et une ligne téléphonique en utilisant le protocole PPP: une configuration très courante. Pour établir la connexion, il faut:

1. Avoir les informations de connexion du fournisseur.
2. Modifier le fichier `/etc/resolv.conf` et vérifier`/etc/nsswitch.conf`.
3. Créer les répertoires `/etc/ppp` et `/etc/ppp/peers` s'ils n'existent pas.
4. Créer le script de connexion, le fichier de négociation (chat) et le fichier d'options de pppd.
5. Créer le fichier d'authentification utilisateur - mot de passe.

Au vu de cette liste, il semble que la procédure soit compliquée et demande beaucoup de travail. En fait, chaque étape est très simple puisqu'elle consiste généralement à modifier, créer ou vérifier quelques petits fichiers texte. Dans l'exemple suivant, nous considérons que le modem est connecté au second port série `/dev/tty01` (COM2 sous DOS).

> En plus des modems connectés aux ports COM (avec `/dev/tty0[012]` sur i386, `/dev/tty[ab]` sur sparc, ...) les modems USB (`/dev/ttyU*`) et pcmcia/cardbus (`/dev/tty0[012]`) peuvent être utilisés.

9.2.1. Les informations de connexion

La première chose à faire est d'obtenir du fournisseur d'accès les informations concernant la connexion. C'est à dire:

- Le numéro de téléphone du point d'accès le plus proche.
- La méthode d'autentification à utiliser.
- Votre nom d'utilisateur et mot de passe pour la connexion.
- L'adresse IP des serveurs de noms.

9.2.2. resolv.conf et nsswitch.conf

Le fichier `/etc/resolv.conf` doit être renseigné avec les informations données par le fournisseur d'accès, en particulier l'adresse des serveurs de noms ou DNS. Dans notre exemple, nous aurons deux serveurs qui auront les adresses "194.109.123.2" et "191.200.4.52".

Exemple 9-1. resolv.conf

```
nameserver 194.109.123.2
nameserver 191.200.4.52
#lookup file bind
```

Note : la dernière ligne ("lookup file bind") indique que le serveur de noms ne sera utilisé que pour les noms absents du fichier `/etc/hosts`. Elle est en commentaire car, à partir de NetBSD 1.4, elle ne sert plus à rien: ce type d'information est désormais définit dans le fichier `/etc/nsswitch.conf`. Le nouveau Selecteur de Service de Noms (Name Service Switch) gère l'accès aux bases de données utilisées par les programmes pour trouver des informations système.

Voici un exemple de fichier `/etc/nsswitch.conf`.

Exemple 9-2. nsswitch.conf

```
# /etc/nsswitch.conf
group:          compat
group_compat:   nis
hosts:          files dns
netgroup:       files [notfound=return] nis
networks:       files
passwd:         compat
passwd_compat: nis
```

Note : seule une ligne a été modifiée: celle commençant par le mot • hosts• . Pour la résolution de noms, la recherche s'effectuera dans le fichier `hosts` puis avec le DNS.

9.2.3. Créer les répertoires de pppd

Les répertoires `/etc/ppp` et `/etc/ppp/peers` vont contenir les fichiers de configuration des connexions PPP. Après une nouvelle installation de NetBSD, ils n'existent pas et doivent être créés à la main (mkdir et chmod 700).

9.2.4. Script de connexion et fichier de négociation

Le script de connexion est utilisé comme paramètre sur la ligne de commandes de pppd. Il se trouve dans `/etc/ppp/peers` et porte généralement le nom du fournisseur d'accès. Par exemple, si le nom de votre fournisseur est • BigNet• et votre nom d'utilisateur • alain• , le fichier peut être:

Exemple 9-3. Script de connexion

```
# /etc/ppp/peers/bignet
connect '/usr/sbin/chat -v -f /etc/ppp/peers/bignet.chat'
noauth
user alan
remotename bignet.it
```

Dans cet exemple, le script spécifie un fichier *chat* (fichier de négociation, prononcez • tchatte•) à utiliser pour la connexion. Les options du script sont détaillées dans la page de manuel pppd(8).

Note : si vous rencontrez des problèmes de connexion, ajoutez la ligne suivante à votre script:

```
debug
kdebug 4
```

vous aurez une trace des opérations effectuées lorsque le système essaie de se connecter. Reportez-vous aux pages de manuel pppd(8) et syslog.conf(5).

Le script de connexion appelle l'application chat pour s'occuper de la connexion physique (initialisation du modem, numérotation...). Les paramètre de chat peuvent être définis dans le script lui-même mais il est préférable de les mettre dans un autre fichier. Si, par exemple, le numéro de téléphone du point d'accès est le "02 99 99 99 99", voici un exemple de fichier de négociation:

Exemple 9-4. Fichier de négociation (chat)

```
# /etc/ppp/peers/bignet.chat
ABORT BUSY
ABORT "NO CARRIER"
ABORT "NO DIALTONE"
″ ATDT0299999999
CONNECT ″
```

Note : si vous rencontrez des problèmes avec ce fichier, vous pouvez essayer de vous connecter manuellement à l'aide du programme cu et vérifier les messages exacts que vous obtenez. Voyez cu(1).

9.2.5. Authentification

Pendant la procédure d'authentification, les deux systèmes vérifient l'identité de l'autre. Vous n'êtes, en principe, pas supposé vérifier celle du fournisseur mais, lui, doit vérifier la votre. Deux méthodes existent:

- login
- PAP/CHAP

La plupart des fournisseurs d'accès utilisent la méthode PAP/CHAP.

9.2.5.1. Authentification avec PAP/CHAP

Les informations utiles se trouvent dans le fichier `/etc/ppp/pap-secrets` pour PAP et dans `/etc/ppp/chap-secrets` pour CHAP. Les lignes sont de la forme:

```
utilisateur * mot-de-passe
```

Par exemple:

```
alan * pZY9o
```

> **Note :** pour des raisons de sécurité, le possesseur des fichiers `pap-secrets` et `chap-secrets` doit être *root* et avoir les permissions • 600• .

9.2.5.2. Authentification avec login

Ce type d'authentification n'est plus très utilisé de nos jours. Si votre fournisseur d'accès l'utilise, votre nom d'utilisateur et votre mot de passe doivent se trouver dans le fichier de négociation (chat) et non dans les fichiers PAP/CHAP car celui-ci simule une connexion (login) interractive. Dans ce cas, donnez les permissions adéquates au fichier de négociation.

Voici un exemple de fichier de négociation pour une authentification avec login:

Exemple 9-5. Fichier de négiciation avec login

```
# /etc/ppp/peers/bignet.chat
ABORT BUSY
ABORT "NO CARRIER"
ABORT "NO DIALTONE"
” ATDT0299999999
CONNECT ”
TIMEOUT 50
ogin: alan
ssword: pZY9o
```

9.2.6. pppd options

La dernière chose qu'il nous reste à faire est d'établir le fichier d'options de **pppd**: `/etc/ppp/options` (chmod 644).

Exemple 9-6. /etc/ppp/options

```
/dev/tty01
lock
crtscts
57600
modem
defaultroute
noipdefault
```

Consultez la page de manuel pppd(8) pour connaitre la signification des options.

9.2.7. Tester le modem

Avant d'activer le lien, nous allons rapidement tester le modem afin de vérifier s'il est bien connecté à l'ordinateur et qu'il communique correctement. Le programme cu peut être utilisé à cette fin, comme dans l'exemple qui vient:

1. Créez le fichier `/etc/uucp/port` contenant les lignes suivantes:

   ```
   type modem
   port modem
   device /dev/tty01
   speed 115200
   ```

   (substitutez les valeurs avez les votres, si elles sont différentes).

2. Tapez la commande **cu -p modem** pour commencer à envoyer des commandes au modem. Par exemple:

   ```
   # cu -p modem
   Connected.
   ATZ
   OK
   ~.

   Disconnected.
   #
   ```

   Dans cet exemple, la commande de réinitialisation (ATZ) a été envoyée au modem. Il a répondu • OK• : la communication fonctionne. Pour quitter cu, tapez "~" (tilde) suivi de "." (point), comme dans l'exemple.

COM, com et tty

Quelques mots sur la différence entre *com*, *COM* et *tty*. Pour NetBSD, • com• est le nom du pilote du port série (celui affiché par **dmesg**) et • tty• est le nom du port. Comme la numérotation commence à partir de 0, com0 est le pilote du premier port série, nommé tty00. Dans le monde DOS, en revanche, COM1 fait référence au premier port série (généralement localisé en 0x3f8), COM2 au deuxième, etc... Ainsi, COM1 (DOS) correspond à tty00 (NetBSD).

9.2.8. Activer le lien

Maintenant que tout est prêt, nous pouvons lancer la connexion avec la commande:

```
# pppd call bignet
```

où `bignet` est le nom du script de connexion décrit plus haut. Pour visualiser les messages de connexion de **pppd**, tapez la commande suivante:

```
# tail -f /var/log/messages
```

Pour raccrocher, effectuez un **kill -HUP** de **pppd**.

9.2.9. Utiliser un script pour connecter et déconnecter

Lorsque la connexion fonctionne correctement, nous pouvons écrire des scripts pour éviter d'avoir à retaper les commandes à chaque fois. Nous allons écrire deux scripts nommés, par exemple, `connecter` et `raccrocher`.

`connecter` sert à se connecter chez le fournisseur d'accès.

Exemple 9-7. connecter

```
#!/bin/sh
MODEM=tty01
POP=bignet
if [ -f /var/spool/lock/LCK..$MODEM ]; then
  echo ppp est d\éj\à actif...
else
  pppd call $POP
  tail -f /var/log/messages
fi
```

`raccrocher` sert à fermer la connexion et raccrocher.

Exemple 9-8. raccrocher

```
#!/bin/sh
MODEM=tty01
if [ -f /var/spool/lock/LCK..$MODEM ]; then
  echo -f killing pppd...
  kill -HUP `cat /var/spool/lock/LCK..$MODEM`
  echo done
else
  echo ppp n'est pas actif
fi
```

Les deux scripts tirent avantage du fait que, lorsque **pppd** est actif, il crée un fichier `LCK..tty01` dans le répertoire `/var/spool/lock`. Ce fichier contient le *numéro de processus* de **pppd**.

Les deux scripts doivent être exécutables:

```
# chmod u+x connecter raccrocher
```

9.3. Mise en place d'un réseau local

Le réseau est l'un des points forts d'Unix et donc de NetBSD: le réseau est à la fois puissant, facile à mettre en place et peu cher car il ne nécessite aucun investissement logiciel supplémentaire ni l'acquisition d'un serveur. La Section 9.4 explique comment configurer NetBSD pour qu'il se comporte comme une passerelle vers un réseau: avec IPNAT, tous les hôtes d'un réseau peuvent atteindre l'internet à partir d'une unique connexion chez un fournisseur d'accès étable par la passerelle. La seule attention à apporter concerne l'acquisition de cartes réseau reconnues par NetBSD (reportez-vous au fichier `INSTALL` qui donne la liste du matériel supporté).

Tout d'abord, les cartes réseaux doivent être installées puis branchées à un concentrateur, un commutateur ou directement (voir Figure 9-1).

Ensuite, vérifiez que les cartes réseaux sont reconnues par le noyau. Pour celà, servez-vous de la commande **dmesg**. Dans l'exemple suivant, le noyau a correctement reconnu un clone NE2000:

```
...
ne0 at isa0 port 0x280-0x29f irq 9
ne0: NE2000 Ethernet
ne0: Ethernet address 00:c2:dd:c1:d1:21
...
```

Si la carte n'est pas reconnue, vérifiez que le pilote est bien activé dans le fichier de configuration du noyau et que l'IRQ de la carte correspond à celui qu'attend le noyau. Dans l'exemple, c'est la ligne isa NE2000 qui doit être activée dans le fichier de configuration. Le noyau s'attend à ce que la carte soit à l'IRQ 9.

```
...
ne0 at isa? port 0x280 irq 9 # NE[12]000 ethernet cards
...
```

Si la configuration de la carte est différente, elle ne sera probablement pas reconnue au démarrage. Dans ce cas, vous avez le choix entre, soit changer la configuration du noyau et de le recompiler, soit changer les paramètrages de la carte (généralement à l'aide d'une disquette de paramètrage ou, pour les ancien matériels, à l'aide de cavaliers sur la carte).

La ligne suivante montre la configuration actuelle de la carte:

```
# ifconfig ne0
ne0: flags=8822<BROADCAST,NOTRAILERS,SIMPLEX,MULTICAST> mtu 1500
   media: Ethernet 10base2
```

La configuration logicielle de la carte est très facile. L'adresse IP "192.168.1.1" (généralement réservée aux réseaux internes) est assignée à la carte de cette façon:

```
# ifconfig ne0 inet 192.168.1.1 netmask 0xffffff00
```

En répétant la commande précédente, nous avons désormais un résultat différent:

```
# ifconfig ne0
ne0: flags=8863<UP,BROADCAST,NOTRAILERS,RUNNING,SIMPLEX,MULTICAST> mtu 1500
   media: Ethernet 10base2
   inet 192.168.1.1 netmask 0xffffff00 broadcast 192.168.1.255
```

L'affichage d'**ifconfig** a changé: l'adresse IP apparait ainsi que deux nouvelles indications, • UP• et • RUNNING• . Si l'interface n'est pas • UP• , elle ne sera pas utilisée pour l'envoi de paquets.

La machine possède l'adresse "192.168.1.1" qui se trouve dans la plage d'adresses réservées aux réseaux internes et, donc, inaccessibles depuis internet. La configuration est terminée et doit être testée. S'il existe une autre machine active sur le réseau, un "*ping*" peut être essayé. Par exemple, si "192.168.1.2" est l'adresse de cette deuxième machine:

```
# ping 192.168.1.2
PING ape (192.168.1.2): 56 data bytes
64 bytes from 192.168.1.2: icmp_seq=0 ttl=255 time=1.286 ms
64 bytes from 192.168.1.2: icmp_seq=1 ttl=255 time=0.649 ms
64 bytes from 192.168.1.2: icmp_seq=2 ttl=255 time=0.681 ms
64 bytes from 192.168.1.2: icmp_seq=3 ttl=255 time=0.656 ms
^C
----ape PING Statistics----
4 packets transmitted, 4 packets received, 0.0% packet loss
round-trip min/avg/max/stddev = 0.649/0.818/1.286/0.312 ms
```

Avec le paramètrage actuel, au prochain démarrage, il faudra recommencer la configuration de la carte réseau. Pour éviter d'avoir à la faire à chaque fois, il y a deux choses à faire: premièrement créer le fichier `/etc/ifconfig.ne0` contenant la ligne suivante:

```
inet 192.168.1.1 netmask 0xffffff00
```

Deuxièmement, dans le fichier `/etc/rc.conf`, mettre l'option suivante:

```
auto_ifconfig=YES
```

Au prochain démarrage, la carte réseau sera automatiquement configurée.

Le fichier `/etc/hosts` est la base de données des adresses IP et des nom d'alias: Il doit contenir les adresses de tous les hôtes du réseau interne. Par exemple:

```
# $NetBSD: hosts,v 1.4 1997/01/09 05:33:14 mikel Exp $
#
# Base de données des hôtes
# Ce fichier doit contenir les adresses et alias
# de toutes les machines qui partagent ce fichier.
# Il n'est utilisé que par ifconfig et d'autres opérations
# avant que le serveur de noms soit en service.
#
#
127.0.0.1  localhost
#
# RFC 1918 indique que ces réseaux sont "internes".
# 10.0.0.0 10.255.255.255
# 172.16.0.0 172.31.255.255
# 192.168.0.0 192.168.255.255

192.168.1.1 ape.insetti.net ape
192.168.1.2 vespa.insetti.net vespa
```

Le fichier `/etc/nsswitch.conf` doit être modifié comme expliqué dans l'Exemple 9-2.

> **Note :** dans cet exemple, le fichier `/etc/ifconfig.ne0` a été créé car la carte réseau a été reconnue en tant que *ne0* par le noyau. Si vous utilisez un adaptateur différent, remplacez ne0 par le votre.

En résumé, pour configurer un réseau, il faut : tout d'abord que les cartes réseaux soient installées et physiquement connectées. Ensuite qu'elles soient configurées (**ifconfig**). Enfin que les fichiers `/etc/hosts` et `/etc/nsswitch.conf` soient adaptés. Ce type de gestion de réseau est très simplifié et n'est destiné qu'à de petits réseaux sans besoins sophistiqués.

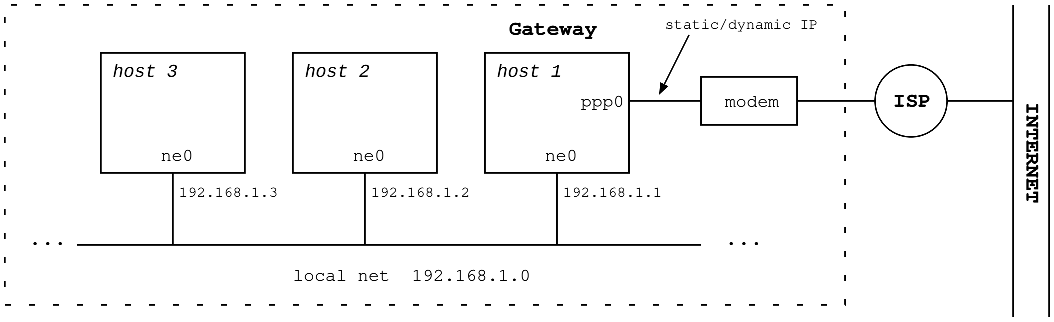
9.4. IPNAT

Derrière les mystérieuses intiales IPNAT, se cache le Protocole de Traduction d'Adresses Réseau IP (Internet Protocol Network Address Translation, on trouvera aussi NAT tout court). Il permet de connecter un réseau interne à un vrai réseau (internet). Cela veut dire qu'avec une • vraie• adresse IP, statique ou dynamique, appartenant à une passerelle possédant IPNAT, il est possible de donner accès à internet simultanément à tous les ordinateurs du réseau interne.

Quelques exemple d'utilisation d'IPNAT se trouvent dans le répertoire `/usr/share/examples/ipf`. Regardez `BASIC.NAT` et `nat-setup`.

Le paramètrage de l'exemple décrit dans cette section est détaillé par la Figure 9-1: *host 1* peut se connecter à internet en appelant un fournisseur d'accès qui lui donne une adresse IP dynamique. *host 2* et *host 3* ne peuvent communiquer avec internet avec un paramètrage normal. IPNAT leur permet de le faire: host 1 joue le rôle d'une passerelle pour host 2 et 3. NdT: ISP est leFournisseur d'Accès à Internet.

Figure 9-1. Réseau avec passerelle



9.4.1. Configurer la passerelle/pare-feu

Pour que NAT soit utilisable, la ligne "pseudo-device ipfilter" du fichier de configuration du noyau doit être activée. Pour vérifier si elle l'est dans le noyau actuel:

```
# sysctl net.inet.ip.forwarding
net.inet.ip.forwarding = 1
```

Si le résultat est "1", comme dans l'exemple précédent, l'option est activée. Autrement, si le résultat est • 0• , elle ne l'est pas. Vous pouvez faire deux choses:

1. Compiler un nouveau noyau dont l'option GATEWAY (passerelle) est activée par défaut.
2. Activer l'option du noyau actuel avec la commande:

```
# sysctl -w net.inet.ip.forwarding=1
```

Si vous placez cette commande dans un script de démarrage (par exemple `/etc/rc.local`) elle sera exécutée à chaque démarrage de la machine.

Le reste de cette section explique comment mettre en place une configuration NAT qui démarre à chaque fois que la connexion avec le fournisseur d'accès se fait par un lien PPP. De cette manière, tous les ordinateurs du réseau familial (par exemple) pourront se connecter à internet par la machine passerelle, même s'ils ne sont pas sous NetBSD.

Tout d'abord, créez un fichier `/etc/ipf.conf` vide à l'aide de la commande:

```
# touch /etc/ipf.conf
```

Ensuite, créez le fichier `/etc/ipnat.conf` contenant les règles suivantes:

```
map ppp0 192.168.1.1/24 -> 0/32 proxy port ftp ftp/tcp
map ppp0 192.168.1.1/24 -> 0/32 portmap tcp/udp 40000:60000
map ppp0 192.168.1.1/24 -> 0/32
```

192.168.1.1/24 sont les adresses réseau qui doivent être gérées (dans notre cas, 192.168.1.0/24 est aussi valable). La première ligne du fichier est optionnelle: elle permet le ftp actif par la passerelle. La seconde ligne est utilisée pour gérer correctement les paquets tcp et udp : la traduction des ports (portmap) est nécessaire à cause de la traduction n vers 1. La troisième ligne permet d'utiliser ICMP, ping...

Dans le fichier `/etc/rc.conf`, portmap doit être activé (ipfilter n'est pas nécessaire).

Créez un fichier `/etc/ppp/ip-up`. Il sera appelé à chaque fois que le lien PPP sera activé.

```
#!/bin/sh
# /etc/ppp/ip-up
/usr/sbin/ipnat -F
/usr/sbin/ipnat -C
/sbin/ipf -E
/usr/sbin/ipnat -f /etc/ipnat.conf
```

Créez un fichier `/etc/ppp/ip-down`. Il sera appelé à chaque fois que le lien PPP sera fermé.

```
#!/bin/sh
# /etc/ppp/ip-down
/sbin/ipf -D
/usr/sbin/ipnat -C
```

Ces deux fichiers, `ip-up` et `ip-down` doivent être exécutables:

```
# chmod u+x ip-up ip-down
```

La machine passerelle est désormais prête.

9.4.2. Configurer les clients

Créez un fichier `/etc/resolv.conf`, comme sur la machine passerelle.

Tapez la commande suivante:

```
# route add default 192.168.1.1
```

192.168.1.1 est l'adresse de la passerelle mise en place à la section précédente.

Bien entendu, vous n'aurez pas envie de taper cette commande à chaque fois. Il nous faut donc définir la "route par défaut" (defaultroute) dans le fichier `/etc/rc.conf` ou bien, ce qui revient au même, écrire l'adresse de la passerelle dans le fichier `/etc/mygate`. La route par défaut sera définie automatiquement pendant l'initialisation du système à l'aide du contenu de `/etc/mygate` ou de l'option
• defaultroute• qui sera utilisé comme argument de la commande **route add default**.

Si l'ordinateur client n'est pas sous NetBSD, la configuration sera différente. Pour un PC sous Windows, la passerelle se déclare, dans les propriétés du protocole TCP/IP, avec l'adresse de la passerelle NetBSD. Sur Macintosh, l'adresse se déclare dans le tableau de bort TCP/IP dans le champ "Adresse du routeur".

C'est tout ce qu'il y a à faire sur les ordinateurs clients.

9.4.3. Quelques commandes utiles

Les commandes suivantes peuvent être utiles pour diagnostiquer des problèmes:

ping

netstat -r

Affiche les tables de routage (similaire à **route show**).

traceroute

Sur le client, il affiche la route suivie par les paquets vers leur destination.

tcpdump

Sur la passerelle, permet de visualiser le traffic TCP/IP.

9.5. Connecter deux PC avec un câble série

S'il vous faut transférer des fichiers d'un ordinateur à l'autre et que ceux-ci ne sont pas connectés à un réseau, il existe une solution pratique quand la copie sur disquette s'avère laborieuse et dépassée: un réseau peut être mis en place à l'aide d'un câble série (un câble *null modem*). La présente section décrit quelques configurations.

9.5.1. Connecter NetBSD avec BSD ou Linux

Le cas le plus simple est lorsque les deux machines tournent sous NetBSD. Une connexion peut être établie grâce au protocole SLIP. Sur la première machine, tapez:

```
# slattach /dev/tty00
# ifconfig sl0 inet 192.168.1.1 192.168.1.2
```

Sur la seconde machine, tapez:

```
# slattach /dev/tty00
# ifconfig sl0 inet 192.168.1.2 192.168.1.1
```

Vous pouvez alors tester la configuration avec **ping**. Par exemple, sur le second ordinateur, tapez:

```
# ping 192.168.1.1
```

Si ça marche, vous avez alors un réseau actif entre les deux machines. Des commandes telles que ftp, telnet et d'autres peuvent être exécutées. Un nom de machine peut être inscrit dans le fichier `/etc/hosts`.

- Dans notre exemple, les deux PC utilisent le premier port série (`/dev/tty0`). Inscrivez le périférique approprié si vous en utilisez un autre.
- Les adresses IP comme 192.168.x.x sont réservées pour des réseaux • internes• . Le premier PC a l'adresse 192.168.1.1 et le second 192.168.1.2.
- Pour avoir une connexion plus rapide, vous pouvez spécifier l'option `-s` *`speed`* à **slattach**.
- **ftp** ne peut être utilisé que si inetd est actif et que le serveur ftpd est lancé.

Linux : si un des deux PC tourne sous Linux, les commandes sont légèrement différentes (uniquement sur la machine Linux). Considérons qu'elle possède l'adresse 192.168.1.2, les commandes suivantes sont nécessaires:

```
# slattach -p slip -s 115200 /dev/ttyS0 &
# ifconfig sl0 192.168.1.2 pointopoint 192.168.1.1 up
# route add 192.168.1.1 dev sl0
```

N'oubliez pas le "&" à la fin de la première commande.

9.5.2. Connecter NetBSD et Windows NT

NetBSD et Windows NT peuvent (souvent) être facilement mis en réseau avec un *câble série*. Le principe consiste à créer une connexion "Accès réseau à distance" sous WindowsNT et de démarrer pppd sous NetBSD.

Démarrez pppd en tant que root, après avoir créé un fichier `.ppprc` dans le répertoire `/root`. Utilisez l'exemple suivant comme modèle:

```
connect '/usr/sbin/chat -v CLIENT CLIENTSERVER'
local
tty00
115200
crtscts
lock
noauth
nodefaultroute
:192.168.1.2
```

La première ligne sera expliquée plus loin. 192.168.1.2 est l'adresse IP qui sera assignée, par NetBSD, à la machine WindowsNT. `tty00` est le port série utilisé pour la connexion (premier port série).

Du côté de Windows NT, un périférique *câble série* (ou null modem) doit être installé à partir du panneau de configuration (icône Modems) et une connexion Accès réseau à distance doit être créée avec ce modem. Le pilote null modem (câble série) est standard sous Windows Nt 4 mais n'est pas 100% null modem: lorsque le lien est activé, Windows envoie le mot CLIENT et s'attend à recevoir la réponse CLIENTSERVER. C'est la raison de la présence de la première ligne du fichier `.ppprc`: **chat** doit répondre à Windows lorsque la connexion est activée sinon elle sera refermée.

Dans la configuration d'Accès réseau à distance, il faut préciser: utiliser le câble série, numéro de téléphone • 1• (qui n'est pas utilisé), serveur PPP, n'activer que le protocole TCP/IP, utiliser l'adresse IP et le serveur de nom du serveur (donc de NetBSD). Choisissez le contrôle de flux matériel et les paramètre de communication à 115200 8N1.

Maintenant, tout est prêt pour établir la connexion.

- Connectez les deux ordinateurs avec le câble série (null modem).
- Lancez **pppd** sur NetBSD. Pour visualiser les messages de pppd: **tail -f /var/log/messages**.
- Activez la connexion Accès réseau à distance de Windows NT.

9.5.3. Connecter NetBSD et Windows 95

La mise en place pour Windows 95 est la même que celle pour Windows NT: Accès réseau à distance sur Windows 95 et serveur PPP sur NetBSD. La plupart (si ce n'est toutes) des versions de Windows 95 ne possède pas de pilote *câble série*, ce qui rend les choses un peu plus compliquées. La solution la plus simple est de trouver un tel pilote sur internet (c'est un petit fichier `.INF`) et répéter les mêmes opérations que pour Windows NT. La seule différence est que la première ligne du fichier `.ppprc` de NetBSD (celle qui appelle **chat**) peut être enlevée.

Si vous ne trouvez pas de pilote câble série pour Windows 95, voici une astuce:

- Créez une connexion Accès réseau à distance, comme le décrit la Section 9.5.2 mais en utilisant un des "Modem standard".
- Dans `.ppprc`, remplacez la ligne qui fait appel à **chat** par:

  ```
  connect '/usr/sbin/chat -v ATH OK AT OK ATE0V1 OK AT OK ATDT CONNECT'
  ```

- Etablissez la connexion comme le décrit la Section 9.5.2.

Ici, le programme **chat**, appelé lorsque la connexion est établie, simule ce que Windows 95 appelle un "Modem standard". Il renvoie à Windows les réponses que feraient un modem standard. Dès que Windows 95 envoie une commande de modem, **chat** répond • OK• .

9.6. NFS

Maintenant que le réseau est en état de fonctionnement, il est possible de partager des fichiers et des répertoires avec les autres ordinateurs, en utilisant NFS. Du point de vue du partage de fichiers, l'ordinateur qui donne accès à ses données est appelé un *serveur*. Les ordinateurs qui utilisent ces données sont appelés des *client*. Un ordinateur peut être à la fois client et serveur.

- Le noyau doit être compilé avec les options appropriées au client et au serveur (elles sont faciles à trouver dans le fichier de configuration).
- Le serveur doit activer ses services RPC dans le fichier `/etc/inetd.conf`.
- Dans `/etc/rc.conf`, le serveur doit activer les démons inetd et portmap ainsi que l'option nfs_server.
- Dans `/etc/rc.conf`, le client doit activer inetd et nfs_client.
- Le serveur doit lister les répertoires à exporter dans le fichier `/etc/exports` puis lancer la commande **kill -HUP ‘cat /var/run/mountd.pid**.

Un client peut accéder à un répertoire distant par NFS si:

- Le serveur exporte ce répertoire.
- Le client monte ce répertoire distant avec la commande **mount server:/xx/yy /mnt**

La commande **mount** possède un riche répertoire d'options concernant les répertoires distants mais elle ne sont pas vraiment intuitives (c'est le moin qu'on puisse dire).

9.6.1. Exemple de mise en place NFS

Avertissement: j'ai tiré cet exemple plutôt complexe d'une liste de diffusion de NetBSD et je ne l'ai pas testé. Il doit cependant donner une idée de la façon dont fonctionne NFS.

Le scénario est le suivant: cinq machines client (cli1, ..., cli5) partagent des répertoires sur le serveur (buzz.toys.org). Certains des répertoires exportés sont réservés à certains clients et les autres sont accessibles à toutes les machines. Tous les clients démarrent depuis le serveur et doivent monter les répertoires.

Les répertoires à exporter du serveur sont:

`/export/cli?/root`

Les cinq répertoires racine des cinq client. Chaque machine possède sont propre répertoire racine.

`/export/cli?/swap`

Les cinq répertoires de mémoire virtuelle (swap) pour les cinq clients.

`/export/common/usr`

Le répertoire `/usr` commun à tous les clients.

`/usr/src`

Le répertoire `/usr/src` commun à tous les clients.

Les fichiers suivants existent sur le serveur:

```
/dev/ra0a on /
/dev/ra0f on /usr
/dev/ra1a on /usr/src
/dev/ra2a on /export
```

Chaque client a besoin des systèmes de fichiers suivants:

```
buzz:/export/cli?/root on /
buzz:/export/common/usr on /usr
buzz:/usr/src on /usr/src
```

La configuration du serveur est la suivante:

```
# /etc/exports
/usr/src -network 123.45.67.0 -mask 255.255.255.0
/export  -alldirs -maproot=root -network 123.45.67.0 -mask 255.255.255.0
```

Sur les clients, chaque fichier `/etc/fstab` contient:

```
buzz:/export/cli?/root / nfs rw
buzz:/export/common/usr /usr nfs rw,nodev,nosuid
buzz:/usr/src /usr/src rw,nodev,nosuid
```

Dans cet exemple, chaque caractère "?" doit être remplacé par le numéro du client correspondant.

Chapitre 10. Le système de Noms de Domaine (DNS)

Jason R Fink

Ce chapitre (Le système de Noms de Domaine (DNS)) est une contribution de Jason R. Fink

Le Système de Noms de Domaines sous NetBSD. Ce chapitre décrit la mise en place d'un domaine simple avec un Système de Noms de Domaine (DNS) sous NetBSD. Il ne donne pas de détails sur ce qu'est un DNS mais de courtes explications sont données. De plus amples renseignements peuvent être trouvés depuis le Répertoire de Ressources DNS (DNSRD) à la page http://www.dns.net/dnsrd/.

10.1. Notes et pré-requis

Les exemples présents font tous référence à l'utilisation du démon de noms internet de Berkeley (Berkeley Internet Naming Daemon ou BIND) version 9.0.0 disponible au Consortium de Logiciels Internet (Internet Software Consortium) http://www.isc.org et, bien entendu, dans la collection de paquetages de NetBSD. BIND 9.0.0 se trouve dans `/usr/pkgsrc/net/bind-9.0.0/`.

BIND 9 a été choisi par l'auteur car il s'agit de la plus récente diffusion de l'ISC.

Le lecteur doit déjà comprendre la répartition des adresses IP sur les hôtes et les spécifications de classes d'adresses IP.

10.2. Qu'est-ce qu'un DNS ?

Le Système de Noms de Domaines convertit des noms de machines en adresses IP. La correspondance se fait de nom en adresse et d'adresse en nom. La différence entre une conversion simple et une conversion par domaine introduit la notion de hiérarchie standardisée de noms. Elle fonctionne de droite à gauche, le plus haut niveau étant à droite. Prenons un exemple:

```
Premier niveau                           .org
                                           |
Niveau intermédiaire                  .diverge.org
                     ______________________|_________________________
                    |                      |                         |
Bas niveau  strider.diverge.org   samwise.diverge.org   wormtongue.diverge.org
```

Le shéma semble simple mais il peut être divisé logiquement à n'importe quel endroit. Dans l'exemple ci-dessus, le domaine "diverge.org" se divise en trois noeuds mais nous aurions pu le diviser en sous domaines comme "strider.net1.diverge.org", "samwise.net2.diverge.org" et "wormtongue.net2.diverge.org". Dans ce cas, nous aurions eu un noeud pour le domaine "net1.diverge.org" et deux pour "net2.diverge.org".

10.3. Les fichiers du DNS

Pour mettre en place notre DNS, nous continerons à utiliser les exemples ci-dessus. Avant de continuer, assurons-nous que:

- La connexion à notre fournisseur d'accès fonctionne correctement.
- IPNAT est actif et fonctionne.
- Tous les ordinateurs utilisent le DNS du fournisseur d'accès.

> **Note :** ce type de configuration est celui décrit au Chapitre 9.

Notre Serveur de Noms sera la machine "strider" qui fait aussi tourner IPNAT et les deux clients l'utilisent comme passerelle. Le type d'interface de • strider• n'est pas très important, disons qu'il utilise une ligne téléphonique à 56k.

Avant d'aller plus loin et avant d'effectuer les modifications pour utiliser le DNS, regardons le fichier • hosts• de strider.

Exemple 10-1. Le fichier /etc/hosts de strider

```
127.0.0.1       localhost
192.168.1.1     strider
192.168.1.2     samwise sam
192.168.1.3     wormtongue worm
```

Ce fichier n'est clairement qu'une ébauche de réseau mais il vaut la peine de constater que les règles que nous allons voir dans ce chapitre s'appliquent pareillement aux réseaux de grande dimension.

10.3.1. /etc/named.conf

Le premier fichier à voir est `/etc/named.conf`. C'est celui qui sert à configurer bind. Le paramètrage du système, tel que nous le voulons, est assez simple. Voici à quoi il doit ressembler:

```
options {
        directory "/etc/named";
        allow-transfer { 192.168.1.0/24; };
        recursion yes;
        allow-query { 192.168.1.0/24; };
        listen-on port 53 { 192.168.1.1; };
};

zone "localhost" {
   type master;
   notify no;
   file "zone.localhost";
};
```

```
zone "0.0.127.in-addr.arpa" in {
   type master;
   notify no;
   file "zone.127.0.0";
};

zone "diverge.org" {
   type master;
   notify no;
   file "zone.diverge.org";
};

zone "1.168.192.in-addr.arpa" {
   type master;
   notify no;
   file "zone.192.168.1";
};

zone "." in {
   type hint;
   file "zone.root";
};
```

A première vue, le contenu a l'air compliqué. En examinant de plus près, nous constatons qu'un certain nombre de lignes sont redondantes, il y aura finalement assez peu de concepts à expliquer.

Passons alors à la section consacrée à `named.conf`:

10.3.1.1. options

Cette section définit quelques paramètres globaux. Le plus important est le répertoire où se trouvent les tables du DNS. Pour ce système, elles seront placées dans `/etc/named`.

Voici le reste des paramètres:

allow-transfer

pour que les DNS du fournisseur d'accès jouent le rôle de serveurs secondaires et qu'ils aient accès aux les fichiers de notre zone.

allow-query

quels réseaux peuvent envoyer des demandes à ce serveur de noms.

listen-on port

le port sur lequel ce serveur named tournera.

Le reste du fichier `named.conf` consiste en une liste de • zones• . Chaque zone est associée à un fichier contenant des tables nécessaires à cette partie (ou • zone•) du domaine. Il va de soi que la forme des différentes sections dédiées aux zones, dans named.conf, est constament la même. Nous allons donc nous intéresser à une seul d'entre elles:

10.3.1.2. La zone "diverge.org"

type

> dans tous les cas, hormis ".", le types est • master• .

notify

> est utilisé pour avertir lorsque les zones subissent des modifications. Dans ce type de mise en place, personne n'a besoin d'être prévenu.

file

> le nom du fichier, dans le répertoire de named, où les renseignements sur cette zone peuvent être trouvés.

10.3.2. /etc/named/zone.localhost

La plus grande partie des fichiers de zones est identique mais chacun possède ses propiétés uniques. Voici à quoi ressemble le fichier `zone.localhost`:

Exemple 10-2. `zone.localhost`

```
 1|@ 1             IN SOA  strider.diverge.org. root.diverge.org. (
 2|                         1       ; Serial
 3|                         8H      ; Refresh
 4|                         2H      ; Retry
 5|                         1W      ; Expire
 6|                         1D)     ; Minimum TTL
 7|;
 8|                         IN NS   strider.diverge.org.
 9|;
10|                         IN A    127.0.0.1
```

Ligne par ligne:

Ligne 1

> Cette ligne est généralement la même pour toutes les zones à part la zone racine. Les parties intéressantes sont "strider.diverge.org" et "root.diverge.org". Evidemment, l'un est le nom du serveur et l'autre le nom du contact pour ce serveur. Dans la plupart des cas, root est ambiguë et il est préférable de mettre un compte de courriel (par exemple, le mien serait jrf.diverge.org.).

Ligne 2

> Cette ligne est un numéro de série. Généralement, les administrateurs indiquent la date à laquelle le serveur a été mis en place, sous la forme MMJJAAAA. Il doit être incrémenté à chaque fois qu'une modification est effectuée dans le fichier *after* l'installation. C'est pourquoi je préfère commencer à 1.

Ligne 3

Le taux de raffraichissement du serveur. Ici, les informations sont remises à jour toutes les huit heures.

Ligne 4

A quel intervelle de temps faut-il ré-essayer.

Ligne 5

Expiration de la recherche.

Ligne 6

La durée de vie minimum (TTL: Time To Live).

Ligne 7

Une ligne blanche pour la lisibilité (un séparateur de sections, si vous voulez).

Ligne 8

La ligne du serveur de noms. Comme nous pouvons le voir, il s’agit de strider.

Ligne 9

Une ligne blanche de séparation.

Ligne 10

L’enregistrement de l’hôte local, c’est à dire 127.0.0.1

ATTENTION

Notez que "strider.diverge.org." possède un point final. Il est essentiel que cette curiosité et les autres soient respectées pour éviter tout problème.

10.3.3. /etc/named/zone.127.0.0

C’est le fichier de recherche inverse de l’hôte local. Il ressemble à:

```
1| @ 1             IN SOA  strider.diverge.org. root.diverge.org. (
 2|                        1       ; Serial
 3|                        8H      ; Refresh
 4|                        2H      ; Retry
 5|                        1W      ; Expire
 6|                        1D)     ; Minimum TTL
 7| ;
 8|                 IN NS   strider.diverge.org.
 9| ;
10| 1               IN PTR  localhost.
```

Dans ce fichier, toutes les lignes sont les mêmes que celles du fichier de zones à part la ligne 10, l'enregistrement inverse. Il est défini dans un fichier séparé car il s'agit de l'hôte local qui possède une adresse totalement différente de celles des autres zones. Laissons cela pour le moment et finissons d'examiner les autres zones.

10.3.4. /etc/named/diverge.org

Ce fichier est peuplé par les enregistrements de tous les ordinateurs de notre réseau 192.168.1.0/24. Voici à quoi il ressemble:

```
1| @ 1             IN SOA  strider.diverge.org. root.diverge.org. (
 2|                         1       ; serial
 3|                         8H      ; refresh
 4|                         2H      ; retry
 5|                         1W      ; expire
 6|                         1D )    ; minimum seconds
 7| ;
 8|                 IN NS   strider.diverge.org.
 9|                 IN MX   10 maila.diverge.org.   ; primary mail server
10|                 IN MX   20 mailb.diverge.org.   ; secondary mail server
11| ;
12| strider         IN A     192.168.1.1
13| maila           IN CNAME strider.diverge.org.
14| samwise         IN A     192.168.1.2
15| www             IN CNAME samwise.diverge.org.
16| mailb
17| worm            IN A     192.168.1.3
```

Nous trouvons ici quelques nouveautés. Passons chaque ligne nouvelle en revue:

Ligne 9 et 10

> Cette ligne indique le gestionnaire de courrier électronique. Dans notre cas, il s'agit de strider mais, pour des raisons pratiques, nous voulons l'appeler • maila• . Comme nous le verrons plus loin, ce n'est pas un problème. Le numéro précédant "maila.diverge.org." indique la priorité. Plus il est petit plus la priorité est grande. Dans notre cas, si strider ne peut gérer le courriel, c'est mailb (en réalité samwise) qui s'en occupera.

Ligne 13

> CNAME veut dire "canonical name" et réprésente un alias. Nous avons ainsi créé les alias suivants:
>
> ```
> maila.diverge.org pour strider.diverge.org
> mailb.diverge.org pour samwise.diverge.org
> www.diverge.org   pour samwise.diverge.org
> ```

Le reste des enregistrements est simplement une correspondance entre une adresse IP et un nom complet.

10.3.5. /etc/named/zone.192.168.1

Ce fichier de zone est le fichier inverse pour tous les enregistrements d’hôtes. Sa forme est similaire à celle de l’hôte local sauf, évidemment, en ce qui concerne l’adresse:

```
@ 1             IN SOA  strider.diverge.org. root.diverge.org. (
                     1       ; serial
                     8H      ; refresh
                     2H      ; retry
                     1W      ; expire
                     1D )    ; minimum seconds
;
                IN NS   strider.diverge.org.
;
1               IN PTR  strider.diverge.org.
2               IN PTR  samwise.diverge.org.
3               IN PTR  worm.diverge.org.
```

10.3.6. /etc/named/zone.root

Ce fichier contient la liste des serveurs racine auxquels doit faire appel votre serveur lorsqu’il reçoit une demande concernant un autre domaine que le sien. En voici les première lignes:

```
$TTL 86400
.       IN      SOA     A.ROOT-SERVERS.NET.     hostmaster.INTERNIC.NET. (
                                1999033000      ;serial
                                1800  ;refresh every 30 min
                                900    ;retry every 15 minutes
                                604800 ;expire after a week
                                86400  ;minimum of a day
                                )
.                         518400 IN  NS    A.ROOT-SERVERS.NET.
A.ROOT-SERVERS.NET.       518400     A     198.41.0.4
.                         518400     NS    H.ROOT-SERVERS.NET.
H.ROOT-SERVERS.NET.       518400     A     128.63.2.53
.                         518400     NS    B.ROOT-SERVERS.NET.
B.ROOT-SERVERS.NET.       518400     A     128.9.0.107
.                         518400     NS    C.ROOT-SERVERS.NET.
C.ROOT-SERVERS.NET.       518400     A     192.33.4.12
.                         518400     NS    D.ROOT-SERVERS.NET.
D.ROOT-SERVERS.NET.       518400     A     128.8.10.90
.                         518400     NS    E.ROOT-SERVERS.NET.
E.ROOT-SERVERS.NET.       518400     A     192.203.230.10
.                         518400     NS    I.ROOT-SERVERS.NET.
I.ROOT-SERVERS.NET.       518400     A     192.36.148.17
.                         518400     NS    F.ROOT-SERVERS.NET.
F.ROOT-SERVERS.NET.       518400     A     192.5.5.241
.                         518400     NS    G.ROOT-SERVERS.NET.
G.ROOT-SERVERS.NET.       518400     A     192.112.36.4
. . .
```

Ce fichier peut être obtenu chez ISC, sur son site http://www.isc.org/et est habituellement fourni dans la distribution de BIND.

10.4. Utilisation du DNS

Dans cette section, nous allons voir comment mettre le DNS en route et paramètrer strider pour qu'il utilise ses propres services.

Tout d'abord, il nous faut lancer le démon name. Il existe plusieurs façons de le faire (dont la manière automatique au démarrage), mais, pour le moment, nous nous contenterons de faire simplement:

```
# /usr/sbin/named -u nobody -g nogroup
```

Pour quelque raison, vous pouvez vouloir le lancer sous un autre utilisateur, sachez tout de même que no* va très bien. Ensuite, nous allons faire pointer strider vers lui-même pour qu'il effectue les recherches. Deux étapes sont nécessaires: en premier, il faut décider l'ordre de recherche. Sur de petits réseaux comme le notre, il est très probable que chaque ordinateur possède sa propre liste des hôtes, l'ordre pourra donc être hôte puis DNS mais sur de grands réseaux, il est plus pratique d'utiliser le DNS. Le fichier dans lequel est défini cet ordre est `/etc/nsswitch.conf` (voir Exemple 9-2). En voici la partie intéressante:

```
. . .
passwd:     files nisplus nis
shadow:     files nisplus nis
group:      files nisplus nis
hosts:      dns files
. . .
```

La ligne qui nous concerne est celle commençant par • hosts• . • Files• indique au système d'utiliser le fichier `/etc/hosts` pour la traduction adresse-nom. L'entrée la plus à gauche (ici dns) indique la première méthode à utiliser.

Le second fichier à utiliser est `/etc/resolv.conf`, le fichier de recherche dns. Son format s'explique par lui-même mais voyons tout de même de quoi il a l'air:

```
nameserver 192.168.1.1
search diverge.org
domain diverge.org
```

La première ligne indique l'adresse du serveur de noms (assez facile). La ligne suivante demande de chercher dans ce domaine mais si une recherche n'aboutit pas, elle sera poursuivie dans la zone racine (celle indiquée par "." dans le fichier named.conf). La dernière ligne est notre domaine.

En théorie, nous voici prêts. Pour rapidement vérifier si notre DNS fonctionne, effectuons une recherche sur un hôte local puis sur une machine hors de notre domaine comme:

```
# host www.blah.net
```

Voici ce que nous obtenons après avoir lancé la commande **host www.yahoo.com**:

```
www.yahoo.com is a nickname for www.yahoo.akadns.net
```

```
www.yahoo.akadns.net has address 216.32.74.50
www.yahoo.akadns.net has address 216.32.74.51
www.yahoo.akadns.net has address 216.32.74.52
www.yahoo.akadns.net has address 216.32.74.53
www.yahoo.akadns.net has address 216.32.74.55
```

La procédure pour paramètrer les clients est la même. Modifiez les fichiers `/etc/nsswitch.conf` et `/etc/resolv.conf`.

Chapitre 11. Courrier électronique et forums de discussion

Ce chapitre explique comment paramètrer NetBSD pour utiliser le courriel (courrier électronique) et les forums de discussion (news). Seule une simple mais très commune mise en place sera décrite: un ordinateur connecté à internet par un modem chez un fournisseur d'accès. Ce chapitre est dans la continuité du Chapitre 9 puisqu'il utilise la même configuration réseau. Même notre • simple• mise en place prouve que c'est difficile si vous ne savez pas par où commencer ou si vous n'avez lu que des introductions ou des documentations techniques. En fait, vous verrez que certains détails sont de vrais défis (par exemple, faire correspondre vos nom internes aux • vrais• noms requière une bonne connaissance de sendmail). Une description générale du courrier et des forums sort du cadre de ce guide. Reportez-vous à un bon livre d'administration Unix (certains très bons ouvrages sont listés sur le site de NetBSD). Le problème est rendu très complexe par la multitude de configurations et de connexions possibles et parce qu'il ne suffit pas configurer un simple programme: il faut que les différents composants coopèrent correctement.

Par ailleurs, ce chapitre décrit brièvement la configuration (mais pas l'utilisation) de deux applications populaires, mutt pour le courriel et tin pour les forums de discussion. L'utilisation n'est pas abordée car ils sont très faciles à utiliser et bien documentés. Bien entendu, ni mutt ni tin ne sont obligatoires: de nombreuses autres applications de ce type existent mais je pense que celles-ci sont un bon point de départ car elles sont largement utilisées, simples, fonctionnent bien et utilisent peu de place sur le disque et en mémoire. Ce sont des applications en mode console, si vous préférez les programmes graphiques, de nombreux choix existent pour X.

En bref, les programmes requis pour la configuration décrite dans ce chapitre sont:

- sendmail
- fetchmail
- m4
- mutt
- tin

Seuls sendmail et m4 sont installés avec le système de base. Il vous faut intaller les autres à l'aide de la collection de paquetages.

Avant de continuer, rappelons qu'aucun de ces programmes ne sont obligatoires: d'autres applications effectuent les mêmes tâches et de nombreux utilisateurs les préfèrent. Vous trouverez d'autres options dans les listes de diffusion. Vous pouvez aussi imaginer d'autres stratégies pour envoyer et recevoir le courrier: celle expliquée ici n'est qu'un point de départ. Une fois que vous aurez compris le fonctionnement, vous adapterez certainement les paramètrages à vos besoins voire adopterez une méthode différente.

A l'autre extrême, vous trouverez des applications telles que Netscape Communicator qui font tout et vous épargnent la configuration de nombreux composants: avec communicator, vous pouvez naviguer sur internet, envoyer et recevoir du courrier et lire les forums de discussion. Pourtant le paramètrage est très simple. Le prix à payer est que Communicator est un programme "fermé" qui ne coopère avec aucun utilitaire standard d'Unix.

Une autre possibilité est d'utiliser emacs pour le courrier et les forums. Il n'a pas besoin d'être présenté aux utilisateurs Unix mais, au cas où il vous serait inconnu, il s'agit d'un éditeur extensible (pourtant, l'appeler un éditeur a quelque chose de réducteur) qui devient un environnement de travail complet utilisable pour le courrier et les forums mais aussi pour effectuer de nombreuses opérations. Pour de nombreuses personnes, emacs est le seul environnement dont elles ont besoin pour travailler. Sa configuration pour le courrier et les forums est décrite dans son manuel.

Dans le reste de ce chapitre, nous considérerons que nous avons un ordinateur connecté à internet par l'intermédiaire d'un fournisseur d'accès avec PPP et un modem.

- Le nom de l'hôte local (de notre ordinateur) est • ape• et le réseau interne est "insetti.net". Le FQDN (Fully Qualified Domain Name: nom internet complet) est donc "ape.insetti.net".
- Le nom d'utilisateur sur "ape" est • carlo• .
- Le fournisseur d'accès est BigNet.
- Le serveur de courrier électronique du fournisseur est "mail.bignet.it".
- Le serveur de forums de discussions est "news.bignet.it".
- Le compte de "carlo" chez le fournisseur d'accès est • alan• avec le mot de passe • pZY9o• .

Tout d'abord, un point sur la terminologie:

MUA (mail user agent): client de courrier électronique.

> tout programme qui permet de lire et écrire du courrier électronique (courriel). Par exemple: mutt, elm, pine, mais aussi le tout simple mailinstallé avec le système de base. En français, on trouvera aussi le mot • mailleur• (pour • mailer• en anglais).

MTA (mail transfer agent): agent de transfert de courrier.

> un programme qui transfert le courriel entre deux ordinateurs mais aussi localement (sur le même ordinateur). Le MTA décide du chemin qu'emprunte un courrier pour arriver à destination. Sur les systèmes BSD (mais pas seulement), le MTA standard est sendmail.

MDA (mail delivery agent): agent de livraison de courrier.

> un programme, habituellement utilisé avec le MTA, qui livre le courriel, c'est à dire qu'il met physiquement les messages dans la boite aux lettres du destinataire. Par exemple, sendmail utilise un ou plusieurs MDA pour livrer le courriel.

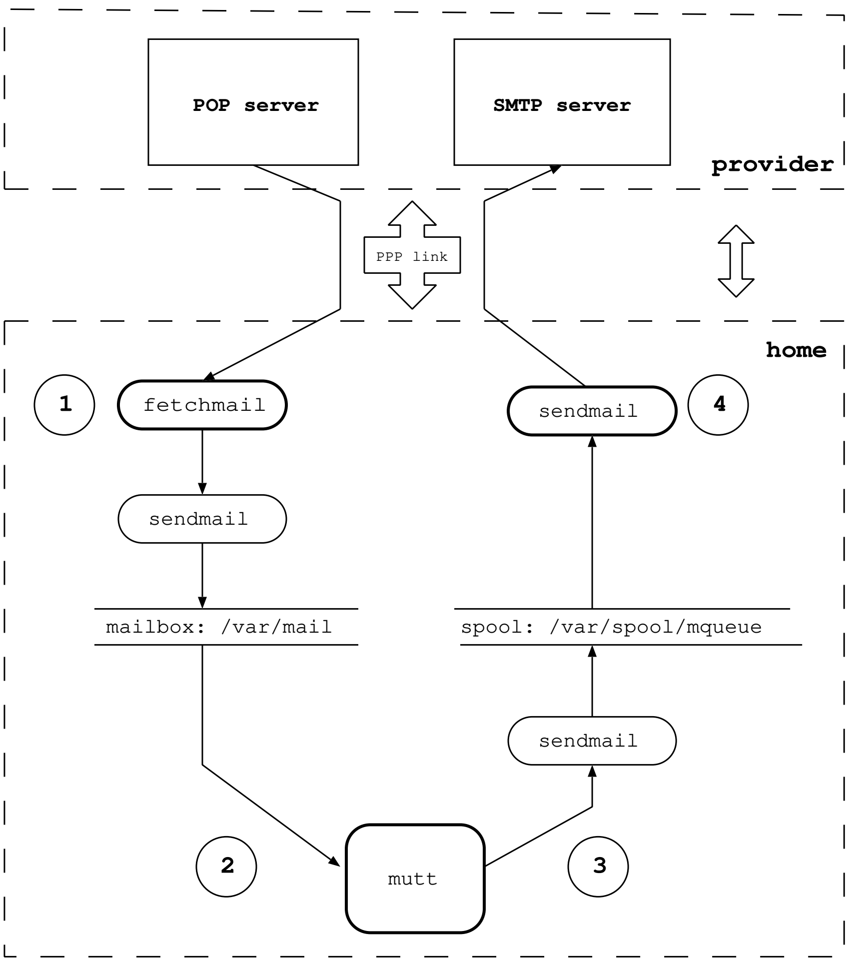
La Figure 11-1 montre le système de courrier électronique que nous allons mettre en place. Entre le réseau local (ou l'ordinateur seul) et le fournisseur d'accès, la connexion se fait par PPP avec un modem. Les • bulles• à bord épais sont les programmes lancés manuellement par l'utilisateur. Les autres bulles sont les programmes lancés automatiquement. Les nombres entourés font référence aux étapes logiques du cycle du courriel.

1. A l'étape (1), le courriel est téléchargé depuis le serveur POP (Post Office Protocol: Protocole de La Poste) du fournisseur en utilisant fetchmail qui, lui-même, utilise sendmail pour mettre le message dans la boite aux lettres de l'utilisateur.
2. A l'étape (2), l'utilisateur lance mutt (ou un autre mailleur) pour lire le courriel, répondre et écrire de nouveaux messages.

3. A l’étape (3), l’utilisateur "envoie" le courriel depuis mutt. Les messages sont accumulés dans la zone de stockage (spool).
4. A l’étape (4), l’utilisateur appelle sendmail pour transférer les messages vers le serveur SMTP (Simple Mail Transfert Protocol: Protocole de Transfert de Courriel) du fournisseur qui se chargera de l’acheminer vers le destinataire (éventuellement en passant par d’autres serveurs de courriel). Le serveur SMTP du fournisseur joue le rôle d’un *relay* pour notre courriel.

La connexion avec le fournisseur d’accès ne se fait qu’aux étapes (1) et (4). Les autres étapes ne demandent pas d’être en ligne.

Figure 11-1. Structure du système de courrier électronique



11.1. sendmail

Lorsque le MTA, sendmail dans la configuration par défaut, veut livrer un message local, il le fait directement. Si le message est à destination d’un domaine différent, il doit retrouver l’adresse du serveur de courriel de ce domaine. Sendmail utilise alors les services du DNS (décrit dans Chapitre 10) puis distribue le courrier au serveur trouvé.

Le plus utilisé des *MTA* dans le monde BSD est probablement sendmail. Il est contrôlé par un ensemble de fichiers et de bases de données dont `/etc/sendmail.cf` est le plus important. En général, si vous n'êtes pas expert, il est préférable de ne pas modifier directement le fichier `/etc/sendmail.cf` mais d'utiliser les macros prédéfinies ainsi que le pré-processeur m4 qui simplifient largement la configuration.

> **Note :** avant la version 1.5 de NetBSD, les fichiers de configuration du courrier électronique se trouvaient dans `/etc` et non dans `/etc/mail`.

Même en utilisant les macros, la configuration de sendmail n'est pas des plus faciles donc la section suivante se bornera à décrire un exemple qui pourra être modifié ou adapté aux besoins que vous éprouverez.

Le premier problème que nous rencontrons est que notre réseau est un réseau interne, c'est à dire pas directement accessible depuis internet. Cela veut dire que les noms utilisés en interne n'ont aucune signification sur internet. En bref, "ape.insetti.net" ne peut pas être atteint par un hôte externe: personne ne pourra répondre à un courriel envoyé avec cette adresse de réponse (certains systèmes vont même rejeter ce messages venant d'un hôte inconnu). La vraie adresse, celle visible par n'importe qui, est assignée par le fournisseur d'accès et il est donc nécessaire de convertir "carlo@ape.insetti.net" en "alan@bignet.it". Sendmail, s'il est correctement configuré, en tiendra compte lors du transfert des messages.

Le second problème est que sendmail doit savoir qu'il doit envoyer le courriel au serveur du fournisseur d'accès. Dans le cas de la configuration de notre chapître, sendmail ne contacte pas directement le serveur de courrier du destinataire (comme c'était décrit plus tôt) mais fait transiter tous les messages vers le serveur de courrier du fournisseur d'accès.

> **Note :** le serveur de courrier du fournisseur d'accès joue le rôle d'un *relais*, c'est à dire qu'il livre tout courriel non destiné au domaine local à un autre serveur. C'est un intermédiaire entre deux serveurs.

Comme la connexion avec le fournisseur n'est pas toujours établie, il n'est pas nécessaire de lancer le démon depuis `/etc/rc.conf`: utilisez la ligne "sendmail=NO". En conséquence, il faudra le lancer manuellement lorsque l'on voudra transférer le courriel chez le fournisseur d'accès. Le courriel local sera livré correctement même si sendmail n'est pas activé comme démon.

Il est temps de se consacrer au fichier de configuration de sendmail. Tout d'abord, changez de répertoire:

```
# cd /usr/share/sendmail/m4
```

11.1.1. Configuration avec genericstable

Ce type de configuration s'effectue dans le fichier (`/etc/mail/genericstable`) qui contient les règles que sendmail devra suivre pour remplacer les noms internes. Le nouveau fichier de configuration de sendmail, que nous appellerons `macf.mc` contient:

```
divert(-1)
include(`../m4/cf.m4')dnl
VERSIONID(`macf.mc (data) created by carlo@ape.insetti.net')dnl
```

```
OSTYPE(bsd4.4)dnl

GENERICS_DOMAIN(ape.insetti.net)dnl
FEATURE(genericstable)dnl
FEATURE(masquerade_envelope)dnl

define(‘SMART_HOST’,‘mail.bignet.it’)dnl

FEATURE(redirect)dnl
FEATURE(nocanonify)dnl

define(‘SMTP_MAILER_FLAGS’,‘e9’)dnl
define(‘confCON_EXPENSIVE’,‘True’)dnl

# ‘for european users”
define(‘confDEF_CHAR_SET’,‘ISO-8859-1’)dnl

MAILER(local)dnl
MAILER(smtp)dnl
```

Cette configuration indique à sendmail de remplacer les adresses de type "ape.insetti.net" en utilisant les vrais noms trouvés dans le fichier `/etc/mail/genericstable`. Il indique aussi que le courriel doit être envoyé au serveur "mail.bignet.it". La signification des options est décrite en détails dans le fichier `/usr/share/sendmail/README`.

Bien entendu, le fichier `/etc/mail/genericstable` doit être créé. Par exemple:

```
carlo:          alan@bignet.it
root:           alan@bignet.it
news:           alan@bignet.it
```

Pour des raisons d’efficacité, le fichier `genericstable` doit être transformé avec la commande:

```
# /usr/sbin/sendmail -bi -oA/etc/mail/genericstable
```

Un autre fichier important est `/etc/mail/aliases` qui peut rester dans sa configuration par défaut bien qu’il faille tout de même exécuter la commande:

```
# newaliases
```

Enfin, le nouveau fichier de configuration doit être généré après avoir sauvegardé la précédente version:

```
# cp /etc/mail/sendmail.cf /etc/mail/sendmail.cf.bak
# m4 mycf.mc > /etc/mail/sendmail.cf
```

> **Note :** dans le répertoire `/usr/share/sendmail/cf`, existe un fichier nommé `netbsd-proto.mc` qui est utilisé pour créer les paramètres par défaut de `/etc/mail/sendmail.cf`. Il peut, au besoin, être reconstruit avec la commande **make**.

Tout est maintenant prêt pour envoyer du courrier.

11.1.2. Tester la configuration

Sendmail est enfin configuré et prêt à fonctionner mais avant de commencer à envoyer du courrier, mieux vaut effectuer quelques tests. Tout d'abord, commençons par envoyer un message localement avec la commande :

```
$ sendmail -v carlo
Subject: test

Preuve
.
carlo... Connecting to local...
carlo... Sent
```

Suivez exactement l'exemple ci-dessus : laissez une ligne blanche après le sujet et finissez le message avec une ligne ne contenant rien d'autre qu'un point. Normalement, vous devriez pouvoir lire ce message dans votre boite aux lettres et vérifier que le champ From: est correctement défini.

```
From: alan@bignet.it
```

Vous pouvez directement vérifier les règles d'adressage en demandant le mode de test d'adresses à sendmail, avec l'option `-bt`. Ce mode montre les analyses effectuées sur une adresse ainsi que sa ré-écriture en fonction des règles édictées dans le fichier de configuration.

```
$ sendmail -bt
ADDRESS TEST MODE (ruleset 3 NOT automatically invoked)
Enter <ruleset> <address>
>
```

Tapez les commandes suivantes :

```
/tryflags ES
/try smtp carlo@ape.insetti.net
```

Vous obtenez quelque chose comme :

```
Trying envelope sender address carlo@ape.insetti.net for mailer smtp
rewrite: ruleset   3   input: carlo @ ape . insetti . net
rewrite: ruleset  96   input: carlo < @ ape . insetti . net >
rewrite: ruleset  96 returns: carlo < @ ape . insetti . net . >
rewrite: ruleset   3 returns: carlo < @ ape . insetti . net . >
rewrite: ruleset   1   input: carlo < @ ape . insetti . net . >
rewrite: ruleset   1 returns: carlo < @ ape . insetti . net . >
rewrite: ruleset  11   input: carlo < @ ape . insetti . net . >
rewrite: ruleset  51   input: carlo < @ ape . insetti . net . >
rewrite: ruleset  51 returns: carlo < @ ape . insetti . net . >
rewrite: ruleset  61   input: carlo < @ ape . insetti . net . >
rewrite: ruleset  61 returns: carlo < @ ape . insetti . net . >
rewrite: ruleset  94   input: carlo < @ ape . insetti . net . >
rewrite: ruleset  93   input: carlo < @ ape . insetti . net . >
rewrite: ruleset   3   input: alan @ bignet . it
```

```
rewrite: ruleset  96   input: alan < @ bignet . it >
rewrite: ruleset  96 returns: alan < @ bignet . it >
rewrite: ruleset   3 returns: alan < @ bignet . it >
rewrite: ruleset  93 returns: alan < @ bignet . it >
rewrite: ruleset  94 returns: alan < @ bignet . it >
rewrite: ruleset  11 returns: alan < @ bignet . it >
rewrite: ruleset   4   input: alan < @ bignet . it >
rewrite: ruleset   4 returns: alan @ bignet . it
Rcode = 0, addr = alan@bignet.it
>
```

Comme vous pouvez le voir, l'adresse locale a été ré-écrite dans la vraie adresse qui apparait dans votre courriel.

Vous obtenez le même résultat avec la commande :

```
/try smtp carlo
```

11.1.3. Utilisation d'un autre MTA

A partir de la version 1.4 de NetBSD, sendmail n'est plus appelé directement :

```
$ ls -l /usr/sbin/sendmail
lrwxr-xr-x  1 root  wheel  21 Nov  1 01:14 /usr/sbin/sendmail@ -> /usr/sbin/mailwrapper
```

La raison d'être de mailwrapper est de permettre l'utilisation d'un MTA différent de sendmail. Si vous êtes dans ce cas, je vous conseille vivement la lecture des pages de manuel de mailwrapper(8) et mailer.conf(5) qui sont très claires.

11.2. fetchmail

Le courrier électronique (courriel) reçu chez le fournisseur d'accès n'est pas automatiquent transféré sur les ordinateurs locaux. Il faut les télécharger. Fetchmail est un programme très populaire qui va chercher le courrier sur un serveur distant et le transmet sur le système local pour préparer la distribution (généralement à l'aide de sendmail). Il est puissant mais facile à configurer et utiliser: après l'installation, il suffit que le fichier `~/.fetchmailrc` soit créé pour que l'application soit prête à fonctionner (`~/.fetchmailrc` contient un mot de passe, il doit donc avoir les permissions appropriées).

Voici un exemple de `.fetchmailrc`:

```
poll mail.bignet.it
protocol POP3
username alan there with password pZY9o is carlo here
flush
mda "/usr/sbin/sendmail -oem %T"
```

> **Note :** la dernière ligne ("mda...") n'est utilisée que si sendmail n'est pas utilisé comme démon sur le système. Veuillez noter que le serveur de courriel indiqué dans ce fichier ("mail.bignet.it") n'est pas forcément le même que le relais de courrier utilisé par sendmail.

La commande suivante est à utiliser pour télécharger et livrer le courrier sur le système local:

```
$ fetchmail
```

Les messages peuvent alors être lus avec mutt.

11.3. Lire et écrire du courrier avec mutt

Mutt est l'un des mailleurs les plus populaires: il est "léger", facile à utiliser et possède de nombreuses possibilités. Sa page de manuel est très épurée, la vraie documentation se trouve dans `/usr/pkg/share/doc/mutt/`, en particulier `manual.txt`.

La configuration de mutt est définie par le fichier `~/.muttrc`. Le plus simple est de faire une copie du fichier d'exemple (`/usr/pkg/etc/Muttrc`) dans le répertoire personnel et de le modifier. L'exemple suivant montre comment:

- Garder une copie du courrier envoyé.
- Définir un répertoire et deux fichiers pour le courrier arrivé et le courrier envoyé sauvegardé par mutt. Dans notre exemple, le répertoire est `~/Mail` et les deux fichiers sont respectivement `incoming` et `outgoing`).
- Définir quelques couleurs.
- Définir un alias.

```
set copy=yes
set edit_headers
set folder="~/Mail"
unset force_name
set mbox="~/Mail/incoming"
set record="~/Mail/outgoing"
unset save_name

bind pager <up> previous-page
bind pager <down> next-page

color normal white black
color hdrdefault blue black
color indicator white blue
color markers red black
color quoted cyan black
color status white blue
color error red white
color underline yellow black
```

```
mono quoted standout
mono hdrdefault underline
mono indicator underline
mono status bold

alias pippo Pippo Verdi <pippo.verdi@pluto.net>
```

Pour lancer mutt:

```
$ mutt
```

> **Note :** Mutt connait les couleurs mais elles dépendent du type de terminal. Sous X, vous pouvez utiliser les couleurs xterm. Par exemple:
>
> ```
> TERM=xterm-color mutt
> ```

11.4. Stratégie pour la réception de courrier

Cette section décrit une méthode simple pour recevoir et lire le courriel. La connexion au fournisseur d'accès n'est établie que le temps nécessaire au téléchargement des messages. La lecture se fait hors ligne.

1. Activez la connexion au fournisseur d'accès.
2. Lancez **fetchmail**.
3. Désactivez la connexion.
4. Lisez le courriel avec mutt.

11.5. Stratégie pour l'envoi du courrier

Lorsque le courriel est écrit puis "envoyé" avec mutt, les messages doivent être transférés chez le fournisseur d'accès avec sendmail. Le courriel est envoyé par mutt avec la commande **y** mais celui-ci ne part pas vraiment. Il est ajouté à la queue dans la zone de stockage (spool). Si sendmail n'est pas actif comme démon, il faut le lancer à la main sinon les messages resteront sur le disque dur. Les étapes sont:

1. Ecrire le courrier avec mutt, l'envoyer et quitter mutt.
2. Etablir la connexion au fournisseur d'accès.
3. Taper la commande **sendmail -q -v** afin de transférer le courriel chez le fournisseur.
4. Désactiver la connexion.

11.6. Outils de courrier électonique avancés

Lorsque vous commencez à utiliser le courrier électronique, vos besoins ne sont probablement pas très sophistiqués et la configuration standard doit vous satisfaire. Mais pour beaucoup d'utilisateurs, le nombre de messages quotidiens ne cesse d'augmenter et une organisation plus rationnelle du stockage du courriel devient nécessaire. Par exemple, trier le courriel dans différentes boites en fonction du thème. Si, par exemple, vous vous inscrivez à une liste de diffusion, vous recevrez de nombreux messages que vous voudrez séparer du courrier habituel. Vous découvrirez que vous perdez beaucoup de temps à chaque jour répéter les mêmes opérations manuelles pour organiser vos boites aux lettres.

Pourquoi répéter les même opérations manuellement alors que des programmes peuvent le faire pour vous ? Il existe de nombreux outils que vous pouvez ajouter à votre messagerie pour améliorer sa flexibilité et pour effectuer des traitements automatiques. Parmis les plus connus et utilisés, nous trouvons:

- procmail, un agent de livraison avancé et de filtrages divers pour le courriel local. Il peut automatiquement traiter le courrier reçu en utilisant des filtres utilisateur. Il s'intègre très bien à sendmail.
- metamail, un outils pour gérer les fichiers attachés.
- formail, pour mettre en page du courriel.

Le reste de cette section présente un extrait de configuration de procmail qui s'applique à un cas courant: distribuer automatiquement le courrier issu d'une liste de diffusion dans une boite aux lettres définie. La configuration de sendmail doit être modifiée pour appeler directement procmail (il sera le *local mailer* utilisé par sendmail) et un fichier de configuration de procmail doit être créé.

Tout d'abord, procmail doit être installé avec le système de paquetages (`mail/procmail`).

Ensuite, la configuration de sendmail doit être changée afin d'utiliser procmail comme client de courrier local. Le fichier de configuration doit être recréé, comme décrit précédament, après avoir ajouté trois lignes à `macf.mc`:

```
define(`PROCMAIL_MAILER_PATH', /usr/pkg/bin/procmail)dnl
FEATURE(local_procmail)dnl
MAILER(procmail)dnl
```

La première ligne indique le chemin d'accès au programme procmail (vous pouvez voir où il a été installé avec la commande **which procmail**). La seconde ligne demande à sendmail d'utiliser procmail comme distributeur de courrier et la troisième ajoute procmail à la liste des mailleurs établie par sendmail.

La dernière étape consiste à établir le fichier de configuration de procmail contenant les instructions.

Disons, par exemple, que nous soyons inscrits à une liste de diffusion sur les roses dont l'adresse est `<roses@flowers.org>` et que chaque message contienne, dans son en-tête, la ligne:

```
Delivered-To: roses@flowers.org
```

Le fichier de configuration de procmail (`.procmailrc`) pourra alors ressembler à:

```
PATH=/bin:/usr/bin:/usr/pkg/bin
MAILDIR=$HOME/mail
LOGFILE=$MAILDIR/from
```

```
:0
* ^Delivered-To: roses@flowers.org
roses_list
```

Ce fichier ne contient qu'une seule condition, commençant avec la ligne ":0". La ligne suivante demande d'identifier tous les messages contenant "Delivered-To: roses@flowers.org" et la dernière ligne indique que les messages correspondants doivent être rangés dans la boite aux lettres "`roses_list`" (qui doit déjà exister dans $MAILDIR). Les autres messages seront distribués dans la boite aux lettres par défaut.

Bien sûr, la liste de diffusion n'est qu'un exemple. Procmail est polyvalent et peut être utilisé pour filtrer le courriel à partir de nombreux critères. Comme d'habitude, pour plus de détails, reportez-vous aux pages de manuel procmail(1), procmailrc(5) et procmailex(5) (cette dernière contient de nombreux exemples de fichiers de configuration).

11.7. Forums de discussion avec tin

Les mots *forum de discussion* indiquent un ensemble d'articles de USENET, un service d'internet (NDT: on trouve d'autres noms comme "groupes de discussion", • nouvelles• , "groupes de nouvelles", • news• , • newsgroup• ...). Chaque forum contient des articles sur un thème spécifique. La lecture d'un forum est différente de la lecture d'une liste de diffusion: lorsque vous vous inscrivez sur une liste de diffusion, vous recevez les articles comme du courriel normal que vous pouvez lire avec des programmes standards comme mutt et auxquels vous pouvez répondre. Les forums, en revanche, sont lisibles directement sur un serveur avec un programme dédié appelé un *lecteur de forums* (newsreader) comme, par exemple, tin. Avec tin, vous pouvez souscrire aux forums de votre choix et suivre le *fil* des discussions.

> **fil :** un fil (thread en anglais) est une suite d'articles dérivés d'un article que nous pouvons appeler • original• . En bref, un message est envoyé au forum, quelqu'un répond puis d'autre personnes répondent aux personnes qui ont répondu et ainsi de suite. Il se crée alors une structure de messages et de réponses sous forme d'arborescence: sans lecteur de forums il est impossible de remettre en place la suite des articles.

Après l'installation de tin (à partir de la collection de paquetages, comme d'habitude), la seule chose qui reste à faire est d'inscrire le nom du serveur NNTP dans le fichier `/usr/pkg/etc/nntp/server`. Par exemple:

```
news.bignet.it
```

Une fois cela fait, le programme peut être lancé avec la commande **rtin**. Sur l'écran, s'affiche quelque chose du genre:

```
$ rtin
Connecting to news.bignet.it...
news.bignet.it InterNetNews NNRP server INN 1.7.2 08-Dec-1997 ready (posting ok).
Reading groups from active file...
Checking for new groups...
Reading attributes file...
Reading newsgroups file...
Creating newsrc file...
```

```
Autosubscribing groups...
Reading newsrc file...
```

Soyez patient lorsque vous vous connectez pour la première fois car tin télécharge une immense liste de forums auxquels vous pouvez souscrire et ceci peut prendre plusieurs minutes. Lorsque que le téléchargement est terminé, l'écran principal du programme s'affiche. Normalement, aucun forum n'apparait. Pour voir la liste, appuyez sur **y**. Pour souscrire à un forum, déplacez-vous jusqu'à son nom et appuyez sur **y**.

Une fois que vous serez inscrit à quelques forums, vous pourrez démarrer plus rapidement en tapant la commande **rtin -Q**. La recherche de la liste des forums sera évitée (`-q`), seuls les forums souscrits seront recherchés (`-n`) et la description des groupe ne sera pas téléchargée : il ne sera plus possible d'utiliser la commande **y** (yank) dans tin. Losqu'il est démarré avec ces options, tin ne peut pas dire si un forum est modéré ou non.

> **Note :** si vous vous connectez depuis un réseau interne (comme dans notre exemple), lorsque vous enverrez un message, son adresse de réponse sera fausse car elle correspond à une adresse interne. Pour résoudre ce problème, utilisez l'option • mail_adress• dans le fichier de configuration de tin (`~/.tin/tinrc`) ou donnez une valeur à la variable d'environnement REPLYTO.

Chapitre 12. Pilotes de console

Dans les versions de NetBSD précédant la 1.4, l'utilisateur pouvait choisir entre deux pilotes différents pour l'écran et le clavier: pccons (spécifique à i386) et pcvt. Avec la version 1.4, est apparu le pilote multi-plateforme wscons qui se substitue aux deux précédents (qui sont toujours supportés).

12.1. wscons

Wscons est le nouveau pilote de console de NetBSD. Il permet les terminaux virtuels, supporte les claviers internationaux, gère la souris... Les capacités de wscons varient suivant le portage (il n'est pas disponible pour tous): la version pour i386 est particulièrement riche en fonctions.

A la fin de l'installation, wscons est actif par défaut et n'a besoin de rien d'autre pour fonctionner. En version 1.4, les consoles virtuelles sont activées par défaut: voyez le Chapitre 4. Le reste de cette section décrit les options de wscons du fichier de configuration du noyau.

Si vous compilez un noyau personnel, pour disposer de wscons, vous devez activer les options adéquates et commenter les options concernant pcvt et pccons (ils ne peuvent pas tourner en même temps). Par exemple:

```
#pc0    at isa? port 0x60 irq 1   # pccons generic PC console driver
#vt0    at isa? port 0x60 irq 1   # PCVT console driver
```

Dans le fichier de configuration du noyau, vous pouvez aussi paramètrer un clavier étranger. Par exemple, pour utiliser un clavier italien par défaut:

```
options     PCKBD_LAYOUT="KB_IT"
```

Note : le clavier italien n'est pas idéal pour la programmation. Pour le modifier, voyez le Chapitre 4.

Le nombre pré-alloué de consoles virtuelles est contrôlé par l'option suivante:

```
options     WSDISPLAY_DEFAULTSCREENS=4
```

D'autres consoles peuvent être ajoutées en activant les bonnes lignes du fichier `/etc/wscons.conf`: le signe de commentaires (#) doit être enlevé des lignes commençant par "screen x". Dans l'exemple suivant, une cinquième console est ajoutée aux quatre précédentes:

```
# screens to create
#       idx     screen  emul
#screen 0       -       vt100
screen 1        -       vt100
screen 2        -       vt100
screen 3        -       vt100
screen  4       -       -
#screen 4       80x25bf vt100
#screen 5       80x50   vt100
```

Le script `rc.wscons` transforme chaque ligne non commentée en appels à la commande **wsconscfg**: chaque colonne devient un paramètre. La colonne *idx* correspond au paramètre `index`, *screen* correspond à `-t type` (qui définit le type d'écran: lignes, colonnes, couleurs...) et *emul* à `-e emul` (qui définit l'émulation). Par exemple:

```
screen 3        -       vt100
```

est transformé en cet appel:

```
wsconscfg -e vt100 3
```

> **Note :** il est possible qu'apparaissent des conflits entre les consoles pré-définies par le noyau et celles définies au démarrage par `/etc/wscons.conf`. Si pendant le démarrage, le système essaie d'allouer un écran déja défini, le message suivant sera affiché:
>
> ```
> wsconscfg: WSDISPLAYIO_ADDSCREEN: Device busy
> ```
>
> La solution est de commenter les lignes incriminées dans `/etc/wscons.conf`.

Les consoles virtuelles doivent aussi être actives dans `/etc/ttys`. Par exemple:

```
console "/usr/libexec/getty Pc"         pc3     off secure
ttyE0   "/usr/libexec/getty Pc"         vt220   on secure
ttyE1   "/usr/libexec/getty Pc"         vt220   on secure
ttyE2   "/usr/libexec/getty Pc"         vt220   on secure
ttyE3   "/usr/libexec/getty Pc"         vt220   off secure
...
```

La ligne

```
ttyE3   "/usr/libexec/getty Pc"         vt220   off secure
```

de `/etc/ttys` est utilisée par le serveur X pour trouver un terminal libre. Pour utiliser un écran différent du numéro 4, un paramètre sous la forme vt*n* doit être passé au serveur (*n* est le numéro de la touche de fonction utilisée pour activer l'écran de X).

Par exemple, "screen 7" doit être activé dans `/etc/wscons.conf` et X démarré avec • vt8• . Si vous utilisez xdm, vous devez modifier `/usr/X11R6/lib/X11/xdm/Xserver`. Par exemple:

```
:0 local /usr/X11R6/bin/X +kb dpms -bpp 16 dpms vt8
```

Pour xdm3d, le chemin est différent: `/usr/X11R6/share/xdm3d/Xservers`.

12.1.1. Mode texte de 50 lignes avec wscons

Un mode texte de 50 lignes peut être utilisé depuis la version 1.4.1 de NetBSD. Il est activé dans `/etc/wscons.conf`. La ligne suivante doit être décommentée:

```
font ibm  -  8  ibm  /usr/share/pcvt/fonts/vt220l.808
```

Les lignes suivantes doivent être modifiées:

```
#screen 0       80x50   vt100
screen  1       80x50   vt100
screen  2       80x50   vt100
screen  3       80x50   vt100
screen  4       80x50   vt100
screen  5       80x50   vt100
screen  6       80x50   vt100
screen  7       80x50   vt100
```

Cette configuration permet d'utiliser huit écrans, accessibles avec la combinaison de touches Ctrl-Alt-F*n* (où *n* varie de 1 à 8). Les pilotes correspondants sont ttyE0 à ttyE7. Pour y accéder et obtenir l'invite de connexion, modifiez `/etc/ttys`:

```
ttyE0   "/usr/libexec/getty Pc"         vt220   on secure
ttyE1   "/usr/libexec/getty Pc"         vt220   on secure
ttyE2   "/usr/libexec/getty Pc"         vt220   on secure
ttyE3   "/usr/libexec/getty Pc"         vt220   on secure
ttyE4   "/usr/libexec/getty Pc"         vt220   on secure
ttyE5   "/usr/libexec/getty Pc"         vt220   on secure
ttyE6   "/usr/libexec/getty Pc"         vt220   on secure
ttyE7   "/usr/libexec/getty Pc"         vt220   on secure
```

Il n'est pas possible de modifier la résolution • 80x25• de l'écran 0, sans doutes pour garantir que, même en cas de problème, il y ait un écran qui fonctionne toujours.

12.1.2. wsmouse

12.2. pccons

C'est le pilote de console qui se trouve sur la disquette d'installation i386. Il ne permet pas les consoles virtuelles et les programmes utilitaires mais prend très peu de place.

12.3. pcvt

Pcvt est un émulateur de terminal VT220 et possède des fonctions plus avancées que celles du simple pccons. Il supporte les claviers étrangers et les consoles virtuelles (avec Ctrl-Alt-F1 à F8 ou avec les touches de fonction F9 à F12). Pour activer pcvt, les lignes suivantes doivent être décommentées du fichier de configuration du noyau:

```
# Enable only one of the following lines
#pc0    at isa? port 0x60 irq 1
vt0     at isa? port 0x60 irq 1

# Options for PCVT console driver
#options FAT_CURSOR
```

```
options PCVT_NETBSD=132
options PCVT_NSCREENS=3
```

Pour utiliser un clavier étranger, vous devez l'activer au démarrage. Il faut aussi choisir le terminal correct. Par exemple:

```
/usr/local/bin/kcon -m i2
TERM=pcvt25;    export TERM
```

`/etc/ttys` doit être modifié en fonction. Par exemple:

```
#console "/usr/libexec/getty Pc"   pcvt25  on secure
ttyv0    "/usr/libexec/getty Pc"   pcvt25  on secure
```

Clavier italien Pcvt : la définition du clavier *i2* est incorrecte, le fichier `/sys/arch/i386/isa/pcvt/Util/keycap/keycap.src` doit donc être modifié. Voici une modification qui a été testée avec NetBSD 1.3.3:

```
i2|italy142|Italian 142 mapping:\
     :A8={:A9=[:A10=]:A11=}:\
     :A12=`:A13=~:\
     :A17=@:A18=#:\
     :tc=italy141:
```

Les paramètrages d'un clavier étranger (ici le clavier italien), doivent être chargés au démarrage, par exemple dans `/etc/rc.local`:

```
KCONP=/usr/local/bin
SCONP=/usr/local/bin
LDFNP=/usr/local/bin
ISPCP=/usr/sbin
CURSP=/usr/local/bin

set_keybd=YES

#-------------------------------------------------------
# En cas, mise en place du clavier italien
#-------------------------------------------------------

if [ X${set_keybd} = X"YES" -a -x $KCONP/kcon ]
then
  echo
  echo 'passage en clavier italien'
  $KCONP/kcon -m i2
fi

echo '.'
```

`/etc/ttys` doit aussi être modifié:

```
#console  "/usr/libexec/getty Pc" pcvt25  on secure
```

```
ttyv0   "/usr/libexec/getty Pc" pcvt25  on secure
ttyv1   "/usr/libexec/getty Pc" pcvt25  on secure
ttyv2   "/usr/libexec/getty Pc" pcvt25  on secure
```

L'utilitaire pcvt doit être compilé et installé:

```
cd /sys/arch/i386/isa/pcvt/Util
make
make install
```

12.3.1. Changer la taille écran

Avec pcvt, vous pouvez changer le nombre de lignes et de colonnes à l'écran. L'exemple de script qui suit peut être utilisé pour passer automatiquement d'une résolution à une autre:

```
#!/bin/sh
# Mettre l'écran à # lignes
case $1 in
  25)
    /usr/local/bin/scon -s 25
    /usr/local/bin/cursor -s13 -e14
    ;;
  28)
    /usr/local/bin/loadfont -c1 -f
    /usr/share/misc/pcvtfonts/vt220l.814
    /usr/local/bin/loadfont -c2 -f
    /usr/share/misc/pcvtfonts/vt220h.814
    /usr/local/bin/scon -s 28
    /usr/local/bin/cursor -s12 -e14
    ;;
  40)
    /usr/local/bin/loadfont -c3 -f
    /usr/share/misc/pcvtfonts/vt220l.810
    /usr/local/bin/loadfont -c4 -f
    /usr/share/misc/pcvtfonts/vt220h.810
    /usr/local/bin/scon -s 40
    /usr/local/bin/cursor -s8 -e10
    ;;
  50)
    /usr/local/bin/loadfont -c5 -i
    /usr/share/misc/pcvtfonts/vt220l.808
    /usr/local/bin/loadfont -c6 -i
    /usr/share/misc/pcvtfonts/vt220h.808
    /usr/local/bin/scon -s 50
    /usr/local/bin/cursor -s6 -e8
    ;;
  *)
    echo "Nombre de lignes invalide (25/28/40/50)"
    ;;
esac
```

Chapitre 13. Edition

Ce chapitre décrit l'éditeur standard de NetBSD, j'ai nommé vi. La première partie présente les bases d'utilisation et les parties suivantes décrivent quelques caractéristiques avancées, avec des références particulières à la version installée sur NetBSD.

13.1. Introduction à vi

Pas besoin de présenter vi : il a été développé par Bill Joy, de l'université Berkeley de Californie, pour devenir l'éditeur standard d'à peu près toutes les versions d'Unix, le compagnon et l'ami de confiance des administrateurs système et des programmeurs mais la bête noire de l'utilisateur débutant.

Il vaut toutefois la peine de placer quelques mots sur son utilisation car il est aussi le seul éditeur présent sur votre machine à la fin de l'installation de NetBSD (à part ed). Il est, de fait, le seul dont vous ayez besoin pour effectuer la configuration du système. Par la suite, naturellement, vous serez libre d'installer l'éditeur que vous préférez ou que vous connaissez le mieux, le choix étant assez nourri dans la collection de paquetages de NetBSD (voir le Chapitre 8).

Cette section est une introduction à l'utilisation de vi mais qui devrait être suffisante au débutant pour créer ou modifier des fichiers texte. Seules les caractéristiques de base et les commandes les plus courantes seront décrites, sans prétention d'exhaustivité ou de précision : pour plus de détails, reportez-vous à la Section 13.2.2.

Le débutant ne doit pas commettre l'erreur de juger vi à partir de cette introduction : utilisé de façon • minimaliste• , il existe certainement des outils plus pratiques. Ses vertus se découvrent lorsque l'on connaît toutes ses commandes (!) et que l'on commence à utiliser ses fonctions avancées. Alors, vi n'est plus l'éditeur de base mais un instrument confortable, puissant et fiable. Ce guide a été écrit à l'aide de l'éditeur vi.

13.1.1. Premières étapes

Pour le lancer, tapez simplement la commande :

```
$ vi fichier
```

où *`fichier`* est le nom du fichier à éditer (qui peut ne pas exister). Si *`nomefile`* est omis, vi utilise un fichier vide (qui n'a pas encore de nom : il faudra lui en donner un au moment de l'enregistrement). Dans cet exemple, nous allons créer un nouveau fichier :

```
$ vi monfichier
```

où *`monfichier`* est le nom du fichier qui n'existe pas encore, dans le répertoire courant.

Une fois que vi est démarré, l'écran montre une colonne de caractères tilde sur la gauche et, tout en bas, le nom du fichier et des informations dont la signification est évidente. Cet écran, sur plus de lignes, ressemble à:

```
~
~
```

```
~
~
~
~
monfichier: new file: line 1
```

Le caractère tilde en début de ligne indique que celle-ci est vide, c'est à dire ne contient rien qui appartienne au fichier. Puisque le fichier chargé (monfichier) n'existe pas encore, toutes les lignes sont vides.

A ce point, appuyer la grande majorité des touches provoque un signal sonore ou un message d'erreur (essayez avec **j** ou **k**). C'est parce que l'éditeur est *modal* et qu'une pression sur une touche n'a pas le même effet suivant le • mode• . Au départ, vi est en *mode commande* et les touches sont interprétées comme des commandes et non comme du texte ajouté au fichier. Certaines commandes permettent de passer du mode commande au *mode insertion* : les touches sont interprétées comme du texte ajouté au fichier. Voyons un exemple.

Appuyez sur **i** pour passer en mode insertion et tapez le texte suivant (en appuyant sur **Entrée** à chaque fin de ligne):

```
Première ligne de texte inséré,
deuxième ligne de texte inséré,
troisième et dernière ligne de texte inséré.
```

Une fois terminé, appuyez sur **ESC** pour retourner en mode commande.

Dans cet exemple, nous avons vu comment passer du mode commande au mode insertion avec la touche **i**, comment insérer quelques lignes de texte en terminant chacune par **Entrée**, puis comment repasser en mode commande avec **ESC**. Maintenant que notre fichier contient quelque chose, nous allons pouvoir expérimenter quelques commandes. Commençons par celles qui nous permettent de naviguer à l'intérieur du fichier.

> **Note :** le texte que nous avons tapé appartient logiquement au fichier `monfichier` (le nom passé comme paramètre à l'éditeur). Mais pour le moment, il ne fait que résider en mémoire vive puisqu'il n'a pas été enregistré dans le fichier. Toutes les opérations effectuées sur le texte (ajout, modification...) ne sont pas enregistrées sur le disque tant que l'utilisateur ne l'a pas explicitement demandé avec une commande d'enregistrement. De cette façon, il est toujours possible d'abandonner les modifications en quittant l'éditeur sans enregistrer et laisser le fichier du disque tel qu'il était. Les commandes pour enregistrer, abandonner, quitter sont décrites à la Section 13.1.9.

13.1.2. Commandes de navigation dans un fichier

En mode commande, quelques commandes permettent de se déplacer à l'intérieur du fichier (c'est à dire changer la position du curseur). La liste suivante résume les principales commandes de navigation :

```
h       curseur vers la gauche      (flèche gauche)
j       curseur vers le bas         (flèche vers le haut)
k       curseur vers le haut        (flèche vers le bas)
l       curseur vers la droite      (flèche droite)
```

```
^F      page suivante               (PgDn)
^B      page précédente             (PgUp)
0       curseur en début de ligne
$       curseur en fin de ligne
b       mot précédent
w       mot suivant
Entrée  premier caractère de la ligne suivante (espace exclus)
+       premier caractère de la ligne suivante (espace exclus)
-       premier caractère de la ligne précédente (espace exclus)
```

Le symbole ^ indique qu'il faut appuyez sur Contrôle en même temps que la touche. **^F** signifie donc Contrôle-F. Suivant la configuration du terminal, il est possible que l'éditeur réussisse à utiliser les touches fléchées et PgUp, PgDn.

Vous avez remarqué que les touches servent à se positionner dans le texte lorsque l'éditeur se trouve en mode commande : utilisées en mode insertion, en revanche, elles servent à écrire du texte dans le fichier (c'est à dire qu'appuyer sur **h** revient à écrire • h•). Nous observons donc un premier fait : en mode insertion, le texte ne peut être ajouté que séquentiellement. Pour se déplacer dans le texte, il est nécessaire de quitter ce mode (avec **ESC**) pour retourner en mode commande. C'est l'une des conséquence du fait que nous ayons affaire à un éditeur modal.

En mode commande, il possible de se rendre directement à la ligne voulue en connaissant son numéro. Par exemple :

```
12G     va à la ligne 12
```

La commande **G** sans numéro envoie le curseur à la fin du fichier.

Préfixe numérique : de nombreuses commandes acceptent un *préfixe numérique* qui modifie l'interprétation d'une partie de l'éditeur. Bien que ces préfixes ne soient pas décrits ici, voici un exemple pour vous donner une idée des possibilités :

```
9k      curseur 9 lignes vers le haut
12dd    efface 12 lignes
3w      le curseur avance de trois mots
5x      efface 5 caractères
```

Dans cette section, ont été décrites les principales commandes de déplacement qui peuvent être utilisée sur du texte déjà écrit. L'éditeur vi possède encore de nombreuses commandes sophistiquées pour se déplacer dans un fichier mais elles ne seront pas abordées dans ce guide.

13.1.3. Commandes de changement de mode

Dans le premier exemple, nous avons vu que pour passer du mode insertion au mode commande, il fallait appuyez sur **ESC**. Si l'on appuie sur cette touche alors que l'on est déjà en mode commande, l'éditeur émet un inoffensif signal sonore. De cette façon, quand on ne sait pas dans quel mode on se trouve, une pression sur la touche Esc garantie de se retrouver en mode commande.

Si Esc est la seule façon de passer en mode commande, il en existe de nombreuses pour passer en mode insertion. Elles dépendent de l'effet voulu, comme le montre le tableau suivant :

`i`	`insère du texte avant le curseur`
`a`	`insère du texte après le curseur`
`A`	`insère du texte en fin de ligne`
`I`	`insère du texte en début de ligne`
`o`	`entame une nouvelle ligne après la ligne courante`
`O`	`entame une nouvelle ligne avant la ligne courante`

La présence de nombreuses commandes qui effectuent des choses similaires est une constante avec vi. Il y a donc plusieurs façons de faire quelque chose. Les experts finissent toujours par trouver la solution pour arriver à quelque chose avec le minimum de fonctions. Par exemple, pour ajouter • fin• en fin de ligne, voici deux séquences équivalentes:

```
$afineESC
AfineESC
```

la première amène le curseur en fin de ligne avec **$**, puis passe en mode insertion avec **a**, ajoute le texte puis retourne en mode commande avec **ESC**. Dans la seconde, la commande **A** combine les fonctions de**$** et **a**.

13.1.4. Commandes d'effacement de texte

Le tableau suivant résume les commandes les plus courantes pour effacer du texte:

`dd`	`efface la ligne courante`
`x`	`efface le caractère sous le curseur`
`X`	`efface le caractère précédant le curseur`
`dw`	`efface jusqu'à la fin du mot`
`D`	`efface du curseur à la fin de la ligne`

Bien entendu, ces commandes ne peuvent être données qu'en mode commande et pas en mode insertion. A partir de maintenant, nous considérerons que ce fait est évident et nous ne le rappellerons plus. Lorsque seront décrites des fonctions directement accessibles du mode insertion, nous l'indiquerons de façon explicite.

13.1.5. Exemple

Voyons maintenant un exemple pratique qui résume l'utilisation des commandes vues dans les sections précédentes. Prenons le fichier de trois lignes que nous avons déjà pris pour exemple. Nous voulons modifier la troisième ligne de comme ceci:

```
troisième ligne de texte inséré dans le fichier.
```

Premièrement, effaçons de la troisième ligne les mots • et dernière• puis ajoutons • dans le fichier• en fin de ligne, avant le point.

Supposons que le curseur soit sur la première ligne (si ce n'est pas le cas, nous pouvons l'y porter avec la commande **1G**). Pour passer de la première à la troisième ligne, l'une des séquences suivantes peut être utilisée (il en existe encore d'autres):

```
<Entrée><Entrée>
jj
3G
/troisième/<Entrée>
```

Une fois arrivé au début de la troisième ligne, déplaçons-nous sur le premier mot à supprimer avec

```
w
```

puis effaçons les deux mots avec:

```
dwdw
```

> **Note :** rappelez-vous que vous pouvez utiliser des préfixes numériques. Ainsi, les deux commandes **dw** peuvent être remplacées par **2dw** qui permet d'effacer un mot deux fois de suite, ou par **d2w**, plus surprenant, qui efface deux mots.

Pour ajouter la fin du texte, allez en fin de ligne avec **$**, puis passez en mode insertion avec **i** pour écrire:

```
dans le fichier
```

Appuyez sur **ESC** pour retourner en mode commande.

13.1.6. Recherche de texte

Pour rechercher une chaîne de caractères dans le fichier courant, utilisez les commandes suivantes:

```
/stringa/
?stringa?
```

La première commande (suivie de Entrée) cherche la chaîne indiquée à partir de la position du curseur. La deuxième commande (aussi suivie de Entrée) cherche la chaîne partir du début du fichier.

Par exemple, reprenons les trois lignes que nous avons insérées initialement et lançons la commande suivante:

```
1G/dernière/
```

appuyez sur Entrée après la dernière barre oblique. La commande **1G** porte le curseur sur la première ligne du fichier (notez que cette commande ne renvoie aucun écho). La séquence **/dernière/** (qui se visualise sur la dernière ligne de l'écran), suivie par **Entrée**, lance la recherche de la chaîne "dernière" dans le fichier et positionne le curseur sur la ligne trouvée.

> **Note :** les commandes de recherche constituent donc une autre méthode de positionnement dans un fichier, permettant de le visualiser rapidement.

Pour répéter la dernière recherche effectuée sans avoir à la retaper, utilisez les commandes:

```
n       recommence la recherche dans la même direction
N       recommence la recherche dans la direction opposée
```

13.1.7. Commandes en mode insertion

Même lorsque vous saisissez du texte, quelques commandes sont disponibles, généralement accessibles avec la touche Contrôle:

```
^H      efface le caractère à gauche du curseur
BS      efface le caractère à gauche du curseur
^W      efface le mot précédent
^U      efface à partir du début de la ligne
ESC     retourne en mode commande
```

13.1.8. Annuler l'effet d'une action

Si, une fois que vous êtes retourné en mode commande, vous vous apercevez que vous avez commis une erreur, il est possible d'utiliser la commande d'annulation *undo* (touche **u**) pour revenir à l'état lors de la dernière commande.

13.1.9. Commandes pour quitter vi

Les principales commandes pour terminer de travailler et quitter vi sont:

```
:w      enregistre le fichier
:q      quitte
:wq     enregistre le fichier puis quitte
:q!     quitte sans enregistrer (les modifications sont perdues)
:w abc  enregistre le fichier sous le nom "abc"
```

Pour changer le nom du fichier en cours (ou pour lui en donner un s'il n'en a pas):

```
:file nom-fichier
```

Les commandes décrites dans cette section sont des exemples du fonctionnement de vi, le *modo ex* qui n'est pas décrit dans ce guide. L'éditeur vi est, en fait, issu de la combinaison de deux éditeurs distincts dont l'un (vi pour le nommer) sert d'interface *plein écran* pour l'autre (ex) qui fournit les fonctions de base de l'édition. Ex peut être pris comme une version évoluée de l'éditeur ed.

13.1.10. Conclusion

Comme le disait l'introduction, seules les commandes les plus élémentaires de vi ont été abordées dans cette section afin que l'utilisateur soit en mesure d'écrire les fichiers de configuration de NetBSD après l'installation. Pour pouvoir utiliser cet éditeur de façon productive (et confortable), il est nécessaire d'approfondir votre connaissance des commandes, par exemple, à l'aide des documents cités à la Section 13.2.2.

13.2. Configurer vi

L'éditeur standard fournit par NetBSD est, est-il besoin de le préciser, vi, l'outils le plus aimé et le plus haï du monde. Si vous n'utilisez pas vi, passez cette section sinon, lisez-la avant d'installer une autre version de vi. Le vi de NetBSD (nvi) a été écrit par Keith Bostic de l'UCB afin d'en avoir une version librement redistribuable, possédant de nombreuses extensions puissantes tout en restant compatible avec la version originale. Nvi est depuis devenu la version standard de vi pour BSD.

Parmi les extensions les plus connues, citons:

- Expressions régulières étendues (style egrep) accessibles avec l'option `extended`).
- Tag stacks.
- Annulation infinie (pour annuler, tapez **u**, pour continuer, tapez **.**).
- Recherche incrémentale accessible avec l'option `searchincr`).
- Défilement gauche-droite des lignes accessible avec l'option `leftright`. Le nombre de colonnes à faire défiler est défini par l'option `sidescroll`).
- Edition de l'historique de la ligne de commandes accessible avec l'option `cedit`).
- Noms de fichiers complétés automatiquementavec l'option `filec`).
- Écrans et affichage de fond.
- Écran d'édition scindé.

13.2.1. Extensions de `.exrc`

Les exemples suivants montrent un fichier .exrc dont certaines options étendues sont activées.

```
set showmode ruler
set filec=^[
set cedit=^[
```

La première ligne active l'affichage de la position du curseur (en lignes et colonnes) et du mode courant (Commande, Insertion, Ajout) sur la ligne d'état. La seconde ligne (où ^[est le caractère Esc), permet de compléter automatiquement les noms de fichiers à l'aide du caractère Esc. La troisième ligne active l'édition de l'*history* de la ligne de commandes (aussi avec le caractère Esc). Par exemple, taper ":" puis Esc ouvre une fenêtre contenant la liste des commandes précédentes qui peuvent être modifiées et exécutées (Appuyez sur Entrée pour exécuter la commande sur laquelle vous êtes).

13.2.2. Documentation

L'archive des sources (`src.tgz`)) contient une grande quantité de documentation sur vi/nvi et ex, dans le répertoire `/usr/src/usr.bin/vi/docs`. Par exemple:

- Edit: A tutorial
- Ex Reference Manual
- Page de manuel de Vi
- An Introduction to Display Editing with Vi par William Joy et Mark Horton
- Ex/Vi Reference Manual par Keith Bostic, le manuel de référence de nvi qui décrit en détail toutes les commandes de l'éditeur.
- Vi Command & Function Reference
- Vi tutorial (débutants et confirmés)

Si vous n'avez jamais utilisé vi, le *Vi tutorial* est un bon point de départ. Il a été conçu pour être lu avec vi en présentant graduellement au lecteur les commandes qui peuvent être essayées dans le document. *An Introduction to Display Editing with Vi* par William Joy et Mark Horton est aussi un bon point de départ. Si vous voulez en apprendre plus sur vi et les extensions nvi, lisez *Ex/Vi Reference Manual* de Keith Bostic qui documente toutes les commandes et options de l'éditeur.

13.3. Utilisation des tags avec vi

Ce thème n'est pas directement lié à NetBSD mais peut s'avérer utile, par exemple, pour lire les sources du noyau.

Lorsque vous consultez du code source dans une arborescence de répertoires et sous-répertoires, vous pouvez simplifier l'exploration à l'aide de la fonction *tag* de vi. La méthode est la suivante:

1. Allez dans le répertoire racine des sources.

   ```
   $ cd /chemin
   ```

2. Tapez les commandes suivantes:

   ```
   $ find . -name "*.[ch]" > listefichiers
   $ ctags -L listefichiers
   ```

3. Ajoutez la ligne suivante à `.exrc`:

   ```
   set tags=/chemin/tags
   ```

   (mettez le chemin d'accès correct à la place de "chemin")

Chapitre 14. X

14.1. Qu'est-ce que X?

Le Système X Window est un environnement graphique existant sur NetBSD et de nombreux systèmes Unix (et non Unix). En fait, c'est même plus que ça : grâce à l'utilisation des protocoles X, le Système X Window est "réseau-transparent" et peut faire tourner des applications distribuées (client-serveur). C'est à dire, en gros, que vous pouvez faire tourner des applications sur un hôte (client) et afficher, de manière transparente, le graphique sur un autre hôte (serveur). De manière transparente signifie que vous n'avez pas besoin de modifier l'application pour arriver à ce résultat. Le Système X Window est produit et maintenu par le Consortium X et la version actuelle est X11R6. La mouture X présente avec NetBSD est XFree86, une implémentation open source librement distribuable du Système X Window.

Dans vos débuts avec X, vous trouverez de nombreux termes qui vous paraîtrons certainement confus. Les éléments de base pour utiliser X sont :

- Matériel vidéo supporté par XFree86.
- Un *serveur X* faisant l'interface avec le matériel. Le serveur X fournit les méthodes standard pour ouvrir des fenêtres, faire des dessins (dont les polices de caractères pour afficher du texte) et gérer la souris, le clavier et autres. X est réseau-transparent, c'est à dire qu'il permet de faire tourner un client X sur une machine et un serveur X (affichage avec carte vidéo) sur une autre machine).
- Un *gestionnaire de fenêtres* tournant sous serveur X. Il est essentiellement un client spécial qui permet de contrôler le placement des fenêtres. Il • decorates• aussi les fenêtre avec des • widgets• (ou "fengets", contraction des mots fenêtre et gadget) qui gèrent les déplacements de fenêtres, la modification de taille, la mise en icônes et d'autres fonctions. Un gestionnaire de fenêtres peut aussi gérer divers "effets spéciaux" d'embellissement, tuer des programmes en cliquant sur leur fenêtre, etc...
- Un gestionnaire de *bureau* (desktop manager), optionnel. KDE et GNOME, par exemple, sont des bureaux : ce sont des suites de programmes plus ou moins intégrés conçus pour donner une interface commune aux programmes. Cela inclus une aide en ligne, un navigateur, un gestionnaire de fichiers, un programme remplaçant xterm, un environnement de développement, la gestion audio-vidéo...
- Toute autre application (clients X tierce partie). Ils inter-agissent avec le serveur X et le gestionnaire de fenêtres. A moins que le gestionnaire de fenêtres ne fasse partie du bureau (s'il y en a un), il ne sera pas plus impliqué que ce que lui demandera l'application (cependant, par exemple, GNOME est capable de détecter que GIMP est installé et d'ajouter un menu pour le lancer).

En résumé : pour utiliser l'environnement graphique, il vous faut :

- le système XFree86.
- un gestionnaire de fenêtres (XFree86 est systématiquement accompagné d'un gestionnaire de base appelé twm).
- Si vous préférez un environnement plus sophistiqué, vous pouvez aussi installer un bureau bien que ça ne soit pas nécessaire. Ils possèdent quelques caractéristiques intéressantes qui aident les utilisateurs

venant d'environnements tels que le Macintosh ou MS-Windows (le bureau KDE, par exemple, est assez proche de MS-Windows).

> **Note :** il doit rester clair que les bureaux tels que GNOME et KDE ne fournissent aucun serveur X. Ils tournent par dessus un serveur X existant géré par XFree86. GNOME et KDE peuvent utiliser leur propre gestionnaire de fenêtres ou un déjà installé.

Normalement, vous ne pouvez lancer qu'un seul gestionnaire de fenêtres sur un serveur X donné (mais vous pouvez faire tourner plusieurs serveur X sur la même machine). Si vous n'avez pas votre propre gestionnaire de fenêtres et que vous lancez GNOME ou KDE, ces derniers en choisirons un pour vous.

14.2. Configuration

Si vous n'avez pas choisi la configuration minimale pendant l'installation, X est présent sur votre ordinateur, dans son incarnation XFree86, prêt à fonctionner. La seule chose à faire est de créer le menaçant fichier `/etc/XF86Config`. Pour avoir une idée de ce à quoi il ressemble, regardez le fichier `/usr/X11R6/lib/X11/XF86Config.eg`. La structure du fichier de configuration est expliquée dans les pages de manuel XF86Config(4 et 5) qui peuvent être consultées avec la commande :

```
# man XF86Config
```

Avant de configurer le système, il serait avisé de consulter soigneusement la documentation du répertoire `/usr/X11R6/lib/X11/doc` : il existe de nombreux • README• sur les cartes vidéo, les souris et certains sont même spécifiques à NetBSD (`README.NetBSD`). Je vous suggère de commencer par lire `QuickStart.doc`. Vous aurez sans doute le sentiment que d'autres systèmes vous permettent de démarrer plus rapidement et sans efforts mais le temps passé à lire la documentation n'est jamais perdu : la connaissance de X et de votre configuration que vous allez acquérir se révélera très utile dans de nombreuses futures occasions et vous pourrez tirer le meilleur parti de votre matériel (et de vos logiciels).

Vous pouvez créer le fichier `/etc/XF86Config` manuellement avec un éditeur ou bien le générer automatiquement avec un programme interactif. Les meilleurs programmes connus sont xf86config et XF86Setup. Le premier est en mode texte et est installé avec le système de base, le deuxième est graphique et doit être installé avec la collection de paquetages.

Une solution mixte est sans doute une bonne approche : tout d'abord créer XF86Config avec l'un des deux programmes puis vérifier et affiner le fichier avec un éditeur.

L'interface des programmes est différente mais ils ont tout les deux besoin des mêmes informations :

- Le type et le pilote de souris à utiliser.
- Le type de clavier et l'organisation des touches.
- Le type de carte vidéo.
- Le type d'écran.

Avant de configurer le système, récupérez ces informations.

14.3. La souris

La première chose à vérifier est le type de souris que vous avez (par exemple, série, PS/2...) et le pilote (par exemple, *wsmouse* requière un protocole différent). Si vous avez une souris série, choisissez le protocole correspondant et le port sur lequel elle est branchée. Par exemple, pour une souris série sur le premier port série :

```
Section "Pointer"
  Protocol    "Microsoft"
  Device      "/dev/tty00"
EndSection
```

Pour une souris utilisant wsmouse, vous aurez :

```
Section "Pointer"
  Protocol    "wsmouse"
  Device      "/dev/wsmouse0"
EndSection
```

Dans le champ «Device» (périphérique), vous pouvez aussi indiquer `/dev/mouse` à condition d'avoir correctement créé le lien. Par exemple :

```
# ln -sf /dev/wsmouse0 /dev/mouse
```

14.4. Le clavier

Même si vous avez déjà configuré le clavier avec wscons, vous devez aussi le faire pour X afin de bénéficier des claviers non-US.

Une solution simple est d'utiliser le protocole XKB, en indiquant le type et l'organisation des touches.

C'est l'une des parties où les programmes de configuration ne sont pas très forts. Vous pouvez alors choisir le clavier standard puis le changer à la main dans l'éditeur.

```
# XkbDisable
# XkbKeymap   "xfree86(us)"

XkbModel        "pc102"
XkbLayout       "de"
XkbVariant      "nodeadkeys"
```

Si vous voulez utiliser les touches «Windows» de votre clavier, utilisez le modèle pc105 au lieu de pc102.

14.5. L'écran

Il est très important d'indiquer des valeurs correctes pour les fréquences horizontales et verticales de l'écran : une définition correcte protège l'écran de dommages que peut causer une incompatibilité entre la configuration et la carte vidéo. Cette information se trouve dans le manuel de l'écran. Le répertoire de documentation de X contient une liste de paramètres de nombreux écrans. Elle peut être utilisée comme point de départ pour personnaliser vos propres paramètrages.

14.6. Carte vidéo et serveur X

La carte vidéo peut être choisie dans la base de données du programme de configuration qui se débrouillera pour déterminer les paramètres.

Lorsque vous aurez choisi la bonne carte vidéo, il vous faudra choisir le serveur correspondant. Habituellement, le programme peut le trouver automatiquement mais certaines cartes peuvent être pilotées par plusieurs serveurs (par exemple, S3 Virge est supportée par les serveurs SVGA et S2V). Dans ce cas, étudiez la documentation de ces serveur et décidez celui que vous préférez : différents serveurs ont généralement différentes capacités et différents degrés de support de la carte vidéo.

14.7. Démarrer X

Lorsque vous quittez le programme de configuration, celui-ci crée le fichier `/etc/XF86Config` qui peut alors être examiné voire modifié à la main.

Avant de lancer X, vous devez :

- vérifier que le lien symbolique `/usr/X11R6/bin/X` pointe vers le bon serveur X :

  ```
  # ls -l /usr/X11R6/bin/X
  ```

- vérifier que la configuration est correcte. Lancez :

  ```
  # X -probeonly
  ```

  et examinez soigneusement le résultat.

Vous pouvez alors démarrer X avec la commande suivante :

```
# startx
```

Si X ne s'allume pas, c'est probablement qu'il y a des erreurs dans le fichier de configuration.

Si X démarre mais ne fonctionne pas comme attendu (par exemple vous ne pouvez pas bouger la souris), vous pouvez quitter précipitamment avec la combinaison de touches Ctrl-Alt-Effacer (pas dans tous les portages). Si tout marche correctement, vous restez dans l'environnement X avec le gestionnaire de fenêtres par défaut (twm) : bien qu'il soit simplifié, beaucoup d'utilisateurs le jugent suffisant pour leurs besoins. Si vous voulez un gestionnaire de fenêtres hautement configurable avec des cloches et des flûtes, vous avez du choix dans la collection de paquetages.

Pour commencer à personnaliser X, essayez la commande suivante, dans un xterm, pour changer la couleur du fond :

```
# xsetroot -solid DarkSeaGreen
```

14.8. Personnaliser X

L’aspect de l’environnement X peut être personnalisé de différentes façons. La méthode la plus simple est de copier le fichier par défaut `.xinitrc` dans votre répertoire personnel et de le modifier. Par exemple :

```
# cp /usr/X11R6/lib/X11/xinit/xinitrc ~/.xinitrc
# vi .xinitrc
```

L’exemple suivant montre comment démarrer le gestionnaire de fenêtres (twm), ouvrir une instance du programme xclock en bas à droite de l’écran et deux fenêtres xterm. La couleur • Bisque4• est utilisée comme fond d’écran.

```
la première partie du fichier est identique
...
# lancer quelques programmes sympa
twm &
xclock -geometry 50x50-1-1 &
xterm -geometry 80x34-1+1 -bg OldLace &
xsetroot -solid Bisque4 &
exec xterm -geometry 80x44+0+0 -bg AntiqueWhite -name login
```

Avec ce genre de paramètrages, pour éditer X, vous devez fermer le dernier terminal (celui dont le titre est • login•).

Même avec ces simples modifications X acquière une allure nettement meilleure. Pour lui en donner une encore meilleure vous pouvez installer des utilitaires à partir de la collection de paquetages. Par exemple :

xcolorsel

> affiche toutes les couleurs définies dans `rgb.txt`. Utilisez-le pour choisir la couleur de la fenêtre racine et des xterms.

xpmroot

> vous permet d’afficher une image en fond d’écran.

xscreensaver

> Un économiseur d’écran pour X.

xdaemon

> aucun bureau ne peut être complet sans ce paquetage : il affiche une image du démon BSD, en deux tailles au choix, qui se déplace sur l’écran.

14.9. Autres gestionnaires de fenêtres

Si vous n’aimez pas twm, qui est un gestionnaire très simple auquel il manque beaucoup de fonctions et n’est pas très paramétrable, vous pouvez en choisir un autre dans la collection de paquetages. Les plus connus sont : fvwm2, olwm/olvwm (Open Look Window Manager), WindowMaker, Enlightenment, AfterStep.

Le reste de cette section prend l'installation de WindowMaker comme exemple. C'est un gestionnaire de fenêtres d'aspect agréable et très configurable. Pour ajouter le programme, le paquetage pré-compilé `windowmaker-0.60.tgz` sera utilisé. Il dépend d'autres paquetages qui devront aussi être installés. Comme d'habitude, **pkg_add** et **make install** iront chercher automatiquement les paquetages requis sans que nous n'ayons besoin de nous en préoccuper.

```
# cd /usr/pkgsrc/x11/windowmaker
# make depends-list
xpm-3.4k
jpeg-6b
pkglibtool-1.2p2
giflib-3.0
libproplist-0.9.1
tiff-3.5.2
```

> **Note :** vous pouvez connaître les dépendances avec la commande :
>
> ```
> # pkg_info -f windowmaker-0.61.0.tgz | grep depends
> ```

Après avoir trouvé les paquetages voulus, WindowMaker et des thèmes pré-définis peuvent être ajoutés :

```
# pkg_add windowmaker-0.61.0.tgz wmthemes-0.6x.tgz
```

WindowMaker est maintenant installé. Pour le démarrer, il vous faut modifier vos fichiers `.xinitrc` ou `.xsession`. Modifiez la ligne qui appelle `wmaker`. Par exemple :

```
# lancer quelques programmes sympa
# lancer WindowMaker
wmaker &
xclock -geometry 50x50-1-1 &
xdaemon2 -geometry +0-70 &
...
```

Dans cet exemple, le programme xdaemon est aussi lancé automatiquement.

Avant de démarrer WindowMaker, le programme de configuration doit être lancé :

```
$ wmaker.inst
$ startx
```

14.10. Connexion graphique avec xdm

Si vous travaillez systématiquement sous X et si la première chose que vous faites, une fois connecté, est de lancer **startx**, vous pouvez mettre en place une invite de connexion (login) graphique qui le fait tout seul. C'est très facile :

1. Créez un fichier `.xsession` dans votre répertoire personnel. Il est identique à `~/.xinitrc` et peut, en fait, être un lien vers ce dernier.
2. Modify `/etc/rc.conf`:

   ```
   xdm=YES       xdm_flags=""       # x11 display manager
   ```

   Si vous préférez (pourquoi ?), vous pouvez ajouter une ligne à la fin du fichier `/etc/rc.local` au lieu de modifier rc.conf :

   ```
   /usr/X11R6/bin/xdm
   ```

   Cette méthode peut être utilisée pour démarrer, par exemple, kdm ou gdm plutôt que xdm.

Les fichiers de configuration de xdm se trouvent dans le répertoire `/usr/X11R6/lib/X11/xdm`. Dans le fichier `Xservers`, X est démarré, par défaut, sur le terminal virtuel vt05. Si vous voulez utiliser un autre terminal, c'est le bon endroit pour indiquer lequel. Pour viter tout conflit entre getty et xdm, mieux vaut dmarrer ce dernier sur un terminal virtuel o getty est dsactiv. Par exemple, si dans `Xservers` vous avez :

```
:0 local /usr/X11R6/bin/X :0 vt04
```

dans `/etc/ttys` vous devrez avoir :

```
ttyE3   "/usr/libexec/getty Pc"         vt220   off secure
```

(notez que vt04 correspond ttyE3 car vt commence 1 et ttyE 0).

Si vous voulez un bel aspect pour votre écran de connexion xdm, vous pouvez modifier son fichier de configuration. Par exemple, pour changer la couleur du fond d'écran, ajoutez la ligne suivante au fichier `Xsetup_0` :

```
xsetroot -solid SeaGreen
```

Plutôt que de mettre une couleur, vous pouvez afficher une image à l'aide du programme xpmroot. Par exemple :

```
xpmroot /path_to_xpm/netbsd.xpm
```

Si vous fouillez un peu le fichier de configuration, vous pourrez établir de nombreux effets visuels et afficher un écran de démarrage plaisant.

Chapitre 15. Émulation Linux

Le portage de NetBSD pour i386 peut exécuter un grand nombre de programmes natifs de Linux en utilisant la couche d'émulation Linux. Généralement, lorsque l'on parle d'émulation, on imagine quelque chose de lent et d'assez peu fiable car, la plupart du temps, l'émulateur doit reproduire les instructions du matériel et même des architectures (souvent d'anciennes machines) de façon logicielle. Le cas de l'émulation Linux est tout à fait différent : il s'agit d'un tout petit logiciel, la grande majorité des appels systèmes étant similaires entre les deux systèmes. Le code de l'application lui-même est exécuté avec la pleine vitesse du processeur dont les performances ne sont pas dégradées par l'émulation. L'impression est la même que pour les applications natives de NetBSD.

Ce chapitre explique comment configurer l'émulation Linux en prenant un exemple : l'installation d'un programme bien connu, Acrobat Reader en version 4.

15.1. Mise en place

L'installation de l'émulateur Linux est décrite dans la page de manuel compat_linux(8). A l'aide du système de paquetages, seules deux étapes sont nécessaires :

1. Configurer le noyau.
2. Installer les bibliothèques de Linux.

15.1.1. Configurer le noyau

Si vous utilisez un noyau GENERIC, vous n'avez rien à faire car la compatibilité avec Linux est déjà assurée.

Si vous utilisez un noyau personnel, vérifiez que les options suivantes sont activées :

```
option COMPAT_LINUX
option EXEC_ELF32
```

Une fois le noyau compilé avec ces options, vous pouvez installer les logiciels.

15.1.2. Installer les bibliothèques de Linux

Vous pouvez les obtenir à partir de n'importe quelle distribution de Linux à condition qu'elle ne soit pas trop ancienne. La méthode conseillée est plutôt de les prendre dans le système de paquetages et de les installer automatiquement (ce sont celles de Suse qui sont utilisées). Voici ce qu'il se passe lors de l'installation :

- Un *deuxième répertoire racine* est créé pour les programmes Linux. Il s'appelle `/emul/linux/`. Les programmes Linux en mode émulation utiliseront ce répertoire comme répertoire racine.
- Les bibliothèques partagées de Linux sont installées. La plupart des applications sont dynamiquement liées et s'attendent à trouver les librairies voulues dans le système. Par exemple, pour Acrobat Reader,

si vous allez dans le répertoire `/usr/pkgsrc/print/acroread`et tapez la commande **make depends**, vous obtiendrez le message suivant :

```
===>  acroread-4.0 requires Linux glibc2 libraries - see compat_linux(8).
```

Ces deux étapes sont automatiquement gérées par le système de paquetages sans intervention manuelle de l'utilisateur (j'imagine que, maintenant, vous commencez à aimer le système de paquetages...)

Pour installer les bibliothèques, un programme doit être présent pour gérer les formats RPM : il s'agit de `rpm-2.5.4`, qui sera utilisé pour extraire les bibliothèques de Suse.

Ensuite, le paquetage `suse_base` doit être installé. Les fichiers RPM de Suse peuvent être téléchargés par le système de paquetages ou, si vous avez le cédérom de Suse, vous pouvez les copier dans le répertoire `/usr/pkgsrc/distfiles/suse` et lancer les commandes **make** et **make install**.

De la même manière, installez `suse_compat`, `suse_libc5` et `suse_x11`. La configuration finale est :

```
# pkg_info -a | grep suse
suse_base-6.1p1     Linux compatibility package
suse_x11-6.1p1      Linux compatibility package for X11 binaries
suse_compat-6.1p1   Linux compatibility package with old shared libraries
suse_libc5-6.1p1    Linux compatibility package for libc5 binaries
```

15.1.3. Installer Acrobat Reader

Maintenant que tout est prêt, nous pouvons commencer à installer Acrobat Reader (ou d'autres programmes Linux). Allez dans le répertoire `/usr/pkgsrc/print/acroread` et tapez les commandes habituelles :

```
make
make install
```

Le script d'installation d'Acrobat Reader vous demande si vous acceptez la licence. Si oui, vous pouvez alors utiliser le programme.

15.2. Structure des répertoires

En examinant le résultat de l'installation des bibliothèques et programmes Linux, nous nous apercevons que `/emul/linux` est, en fait, un lien symbolique vers le répertoire `/usr/pkg/emul/linux`, dans lequel les sous-répertoires suivants ont été créés :

```
bin/
boot/
cdrom/
dev/
etc/
floppy/
home/
```

```
lib/
mnt/
opt/
proc/
root/
sbin/
usr/
```

> **Note :** prenez toujours le répertoire `/emul/linux` comme référence et non `/usr/pkg/emul/linux`. Ce dernier est un détail dans l'implémentation et peut changer dans le futur.

Combien de place faut-il pour le logiciel d'émulation Linux ? Sur mon système, voici ce que j'obtiens :

```
# cd /usr/pkg/emul
# du -k linux
...
60525   linux/
```

Acrobat Reader, le programme, a été installé dans le répertoire normal de tous les binaires de paquetages : `/usr/pkg/bin/`.

Chapitre 16. Audio

Manolo De Santis

Contribution de Manolo De Santis

Ce chapitre est une rapide introduction à l'utilisation des périphériques audio sur NetBSD (qui voudrait d'un ordinateur muet ?)

16.1. Eléments matériels de base

Pour que votre système puisse jouer de l'audio, vous devez savoir de quelle carte audio il est muni. Malheureusement, il n'est souvent pas suffisant de connaitre la marque et le modèle, de nombreuses cartes étant équipées de puces fabriquées ailleurs. Connaitre ces puces peut donc parfois s'avérer utile. Le noyau de NetBSD reconnait de nombreuses puces et un rapide coup d'oeil à **dmesg** est généralement suffisant.

Tapez donc la commande :

```
# dmesg | more
```

et recherchez votre carte. Si vous avez de la chance, vous n'aurez rien à faire car NetBSD détecte et configure tout seul de nombreuses cartes audio.

Parfois, l'audio ne fonctionne pas parce que la carte n'est pas gérée ou bien sa reconnaissance par NetBSD nécessite des manipulations de votre part. De nos jours, les cartes audio sont très bon marché et il vaut parfois la peine d'en acheter une autre. Toutefois, avant de le faire, essayez certaines choses permettant éventuellement de la faire reconnaitre par NetBSD.

16.2. Paramètres du BIOS

Cette section ne concerne que les possesseurs de PC i386. Les autres architectures (par exemple Amiga) ne possèdent pas cette caractéristique. La première chose importante à faire est de déterminer le type de bus sur lequel est branchée la carte.

Les interfaces les plus courantes sont ISA et PCI.

Les cartes ISA sont généralement plus délicates à configurer en raison de leur interaction avec le BIOS de l'ordinateur.

Sur les machines récentes (contruites après 1997) existent des options de BIOS qui causent de nombreux maux de têtes pour configurer les cartes audio ISA (et pas uniquement les cartes audio) : cette options est appelée • OS PNP Installé• (le nom peut être différent sur votre BIOS). En règle générale, il vaut mieux la déctiver (la mettre sur • NON•) pour NetBSD.

> **Note :** Sur de nombreux systèmes, tout marche correctement même si cette option est activée. Cela dépend du système d'exploitation.

16.3. Configurer le périphérique audio

Pendant l'installation de NetBSD, les périphériques sont créés dans le répertoire `/dev`. Ceux qui nous intéressent sont :

```
/dev/audio
/dev/sound
/dev/mixer
```

S'ils ne sont pas présents, ils peuvent être créés avec :

```
# cd /dev
# ./MAKEDEV all
```

Cette commande crée tous les périphériques, dont les audio.

La carte audio est maintenant certainement prête à être utilisée sans travail supplémentaire.

Effectuons un test rapide en envoyant un fichier audio sur le périphérique (ces fichiers ont souvent l'extension `.au`). Si vous n'en avez pas, envoyez simplement un fichier texte ou binaire (évidemment, vous n'entendrez rien d'intelligible...). Utilisez `/dev/audio` ou `/dev/sound` :

```
# cat fichier > /dev/audio
```

ou

```
# cat fichier > /dev/sound
```

Si vous entendez quelque chose, c'est que votre carte est gérée et reconnue par NetBSD et qu'elle a été configurée par le noyau au démarrage. Autrement, vous devez configurer le noyau pour le périphérique audio installé (à condition que la carte et la puce soient gérées).

16.4. Configurer le périphérique audio

NetBSD gère une grande variété de cartes audio et le noyau GENERIC en active et configure la plupart.

De nombreux PC sont dépourvus de carte son mais possèdent une audio interne. Cela veut dire que la puce se trouve sur la carte mère. Ce cas n'est pas géré par le noyau GENERIC. Vous devez compiler votre propre noyau afin d'utiliser cette puce. Recherchez les lignes suivantes (ou similaires) dans le fichier GENERIC :

```
# Plug-and-Play BIOS and attached devices

#pnpbios*       at mainbus?

# mainboard audio chips
```

```
#ess*           at pnpbios? index ?     # ESS AudioDrive
#sb*            at pnpbios? index ?     # NeoMagic 256AV in sb mode
#wss*           at pnpbios? index ?     # NeoMagic 256AV in wss mode
#ym*            at pnpbios? index ?     # OPL3-SA3
```

Décommentez (effacez le # de) la ligne • pnpbios• ainsi que celle décrivant le périphérique de votre carte mère.

> **Note :** lorsque vous créez votre noyau personnel, travaillez sur une copie du fichier GENERIC, comme décrit dans Chapitre 7.

Il est parfois nécessaire de paramètrer manuellement l'IRQ et DMA du périphérique.

Si vous rencontrez encore des problèmes, essayez d'activez tous les périphériques. Certaines cartes audio peuvent fonctionner en émulant une autre carte.

De nombreuses puces font usage de la compatibilité SoundBlaster et OPL mais la plupart fonctionnent avec l'émulation WSS.

OPL est un synthétiseur MIDI produit par Yamaha. Il en existe plusieurs variantes (OPL2, OPL3SA, OPL3SA2, etc...). De nombreuses cartes audio dépendent de ce composant ou d'un compatible. Par exemple, les puces produites par Crystal (dont la commune CS423x) le possèdent, raison pour laquelles elles fonctionnent avec NetBSD.

WSS n'est pas une micro-puce, c'est l'abréviation de Windows Sound System. Il s'agit aussi du pilote de NetBSD qui gère le système audio de Microsoft Windows. De nombreuses cartes audio fonctionnent avec Windows car elles adhèrent à cette norme (WSS). Il en va de même pour NetBSD.

Parmi les nombreuses cartes que j'ai testées avec NetBSD, la plupart fonctionnent si `opl*` et `wss*` sont actifs dans le noyau.

Vous n'aurez aucun problèmes à faire marcher une Creative SoundBlaster avec NetBSD: presques toutes sont gérées, dont la Sound Blaster Live 1024 !

Une fois que tout fonctionne correctement, vous pouvez désactiver du noyau les périphériques dont vous n'avez pas besoin.

16.5. Commandes avancées

NetBSD connait quelques commandes concernant l'audio. Ce sont :

- audioctl
- mixerctl
- audioplay
- audiorecord

16.5.1. audioctl

Audioctl est apparu avec NetBSD 1.3. Elle sert à paramètrer manuellement certaines variables E/S comme la fréquence de lecture et d'enregistrement. La liste des variables accessibles s'affiche avec :

```
# audioctl -a | more
```

Par exemple, pour écouter une musique avec la qualité CD, tapez :

```
# audioctl -w play=44100,2,16,slinear_le
```

Cette commande met la fréquence à 44100Hz, sur 2 canaux, 16 bits, encodage slinear_le.

Les encodages reconnus sont listés avec :

```
# audioctl encodings
```

Cette commande affiche tous les encodages gérés par la carte audio de votre système.

16.5.2. mixerctl

Cette commande sert à configurer le mixage audio. Son interface est similaire à celle de audioctl.

16.5.3. audioplay

Cette commande sert à jouer des fichiers audio. Pour des besoins plus sophistiqués, vous pouvez installer l'un des nombreux programmes du système de paquetages, qui vous permettra de lire différents formats (MP3...).

16.5.4. audiorecord

Sans surpise, cette commande sert à enregistrer des fichiers audio.

Chapitre 17. Obtenir des sources par CVS

Reinoud Koornstra

Ce chapitre (Obtenir des sources par CVS) est une contribution de Reinoud Koornstra

CVS (Concurrent Versions System) peut être utilisé pour aller chercher ou pour conserver à jour l'arborescence des sources de NetBSD, tout en respectant les modifications effectuées dans celle-ci. Il existe trois arborescences sur lesquelles appliquer CVS : Tout d'abord l'arborescences des sources courantes (-current) dans laquelle les derniers développements peuvent être suivis ou testés. Ensuite, l'arborescence des sources 1.5 dans laquelle se trouve les correctifs à appliquer pour résoudre les problèmes signalés et reboucher les trous de sécurité trouvés (par exemple, le problème de fragmentation de ipf a été résolu) et dans laquelle certaines choses ont été ajoutées ou remplacées par de nouvelles versions, stables et sûres (par exemple, il est désormais possible de supprimer complètement la table arp avec la commande arp, c'est une caractéristique supplémentaire, pas une correction). Enfin, l'arborescence des sources 1.4 dans laquelle ne se trouvent que les correctifs à appliquer. Elle ne comportera plus aucun développement.

17.1. Récupérer les sources du système et des applications

CVS n'est pas nativement sur NetBSD 1.4.x et 1.5.x. Il fera en revanche partie du système 1.6 et est déjà présent la branche -current. Pour installer CVS (si vous ne l'avez pas déjà), tapez simplement :

```
% pkg_add ftp://ftp.netbsd.org/pub/NetBSD/packages/<OS Ver>/<arch>/All/cvs-1.11nb2.tgz
```

Où <OS Ver>, et <arch> peuvent être obtenus en tapant

```
% sysctl kern.osrelease hw.machine_arch
```

Pour récupérer les sources à partir de zéro, le répertoire `/usr/src` étant vide :

```
% setenv CVSROOT :pserver:anoncvs@anoncvs.netbsd.org:/cvsroot
% cd /usr
% cvs login
password: anoncvs
% cvs checkout -rnetbsd-1-5 -PA src
```

Ou bien utilisez ssh, afin d'encrypter les données :

```
% setenv CVS_RSH ssh
% setenv CVSROOT anoncvs@anoncvs.netbsd.org:/cvsroot
% cd /usr
% cvs checkout -rnetbsd-1-5 -PA src
```

Les sources de la branche 1.5 seront ainsi récupérées. Pour obtenir les sources de -current, omettez juste • -rnetbsd-1-5• dans la dernière ligne. Pour obtenir les sources de la branche 1.4, utilisez • -rnetbsd-1-4• dans la dernière ligne.

Pour mettre à jour les sources de la branche 1.5 que vous avez déjà récupérée :

```
% setenv CVSROOT :pserver:anoncvs@anoncvs.netbsd.org:/cvsroot
% cd /usr
% cvs login
password: anoncvs
% cvs -d $CVSROOT update -rnetbsd-1-5 -PAd src
```

Ou par ssh :

```
% setenv CVS_RSH ssh
% setenv CVSROOT anoncvs@anoncvs.netbsd.org:/cvsroot
% cd /usr
% cvs -d $CVSROOT update -rnetbsd-1-5 -PAd src
```

Pour mettre à jour la branche -current, omettez • -rnetbsd-1-5• . Pour mettre à jour la branche 1.4, utilisez • -rnetbsd-1-4• .

Lorsque vous voulez effectuer une mise à jour à partir d’une arborescence non propre, par exemple lorsque vous recompilez une partie des sources ou même les sources complètes ou le noyau sans effectuer de «make cleandir», vous devez faire des fichiers objet.

Les répertoires objet sont nécessaires pour effectuer un • cvs update• d’arborescence non propre. Une arborescence non propre existe lorsque vous compilez une partie de celle-ci, partiellement ou complètement, sans taper ensuite • make clean• pour cette partie ou bien • make cleandir• pour l’arborescence complète. Dans ce cas, cvs voudra créer des répertoires portant le même nom que des binaires et échouera. Par exemple, vous avez un répertoire "groff" et vous recompilez le binaire "groff". cvs devra créer tous les répertoires vides avant de les élaguer.

Donc, dans `/usr/src`:

```
% makedir /usr/obj
% make obj
```

Vous êtes maintenant prêt à faire une mise à jour cvs. Tapez toutefois **make cleandir** dans `/usr/src` avant de commencer, c’est plus facile et demande moins de travail que la méthode des répertoires objet. CVS est plus rapide que sup mais je ne sais pas combien de temps peut prendre la récupération de toutes les sources. Je ne l’ai testé que sur des ligne T1 ou plus rapides et cela prend un peu plus d’une heure suivant, évidemment, la qualité instantanée de la connexion. Je n’ai aucune expérience de rapatriement par modem mais, dans ce cas, vous demanderez de compresser et décompresser les données une fois transférées. Pour cela, tapez :

```
% cvs -z5 checkout .........
```

ou

```
% cvs -z5 -d $CVSROOT update ......
```

Le 5 définit le niveau de compression. Vous pouvez demander un chiffre entre 1 et 9 où 1 utilise la méthode la plus rapide et 9 la meilleure mais plus lente. Gardez à l'esprit que cette requète occasionne une surcharge du serveur cvs.

> **Note :** Ensuite, allez dans `/usr/src/sys/arch/$arch/compile/$kernel_conf_name` et tapez **make clean**. Effacez ce répertoire avant d'effectuer une mise à jour cvs afin d'éviter qu'il soit non-propre par la suite. Assurez-vous que les sources du noyau passent correctement durant la mise à jour cvs.

17.2. Récupérer pkgsrc

Pkgsrc ("package source" ou sources des paquetages) est un ensemble de logiciels et de bibliothèques portés pour NetBSD. C'est une façon très simple d'installer et de désinstaller des logiciels sur votre système NetBSD : il récupère les fichiers requis, corrige les sources si nécessaire, les configure, compile les binaires puis les installe avec leurs pages de manuel. Une base de données des logiciels installés est tenue à jour qui sait exactement quels fichiers appartient à quel paquetage et où ce dernier se trouve.

Pour récupérer pkgsrc à partir de zéro :

```
% setenv CVSROOT :pserver:anoncvs@anoncvs.netbsd.org:/cvsroot
% cd /usr
% cvs login
(the password is: "anoncvs")
% cvs checkout -PA pkgsrc
```

Ou par ssh:

```
% setenv CVS_RSH ssh
% setenv CVSROOT anoncvs@anoncvs.netbsd.org:/cvsroot
% cd /usr
% cvs checkout -PA pkgsrc
```

Le répertoire `pkgsrc` est créé dans votre `/usr` et toutes les sources des paquetages seront stockées sous `/usr/pkgsrc`.

Pour mettre pkgsrc à jour :

```
% setenv CVSROOT :pserver:anoncvs@anoncvs.netbsd.org:/cvsroot
% cd /usr
% cvs login
(the password is: "anoncvs")
% cvs -d $CVSROOT update -PAd pkgsrc
```

Ou par ssh:

```
% setenv CVS_RSH ssh
```

```
% setenv CVSROOT anoncvs@anoncvs.netbsd.org:/cvsroot
% cd /usr
% cvs -d $CVSROOT update -PAd pkgsrc
```

Assurez-vous bien que le répertoire pkgsrc est propre lorsque vous commencez la mise à jour. Pour cela, tapez **make clean** dans `/usr/pkgsrc` si vous n'êtes pas certain.

Chapitre 18. Opérations diverses

Ce chapitre rassemble, comme ils viennent, les divers sujets qui n'ont pas trouvé leur place dans les précédents.

18.1. Créer une disquette d'installation pour i386

Cette section (Créer une disquette d'installation pour i386) est une contribution de Eric Delcamp

Avant toute chose, il vous faut avoir un noyau dont le pseudo-périférique *vnd* est actif (c'est le cas du noyau GENERIC).

1. Tout d'abord, créons un noyau valide pour le placer sur la disquette. Appelons-le DISQUETTE. Il doit être dérivé d'un modèle INSTALL. Après la compilation, vous obtenez un fichier `/sys/arch/i386/compile/FLOPPY/netbsd`.
2. Allez dans le répertoire `/usr/src/distrib/i386/floppies/ramdisk` et tapez:

   ```
   # make
   ```

   afin de créer un fichier `ramdisk.fs` dans le répertoire.
3. De même, allez dans le répertoire `/usr/src/distrib/i386/floppies/fdset` et tapez:

   ```
   # make KERN=/sys/arch/i386/compile/FLOPPY/netbsd
   ```

   afin de créer, suivant la taille du noyau, un ou deux fichiers nommés `boot1.fs` et `boot2.fs`.
4. Transférez ces fichiers sur la ou les disquettes avec les commandes:

   ```
   # dd if=boot1.fs of=/dev/fd0a bs=36b
   # dd if=boot2.fs of=/dev/fd0a bs=36b
   ```

5. Mettez la première disquette dans le lecteur et allumez!

18.2. Graver un cédérom

Pour créer un cédérom de données, nous pouvons utiliser les programmes mkisofs et cdrecord: les graveur SCSI et IDE sont reconnus. Les lecteur IDE/ATAPI sont supportés par NetBSD sans recourir à une émulation car le pilote peut directement recevoir des commandes ATAPI, ce qui est une solution simple et élégante.

Deux étapes sont nécessaires pour graver un cédérom: premièrement établir l'image ISO du CD sur le disque dur avec le programme mkisofs. Deuxièmement, graver l'image sur le CD avec le programme cdrecord. Dans l'exemple suivant, nous allons utiliser un graveur IDE/ATAPI reconnu. Voici ce que donne dmesg:

```
cd1 at atapibus1 drive 0: <HP CD-Writer Plus 8100> type 5 cdrom removable
```

Note : lors de la gravure d'un cédérom, la vitesse d'exécution est critique: le flot de données doit être constant et ne supporter aucune interruption. Le tampon de données de cdrecord ne doit jamais être vide. Cela veut dire qu'il est préférable de graver un CD lorsque le système ne fait rien ou presque rien d'autre (ne recompilez pas un noyau ou n'encodez pas du MP3 pendant la gravure...)

18.2.1. Etablir une image ISO

Système de fichiers de cédérom

Un cédérom peut être gravé en utilisant des système de fichiers différents (et parfois incompatibles). Ainsi, un cédérom gravé sur un système peut ne pas être lu (ou avec perte de données) sur un autre système. Cette situation peut survenir, par exemple, lorsque vous lisez sous NetBSD un cédérom gravé sous Windows. Les paragraphes suivants donnent un bref aperçu des systèmes de fichiers les plus communs sur cédérom.

L'*ISO9660* est le premier format, apparu en 1988, et représente une sorte de dénominateur commun entre les différents systèmes d'exploitation: Apple, MS-DOS, Unix et VMS. Un cédérom gravé, dans ce format, sous windows possède quelques limitations: des noms sous la forme 8.3 et seulement 8 niveaux de sous-répertoires. Pour cette raison, si, par exemple, vous gravez un cédérom d'installation de NetBSD sous Windows, les noms de fichiers apparaitront normalement dans l'explorateur mais seront tronqués sur le CD.

Pour contourner cette limitation, des extensions ont été ajoutées au format ISO9660 original. L'inconvénient est qu'il n'est plus possible de partager un tel cédérom entre les plateformes.

Le format Joliet, introduit par Microsoft pour DOS/Windows, permet la gravure de noms longs et de plus de répertoires. Ce format n'est généralement pas reconnu par les systèmes Unix (il l'est sous NetBSD-current).

L'extension Rock Ridge a été introduite par les systèmes Unix pour prendre en comptes ses conventions propres sans perdre la compatibilité avec le format ISO9660 original. C'est le format à utiliser pour la gravure de cédéroms sous NetBSD et les autres Unix (dans les paragraphes suivants, nous verrons comment graver un cédérom compatible à la fois avec Joliet et Rock Ridge, avec le programme mkhybrid.

Il existe encore d'autres standards en plus de ceux-ci (par exemple *El Torito* qui est utilisé pour produire des cédéroms démarrables sur les PC récents).

Comme les images ISO ont tendance à être très volumineuses, mieux vaut préalablement vérifier si le disque dur possède suffisament de place libre (jusqu'à 700 Mo pour un image). Considérons que les données à graver se trouvent dans le répertoire `agraver`. Pour créer l'image, tapez:

```
# mkisofs -aflrTv -o image-cd agraver/
```

Lorsque le fichier `image-cd` est terminé, il peut être visualisé et parcouru comme tout système de fichier normal afin vérifier qu'il n'y a pas d'erreurs avant de lancer la gravure. Par exemple:

```
# ls -l image-cd
```

```
-rw-rw-r--  1 auser     user  284672 Dec  1 11:58 image-cd
# vnconfig -v vnd0 image-cd 512/556/1/1
# mount -r -t cd9660 /dev/vnd0c /mnt
  ... browsing su /mnt ...
# umount /mnt
# vnconfig -u vnd0
```

La valeur 556 est la taille du cédérom divisée par 512.

> **Note :** pour graver un cédérom qui soit aussi lisible par les systèmes Microsoft, utilisez le programme mkhybrid avec l'option `-J` au lieu de mkisofs. Par exemple:
>
> ```
> mkhybrid -a -r -J -v -o cd.iso -b NetBSD-1.4.2/i386/installation/floppy/boot-big.fs ./
> ```

18.2.2. Graver l'image sur le CD

Dans la seconde étape, l'image est gravée sur le CD avec la commande:

```
# cdrecord -v speed=2 dev=/dev/rcd1d image-cd
```

> **Note :** pour les graveur ATAPI, utilisez *rcd#d* car le périférique *a* ne comprend pas les commandes ATAPI.

Avant de graver l'image, il est possible de lancer un test en désactivant le laser: ajoutez les options `-dummy` et `-nofix` sur la ligne de commandes. Par exemple:

```
# cdrecord -v -dummy -nofix speed=2 dev=/dev/rcd1d image-cd
```

Les deux étapes, création puis gravure de l'image, peuvent être combinées en une seule commande, sans avoir besoin de passer par un (gros) fichier temporaire sur le disque dur. La commande ressemble à:

```
# (nice -18 mkisofs -aflrT agraver/) | cdrecord -v fs=16m speed=2 dev=/dev/rcd1d -
```

L'option `fs=16m` est utilisée pour allouer un *fifo* plus gros. Cela permet à cdrecord d'éviter les erreurs dues à un flux trop réduit (c'est à dire qu'il n'a pas assez de données à graver).

18.2.3. Dupliquer un cédérom

Pour directement copier un CD, l'option `-isosize` doit être donnée à cdrecord. Par exemple:

```
# cdrecord -v fs=16m -isosize speed=2 dev=/dev/rcd1d /dev/rcd0d
```

Note : si à la fois le lecteur et le graveur sont IDE/ATAPI, il est préférable qu'ils soient connectés à deux contrôleurs différents (un sur le primaire, l'autre sur le secondaire) pour améliorer le flux de données. Voici un exemple de configuration:

```
wd0: disque dur, IDE primaire maitre
cd0: lecteur CD, IDE primaire esclave
cd1: graveur CD, IDE secondaire maitre
```

18.2.4. Graver un cédérom démarrable

Que le cédérom soit démarrable ou non ne dépend que de la présence ou non d'un fichier binaire de démarrage sur celui-ci: ce fichier simule une disquette. L'option `-b` de mkisofs et mkhybrid est utilisable dans ce but. Par exemple:

```
# mkisofs -avr -b boot.fs -o cdimage agraver/
```

`boot.fs` est le binaire de démarrage pour le cédérom. Notez que le chemin d'accès à `boot.fs` est relatif par rapport au répertoire `agraver/`.

18.3. Synchroniser l'horloge système

Il n'est pas rare de constater que l'horloge système est déréglée de quelques minutes. Pour d'étranges raisons, il semble que l'horloge interne des ordinateurs ne soit pas très précise. Le problème devient gènant si vous administrez de nombreuses machines: la synchronisation des horloges peut facilement devenir un cauchemar. Pour le résoudre, le protocole NTP (version 3) vous vient en aide. Il sert à synchroniser les horloges d'un réseau de stations de travail à l'aide d'un ou plusieurs serveurs NTP.

Grâce au protocole NTP, il est possible de règler l'horloge d'une station mais aussi synchroniser un réseau entier. Ce protocole est assez complexe puisqu'il définit une hiérarchie de serveurs maitre-esclave divisés en strates: la plus haute partie de la hiérarchie est occupée par les serveurs de strate 1 connectés à une horloge externe (horloge radio, par exemple) pour garantir une grande précision. Ensuite, les serveurs de strate 2 se synchronisent sur les serveurs de strate 1 et ainsi de suite. La précision diminue au fur et à mesure des niveaux. La structure hiérarchique permet d'éviter les engorgements que pourraient causer les nombreux clients qui se connecteraient aux mêmes serveurs (peu nombreux) de la strate 1. Par exemple, si vous voulez synchroniser un réseau entier, vous ne connectez pas toutes les machines au même serveur public de strate 1: vous mettez en place un serveur qui se connecte au serveur public et toutes les machines se synchronisent sur ce serveur local.

Heureusement, pour utiliser les outils NTP, nul n'est besoin de comprendre les détails du protocole et de ses implémentations (cependant, si vous êtes intéressés, consultez le RFC 1305) mais simplement la configuration et le lancement de quelques programmes. Le système de base de NetBSD contient tous les outils nécessaires au protocole (ainsi que d'autre protocoles liés, comme nous le verrons) dérivé de l'implémentation xntp. Cette section décrit une façon simple d'avoir toujours l'heure exacte.

Tout d'abord, il est nécessaire de trouver l'adresse des serveurs NTP publics à utiliser comme référence. Une liste détaillée se trouve dans la page http://www.eecis.udel.edu/~mills/ntp/servers.html. Par exemple, pour l'Italie, les deux serveur de strate 1, tempo.cstv.to.cnr.it et time.ien.it, peuvent être utilisés (NDT: pour la France, vous avez, par exemple, ntp-p1.obspm.fr ou ntp-sop.inria.fr).

Ensuite, sous root, pour règler l'horloge système, lancez la commande:

```
# ntpdate -b tempo.cstv.to.cnr.it time.ien.it
```

(remplacez le nom des serveurs de cet exemple par ceux que vous voulez utiliser). L'option `-b` demande à **ntpdate** de règler l'heure à l'aide de l'appel système settimeofday au lieu de adjtime par défaut. Elle est conseillée si le décalage avec l'heure correcte est considérable.

Comme vous le voyez, ntpdate n'est pas dificile à utiliser. L'étape suivante consiste à le lancer automatiquement afin de toujours avoir l'heure exacte. Si vous avez une connexion permanente à internet, vous pouvez lancer le programme au démarrage en ajoutant la ligne suivante dans `/etc/rc.conf`:

```
ntpdate=YES      ntpdate_hosts="time.ien.it"
```

Le nom du serveur à utiliser est spécifié par la variable ntpdate_hosts. Si vous laissez le champ vide, le script de démarrage tentera d'extraire le nom du fichier `/etc/ntp.conf`.

Si vous n'avez pas de connexion permanente à internet (par exemple une connexion par ligne téléphonique chez un FAI), vous pouvez lancer ntpdate depuis le script `ip-up` que nous avons mis en place au Chapitre 9. Ajoutez-lui la ligne:

```
/usr/sbin/ntpdate -s -b time.ien.it
```

(Le chemin d'accès est obligatoire sinon le script ne trouvera sans doutes pas le programme.) L'option `-s` détourne l'affichage de la sortie standard (par défaut) vers syslogd(3). Les messages de ntpdate seront écrits dans le fichier `/var/log/messages`.

Il existe d'autres commandes utiles en plus de ntpdate. Il est aussi possible de transformer l'une des machines du réseau en serveur NTP pour le reste des ordinateurs. Il synchronisera son horloge avec celle d'un serveur public. Pour ce type de configuration, il vous faut utiliser le démon **xntpd** et créer un fichier `/etc/ntp.conf`. Par exemple:

```
server time.ien.it
server tempo.cstv.to.cnr.it
```

Xntpd peut être démarré à partir de `rc.conf` en utilisant l'option adéquate:

```
xntpd=YES
```

NTP n'est pas l'unique solution pour synchroniser les horloges d'un réseau: il existe aussi le démon timed, développé par 4.3BSD. Il utilise aussi une hiérarchie maitre-esclave: lorsqu'il démarre sur un ordinateur, timed demande l'heure réseau à un maitre et règle son horloge en fonction de la réponse. Une structure mélageant timed et xntpd peut être mise en place. Un des hôtes locaux se met à l'heure à l'aide d'un serveur NTP public et devient un maitre timed auquel se synchroniseront les autres ordinateurs du réseau, par timed. Cela veut dire que le serveur local doit faire tourner à la fois NTP et timed. Prenez bien soin de ne pas les faire interférer (timed doit être démarré avec l'option `-F hostname` pour l'empêcher de règler l'horloge).

18.4. Installer le gestionnaire de démarrage

Sysinst, le programme d'installation de NetBSD, peut mettre en place le gestionnaire de démarrage de NetBSD sur le disque dur. Il peut aussi être installé ou configuré, plus tard au besoin, avec la commande **fdisk**. Par exemple:

```
# fdisk -B wd0
```

Si NetBSD ne démarre pas depuis le disque dur, vous pouvez démarrer depuis la disquette d'installation puis lancer le noyau du disque dur. Insérez la disquette et, à l'invite de démarrage, tapez:

```
> boot wd0a:netbsd
```

Cela permet de lancer le noyau du disque dur (utilisez le périférique correct, comme par exemple, sd0a pour un disque SCSI).

> **Note :** parfois, **fdisk -B** ne donne pas le résultat attendu (en tout cas, ça m'est arrivé). C'est sans doutes le cas si vous avez installé ou désinstallé d'autres systèmes d'exploitation, comme Windows 95. Essayez alors de lancer **fdisk /mbr** depuis DOS puis de renouveler **fdisk** depuis NetBSD.

18.5. Effacer le disklabel

Bien que ça ne soit pas une opération que vous avez besoin de faire fréquemment, il peut être utile de savoir comment faire. Soyez exactement sûr de ce que vous faites avant d'effectuer ce type de traitement. Par exemple:

```
# dd if=/dev/zero of=/dev/rwd0c bs=8k count=1
```

Cette commande efface le disklabel (pas la table de partitions MBR). Pour entièrement effacer le disque, utilisez le périférique `wd0d`. Par exemple:

```
# dd if=/dev/zero of=/dev/rwd0d bs=8k
```

18.6. Haut-parleurs

J'ai trouvé ce truc sur une liste de diffusion (j'en ai oublié l'auteur). Pour émettre un son sur le haut parleur (par exemple à la fin d'un long traitement), le périférique *spkr* peut être utilisé. Il correspond à `/dev/speaker`. Par exemple:

```
echo 'BPBPBPBPBP' > /dev/speaker
```

> **Note :** le périférique *spkr* n'est pas présent sur le noyau générique. Il vous faut un noyau personnalisé.

18.7. Perdu votre mot de passe ?

Si vous avez oublié le mot de passe "root", tout n'est pas perdu, vous pouvez toujours retrouver le système en suivant les étapes: démarrer en mono-utilisateur, monter / et changer le mot de passe • root• . En détail:

1. Démarrer en mono-utilisateur: lorsque l'invite de démarrage apparait et que le compte à rebours de cinq secondes démarre, tapez la commande suivante:

   ```
   > boot -s
   ```

2. A l'invite suivante,

   ```
   Enter pathname of shell or RETURN for sh:
   ```

   appuyez sur Entrée.

3. Tapez les commandes suivantes:

   ```
   # fsck -y /
   # mount -u /
   # fsck -y /usr
   # mount /usr
   ```

4. Changez le mot de passe "root" avec **passwd**.
5. Utilisez la commande **exit** pour passer en mode multi-utilisateur.

18.8. Ajouter un nouveau disque dur

Cette section décrit la façon d'ajouter un disque dur à un système NetBSD existant. Dans l'exemple suivant, un nouveau contrôleur SCSI et un nouveau disque dur connecté dessus sont ajoutés. Si vous n'avez pas besoin d'ajouter de contrôleur, allez direcement à la partie concernant le disque dur. L'installation d'un disque IDE est identique, seul change le nom du périférique (`wd#` au lieu de `sd#`).

Comme toujours, avant d'acheter du nouveau matériel, consultez la liste des compatibilités avec NetBSD et soyez sûr que le périférique puisse être reconnu par le système.

Une fois que le contrôleur SCSI est physiquement branché et le disque dur connecté, redémarrez votre ordinateur puis vérifiez s'ils sont correctement détectés avec la commande **dmesg**. Voici un extrait concernant un contrôleur NCR-875:

```
ncr0 at pci0 dev 15 function 0: ncr 53c875 fast20 wide scsi
```

```
ncr0: interrupting at irq 10
ncr0: minsync=12, maxsync=137, maxoffs=16, 128 dwords burst, large dma fifo
ncr0: single-ended, open drain IRQ driver, using on-chip SRAM
ncr0: restart (scsi reset).
scsibus0 at ncr0: 16 targets, 8 luns per target
sd0(ncr0:2:0): 20.0 MB/s (50 ns, offset 15)
sd0: 2063MB, 8188 cyl, 3 head, 172 sec, 512 bytes/sect x 4226725 sectors
```

Si le périférique n'apparait pas dans l'affichage, vérifiez qu'il soit supporté par le noyau que vous utilisez. Si nécessaire, recompilez un noyau personnalisé (voyez le Chapitre 7).

Nous pouvons maintenant partitionner le disque à l'aide de **fdisk**. En premier lieu, vérifiez l'état actuel du disque:

```
# fdisk sd0
NetBSD disklabel disk geometry:
cylinders: 8188 heads: 3 sectors/track: 172 (516 sectors/cylinder)

BIOS disk geometry:
cylinders: 524 heads: 128 sectors/track: 63 (8064 sectors/cylinder)

Partition table:
0: sysid 6 (Primary 'big' DOS, 16-bit FAT (> 32MB))
    start 63, size 4225473 (2063 MB), flag 0x0
        beg: cylinder    0, head   1, sector  1
        end: cylinder  523, head 127, sector 63
1: <UNUSED>
2: <UNUSED>
3: <UNUSED>
```

Dans cet exemple, le disque contient déjà une partition DOS qui sera effacée et remplacée par une partition NetBSD. La commane **fdisk -u sd0** permet de modifier interractivement les partitions. Les données modifiées ne seront écrites sur le disque que lorsque fdisk quittera et il vous demandera confirmation avant de faire quoi que ce soit. Vous pouvez donc travailler tranquillement.

Géométrie des disques

La géométrie du disque rapportée par fdisk peut paraitre confuse. Dmesg indique 4226725 secteurs avec 8188/3/172 pour C/T/S mais 8188*3*172 donne 4225008 et non 4226725. En fait, les disques modernes n'ont pas de géométrie fixée. Le nombre de secteurs par piste change suivant le cylindre. Le seul paramètre intéressant est donc le nombre de secteurs. Le disque indique des valeurs C/T/S concernant une géométrie fictive: la valeur 172 est le résultat du nombre total de secteurs (4226725) divisé par 8188 puis par 3.

Pour rendre les choses encore plus confuses, le BIOS utilise une autre • fausse• géométrie (C/T/S 524/128/63) qui donne un total de 4225536, qui s'approche déja plus des 425008 réels. Pour partitionner le disque, nous utiliseront la géométrie BIOS pour maintenir la compatibilité avec les autres systèmes d'exploitation. Nous allons donc perdre quelques secteurs (4226725 - 4225536 = 1189 secteurs = 594 Ko).

Pour créer une partition BIOS, utilisez la commande **fdisk -u**. Voici ce que cela donne:

```
Partition table:
0: sysid 169 (NetBSD)
    start 63, size 4225473 (2063 MB), flag 0x0
        beg: cylinder    0, head   1, sector  1
        end: cylinder  523, head 127, sector 63
1: <UNUSED>
2: <UNUSED>
3: <UNUSED>
```

Maintenant, créons le disklabel pour NetBSD. Les étapes à suivre sont:

```
# disklabel sd0 > fichiertemp
# vi fichiertemp
# disklabel -R -r sd0 fichiertemp
```

Si vous essayez de directement créer le disklabel avec

```
# disklabel -e sd0
```

vous aurez le message suivant:

```
disklabel: ioctl DIOCWDINFO: No disk label on disk;
use "disklabel -r" to install initial label
```

car celui-ci n'existe pas encore sur le disque.

Nous allons maintenant créer le disklabel en éditant le fichier `fichiertemp`, comme déjà expliqué. Le résultat est:

```
#       size   offset    fstype [fsize bsize   cpg]
a:   2048004       63    4.2BSD   1024  8192    16 # (Cyl.   0*- 3969*)
c:   4226662       63    unused      0     0       # (Cyl.   0*- 8191*)
d:   4226725        0    unused      0     0       # (Cyl.   0 - 8191*)
e:   2178658  2048067    4.2BSD   1024  8192    16 # (Cyl.   3969*- 8191*)
```

Note : lorsque le disklabel est créé, il est possible de l'optimiser en s'aidant de la sortie de la commande **newfs -N /dev/sd0a**. Elle avertit en cas de présence de secteurs inutilisés en fin de partition. Les valeurs rapportées peuvent être utilisées pour ajuster la taille des partition en recommençant le processus.

Enfin, nous allons créer le système de fichiers des partitions nouvellement définies (*a* et *e*).

```
# newfs /dev/sd0a
# newfs /dev/sd0e
```

Le disque est maintenant prêt à être utilisé et deux partitions peuvent être montées. Par exemple:

```
# mount /dev/sd0a /mnt
```

18.9. Fichier de mots de passe occupé ?

Si vous essayez de changer de mot de passe et que vous obtenez le mystérieux message: "Password file is busy" (le fichier de mots de passe est occupé), cela veut certainement dire que le fichier `/etc/ptmp` n'a pas été effacé du système. Il contient une copie temporaire du fichier `/etc/master.passwd`. Vérifiez que vous ne perdez pas d'informations importantes et effacez-le (`ptmp`, pas `master.passwd`).

> **Note :** si le fichier `/etc/ptmp` existe, vous pouvez aussi recevoir un avertissement au démarrage du système. Par exemple:
>
> ```
> root: password file may be incorrect - /etc/ptmp exists
> ```

Annexe A. Informations

A.1. Histoire de ce guide

Ce guide est né d’une série d’annotations devenuent de plus importantes qui me servaient de pense-bête. Elles étaient originellement écrites au format troff en utilisant les macros *ms*. Avec le temps, je me suis rendu compte qu’elles pouvaient constituer une aide à l’utilisateur de NetBSD. J’ai donc décidé de les mettre sur mon site une fois que j’aurais trouvé l’opportunité de les mettre en forme. J’ai alors choisi d’adopter le format *SGML/DocBook*.

Le présent document est écrit avec l’éditeur vi (voir Section 13.2) et quelques simples macros. La version HTML a été réalisée à partir des sources *SGML/DocBook* à l’aide du programme jade et des *feuilles de style modulables* de Norman Walsh.

Annexe B. SGML/DocBook

Annexe C. Remerciements

Certaines personnes m'ont aidé dans le développement de ce guide, j'aimerais, ici, les remercier et leur donner crédit. Elles m'ont apporté leurs contributions ou envoyé des suggestions. Elles ont relu le guide ou aidé à la mise en place des programmes nécessaires à la production de ce guide.

En particulier, je voudrais remercier :

- Paulo Aukar
- Manolo De Santis
- Eric Delcamp
- Hubert Feyrer
- Jason R. Fink
- Reinoud Koornstra (Mipam)
- David Magda
- Guillain Seuillot
- Wojciech Puchar
- Oliver Tonnhofer

Si j'ai oublié d'inscrire votre nom dans cette liste, veuillez me le faire savoir.