Le Guide du ROOTard pour Linux

Eric Dumas, Eric.Dumas@freenix.fr

Version 2.5, 14 Décembre 1996.

Table des matières

1 Préambule **6**

1.1 Histoire de ce guide . . . 6
1.2 Présentation du contenu . . . 6
1.3 Quoi de neuf? . . . 7
1.4 Copyright . . . 7
1.5 Sites officiels de distribution . . . 7
1.6 Evolution du format de ce guide . . . 8

2 Introduction **8**

3 Feuille d'Information Linux (v2.31) **9**

3.1 Résumé . . . 9
3.2 Linux - Introduction . . . 9
3.3 Caractéristiques de Linux . . . 11
3.4 Matériel . . . 13
3.4.1 Configuration minimale . . . 13
3.4.2 Configuration recommandée . . . 13
3.4.3 Matériel pris en charge . . . 14
3.5 Liste incomplète de programmes disponibles . . . 15
3.6 Qui utilise Linux? . . . 16
3.7 Documentation . . . 17
3.8 Comment se procurer Linux en France . . . 17
3.8.1 Par l'Internet . . . 17
3.8.2 Par téléphone . . . 18
3.8.3 Sur CD-ROM par correspondance . . . 19
3.9 Comment débuter . . . 19

3.10 Comment se tenir au courant . . . 21
3.11 The Linux Journal . . . 22
3.12 Statut légal de Linux . . . 22
3.13 Ce document . . . 23

4 Introduction à l'installation 23

4.1 Récupération . . . 23
4.2 Un mot sur le format des binaires . . . 24
4.3 Les différentes possibilités d'installations . . . 24

5 Distributions 25

5.1 Introduction . . . 25
5.2 Disquettes d'amorçages . . . 26
5.3 Slackware : description . . . 27
5.3.1 Disquettes Boot et Root . . . 27
5.3.2 La distribution . . . 29
5.3.3 Bogues de la Slackware . . . 40
5.4 RedHat : description . . . 40
5.4.1 Disquette de boot . . . 40
5.4.2 Distribution . . . 40
5.4.3 Utilisation du programme `rpm` . . . 55

6 Installation 57

6.1 Installation initiale . . . 57
6.2 Partitionnement du disque . . . 57
6.3 Installation avec la Slackware . . . 58
6.4 Installation avec la RedHat . . . 59

7 Administration sous Linux 59

7.1 Clavier Français . . . 59
7.2 LILO . . . 59
7.3 Montage automatique de partitions : /etc/fstab . . . 61
7.4 Montage manuel de partitions . . . 62
7.5 Mise en place des quota . . . 63

7.5.1 Logiciels nécessaires . . . 63
7.5.2 Compilation des utilitaires . . . 63
7.5.3 Activation du support des quota dans le noyau . . . 63
7.5.4 Activation des quota sur les systèmes de fichiers . . . 63
7.5.5 Attribution d'un quota à un utilisateur . . . 64
7.5.6 Statistiques sur les quota . . . 65
7.5.7 Autres documentations sur les quota . . . 65
7.6 mtools . . . 65
7.7 Swap : création de la partition . . . 66
7.8 Swap : fichier . . . 66
7.9 Ajouter un utilisateur, un groupe . . . 66
7.10 Lancements de programmes personnels lors du boot : /etc/rc.d/rc.local . . . 67
7.11 Ordonnancement de travaux : la crontab . . . 68
7.12 Imprimer ! . . . 69
7.13 Timezone . . . 71
7.14 Nouveaux changement d'heure . . . 72
7.15 locate . . . 73
7.16 Un second contrôleur IDE . . . 73
7.17 CD-ROM ATAPI . . . 73
7.18 Mettre une console externe comme console . . . 74
7.19 Deux cartes réseaux . . . 74
7.20 Les fichiers d'initialisation des interpréteurs de commandes . . . 74
7.21 Les sauvegardes sous Linux . . . 75
7.21.1 dump et restore . . . 76
7.21.2 tar . . . 77
7.22 Num Lock au démarrage . . . 77
7.23 Les fichiers core . . . 77
7.24 Accounting - lastcomm . . . 77

8 X Window **78**

8.1 XF86Config . . . 78
8.2 Clavier français sous X : Xmodmap . . . 82
8.3 Clavier français sous X... 2ième possibilité . . . 86

8.4 xdm . . . 86
8.5 Comment démarrer plusieurs sessions X? . . . 87
8.6 Comment récupérer les erreurs sur la console? . . . 87
8.7 XFree et XInput . . . 87
8.8 Le fichier .Xauthority . . . 89
8.9 Problème assez courant . . . 89

9 Sécurité 89

9.1 Présentation . . . 89
9.2 Problèmes de sécurité . . . 90
9.3 Protection de la machine contre l'extérieur . . . 90
9.4 Quelques pointeurs . . . 90

10 Recompiler le noyau 90

10.1 Pentium . . . 90
10.2 Compilation . . . 91

11 Les fichiers spéciaux 96

12 Communiquer 112

12.1 UUCP et modem . . . 112
12.2 Appels entrants . . . 113
12.3 PPP . . . 113
12.3.1 Introduction . . . 114
12.3.2 Accès à un serveur PPP . . . 114
12.3.3 Configuration d'un serveur PPP . . . 117
12.3.4 Bibliographie . . . 117
12.4 Un réseau derrière une seule adresse . . . 117
12.5 Sendmail . . . 118
12.6 Configuration du mail lors d'une connexion avec un provider . . . 119

13 ELF 120

13.1 Introduction . . . 120
13.2 Installation . . . 121

13.3 Migration vers un système tout-ELF . . . 122
13.4 Utilisation des utilitaires ELF . . . 123
13.5 Sources d'information sur ELF . . . 123

14 Catastrophe ! 123

14.1 Impossible de booter ! . . . 124
14.2 Formattage accidentel d'une partition . . . 124
14.3 Problèmes avec le disque dur : super bloc . . . 125
14.4 Problèmes avec le disque dur : blocs défectueux . . . 125
14.5 fatal signal 11 . . . 126

15 Divers 126

15.1 Comment se faire une disquette de Boot . . . 126
15.2 Mon mot de passe, où est mon mot de passe? . . . 126
15.3 Comment limiter le reboot en single user? . . . 127
15.4 Fenêtres virtuelles . . . 127
15.5 Francisation . . . 128
15.5.1 locale // nls . . . 128
15.5.2 ispell . . . 130
15.5.3 Les accents sous bash . . . 130
15.5.4 less . . . 131
15.5.5 Formatage de textes . . . 131
15.5.6 elm . . . 131
15.6 TeX et les polices de caractères . . . 131
15.7 Une carte son bien configurée . . . 132
15.8 Bug gcc . . . 133
15.9 Emacs . . . 134
15.10 Vérouillage des ports série . . . 136
15.11 Minitel . . . 137
15.12 Fax . . . 137
15.13 Modules Chargeables . . . 137
15.13.1 Structure du noyau . . . 137
15.13.2 Qu'est-ce qu'un module chargeable? . . . 137

15.13.3 Compilation du noyau . . . 138
15.13.4 Chargement : méthode manuelle . . . 139
15.13.5 Chargement automatique : `kerneld` . . . 139

16 Pour aller plus loin **142**
16.1 « Bien débuter sous **Linux** » (V 2.2.2) . . . 142
16.2 Livres . . . 143
16.3 Les HowTo's . . . 143

17 Adresses **144**

18 Contributions **145**

19 Si tu veux participer à Linux **146**

20 That's all folks **146**

1 Préambule

Et oui, voici la version 2.5 du Guide du ROOTard. Cette version est quelque peu particulière : c'est la dernière que j'écrit en tant qu'étudiant. Pourtant, j'ai fait ce que j'ai pu pour faire de longues études dans la bonne vieille ville (mais charmante) de Bordeaux... mais 5 années, ça suffit ! Il est fort probable que la prochaine version ne soit réalisée que dans quelques mois, pour cause de vacances forcées au frais de l'état (service militaire... et oui, certains n'y coupent pas).

Donc ne m'en veut pas si j'ai un peu de retard dans mon courrier. J'y réponds généralement toujours... mais les délais risquent être très longs (une bonne quinzaine de jours)...

1.1 Histoire de ce guide

Ce guide a initialement été conçu par Julien SIMON (Julien.Simon@freenix.fr), que je tiens à remercier pour le travail extraordinaire qu'il a réalisé. Bon nombre de passages de ce guide ont été empruntés et adaptés des versions précédentes. Le dernier chapitre indique les noms des personnes ayant contribuées à la réalisation de ce document. Au fil des versions, cette liste s'allonge : c'est bon signe.

1.2 Présentation du contenu

Ce guide a pour but de fournir des réponses claires et concrètes aux problèmes que peut poser l'installation de **Linux**. Il regroupe des trucs et astuces fournis par des utilisateurs, le plus souvent lues dans le newsgroup *fr.comp.os.linux* (f.c.o.l) ou dans les *HowTo*, documents indiquant comment installer, configurer ou

améliorer un élément logiciel ou matériel donné. Merci à tout ceux qui m'envoient des corrections, des mises à jours ou même des paragraphes entiers. C'est très sympatique, et c'est la bonne manière pour faire évoluer positivement ce guide.

Histoire de simplifier les choses, on se tutoie : c'est plus simple, et en plus, c'est une tradition sur l'Internet.

1.3 Quoi de neuf?

Cette version du Guide du ROOTard possède pas mal de nouveautés :

- description de la Redhat 4.0, dite « Colgate » ;
- description de la Slackware 3.1, dite « 96 » ;
- noyau 2.0.27 ;
- XFree 2.0 ;
- de nombreux compléments d'un point de vue réseau, sendmail, etc. Merci à Stéphane Bortzmeyer pour ses nombreux conseils et contributions.
- les modules chargeables (`kerneld`) ;
- et bon nombre d'autres choses...

De plus, j'ai intégré quelques articles provenant du Journal Linux français (`http://echo-linux.alienor.fr`).

Cette nouvelle version dépasse allègrement les 140 pages... chose que je n'aurais jamais imaginé. Cela commence à faire un guide assez important qui se veut destiné à n'importe qui.

1.4 Copyright

Le guide du ROOTard est *freeware* et je t'encourage donc vivement à le distribuer tel quel et gratuitement. Par contre, tu ne peux pas le modifier ou le vendre. J'insiste particulièrement sur le fait que tu ne peux pas tirer de bénéfice financier de ce document.

Aucune publication n'est autorisée sans l'accord préalable de l'auteur et des contributeurs, et il en va de même pour l'inscription sur CD-ROM, etc. Envoies-moi un courrier pour toute demande particulière.

1.5 Sites officiels de distribution

- FTP : `ftp.ibp.fr` dans le répertoire `/pub/linux/french/docs`.
- WWW : `http://www.freenix.fr/linux/Guide`

Tu peut également te procurer ce document sur des sites miroirs comme par exemple :

- ftp.calvacom.fr

- ftp.ese-metz.fr
- ftp.loria.fr
- ftp.switch.ch
- ftp.univ-angers.fr
- stef.u-picardie.fr
- lirftp.insa-rouen.fr
- sunsite.mff.cuni.cz

(*Nota : si j'en oublie, contactes-moi* !)

Il est disponible en plusieurs formats :

- texte simple : aucun accent, idéal pour commencer (mais avec une mise en page quelque peu hasardeuse) ;
- PostScript : le format le plus agréable à consulter ;
- HTML : à lire avec des outils tels que Mosaic, Netscape, lynx ou arena ;
- SGML avec accents normaux : il s'agit du code source original du Guide. Si tu souhaites m'envoyer un patch, tu peux utiliser directement ce fichier.

1.6 Evolution du format de ce guide

Depuis que j'ai repris le GRL (Guide du ROOTard pour Linux), je l'ai convertit au format *sgml*. Il s'agit du format dans lequel sont écrit les HowTo's ainsi que les documentations **Linux**. Il permet de générer le texte dans des formats aussi variés qu'*html*, *texte*, *dvi*, etc. Toutefois, ce format est très limité car, il ne permet pas l'inclusion de figures (et dans certains cas, cela serait très pratique), et surtout ne respecte pas la typographie française. De plus, les seules générations vraiment propres et lisibles sont les versions LaTeX (PostScript) et Html.

Je te conseille de lire plutôt la version en mode texte si tu n'as rien installé pour le moment, mais la lisibilité en est assez réduite.

Il est peut-être possible que la prochaine version sera complètement écrite en LaTeX, puisque les convertisseurs vers Html sont assez au point. Enfin bon, ce n'est qu'une idée personnelle. Si j'ai du temps à perdre un de ces jours...

2 Introduction

Si tu trouves des erreurs (g toujours été très phaché avec l'aurtografe :-)), des commentaires, des lacunes, ou juste pour me dire que ça t'a bien aidé, n'hésite pas à m'envoyer un mail à l'adresse suivante : `Eric.Dumas@freenix.fr`.

Tu peux toujours poster un article en français dans le groupe Usenet consacré à **Linux** : *fr.comp.os.linux* si tu as des problèmes, mais après avoir lu ce document !

Les conseils prodigués par le présent document n'engagent que ceux qui les suivent : si tu casses ta machine, si ton écran explose... c'est ta faute, pas la mienne !

Toutes les remarques citées appartiennent à leur propriétaire (Comme le dit Juju, « profond ça ! ») *et ne s'usent que si l'on s'en sert.*

Je suis bien évidement ouvert à toute suggestion permettant l'évolution, l'amélioration et la meilleur diffusion de ce document. N'hésites donc surtout pas à me contacter : c'est toujours un plaisir.

3 Feuille d'Information Linux (v2.31)

3.1 Résumé

Ce document fournit des informations de base sur le système d'exploitation **Linux**, notamment une description de **Linux**, une liste de ses fonctionnalités et de ses exigences, et quelques pointeurs sur des informations complémentaires.

3.2 Linux - Introduction

Linux est une libre implémentation totalement gratuite des spécifications POSIX, avec des extensions System V et Berkeley (ce qui signifie qu'il ressemble à Unix, mais ne provient pas du tout des mêmes sources), qui est disponible à la fois sous forme de code source et de binaires. Il est propriété de Linus B. Torvalds (`torvalds@cs.helsinki.fi`) ainsi que des autres contributeurs à ce projet et est librement diffusable selon les termes de la « GNU Public License ».

Linux ne se classe ni dans le domaine public, ni dans le *partagiciel* : il fait partie du « *logiciel libre* », appelé couramment « *graticiel* » ; vous pouvez en distribuer des copies mais vous devez fournir le code source correspondant ou au moins le rendre disponible de la même façon. Si vous y faites des modifications que vous diffusez, vous êtes légalement tenu de distribuer le code source de vos modifications. Consultez la « *licence publique générale GNU* » (GPL) pour plus de détails. Vous en trouverez une copie dans l'arborescence des sources de **Linux** (fichier nommé "COPYING") ou par FTP sur le site `prep.ai.mit.edu`.

Linux est libre et continuera de l'être. En raison de la nature du copyright GNU qui protège ce système, vouloir tout-à-coup changer cet état de fait serait illégal. Notez bien cependant qu'il est parfaitement autorisé de demander de l'argent pour distribuer **Linux**, du moment que vous fournissez aussi le code source. Si vous désirez plus de précisions sur ce sujet, lisez la GPL ; nous n'avons fait que résumer les grandes lignes de cette licence.

Linux fonctionne sur des machines 386/486/Pentium possédant un bus ISA, EISA ou PCI.

MCA (le bus propriétaire d'IBM) est pour l'instant mal pris en charge car il y a peu de documentation disponible sur ce dinosaure et peu d'intérêt de la part des développeurs **Linux**, mais il existe malgré tout des « patches » pour certaines machines. Si vous êtes intéressé, voyez :

`(http://www.undergrad.math.uwaterloo.ca/ cpbeaure/mca-linux.html)`

Un portage vers les plateformes basées sur les processeurs Motorola 680x0 (actuellement des Amiga et Atari) est en cours. Cette version fonctionne d'ores et déjà de manière satisfaisante et nécessite un 68020 avec un gestionnaire de mémoire externe (MMU), un 68030, un 68040, ou un 68060, ainsi qu'un coprocesseur arithmétique. Le réseau et X sont fournis.

Linux fonctionne également bien sur les processeurs Alpha de DEC. Les plateformes « Jensen », « NoName », « Cabriolet », « Universal Desktop Box » (plus connue sous le nom de « Multia »), et quelques autres.

Le portage de **Linux** vers Sparc progresse rapidement. **Linux** fonctionne sur la plupart des architectures Sparc, et la distribution Red Hat 4.0 inclut une version Sparc (en plus des versions x86 et AXP).

Les portages vers d'autres machines, dont MIPS, PowerPC, et PowerMAC, sont en cours à des stades divers. Ne soyez pas trop impatient, mais si vous être intéressé et capable de contribuer à ces portages, vous pouvez trouver d'autres développeurs qui souhaiteraient travailler avec vous.

Pour PowerMac, Apple et OSF ont effectué un portage de **Linux** au dessus du micro-noyau **Mach**. Le résultat, appelé **Mklinux**, est encore en béta-test, mais peut déjà être utilisé.

Linux n'est plus considéré comme un système en béta-test, depuis que la version 1.0 a été rendue disponible le 14 Mars 1994. Il existe toujours des bogues dans le système et de nouveaux bogues apparaîtront et seront corrigés au fur et à mesure du développement. Puisque **Linux** suit un modèle de développement ouvert, toutes les nouvelles versions seront accessibles au public, qu'elles soient considérées comme suffisament stables ou non. Cependant, afin d'aider les utilisateurs à déterminer si ue version donnée est ou non considérée comme stable, une convention de numérotation spéciale a été mise au point. Les versions x.y.z, où y est un nombre pair, sont stables, et seules des corrections de bogues seront appliquées lorsque z est incrémenté. Par exemple, entre les versions 1.2.2 et 1.2.3, il y a eu uniquement des corrections, et aucun ajout de fonctionnalités. Les versions x.y.z, où y est un nombre impair, sont des versions en béta-test destinées aux développeurs uniquement, peuvent être instables, et contiennent de nouvelles fonctionnalités qui sont ajoutées au cours du développement. De temps en temps, quand le développement du noyau se stabilise, un « gel » intervient pour fournir une nouvelle version « stable » (paire), et le développement continue sur une nouvelle version (impaire).

La version stable actuelle est 2.0.x (x changeant au fur et à mesure que de nouvelles corrections de bogues sont intégrées au noyau), et le développement devrait bientôt commencer sur des noyaux expérimentaux, numérotés 2.1.x. Si 2.0.x est trop récent pour vous, vous pouvez continuer d'utiliser la version 1.2.13 (version stable précédente).

Le noyau est très stable depuis longtemps et la majorité des versions récentes peut être utilisée 24H/24 pendant des mois sans aucun réamorçage, plantage ou erreur fatale. Certains sites ne réamorcent leurs machines **Linux** qu'à l'occasion d'un changement majeur dans le noyau.

Il faut garder à l'esprit que **Linux** est developpé selon un modèle ouvert et réparti, contrairement à la plupart des logiciels connus qui évoluent souvent selon un modèle fermé et centralisé. Ceci signifie que la version courante de développement est toujours publique (avec une ou deux semaines de retard) afin que tout le monde puisse l'utiliser. Une version apportant de nouvelles fonctionnalités contient par conséquent presque toujours des bogues, mais ceux-ci sont découverts et corrigés rapidement, souvent en quelques heures, car ceux qui y travaillent sont nombreux. Il est donc facile pour un utilisateur final de les éviter.

A l'opposé, le modèle fermé et centralisé signifie que seule une personne ou une équipe travaille sur le projet et qu'elle ne diffuse un programme que lorsqu'elle considère qu'il fonctionne bien. Ceci implique souvent de

longs intervalles entre les versions, de longs délais avant correction des bogues et un développement moins rapide. Bien sûr, la version la plus récente d'un programme réalisé ainsi est souvent de bonne qualité, mais le développement en est bien plus lent.

Au 1 Décembre 1996, la version stable de **Linux** est 2.0.27, et la version de développement est 2.1.14.

La version 2.0 est la référence stable destinée à servir de plate-forme fiable lors du développement de la version 2.1, qui va permettre d'ajouter de nouvelles possibilités et d'essayer des solutions audacieuses et modernes au coeur de **Linux** (IP v6 entre autres). Les versions 1.0 et 1.2 sont maintenant obsolètes. Une fois arrivée à maturité, cette version 2.1 donnera naissance à **Linux** 2.2 ou 3.0, et le jeu continuera de plus belle.

3.3 Caractéristiques de Linux

- Multi-tâches : exécute plusieurs programmes en pseudo-parallélisme.
- Multi-utilisateurs : plusieurs utilisateurs actifs sur la même machine en même temps (et sans licence multi-utilisateurs !).
- Multi plates-formes : il fonctionne sur différents processeurs, et pas seulement sur Intel.
- Exécution en mode protégé sur les processeurs x86.
- Protection de la mémoire entre les processus, afin qu'un programme ne puisse à lui seul compromettre le fonctionnement de l'ensemble du système.
- Chargement des exécutables à la demande : **Linux** ne lit sur le disque que les parties effectivement utilisées d'un programme.
- Partage des pages entre exécutables avec copie en écriture. Cela signifie que des processus multiples peuvent partager la même mémoire. Lorsque l'un d'eux tente d'y écrire, cette page (4Ko de mémoire sur processeurs x86) est copiée ailleurs. Ceci accroît les performances et réduit l'occupation mémoire.
- Mémoire virtuelle utilisant la pagination (et non pas le « swap » de processus entiers) sur disque, grâce à une partition, un fichier ou les deux, avec la possibilité d'ajouter et de retirer de nouvelles zones de swap dynamiquement. Jusqu'à 16 zones de 128 Mo peuvent être utilisées simultanément, soit un total de 2 Go de swap disponible.
- Réserve de mémoire commune aux programmes utilisateur et au cache disque (afin que toute la mémoire libre puisse être utilisée pour le cache qui ne sera réduit que lors de l'exécution de gros programmes) : **Linux** utilise au mieux à chaque instant TOUTE la mémoire disponible sur la machine.
- Bibliothèques partagées liées dynamiquement (DLL a.out, et ELF) ; des bibliothèques statiques sont bien entendu également disponibles.
- Sauvegarde d'image-mémoire (*core-dump*) lors du plantage d'un programme, à des fins d'analyse post-mortem effectuée grâce à un débogueur.
- Très conforme à POSIX, compatible System V et BSD au niveau programme source. Support des binaires COFF et ELF.

- Bonne compatibilité binaire avec SCO, SVR3 et SVR4, grâce à un module d'émulation conforme iBCS2.
- Tous les sources sont disponibles : le noyau, les pilotes, les outils de développement et tous les programmes utilisateur. De plus, ils sont librement diffusables. Il existe bien quelques applications commerciales pour **Linux** diffusées sans leurs sources, mais tout ce qui a été libre l'est toujours et le restera.
- Contrôle de processus POSIX (*job control*).
- Pseudo-terminaux (*pty's*).
- Émulation du coprocesseur dans le noyau afin qu'il ne soit pas nécessaire de l'inclure dans les programmes. Tout ordinateur exécutant **Linux** semble ainsi doté d'un coprocesseur mathématique. Bien entendu, si votre ordinateur comporte déjà un coprocesseur, **Linux** l'utilisera à la place de l'émulation. Vous pourrez recompiler votre noyau pour éliminer le code de l'émulateur (dont les fonctions seront prises en charge par le FPU), et gagner un peu de mémoire vive.
- De nombreux claviers nationaux ou personnalisés sont définis et il est assez facile d'en ajouter de nouveaux.
- Consoles virtuelles multiples : plusieurs sessions indépendantes accessibles par une combinaison de touches (indépendamment de la carte vidéo utilisée). Elles sont allouées dynamiquement et l'on peut en utiliser jusqu'à 63 simultanément.
- Plusieurs systèmes de fichiers reconnus comme `minix-1`, *Xenix*, et les plus courants de System V, en plus d'un système très perfectionné (`Ext2fs`), offrant des noms de 255 caractères et une taille pouvant atteindre 4 To.
- Accès transparent aux partitions MS-DOS (ou aux partitions FAT OS/2) via un système de fichiers spécial : vous n'avez pas besoin de commandes particulières pour utiliser une partition MS-DOS. Elle ressemble a une partition Unix (sauf pour la taille du nom des fichiers, les permissions d'accès, etc). Les partitions compressées de MS-DOS 6 ne sont pas encore reconnues mais il existe un *patch* (dmsdosfs). Le type VFAT (MS-Windows NT, MS-Windows 95) est pris en charge depuis la version 2.0.
- Système de fichiers spécial nommé UMSDOS, autorisant l'installation de **Linux** directement sur une partition MS-DOS (les performances sont bien sûr grevées mais c'est très utile pour une installation provisoire).
- Lecture seule du système de fichiers HPFS-2 de OS/2 V 2.1.
- Lecture seule du système de fichiers HFS (Macintosh) disponible sous forme de « patch ».
- Système de fichiers ISO 9660, qui permet de lire tous les CD-ROM conformes à ce standard quasi universel.
- TCP/IP, incluant tous les outils comme `ftp`, `telnet`, NFS, etc.
- Pile de protocoles réseau *Appletalk*.
- Client et serveur *Netware*.

- Client et serveur *Lan Manager* (SMB).
- X Window System (X11R6) sous la forme de XFree86 3.2, gérant l'essentiel des cartes vidéo et des souris disponibles sur PC (y compris la carte monochrome Hercules...).
- Gestion des principales cartes sonores.

3.4 Matériel

3.4.1 Configuration minimale

La configuration suivante est probablement la plus petite configuration possible sur laquelle **Linux** peut fonctionner : 386SX/16, 2 Mo RAM, un lecteur 1.44 Mo ou 1.2 Mo, n'importe quelle carte vidéo reconnue. Elle devrait vous permettre d'amorcer le système et de tester si **Linux** tourne sur votre machine, mais vous ne serez pas en mesure de faire quoi que ce soit d'utile. (NDR : 4 Mo sont quasiment indispensables.)

Pour y parvenir, vous aurez aussi besoin d'espace disque : 5 à 10 Mo devraient suffire pour un système minimal ne comportant que les commandes les plus importantes et peut-être une ou deux petites applications, comme par exemple un programme d'émulation de terminal. Ceci reste très, très limité et très inconfortable, puisqu'il ne reste pas assez de place pour faire quoi que ce soit, sauf grâce à de minuscules applications. Ceci n'est généralement pas recommandé, sauf pour tester si le noyau fonctionne et bien sûr pour pouvoir se vanter des faibles ressources matérielles nécessaires.

3.4.2 Configuration recommandée

Si vous souhaitez utiliser des programmes travaillant intensivement, comme **gcc**, **X** et **TeX**, vous voudrez probablement un processeur plus rapide qu'un 386SX/16, qui devrait cependant suffire si vous êtes patient.

En pratique, vous avez besoin d'au moins 4 Mo si vous n'utilisez pas X11 et d'au moins 8 Mo si vous l'utilisez. De plus, si vous voulez que soient simultanément actifs plusieurs utilisateurs ou plusieurs gros programmes (comme des compilations), plus de 4 Mo sont souhaitables. Tout fonctionnera avec moins de RAM (même avec 2 Mo), mais Linux utilisera alors la mémoire virtuelle (le disque dur servant de mémoire lente) et l'ensemble s'en trouvera considérablement ralenti, au point de s'avérer inutilisable.

La quantité d'espace disque nécessaire dépend des programmes que vous voulez installer. La série de base des utilitaires, shells et programmes d'administration devrait tenir sur un peu moins de 10 Mo, plus un peu de place pour les fichiers utilisateurs. Pour un système plus complet, Il est courant d'obtenir des valeurs de l'ordre de 20/30 Mo si vous n'utilisez pas X Window, et 40 Mo sinon (il n'est ici question que des binaires, les sources demeurent « nettement » plus encombrants). Ajoutez l'espace disque que vous voulez réserver aux utilisateurs. Etant donné le prix de la mémoire de masse de nos jours, si vous achetez une nouvelle machine, il est ridicule de l'équiper d'un disque dur de faible capacité. Prenez au moins 500 Mo, voire 1 Go ou plus, vous ne le regretterez pas.

Ajoutez plus de mémoire vive, de disque dur, un processeur plus rapide selon vos besoins, vos souhaits et votre budget pour aller au-delà d'un système à peine utilisable. En général, la grosse différence entre **Linux** et MS-DOS est qu'ajouter de la mémoire à **Linux** entraîne un gros gain de performances, ce qui n'est pas

vraiment le cas avec MS-DOS. Ceci est bien sûr en rapport avec la limite des 640 Ko directement exploitables sous MS-DOS, qui n'a aucun sens sous **Linux**.

Si vous trouvez votre système trop lent, AJOUTEZ DE LA MEMOIRE avant de dépenser des sommes folles pour un processeur plus puissant. Sur un 386, passer de 4 à 8 Mo de mémoire peut multiplier la vitesse d'exécution par un facteur de 10 ou 20...

3.4.3 Matériel pris en charge

Unités centrales :

toutes celles qui exécutent des programmes en mode protégé 386 (tous les modèles de 386, 486, 586 et 686) conviennent. Les processeurs 286 et inférieurs pourront peut-être être exploités un jour par un noyau plus petit appelé ELKS (*Embeddable Linux Kernel Subset*), mais n'espérez pas disposer alors des mêmes fonctionnalités. Une version pour les processeurs 680x0 (avec x=2 plus MMU externe, 3 et 4) pour Amiga et Atari se trouve dans le répertoire `680x0` des sites **Linux**. De nombreuses architectures Alpha sont compatibles avec Linux, ainsi que certaines machines Sparc. Des portages vers PowerPC, ARM et MIPS sont aussi en cours.

Architecture :

bus ISA et EISA. Le support du bus MCA est incomplet (voir plus haut). Les bus locaux (VLB et PCI) sont supportés. **Linux** exploite le matériel de manière plus fine que MS-DOS, MS-Windows, et les autres systèmes d'exploitation en général. Cela signifie que certains matériels marginaux qui conviennent pour d'autres systèmes d'exploitation peuvent se révéler non fiables sous **Linux**. **Linux** est, entre autres, un excellent testeur de mémoire...

RAM :

en théorie, jusqu'à un 1 Go; plus sur les plateformes 64 bits, mais cette quantité de mémoire n'a jamais été testé. Attention : certaines personnes (dont Linus Torvalds) ont remarqué qu'ajouter de la mémoire sans dans le même temps augmenter la quantité de cache peut ralentir la machine dans des proportions considérables. Au-delà de 64 Mo, il faut spécifier la quantité de mémoire en paramètre lors de l'amorçage car le BIOS des PC est limité et ne sait pas indiquer plus de 64 Mo.

Stockage des données :

les disques AT génériques (IDE, contrôleurs de disque dur 16 bit avec MFM ou RLL) sont reconnus, comme les disques durs SCSI et les CD-ROMs dotés d'une interface SCSI adéquate. Les contrôleurs génériques XT (contrôleurs 8 bits avec MFM ou RLL) nécessitent un pilote spécial intégré à la distribution standard du noyau. Les cartes SCSI supportées sont : Adaptec 1542, 1522, 1740, et séries 27xx et 29xx, les contrôleurs Buslogic (à l'exception du « Flashpoint »), les contrôleurs basés sur le circuit NCR53c810, les contrôleurs DPT, Seagate ST-01 et ST-02, les séries Future Domain TMC-88x (et toute carte basé sur le processeur TMC950) et TMC 1660/1680, Ultrastor 14F, 24F, et 34F, Western Digital wd7000, et d'autres. Les lecteurs de cartouches SCSI, QIC-02 et certains QIC-80 sont également supportés. Plusieurs lecteurs de CD-ROM sont également gérés, dont Matsushita/Panasonic, Mitsumi, Sony, Soundblaster, Toshiba, ATAPU (EIDE), et les lecteurs SCSI. Pour les références exactes des modèles supportés, consultez le « Hardware HOWTO ».

Vidéo :

les cartes VGA, EGA, CGA et Hercules fonctionnent en mode texte. Pour les modes graphiques, les cartes Hercules, EGA, VGA normales, quelques cartes super VGA (la plupart des cartes à base de puces ET-3000, ET-4000, Paradise et Trident), de nombreuses cartes S3 sont reconnues, ainsi que 8514/A, ATI MACH8, ATI MACH32. **Linux** utilise le serveur X **XFree86**, ce qui détermine quelles cartes fonctionnent. Les performances de certaines cartes accélérées sont impressionnantes.

N'achetez pas de carte vidéo de marque Matrox : ce constructeur a longtemps refusé de fournir les spécifications nécessaires à la programmation d'un pilote gratuit et en sources, permettant de les exploiter sous Linux. Elles ne sont, pour l'instant, supportées efficacement que par des versions commerciales de X Window (de marque « Metro-X » ou « Xinside »). Il existe, certes, un pilote pour ces cartes dans XFree86 3.2, mais il est encore minimal (il n'exploite pas toutes les fonctionnalités), et est considéré comme une version béta.

Réseau :

Western Digital 80x3, ne1000, ne2000, 3com503, 3com509, 3com589 PCMCIA, Allied Telliesis AT1500, la plupart des cartes LANCE, adaptateurs d-link pocket, SLIP, CSLIP, PLIP (IP sur ports parallèles), PPP, et bien d'autres... De nouveaux pilotes paraissent régulièrement.

Ports série :

la plupart des cartes utilisant des UART 16450 et 16550, dont entre autres les *AST Fourport*, *Usenet Serial Card II*, etc. Parmi les cartes série intelligentes reconnues, citons les gammes *Cyclades Cyclom*, *Comtrol Rocketport* et *Stallion* (toutes gammes supportées par leurs constructeurs), et *Digi* (certains modèles). Certains équipements RNIS, frame relay et pour ligne louée, sont également supportés.

Autres matériels :

Cartes Soundblaster, ProAudio Spectrum 16, Gravis Ultrasound, la plupart des autres cartes son, plusieurs types de souris bus (Microsoft, Logitech, PS/2).

3.5 Liste incomplète de programmes disponibles

La plupart des outils et des programmes classiques d'Unix ont été portés vers **Linux**, ce qui comprend presque tous les programmes GNU et de nombreux clients X provenant de diverses sources. En fait, le mot « portage » est souvent exagéré, puisqu'un grand nombre d'applications ont été compilées sans modification ou presque en raison de la forte conformité à POSIX de **Linux**. Malheureusement, il n'existe pour le moment qu'assez peu d'applications pour l'utilisateur final, mais cela a récemment commencé à changer. Voici néanmoins une liste très réduite de programmes qui fonctionnent sous **Linux**.

Commandes Unix de base :

ls, tr, sed, awk, etc ... Pratiquement tout y est.

Outils de développement :

gcc, gdb, make, bison, flex, perl, rcs, cvs, prof.

Langages et environnements :

C, C++, Objective C, Modula-3, Modula-2, Oberon, Ada95, Pascal, Fortran, ML, scheme, Tcl/tk, Perl, Python, Common Lisp, et bien d'autres.

Environnements graphiques :

X11R5 (XFree86 2.x), X11R6 (XFree86 3.x), MGR.

Editeurs :

GNU Emacs, XEmacs, MicroEmacs, jove, ez, epoch, elvis (GNU vi), vim (clone vi), vile, joe, pico, jed et bien d'autres.

Shells :

Bash (Bourne-shell compatible POSIX), zsh (incluant la compatibilité ksh), pdksh, tcsh, csh, rc, es, ash (Bourne-shell utilisé par BSD) et bien d'autres.

Télécommunications :

Taylor (compatible BNU) UUCP, SLIP, CSLIP, PPP, kermit, szrz (Zmodem), minicom, pcomm, xcomm, term/slap (exécute plusieurs shells, redirige les connexions réseau, et permet les affichages de fenêtres X-Window, sur une seule ligne téléphonique) Seyon (programme de communication sous X-Window) et de nombreux programmes de télécopie et de transmission de la voix. Bien entendu, les accès distants en mode terminal par ligne série sont supportés en standard.

News et mail :

C-news, INN, trn, nn, tin, sendmail, smail, elm, mh, pine, etc.

Formateurs de texte :

TeX, LaTeX, groff, doc, ez, Linuxdoc-SGML, etc.

Jeux :

Nethack, de nombreux jeux sous X11, dont DOOM. L'un des jeux les plus passionnants consiste à rechercher tous ceux qui sont disponibles sur les sites diffusant **Linux**...

Tous ces programmes (qui ne représentent pas le centième de ce qui existe) sont bien entendu disponibles gratuitement. Les applications commerciales commencent à exister, demandez à votre fournisseur si votre logiciel préféré a été porté sous **Linux**.

(NDR : une liste des programmes portés pour **Linux** est disponible. Elle est contenue dans le fichier LSM*.*, ou *Linux Software Map*)

3.6 Qui utilise Linux?

Linux est disponible gratuitement et il n'est demandé à personne d'enregistrer ses copies à une autorité quelconque ; aussi est-il très difficile de savoir combien de personnes dans le monde utilisent ce système d'exploitation. Plusieurs sociétés commerciales se consacrent exclusivement à la vente et au support de **Linux** (ce n'est pas interdit) mais leur clientèle ne constitue qu'une minorité des utilisateurs de Linux.

Les groupes **Linux** sur Usenet sont parmi les plus lus selon les statistiques, on pourrait donc penser que plusieurs centaines de milliers d'utilisateurs ou de personnes intéressées.

Toutefois, Harald T. Alvestrand a décidé d'essayer de compter tant bien que mal les utilisateurs de **Linux**. Il demande que vous postiez un courrier électronique à l'adresse `linux-counter@uninett.no`, avec l'une des phrases suivantes dans le SUJET du message: « I use Linux at home », « I use Linux at work », ou « I use Linux at home and at work ». Il compte aussi les votes contenant « I don't use Linux », au passage. Il diffuse périodiquement ces statistiques dans comp.os.linux.misc. Il est possible de remplir un formulaire plus détaillé, pour enregistrer des amis n'ayant pas la possibilité de poster du courrier électronique.

Ce compteur est fatalement assez pessimiste. Lors de la conférence **Linux** de Berlin de 1995, il a été annoncé une estimation à plus d'un million d'utilisateurs actifs dans le monde; sans autre précision quant au mode de détermination de ce nombre.

3.7 Documentation

Matt Welsh a écrit un guide de plus de 300 pages decrivant comment installer et configurer **Linux**. Il est disponible sur les principaux sites diffusant le système, sous la forme de sources LaTeX aussi bien que DVI, PostScript, et ASCII. (Cet ouvrage a été traduit en français, vous le trouverez sur les sites diffusant **Linux** en france en sources LaTeX, sortie DVI et PostScript sous le nom `install-guide-french-2.2.2.*`).

Il est aussi vendu par certaines personnes ou organismes sous forme imprimée. De même, le projet de documentation **Linux** (Linux Doc Project, LDP), diffuse plusieurs autres ouvrages dans un état d'avancement plus ou moins complet. Lisez le groupe comp.os.linux.announce, **Linux** est en passe d'être aussi bien documenté que ses équivalents commerciaux... Et ceci gratuitement bien sûr.

Dans le commerce, il faut noter la sortie de deux ouvrages en version française chez O'Reilly: « Le système Linux », guide de plus de 600 pages par Matt Welsh, et « Administration Réseau sous **Linux** », par Olaf Kirch. Les titres originaux de ces livres sont respectivement « Running Linux » et « Linux Network Administration Guide ». Ce dernier est également disponible en téléchargement gratuit, sous forme de son code source LaTeX (l'archive s'appelle `nag.french.eoit-1.0.tar.gz`).

La série des documents *HOWTO's*, et le *Guide du Rootard*, disponibles librement, sont quasiment indispensables, consultez la section « *comment débuter* » pour plus de détails à ce sujet.

3.8 Comment se procurer Linux en France

3.8.1 Par l'Internet

Si vous avez la chance d'accéder à une machine connectée à l'Internet, vous pouvez récupérer **Linux** facilement sur l'un de ces sites:

Nom du site		Adresse IP	Répertoire Linux
ftp.ibp.fr	(FRANCE)	132.227.77.2	/pub/linux
ftp.calvacom.fr	(FRANCE)	194.2.168.3	/pub/linux/slackware
ftp.cnam.fr	(FRANCE)	163.173.128.15	/pub/Atari/Linux68k

ftp.ese-metz.fr (FRANCE)	193.48.224.106	/pub/Linux
ftp.iut-bm.univ-fcomte.fr	193.52.61.33	/pub/linux
ftp.loria.fr (FRANCE)	152.81.10.10	/pub/linux
ftp.univ-angers.fr (FRANCE)	193.49.144.10	/pub/Linux
lirftp.insa-rouen.fr(FRANCE)	192.54.152.136	/pub/linux
stef.u-picardie.fr (FRANCE)	193.49.184.23	/pub/linux
tsx-11.mit.edu	18.172.1.2	/pub/linux
sunsite.unc.edu	152.2.22.81	/pub/Linux
ftp.funet.fi	128.214.248.6	/pub/Linux
net.tamu.edu	128.194.177.1	/pub/linux
ftp.mcc.ac.uk	130.88.203.12	/pub/linux
src.doc.ic.ac.uk	146.169.2.1	/packages/linux
fgb1.fgb.mw.tu-muenchen.de	129.187.200.1	/pub/linux
ftp.informatik.tu-muenchen.de	131.159.0.110	/pub/comp/os/linux
ftp.dfv.rwth-aachen.de	137.226.4.111	/pub/linux
ftp.informatik.rwth-aachen.de	137.226.225.3	/pub/Linux
ftp.Germany.EU.net	192.76.144.75	/pub/os/Linux
ftp.uu.net	137.39.1.9	/systems/unix/linux
wuarchive.wustl.edu	128.252.135.4	/mirrors/linux
ftp.win.tue.nl	131.155.70.100	/pub/linux
ftp.stack.urc.tue.nl	131.155.2.71	/pub/linux
srawgw.sra.co.jp	133.137.4.3	/pub/os/linux
cair.kaist.ac.kr		/pub/Linux
ftp.denet.dk	129.142.6.74	/pub/OS/linux
NCTUCCCA.edu.tw	140.111.1.10	/Operating-Systems/Linux
nic.switch.ch	130.59.1.40	/mirror/linux
cnuce_arch.cnr.it	131.114.1.10	/pub/Linux
ftp.monash.edu.au	130.194.11.8	/pub/linux
ftp.dstc.edu.au	130.102.181.31	/pub/linux
ftp.sydutech.usyd.edu.au	129.78.192.2	/pub/linux

3.8.2 Par téléphone

Si vous ne possédez qu'un modem (au moins 9600 bps, il faut être raisonnable et vivre avec son temps) vous pouvez télécharger **Linux** sans restriction d'accès sur différents BBS ou différentes machines Unix ou **Linux** de particuliers, mettant leur système à disposition de tous pour rendre service :

brasil	01 44 67 08 44	Bulletin Board System
polux	05 61 26 08 27	Site Linux
elfix	05 61 73 57 06	Site Linux
barberouge	05 61 61 26 86	Site Linux
gyptis	04 91 42 95 30	
shagshag	01 40 30 04 68	Site Linux
stdin BBS	04 72 34 54 37	Bulletin Board System
Le Lien	04 72 08 98 79	Bulletin Board System
Suptel Nancy	03 83 53 16 17	Bulletin Board System

```
             03 83 53 20 21
MtelTls      05 61 37 11 62          Bulletin Board System (Toulouse)
             05 61 37 07 52
cygnux BBS   01 39 94 95 76          Bulletin Board System
             01 39 94 29 47
zenux        04 78 36 10 01 (VF 24k) Passerelle Internet Dedal.
                                       Site Linux uucp/SLIP/PPP/shell
                                       RadioAm TCP/IP et AX25
afflynn      01 45 65 09 21
Viking       04 76 93 13 48          (Acces PPP en 33 600 - Grenoble)
             3 lignes groupees       (Info http://www.voiron.com/viking/)

alphanet     +41 (0) 32 8414081      V34        Acces BBS (guest),
             +41 (0) 32 8414037      ISDN X.75  ou nuucp (~/archives/README,
             +41 (0) 32 8414004      V32bis     ~/archives/ls-laR.gz)
```

Certaines machines appartiennent à des particuliers et ne sont pas forcément en service 24H/24. D'autres BBS français possèdent des fichiers **Linux**, toutefois ils ne sont pas automatiquement (ou régulièrement) mis à jour comme le sont ceux cités ici.

Si vous ne possédez même pas de modem, commencez par en regarder les prix, c'est un périphérique très utile à notre époque. Vous pouvez également vous procurer **Linux** sous forme de CD-ROM vendu par différents distributeurs de shareware/freeware. Toutefois, il vous faut savoir que les délais de fabrication ne permettent pas de tenir ces supports à jour. Certains sont parfaitement utilisables et très complets mais d'autres sont totalement dépassés.

3.8.3 Sur CD-ROM par correspondance

Certains fournisseurs vendent des CD-ROMS contenant Linux, quelques-uns seront bientôt référencés dans ce document ; pour l'instant, reportez-vous à leurs annonces dans les *News Usenet*.

En France, citons DP-Tool, qui fournit un service sérieux pour des prix raisonnables :

```
DP-TOOL club
BP 11
59155 Faches-Thumesnil
Tel : 20 66 37 60
Fax : 20 60 37 23
```

3.9 Comment débuter

Comme il est expliqué ci-dessus **Linux** n'est pas géré de manière centralisée. De par ce fait, il n'en existe pas de « distribution officielle » (certains y travaillent toutefois). A la place, il existe différentes « distributions »,

qui sont des collections plus ou moins complètes de programmes **Linux**, destinées à installer un système en état de fonctionnement.

Il faut commencer par récupérer et LIRE la liste des « Frequently Asked Questions » (FAQ) disponible sur l'un des sites cités précédemment, ou pris sur l'un des sites archivant les FAQ'S Usenet en général (par exemple, `rtfm.mit.edu` ou l'un de ses miroirs français comme `ftp.ibp.fr` dans le répertoire `/pub/doc/faqs`). Ces documents contiennent des centaines de renseignements permettant de résoudre la plupart de vos problèmes.

La lecture du « GUIDE DU ROOTARD », créé par Julien Simon et désormais maintenu par Eric Dumas, est chaudement recommandée, surtout si vous ne lisez pas bien l'anglais. Ce fichier, entièrement rédigé en français, rassemble la traduction de considérations essentielles sur **Linux**, ainsi que nombre de trucs astuces, expliqués quotidiennement dans les *newsgroups* ou conférences francophones dediés à **Linux**. Il est bien entendu disponible sur la majorité des sites francais diffusant le système, sa source officielle étant `ftp.ibp.fr`, répertoire `/pub/linux/french/docs`.

Il est également disponible sur le Web à l'URL (`http://www.freenix.fr/linux/Guide`).

Les « HOWTO », petits guides traitant chacun d'un domaine de **Linux**, sont une lecture indispensable. La plupart d'entre eux est disponible en version française dans le répertoire `/pub/linux/french/docs/HOWTO` sur le site `ftp.ibp.fr`.

Si vous avez accès au Web vous pouvez également trouver de précieux renseignements sur les URL :

- (`http://www.freenix.fr/linux`)
- (`http://www.loria.fr/linux`)

Si le monde UNIX est tout nouveau pour vous, et que **Linux** est votre premier contact avec ce domaine passionnant, suivez **absolument** les conseils suivants :

- Ne vous ruez pas sur les forums de discussion **Linux** pour poser mille questions auxquelles tout le monde est lassé de répondre. Lisez les quelques semaines auparavant, voyez ce qui s'y passe, consultez-en les archives mises à disposition sur certains sites, et il y a fort à parier que vous trouverez ce que vous cherchez sans déranger personne.
- **Linux** est un système UNIX parmi d'autres, et par conséquent toute la littérature spécifique à UNIX s'y applique. On n'apprend pas à maîtriser un nouveau système d'exploitation sans un minimum d'effort de documentation, souvenez-vous du temps qu'il vous a fallu pour maîtriser celui avec lequel vous travailliez auparavant, et des nombreux ouvrages que vous avez du consulter pour arriver à vos fins.
- Procurez-vous et lisez attentivement tous les ouvrages gratuits rédigés sur **Linux**, en particulier le guide d'installation et d'initiation de Matt Welsh et le manuel d'administration réseau d'Olaf Kirch, disponibles en version française, ainsi que le « Guide du Rootard ». Ils contiennent les bases indispensables à votre apprentissage du système. Il ne vous sera pas pardonné de poser une question dont la réponse est contenue dans l'un de ces livres, ou dans l'un des documents complémentaires « HOWTO » disponibles tout aussi librement sur les sites diffusant **Linux**.
- Expérimentez, tentez de comprendre et de résoudre votre problème vous-même : c'est le meilleur apprentissage. Vous n'arriverez jamais à rien si vous faites tout faire par les autres. **Linux** et ses différentes

documentations représentent des milliers d'heures de travail bénévole de centaines de développeurs, rédacteurs, traducteurs, testeurs ; ne condamnez pas le système si vous n'arrivez pas à en tirer parti parce que vous refusez de faire à votre tour un tout petit effort.

3.10 Comment se tenir au courant

Linux fait l'objet sur Usenet de 10 *newsgroups* en langue Anglaise et d'un en langue française, diffusé au moins en France, Suisse, Belgique et au Canada.

Ce groupe destiné aux francophones est `fr.comp.os.linux`. Il accueille les discussions sur **Linux** et est diffusé, hors Usenet, par certains BBS, sous forme de conférence dite « Linux » ou « Linux.fr », ce qui explique le format parfois inhabituel de certains articles.

Les articles postés dans `fr.comp.os.linux` sont archivés régulièrement sur le site `ftp.ibp.fr` dans le répertoire `/pub/linux/french/fr.comp.os.linux`.

Le groupe `comp.os.linux.announce` est modéré par Lars Wirzenius. Il recèle les annonces concernant **Linux** (nouveaux programmes, corrections de bogues, etc). L'adresse à laquelle les soumissions doivent être envoyées est `linux-announce@news.ornl.gov`.

Les diverses documentations sont régulièrement postées dans le groupe modéré `comp.os.linux.answers`.

Le groupe non modéré `comp.os.linux.setup` est prévu pour les discussions sur la mise en oeuvre de systèmes **Linux**.

Le groupe non modéré `comp.os.linux.admin` est prévu pour les discussions sur l'administration de systèmes **Linux**.

Le groupe non modéré `comp.os.linux.development.system` est prévu pour les discussions sur le développement du *noyau* **Linux**. Les questions portant sur le développement d'applications doivent être dirigées vers `comp.os.linux.development.apps` ou un groupe lié à Unix.

Le groupe non modéré `comp.os.linux.development.apps` est prévu pour les discussions sur le développement d'applications spécifiques à **Linux**. Il n'est pas prévu pour contenir les demandes d'applications pour **Linux**.

Le groupe non modéré `comp.os.linux.hardware` est prévu pour les questions portant sur les aspects matériels spécifiques à **Linux**.

Le groupe non modéré `comp.os.linux.networking` est prévu pour les discussions sur le développement et la configuration des outils de gestion du réseau sous **Linux**.

Le groupe non modéré `comp.os.linux.x` est prévu pour les discussions sur X-window spécifiques à **Linux**.

Le groupe non modéré `comp.os.linux.advocacy` est prévu pour les fanatiques désireux de chanter les louanges de leur système préféré. Il est conseillé de l'éviter car il contient principalement des querelles stériles en tous genres.

Le groupe non modéré `comp.os.linux.misc` est prévu pour contenir toutes les discussions et questions qui ne correspondent à aucun des thèmes des autres groupes.

En général, **ne** *crosspostez* **pas** entre plusieurs groupes **Linux**. Le **seul** cas où les *crossposts* sont tolérés est

celui où un article est posté dans `comp.os.linux.announce` et un autre groupe. Le but de la division du groupe original `comp.os.linux` en plusieurs groupes était de réduire le trafic dans chaque groupe. Ceux qui ne respectent pas cette règle risquent d'être insultés sans pitié ...

Attention : Avant d'intervenir dans les groupes Usenet, soyez sûr de connaître les règles de savoir-vivre Usenet, la fameuse « Netiquette ». Consultez notamment l'URL (`http://www.fdn.fr/fdn/doc-misc`).

De plus, lorsque vous souhaitez poster une question dans un groupe Usenet afin d'exposer un problème rencontré, veillez à inclure le maximum d'informations dans votre question : version du noyau (commande « `uname -a` »), de la bibliothèque C (en tant que `root`, commande « `/sbin/ldconfig -v` »), du ou des programmes causant le problème, fichiers de configuration utilisés, distribution Linux installée, traces d'exécution, extraits des fichiers d'historique (situés dans les répertoire `/var/adm` et/ou `/var/log`, etc.

3.11 The Linux Journal

Un magazine, rédigé en anglais et appelé *Linux Journal* a été lancé en 1994, à l'initiative d'utilisateurs travaillant dans la presse informatique aux USA.

Il contient des articles pour lecteurs de niveaux divers et est fait de manière à être utile à tous les utilisateurs de **Linux**.

Ce journal est vendu pour la somme de 32 dollars pour un an d'abonnement (12 numéros) ou 54 dollars pour deux ans.

On peut le commander depuis la France par courrier électronique à l'adresse :

`sales@ssc.com`

Questions et commentaires à adresser (en anglais) à `linux@ssc.com`.

Adresse postale :

```
Linux Journal  P.O Box 85867
Seattle, WA 98145-1867
Fax: +1 206-526-0803
Tel: +1 206-524-8338
```

Linux Journal est également en vente à Paris à la librairie « Le monde en Tique ».

3.12 Statut légal de Linux

Bien que **Linux** soit diffusé avec son code source au complet, le système est « copyrighté » et ne relève donc pas du domaine public. Toutefois, il est disponible gratuitement selon les termes de la « GPL » (Gnu Public License). Lisez cette « GPL » pour obtenir davantage d'informations. (En quelques mots, cette licence assure que nul ne pourra retirer certains droits à l'utilisateur du *logiciel libre*, comme par exemple l'accès gratuit à son code source. Mais elle n'interdit nullement la vente des programmes auxquels elle s'applique).

Les programmes qui tournent sous **Linux** ont chacun leur copyright propre, bien que beaucoup d'entre eux utilisent également la GPL. X Window dépend du « MIT X Copyright », et quelques utilitaires, de celui

de BSD. Dans tous les cas, tous les programmes disponibles sur les sites diffusant **Linux** sont librement diffusables (sinon ils ne s'y trouveraient pas!).

3.13 Ce document

Ce document était au départ une adaptation française par Julien Simon et retouchée par René Cougnenc, de la « *Linux INFO-SHEET* » maintenue par Michael K. Johnson (`Johnsonm@Sunsite.unc.edu`). Cette feuille d'informations est désormais maintenue par Rémy Card.

Ce document est, bien entendu, mis dans le domaine public. Toutefois, soyez assez courtois pour ne pas vous en approprier la propriété si vous le diffusez... Veuillez également nous prévenir si vous utilisez ce document dans le cadre d'une diffusion commerciale de Linux.

Il a évolué petit à petit en s'éloignant du document original (bien que les modifications dans la version originale soient périodiquement intégrées dans ce document), et est dorénavant réalisé en SGML selon la même méthode que les *HOWTOs* **Linux**, ce qui permet de le rendre disponible aussi bien sous forme ASCII ordinaire que dans les formats plus agréables à lire PostScript, DVI et hypertexte HTML.

Les outils utilisés ne permettent pas de respecter parfaitement les conventions typographiques en vigueur en France et la génération de la version ASCII donne parfois lieu à un formatage hasardeux ; le résultat est néanmoins acceptable pour ce type de document utilitaire.

Critiques et louanges à adresser à :

Rémy Card <Remy.Card@linux.org>
René Cougnenc
Éric Dumas <Eric.Dumas@freenix.fr>
Julien Simon <Julien.Simon@freenix.fr>

4 Introduction à l'installation

Avant de commencer, deux trois petites choses sont à préciser sur la manière d'effectuer l'installation de ta machine.

4.1 Récupération

La récupération des distributions **Linux** peut se faire de plusieurs manières :

1. tu as accès à l'Internet, ou tu as un modem : regarde la feuille d'informations plus haut ;
2. tu as un pote qui a les disquettes ou les CD-ROM, et dans ce cas-là, tu lui empruntes :-) ;
3. si tu n'as ni l'un ni l'autre, je te déconseille de télécharger (à moins que tu sois actionnaire de France Télécom :-)). Tu peux acheter les CD-ROM. Je te conseille les lots de CD édités par Infomagic (5 CD pour 100 Frs environ) ou Yggdrasil (6 CD pour le même prix).

Bon nombre de commerces proposent des distributions CD qui sont souvent beaucoup plus chères, plus âgées. Je n'ai pas de pourcentage chez les deux fournisseurs que je vais citer. Mais je n'ai pas trouvé moins cher, plus rapide et plus sérieux, c'est pour ça que je te donne leurs adresses :

- **DP-TOOL club**

```
BP 11
59155 Faches-Thumesnil
Tel : 20 66 37 60
Fax : 20 60 37 23
```

  Url `http://ourworld.compuserve.com/homepages/dp_tool/`.

- **Logiciels du soleil**

```
1, rue Pasqualini
06800 Cagnes sur Mer
```

  Contacts à l'adresse email `kheops@kheops.com` ou via l'Url `http://www.kheops.com`.

4.2 Un mot sur le format des binaires

Au cours de l'histoire de **Linux**, le format des binaires a changé. Le format des binaires des premières distributions était en **a.out**. Depuis environ 1 an, le format des binaires a changé: il est passé en **ELF**. Si jamais tu n'avais pas **Linux** sur ta machine, tu n'y verras que du feu. Si jamais tu veux mettre à jour ta machine et que tu es en a.out, tu as deux solutions :

1. tout casser et réinstaller... c'est le plus simple ;
2. faire une mise à jour progressive. Dans ce cas, lis tout d'abord le chapitre consacré à ELF plus bas.

Pour connaître le format des binaires de ta machine, lance par exemple la commande suivante :

```
gandalf# file /bin/ls
/bin/ls: ELF 32-bit LSB executable i386 (386 and up) Version 1
```

Dans ce cas, les binaires sont en ELF. Si le message indique un format `Linux/i386 executable`, alors il est fort probable que ta machine soit au format a.out.

4.3 Les différentes possibilités d'installations

Il existe en fait plusieurs possibilités pour installer Linux... En voici un court résumé :

- Installation classique : tu as une partition libre pour y mettre Linux. Dans ce cas, pas de problème: il suffit de suivre les instructions des sections suivantes.

- Aïe : j'avais pas prévu, et je n'ai qu'une partition Dos. Dans ce cas, il existe deux solutions :
  - FIPS : toutes les distributions donnent un programme DOS qui permet en fait de repartitionner ta partition Windows sans avoir à tout réinstaller. Comment ça marche? C'est simple. Il faut dans un premier temps passer un coup de `scandisk` puis un coup de `defrag` (optimisation complète : tous les fichiers doivent être remontés). Ensuite, tu rebootes avec une disquette DOS (avec le moins de programmes résidents que possible). Il suffit ensuite de lancer FIPS et de suivre les instructions. Cela permet de réduire la taille de ta partition Dos, et ainsi d'installer Linux sur la place restante.
  - Installer Linux en utilisant *umsdos*. Ce système est très pratique (je l'utilise pour le moment à l'armée). Cela permet d'installer un système Linux, sans toucher les partitions : Linux est installé en fait sur la partition Dos. Toutefois, cette installation est très limitée car le système de fichier umsdos est très très lent. Cette solution peut être envisagée dans des cas extrèmes... mais qui peuvent être très pratiques.
- une machine sans disque ! Il peut arriver que tu possède déjà une machine sur le réseau qui ai Linux de parfaitement installé. Tu peux alors faire en sorte que ta machine utilise l'autre machine comme serveur. Dans ce cas, il te suffit d'une disquette de boot ! Le mini Howto *nfs-root* détaille complêtement cete opération.

Bien... après ces préliminaires, passons aux distributions.

5 Distributions

5.1 Introduction

On appelle « distribution » un ensemble de supports (disquettes, CD-ROM ...) contenant de quoi installer un logiciel sur une machine. Voir, à ce propos, la section « Comment débuter » de la *feuille d'informations Linux* reproduite ci-dessus. Il n'existe pas de distribution officielle de **Linux**. Seules des distributions circulent. Dans ce guide, seules la Slackware et la Redhat seront détaillées. Toutefois, voici quelques mots sur les principales distributions existantes :

- Slackware : c'est la distribution qui faisait historiquement référence dans la mesure où c'était la seule distribution à peu près correctement faite il y a un ou deux ans. De nos jours, elle est fortement concurrencée par de nouvelles distributions comme Red Hat et on peut craindre qu'elle finisse comme la SLS en raison de sa difficulté à évoluer. De plus, elle ne gère qu'une seule architecture (Intel).

  Elle fait près de 113 Mo (compressée), c'est-à-dire 89 disquettes, auquel il faut rajouter les 6 disquettes root et la bonne vingtaine de disquettes de boot.

  La dernière version de la Slackware est la 3.1. Elle est tout ELF. Elle utilise la libc 5.3.12, le noyau 2.0.0, et XFree86 3.2.0. Sa dernière mise à jour date de début Décembre 1996. C'est cette version qui sera détaillée ici.

  Site ftp officiel : `ftp.cdrom.com`Site Web : `http://www.cdrom.com`

- Red Hat : C'est une distribution initialement commerciale, mais qui est également disponible par téléchargement sur les sites FTP. Elle semble bien plus professionnelle que Slackware. De plus, elle est disponible pour d'autres architectures processeurs « non Intel » (Alpha, Sparc). Version 4.0 (Colgate). Dernière mise à jour : 8 Décembre 1996.

  Cette distribution se disitingue par son sérieux, sa simplicité d'installation, et surtout par son extrème convivialité. De plus, la mise à jour du système est gérée par un système de paquetages.

  Site ftp officiel : `ftp.redhat.com`Site Web : `http://www.redhat.com`

- Debian : C'est la version de Linux soutenue par la FSF (Free Software Foundation). Elle est basée sur des programmes relativement anciens; les auteurs de cette distribution voulant principalement une distribution « stable » au détriment de nouvelles fonctionnalités. Version : 1.1.8 Dernière mise à jour : 8 Décembre 1996

- MCC : Version 1.2+. Dernière mise à jour : Mai/Juin 1995. (semble tombée dans l'oubli...)

- Bogus : C'est une distribution stable et assez bien faite. Le seul problème avec Bogus est que sa fréquence de mise à jour est assez irrégulière. Version 1.0.1 Dernière Mise à jour : Juillet 94

- SLS : Première grande distribution de **Linux**, la SLS semble être désormais obsolète : Version 1.06. Dernière mise à jour : Novembre 1994.

- Mini-Linux : Comme son nom l'indique, il s'agit d'une distribution très réduite de Linux (en ce qui me concerne, je la trouve justement trop réduite mais bon...). Elle se trouve sur `ftp.loria.fr`, en version française. Elle contient 5 disquettes et à été mise à jour pour la dernière fois le 10 Avril 1995.

- Jurix : C'est une distribution créée en Allemagne à partir des dernières versions des programmes disponibles. Elle est donc toujours « à la pointe du progrès » mais peut parfois causer des problèmes car les programmes qu'elle contient ne sont pas toujours complètement débogués. Dernière mise à jour : 8 Décembre 1996.

5.2 Disquettes d'amorçages

Que ça soit avec la RedHat ou la Slackware, tu dois utiliser une disquette de boot (et une disquette root pour la Slackware). La manière de copier ces fichiers sur disuqette est commune aux deux distributions.

Après avoir sélectionné la ou les images à copier sur disquette pour permettre l'amorçage de la machine, il est nécessaire de la copier sur disquette. Pour copier cette disquette, plusieurs solutions existent :

1. copier le fichier sur la disquette avec un `cat image >/dev/fd0` (`/dev/fd0` correspondant au périphérique de ton lecteur de disquettes.). Il est possible que tu doives décompresser le fichier. Pour cela, utilises le programmes `gunzip` : `gunzip bare.i.gz`.

2. si jamais tu es sous MS-DOS, tu dois te servir du programme *RAWRITE*. Pour cela : décompacter le fichier si nécessaire (le nom se termine par « .gz ») grâce à *GZIP.EXE*, formater la disquette MS-DOS qui ne doit pas présenter le moindre défaut de surface (« secteurs défectueux ») puis utiliser *RAWRITE* pour écrire le (ou les fichiers) sur les disquettes.

5.3 Slackware : description

Dans cette partie, on suppose que tu as la version de la Slackware 3.1. Voici une description brève des disquettes (en bref, ce que tu vas y trouver, pour éviter que tu installes des trucs dont tu n'as pas besoin).

5.3.1 Disquettes Boot et Root

Deux disquettes sont fondamentales pour installer **Linux** avec la Slackware: la disquette de boot et la disquette root. Linux s'installe en effet ... sous Linux, de sorte qu'il faut pouvoir le lancer « à partir d'une disquette » avant de commencer l'installation. La disquette *boot* contient un noyau Linux et la disquette *root* fournit au Linux ainsi lancé quelques fichiers absolument nécessaires à son fonctionnement. Plusieurs types de disquettes *boot* et *root* existent, il faut choisir en fonction du matériel employé. Le contenu de chaque disquette est livré sous forme d'un fichier de type *image*, stockant le contenu de tous les secteurs physiques d'une disquette.

Voici un résumé des étapes préalables à une installation :

- Choix d'un fichier *boot* et d'un *root* approprié au matériel (voir tableau ci-dessous) ;
- formatage d'une paire de disquettes (sans secteurs défectueux), par exemple avec MS-DOS « format » ;
- inscription du fichier boot sur une disquette ;
- inscription du fichier root sur une disquette ;
- démarrage du système grâce à la disquette *boot* ;
- insertion, sitôt que le système le demande, de la disquette *root*.

Il y a un bon nombre de disquettes de boot. Voici un tableau récapitulatif qui devrait te permettre de choisir celle qui te conviendra le mieux.

Les sources sont sur : \ Contrôleur Disque :	IDE	SCSI	MFM
Disque dur	bare.i	Utiliser une disquette SCSI dans la liste ci-dessous.	xt.i
CD-ROM SCSI	Utiliser une disquette SCSI	idem	
CD-ROM IDE/ATAPI	bare.i	idem	
Aztech, Orchid, Okano, Wearnes,	aztech.i	aztech.s	

Conrad, CyCDROM non-IDE CD-ROM			
Sony CDU31a, Sony CDU33a CD-ROM	cdu31a.i	cdu31a.s	
Sony CDU531, Sony CDU535 CD-ROM	cdu535.i	cdu535.s	
Philips/LMS cm206 CD-ROM	cm206.i	cm206.s	
Goldstar R420 CD-ROM	goldstar.i	goldstar.s	
Mitsumi non-IDE CD-ROM	mcdx.i mcd.i	mcdx.s mcd.i	
Optics Storage 8000 AT CD-ROM ("Dolphin")	optics.i	optics.s	
Sanyo CDR-H94A CD-ROM	sanyo.i	sanyo.s	
Matsushita, Kotobuki, Panasonic, CreativeLabs (Sound Blaster), Longshine and Teac non-IDE CD-ROM	sbpcd.i	sbpcd.s	
NFS	net	scsinet	
Tape	bare (pour lecteur non SCSI. Sinon utiliser une disquette SCSI - voir ci-dessous.	Utiliser une disquette SCSI de la liste ci-dessous.	xt.i (pour lecteur non SCSI)

```
------------------+---------------+--------------------------------------
```

Gestionnaires SCSI : 7000fast.s, advansys.s, aha152x.s, aha1542.s, aha1740.s, aha2x4x.s, am53c974.s, buslogic.s, dtc3280.s, eata_dma.s, eata_isa.s, eata_pio.s, fdomain.s, in2000.s, iomega.s, n53c406a.s, n_5380.s, n_53c7xx.s, pas16.s, qlog_fas.s, qlog_isp.s, seagate.s, trantor.s, ultrastr.s et ustor14f.s

Si tu ne sait pas quelle est la carte SCSI installée sur ta machine, essayes alors la disquette d'amorçage générique `scsi.s`. Son nom sera affiché dans les messages lors de l'amorçage. Ensuite, utilises la disquette correspondante (car la disquette scsi.s utilise beaucoup de mémoire).

Passons ensuite à la disquette root. Le fichier correspondant doit être copié sous sa forme compressée : **Linux** le décompresse tout seul comme un grand. Il existe six disquettes root possibles :

- color.gz : permet d'avoir les couleurs sur la console ;
- umsdos.gz : Utilise umsdos (évite de repartitionner ton disque). Attention : c'est beaucoup plus lent que l'ext2fs natif.
- text.gz : disquette root de base ;
- tape.gz : expérimental, pour l'installation à partir de cartouches (je n'ai jamais essayé) ;
- pcmcia.gz : conçu pour l'installation via NFS pour les portables ayant une carte Ethernet de type PCMCIA ;
- rescue.gz : disquette avec quelques programmes : *vi, lilo, e2fsck, etc.*. Très utile si tu as planté ta machine...

ATTENTION : La disquette Root ne doit pas être décompressée !!

5.3.2 La distribution

Bon, on a vu les deux premières disquettes. Voici la description de la distribution. Comme c'est long, voici un résumé. Entre parenthèses, le nombre de disquettes :

- A (8) : Base du système **Linux** ;
- AP (5) : Applications courantes ;
- D (13) : Compilateurs (gcc 2.7.2) ;
- E (8) : Emacs (19.31) ;
- F (2) : FAQ en anglais ;
- K (6) : Noyau (2.0.0) ;
- N (6) : Réseau ;
- T (9) : TeX et LaTeX (2e) ;

- TCL(2) : Tcl (7.5) Tk (4.1) :
- X (21) : X Window X11R6 - Serveur XFree86 3.2.0 ;
- XAP (4) : Applications X Window ;
- XD (4) : X Window dev. : linkkit + PeX ;
- XV (3) : XView pour les adorateurs du Look OPEN LOOK ;
- Y (4) : Jeux (doom, etc).

Dans la liste, tu vas trouver trois champs :

- ADD : Le programme est automatiquement installé ;
- REC : Recommandé ;
- OPT : Optionnel.

Bon, on est partis :-) !

- A : *Base*
  - A1 :
    - aaa_base : (*ADD*) Mise en place de l'arborescence de base. Programmes de base (GNU mtools 2.0.7, zip201, unzip512).
    - aoutlibs : (*ADD*) bibliothèques a.out (libc 4.7.6) ;
    - bash : (*ADD*) (1.14.7) interpréteur de commandes ;
    - comms : (*REC*) outils de communications séries ;
    - cpio : (*ADD*) (2.3) utilitaires d'archivage de fichiers ;
    - devs : (*ADD*) les fichiers spéciaux situés dans `/dev` ;
    - etc : (*ADD*) fichiers de configuration situés dans le répertoire `/etc` ;
    - grep : (*ADD*) (2.0) recherche de chaînes dans les fichiers.
  - A2 :
    - bin : (*ADD*) binaires (/bin et /usr/bin) (GNU fileutils 3.12, man, patch, sed, etc.) ;
    - e2fsbn : (*ADD*) (1.5) utilitaires pour le gestionnaire de fichier ext2fs ;
    - gzip : (*ADD*) (1.2.4) GNU zip compression.
  - A3 :
    - elflibs : (*ADD*) bibliothèques ;
    - find : (*ADD*) (4.1) outils de recherche de fichiers ;
    - getty : (*OPT*) getty_ps 2.0.7i (Contrôle le login sur le système) ;
    - gpm : (*REC*) (1.09) Permet de réaliser du copier-coller avec la souris dans les fenêtres en mode texte ;
    - hdsetup : (*ADD*) (3.1.0) programme setup de maintenance/d'installation du système.

- A4 :
  - ibcs2 : (*OPT*) permet d'exécuter des binaires provenant d'autres plates-formes (SVR4, SCO, Wyse, Xenix, etc.) ;
  - ide : (*REC*) noyau IDE (2.0.0) sans SCSI ;
  - keytbls : (*OPT*) (0.90) fichiers de configuration pour le clavier. Contient la définition du clavier français.
  - ldso : (*ADD*) `ld.so.1.8.2`, c'est à dire l'éditeur de liens dynamique ;
  - syslogd : (*ADD*) (1.3) enregistre/redirige les messages d'erreurs système.
- A5 :
  - less : (*ADD*) less-290 (Mieux que more) ;
  - lilo : (*ADD*) (19) chargeur du noyau ;
  - loadlin : (*REC*)(1.6) charge les noyaux sous MS-DOS ;
  - pcmcia : (*OPT*) (2.8.17) pour les cartes PCMCIA ;
  - ps : (*ADD*) procps 0.99a (Utilitaires pour l'affichage des processus) ;
  - sh_utils : (*ADD*) (1.12) commandes utilitaires ;
  - sysvinit : (*ADD*) (2.62) programme init + scripts /etc/rc.d ;
  - umsprogs : (*ADD*)(0.8) pour gérer le gestionnaire de fichier umsdos.
- A6 :
  - modules : modules chargeables pour le noyau 2.0.0.
- A7
  - scsi : (*REC*) (2.0.0) noyau SCSI + IDE ;
  - tar : (*ADD*) (1.11.8) GNU tar : gestion d'archives ;
  - tcsh : (*OPT*)(6.06) shell compatible csh.
- A8
  - txtutils : (*ADD*) (1.13) outils de manipulation de textes (cat, head, etc) ;
  - util : (*ADD*) util-linux 2.5 ;
  - zoneinfo : (*ADD*) système de gestion des fuseaux horaires.

- AP : *Applications courantes*
  - AP1 :
    - ash : (*OPT*) autre shell, très petit en mémoire ;
    - bc : (*OPT*) (1.03) calculatrice ;
    - diff : (*REC*) (2.7) permet de donner la liste des différences entre fichiers ;
    - ftape : (*OPT*) (2.03b) permet de manipuler les lecteurs QIC-40 et QIC-80 ;
    - ghostscr : (*OPT*) (2.6.2) GNU Ghostscript permet de visualiser, de transformer des fichiers postscript ;
    - gp9600 : (*OPT*) script de configuration de la vitesse du modem ;
    - jed : (*OPT*) (0.96) petit éditeur de texte ;

  - mt_st : (*OPT*) (0.2) gestionnaire de manipulation de lecteurs de cartouches ;
  - sudo : (*OPT*) (1.2) permet l'exécution de quelques commandes en tant que Root pour certains utilisateurs.
- AP2 :
  - groff : (*REC*) (1.09) outils de formatage de documents. Utilisé par man ;
  - gsfonts1 : (*OPT*) polices de caractères pour Ghostscript ;
  - jove : (*OPT*) (4.14.10) petit éditeur de texte compatible Emacs ;
  - workbone : (*OPT*) (2.3) lecteur de CD audio.
- AP3 :
  - gsfonts2 : (*OPT*) polices de caractères pour Ghostscript(2/2) ;
  - manpags : (*REC*) (1.8) pages de manuel des sections 1, 4, 5, 6, 7, 8 et 9.
- AP4 :
  - ispell : (*OPT*) (3.1.08) vérificateur orthographique (Anglais) ;
  - mc : (*OPT*) (2.1) outil de manipulation de fichiers, répertoires ;
  - sc : (*OPT*) (6.21) petit tableur ;
  - shlbsvga : (*REC*) libsvga.so.1.2.7 (Nécessaire pour Ghostscript en mode texte) ;
  - termbin : (*OPT*) (2.3.5) exécutables term, qui émulent une connexion TCP/IP sur une connexion modem ;
  - termnet : (*OPT*) utilitaires réseau compilés pour utiliser le système `term` de multiplexage des connexions sur une ligne série (`tncftp`, `ttelnet`, `tfinger`, `twhois`).
- AP5 :
  - termsrc : (*OPT*) sources de termbin: (2.3.5) ;
  - texinfo: (*REC*) (3.6) générateur de documentation. Nécessaire pour lire les fichiers info.
  - vim : (*OPT*) (3.0) Version améliorée de vi (vi pur beurre (ou presque) étant fourni avec elvis). Préférer de loin cette version.
  - zsh : (*OPT*) (2.4) shell zsh.

- D : *Compilateurs (gcc 2.7.2)*
  - D1 :
    - binutils : (*ADD*) (2.6.0.14) ar, as, c++filt, ld, nm, ranlib, etc ;
    - bison : (*REC*) (1.22) analyseur syntaxique (compatible yacc) ;
    - byacc : (*REC*) yacc, version BSD ;
    - flex : (*ADD*) (2.5.3) analyseur lexical ;
    - m4 : (*REC*) (1.4) expanseur de macros.
  - D2 :
    - g77 : compilateur Fortran ;
    - gmake : (*ADD*) (3.74) GNU make ;
    - man2 : (*REC*) (1.8) pages de manuel des appels système ;

  - pmake : (*ADD*) (4.4) make BSD.
- D3 :
  - gcc272 : (*REC*) compilateur C ELF optimisé pour 486 ;
  - gccaout : (*OPT*) version du compilateur C générant des binaires au format a.out ;
  - strace : (*OPT*) (3.0) trace les appels système et les signaux.
- D4 :
  - gcl : (*OPT*) (2.2) interpréteur Lisp ;
  - man3 : (*REC*) (1.8) pages de manuel des fonctions de la bibliothèque C.
- D5 :
  - gdb : (*REC*) (4.16) débogueur ;
  - gxx272 : (*REC*) (2.7.2) compilateur C++, optimisé pour 486 ;
  - p2c : (*OPT*) (1.19) convertisseur Pascal To C.
- D6 :
  - gxxaout : (*OPT*) (2.7.2) version du compilateur C++ générant des binaires au format a.out ;
  - objc270 : (*OPT*) (2.7.2) compilateur Objective-C.
- D7 :
  - libaout : (*REC*) bibliothèques pour le support du format a.out (4.7.6) ;
  - linuxinc : (*ADD*) fichiers d'entête de Linux 2.0.0.
- D8 :
  - libc : (*ADD*) (libc 5.3.12) bibliothèques nécessaires pour la compilation + exécution. Inclue libcurses 1.0.0, libgdbm 1.7.3, libtermcap 2.0.8, et db 1.85.4 ;
  - rcs : (*OPT*) (5.7) système de gestion de versions des fichiers ;
  - terminfo : (*OPT*) descendant du fichier termcap de 4.4 BSD.
- D9 :
  - libcinfo : (*OPT*) informations sur la bibliothèque C ;
  - objcaout : (*OPT*) objective C, format a.out.
- D10 :
  - libgxx : (*REC*) (2.7.1.4) bibliothèque libg++ + fichiers includes pour c++.
- D11 :
  - ncurses : (*REC*) (1.9.9e) bibliothèque de gestion de l'écran ;
  - svgalib : (*OPT*) (v1.2.10) bibliothèque graphique.
- D12 :
  - perl1 : (*OPT*) (5.003) interpréteur Perl ;
  - tools : (*OPT*) (2.17) bibliothèque de génération de DLL.
- D13 :
  - perl2 : (*OPT*) (5.003) interpréteur Perl.

- E : *Emacs (19.31)*
  - E1 :
    - elisp1 : (*OPT*) sources Emacs-Lisp.
  - E2 :
    - elisp2 : (*OPT*) sources Emacs-Lisp ;
    - elisp3 : (*OPT*) sources Emacs-Lisp ;
    - elispc2 : (*REC*) fichiers Emacs-Lisp compilés.
  - E3 :
    - elisp4 : (*OPT*) sources Emacs-Lisp ;
    - elispc3 : (*REC*) fichiers Emacs-Lisp compilés.
  - E4 :
    - elispc1 : (*REC*) fichiers Emacs-Lisp compilés.
  - E5 :
    - emac_nox : (*OPT*) emacs sans X Window.
  - E6 :
    - emacinfo : (*REC*) fichiers d'information.
  - E7 :
    - emacmisc : (*REC*) fichiers pour Emacs.
  - E8 :
    - emacsbin : (*ADD*) (19.31) Binaires d'Emacs. Utilise X Window.
- F : *FAQ en anglais*
  - F1 :
    - manyfaqs: FAQs : (*ADD*) FAQ comp.lang : Fortran, GCC, G++, TCL, Objective-C, Lisp, Scheme, et Perl ;FAQ PostScript ;
      Linux FAQ ;
      FAQ comp.windows.x.
  - F2 :
    - howto : (*ADD*) Tous les howtos du **Linux** Documentation Project, en anglais bien sûr.
- K : *Sources du noyau*
  - K1 à K6 : sources de la version 2.0.0.
- N : *Réseau*
  - N1 :
    - bind : (*REC*) (4.9.3 beta 26) bind, dig, dnsquery, host, nslookup, ... ;
    - cnews : (*OPT*) système de transport de news ;

  - dip : (*OPT*) (3.3.7o) gestionnaire de connections IP pour SLIP, CSLIP ;
  - elm : (*OPT*) (2.4 pl 25) mailer interactif ;
  - mailx : (*REC*) (5.5) mailer ;
  - metamail : (*REC*) (2.7) gestionnaires du format MIME ;
  - procmail : (*OPT*) (3.10) programme de gestion de courrier électronique.
- N2 :
  - inn : (*OPT*) (1.4) système de transport de news et serveur NNTP ;
  - lynx : (*OPT*) (2.5) arpenteur (browser) en mode texte ;
  - nn-nntp : (*OPT*) (6.4.18) lecteur de news nn compilé pour utiliser NNTP ;
  - rdist : (*OPT*) programme pour maintenir des parties d'arborescences identiques entre machines.
- N3 :
  - nn-spool : (*OPT*) (6.4.18) autre système de news ;
  - pine : (*OPT*) (3.95) mailer basé sur elm ;
  - PPP : (*OPT*) (2.2.0f) démon et gestion du protocole PPP ;
  - smailcfg : (*OPT*) fichiers de configuration pour sendmail ;
  - tin : (*OPT*) (1.2pl2) lecteur de news plein écran.
- N4 :
  - netpipes : (*OPT*) (3.1) utilisation de tcp/ip dans des scripts ;
  - sendmail : (*REC*) (BSD 8.8.3) processus de transport de courier. Gère TCP/IP et UUCP.
  - trn-nntp : (*OPT*) (3.5) lecteur de news trn compilé pour utiliser un serveur NNTP ;
  - trn : (*OPT*) (3.5) lecteur de news trn compilé pour utiliser le spool local.
- N5 :
  - tcpip : (*REC*) programmes de gestion réseau.
- N5 :
  - apache : (*OPT*) (1.0.0) serveur WWW ;
  - uucp : (*OPT*) (1.06.1) Taylor UUCP, système de communication UUCP.

- T : *TeX et LaTeX : distribution*
  - T1 :
    - gentle : (*REC*) manuel d'introduction à TeX ;
    - ntf-ams ntf-astr ntf-bard ntf-bbm ntf-bl ntf-call ntf-cher ntf-cmas ntf-cmb ntf-cmca ntf-cmcy ntf-cml ntf-cmoe ntf-cmpc ntf-cms ntf-cmts ntf-cmu ntf-conc ntf-cypr ntf-ding ntf-duer ntf-elvi ntf-engw ntf-futh ntf-geor ntf-go ntf-hand ntf-hge ntf-kart ntf-klin ntf-la ntf-logi ntf-logo ntf-ogha ntf-okud : polices de caractères.
  - T2 :
    - ntf-dc ntf-fc ntf-goth ntf-ocm ntf-ocra ntf-osma ntf-pand ntf-phon ntf-punk ntf-recy ntf-rsfs ntf-rune ntf-stma ntf-teng ntf-thai ntf-twca ntf-ugar ntf-wasy ntf-wsui ntf-xcmr : polices de caractères ;

- ntb-b2d : (*OPT*) (1.0) transforme un .bib en .dvi ;
- ntb-djc : (*REC*) (0.1) programmes pour une sortie des fichiers dvi sur HP 500, 500c et 550c ;
- ntb-dps : (*REC*) (1.2) dvips (Transforme un fichier .dvi en PostScript) ;
- ntb-gsfp : (*REC*) (1.0) conversion de fontes postscript en pk ;
- ntb-html : (*OPT*) (1.1) il s'agit tout simplement du convertisseur html2latex.

– T3 :

- ntf-ps : (1.2) polices de caractères Postscript ;
- ntb-indx : (*REC*) (1.1) programmes makeindex, ... ;
- ntm-deva : (1.1) (*OPT*) programmes pour l'impression avec TeX pour certaines langues nordiques, indiennes, ... ;

– T4 :

- ntb-bibt : (*REC*) (1.1) bibtex ;
- ntb-dlj : (*REC*) (1.2) programmes dvi2xx ;
- ntb-kpat : (*OPT*) (1.1) version binaire de la bibliothèque kpathsea ;
- ntb-tex : (*ADD*) (1.2) binaires pour TeX (3.1415), MakeTeX,... ;
- ntb-traf : (*OPT*) (1.1) outil de conversion de fichiers au format .fig (xfig) au format epsf ou LaTeX ;
- ntb-utl1 : (*REC*) (1.1) dvicopy, dvitype, pooltype, ... ;
- ntb-xdvi : (*REC*) (1.2) programme pour afficher les fichiers .dvi sous X (xdvi).

– T5 :

- ntb-mf : (*ADD*) (1.2) binaires pour MétaFont ;
- ntb-utl2 : (*REC*) (1.1) gftodvi, gftopk, gftype, mft, pktogf, ... ,
- ntb-utl3 : (*REC*) (1.1) pltotf, tftopl, vftovp, ... ,
- ntb-xfig : (*REC*) binaires pour xfig : logiciel de dessin vectoriel ;
- ntm-amst, ntm-arab,ntm-germ, ntm-pict : (*OPT*) Feuilles de styles, macros.

– T6 :

- ntm-etex : (*REC*) (1.2) extention de TeX : ETeX ;
- ntm-gree, ntm-hebr, ntm-ltx3, ntm-tami, ntm-viet: (*OPT*) styles supplémentaires.

– T7 :

- ntm-hier,ntm-ltx2 : (*OPT*) styles.

– T8 :

- ntm-ltx1, ntm-plft,ntm-turk : (*OPT*) styles.

– T9 :

- ntm-tex, ntm-ltx4 : (1.1) styles et formats pour TeX.

– Tcl (7.5) Tk (4.1)

– TCL1 :

- tcl : (*ADD*) (7.5) ;

    - tk : (*REC*) (4.1).
  - TCL2 :
    - tkdesk : (*OPT*) (1.0a2) gestionnaire de fichiers ;
    - tclx : (*REC*) (7.5.0) extension de Tcl.
- X : *X Window X11R6 - Serveur XFree86 3.2*
  - X1 :
    - fnon_1 : (*OPT*) polices de caractères divers ;
    - fnt100_1 : (*OPT*) polices de caractères 100 dpi ;
    - fvwmicns : (*OPT*) icones xpm3icons.tar.Z. Utilisées par fvwm ;
    - xlock : (3.11) (*ADD*) programme de verrouillage du serveur X.
  - X2 :
    - fnon_2 : (*OPT*) polices de caractères divers ;
    - oldlibs5 : (*REC*) vieilles bibliothèques XFree86 2.1.1 (X11R5).
  - X3 :
    - fnon_3 : (*OPT*) polices de caractères diverses.
  - X4 :
    - fnt100_2 : (*OPT*) polices de caractères 100 dpi ;
    - x32html : (*OPT*) docmentation XFree86 3.2 au format HTML.
  - X5 :
    - oldlibs6 : (*REC*) bibliothèques au format a.out ;
    - x328514 : (*REC*) serveur X Window pour les cartes utilisant IBM8514.
  - X6 :
    - x32agx : (*REC*) serveur X Window pour les cartes utilisant IIT AGX-016, AGX-015, et AGX-014 ;
    - x32cfg : (*ADD*) fichiers de configuration pour X, xdm et fvwm.
    - x32doc : (*OPT*) documentation pour XFree 3.2.
  - X7 :
    - x32bin : (*ADD*) programmes X.
  - X8 :
    - x32fcyr (*OPT*) polices de caractères cyrilliques ;
    - x32fnts1 : (*ADD*) polices de caractères ;
    - x32ma8 : (*REC*) serveur X Window pour les cartes utilisant Mach8.
  - X9 :
    - x32fnts2 : (*ADD*) polices de caractères.
  - X10 :
    - x32fscl : (*OPT*) polices de caractères ;

    - xpm : (3.4c) (*ADD*) bibliothèque de gestion de fichier images au format xpm.
  - X11 :
    - x32ma32 : (*REC*) serveur X Window pour les cartes utilisant Mach32 ;
    - x32i128 : (*REC*) serveur X Window pour les cartes utilisant Nine Imagine 128.
  - X12 :
    - x32lib : (*ADD*) bibliothèques dynamiques et fichiers de configuration pour XFree86 3.2.
  - X13 :
    - x32ma64 : (*REC*) serveur X Window pour les cartes utilisant Mach64 ;
    - x32mono : (*REC*) serveur X Window monochrome.
  - X14 :
    - x32man : (*REC*) pages de manuel pour le serveur XFree86 3.2.
  - X15 :
    - x32p9k : (*REC*) serveur X Window pour les cartes utilisant P9000 ;
    - x32nest : (*OPT*) serveur expérimental.
  - X16 :
    - x32s3 : (*REC*) serveur pour les cartes utilisant S3 ;
    - x32prog : (*REC*) fichiers d'en-têtes et Imakefile.
  - X17 :
    - x32ps : (*OPT*) documentation XFree 3.2 au format PostScript.
  - X18 :
    - X32s3v : (*REC*) serveur pour cartes S3 Virge ;
    - X32setup : (*REC*) outil de configuration graphique pour X.
  - X19 :
    - x32svga : (*REC*) serveur X pour cartes SuperVga.
  - X20 :
    - X32xfb : (*OPT*) serveur particulier...
  - X21 :
    - x3216 : (*OPT*) Serveur 16 couleurs pour les modes EGA/VGA ;
    - x32w32 : (*OPT*) Serveur pour cartes ET4000/W32.
- XAP : *Applications X Window*
  - XAP1 :
    - gchess : (*OPT*) GNU chess (v. 4.00pl 69) + xboard (v.3.0 pl 9) le fameux jeux d'échec ;
    - ghstview : (*OPT*) (1.5) Ghostview permet de visualiser les fichiers au format Postscript. A besoin du pack gs_x12.
    - gnuplot : (*OPT*) (3.5) programme pour tracer des courbes mathématiques ;
    - gs_x11 : (*REC*) permet de visualiser les fichiers Postscript sous X ;

      - seyon : (*OPT*) (2.14c) outils de télécommunication sous X ;
      - xxgdb : (*OPT*) (1.12) débogueur en mode X.
    - XAP2 :
      - arena : (*OPT*) (béta-2b) arpenteur Web (browser) ;
      - libgr : (*REC*) (1.3) bibliothèques de manipulation des fichiers GIF, TIFF, JPEG. Elles sont utilisées par XV et Interviews.
      - x3270 : (*OPT*) (3.0.1.3) outil de communication avec un IBM 3278 ;
      - xfm : (*OPT*) (1.3.2) gestionnaire graphique de fichiers ;
      - xpaint : (*OPT*) (2.1) programme de dessin bitmap.
    - XAP3 :
      - fvwm95 : (*OPT*) (2.0.41f) fvwm 2.0.41 modifié pour ressembler au gestionnaire de fenêtres de Windows 95 ;
      - xgames : (*OPT*) quelques jeux ;
      - xspread : (*OPT*) (2.1) un tableur rudimentaire ;
      - xfract : (*OPT*) (3.00) logiciel de génération de fractales ;
      - xfileman : (*OPT*) (0.5) gestionnaire de fichiers.
    - XAP4 :
      - xv : (*OPT*) (3.10a) programme de visualisation de fichiers d'images + conversions + filtres d'images, etc.
  - XD : *X Window dev. : linkkit + PeX*
    - XD1 :
      - x32lkit1 : (*OPT*) XFree86 3.2 Linkkit 1/3.
    - XD2 :
      - x32lkit2 : (*OPT*) XFree86 3.2 Linkkit 2/3.
    - XD3 :
      - x32lkit3 : (*OPT*) XFree86 3.2 Linkkit 3/3.
    - XD4 :
      - x32stat : (*OPT*) bibliothèques statiques pour XFree86 3.2.
  - XV : *XView pour les adorateurs du Look OPEN LOOK*
    - XV1 :
      - xv32_so : (*ADD*) bibliothèques dynamiques pour xview3.2p1-X11R6 ;
      - xvol32 : (*ADD*) binaires pour XView 3.2p1.
    - XV2 :
      - xv32_a : (*OPT*) bibliothèques statiques xview3.2p1-X11R6 ;
      - xvinc32 : (*OPT*) fichiers d'en-tête xview3.2p1-X11R6 ;
      - xvmenus : (*ADD*) menus et fichiers de configuration pour OpenLook ;

- workman : (*OPT*) (1.2.2a) programme de manipulation de CD-ROM audios ;
- xv32exmp : (*OPT*) programmes de démonstrations.

– XV3 :

- sspkg : Extention SlingShot (2.1). (*OPT*)

– Y : *Jeux*

– Y1 :

- bsdgames, lizards, sastroid et tetris. (*OPT*)

Ca y est, c'est fini !

5.3.3 Bogues de la Slackware

La Slackware, comme toute distribution, possède un certain nombre de bogues. Une liste a été faite et peut-être consultée à l'adresse suivante :

`http://www.cuug.ab.ca:8001/~barkers/programs/slackware-fixes/`

Elle est mise à jour assez régulièrement, et je ne vais pas la détailler ici.

5.4 RedHat : description

5.4.1 Disquette de boot

La RedHat utilise un système très différent de la Slackware : elle met à profit les modules chargeables pour l'amorçage de la machine. Toutefois, il est à noter qu'elle permet pas d'être installée à partir de disquettes (et dans ce cas, la Slackware s'impose). Enfin, sa dernière limitation est qu'elle impose d'avoir une partition Linux native (ext2fs).

Il n'existe en fait qu'une seule disquette de boot : `images/boot.img`. Si tu installes ta machines depuis un Pc utilisant du PCMCIA, tu devras utiliser une disquette suppémentaire : `images/supp.img`. Attention : dans la première version du CD de la RedHat 4.0, il y a des problèmes avec les cartes SCSI Adaptec 1520 et 1740. Dans ce cas, tu devras récupérer une version plus récente de l'image de boot.

Une fois que tu as copié la disquette de boot (et enventuellement la seconde si nécessaire), tu bootes... et tu tombes directement sur le système d'installation. Difficile de faire plus simple ! De plus, ce système de paquetage permet de gérer les dépendances entre les bibliothèques et les outils.

5.4.2 Distribution

La distribution RedHat est composée d'un ensemble de paquetages (un peu plus de 430) regroupés sous une forme de centres d'intérets (en fait une arborescence de groupes). Tu trouveras ci-dessous la liste des paquetages par ordre alphabétique de groupes qui peuvent ainsi être installés sur ta machine. Les noms de groupes sont conservés en anglais car ils sont tels quels sur ta machine, et sont en gras.

La manipulation « directe » des paquetages (archives rpm) est détaillée dans la prochaine section.

Il est a noter qu'il existe bien souvent un répertoire du nom de `upgrade` dans lequel se trouvent des fichiers qui permettent de mettre à jour (évolution et correction de bogues éventuelles) la distribution.

- **Applications**
  - **Communications** :
    - efax (0.7a r6) : programme d'envoie et de réception de FAX via des modems de classe 1 ou 2 ;
    - ircii (2.8.2 r7) : client IRC (Internet Relay Chat) ;
    - ircii-help (2.8.2 r7) : fichiers d'aide pour le paquetage `ircii` ;
    - lrzsz (0.12a r7) : ensemble de commandes permettant de télécharger des fichiers en utilisant les protocoles Z, X et Y ;
    - minicom (1.75 r2) : programme de communication ressemblant au programme Telix MsDos. Emulation VT100 et ANSI.
  - **Databases**
    - mb (5.0 r6) : base de données relationnelle (MetalBase). Système client serveur.
  - **Editors**
    - ed (0.2 r5) : éditeur en ligne GNU ;
    - jed (0.97.14 r2) : petit éditeur basé sur slang ;
    - jed-xjed (0.97.14 r2) : version de `jed` pour X Window ;
    - joe (2.8 r6) : petit éditeur de texte très pratique ;
    - vim (4.2 r5) : éternel `vi`... un peu amélioré ;
    - vim-X11 (4.2 r5) version X Windows de `vim`.
  - **Editors/Emacs**
    - emacs (19.34 r1) : éditeur de texte ultra puissant, qui permet également de faire son courrier électronique, poster des niouses, faire du ftp, ...
    - emacs-el (19.34 r1) : sources des fichiers lisp pour `emacs` ;
    - emacs-nox (19.34 r1) : `emacs`, version mode texte ;
    - emacs-X11 (19.34 r1) : `emacs`, version X Window ;
  - **Emulators**
    - dosemu (0.63.1.66 r2) : émulteur de programmes MS-DOS ;
    - xdosemu (0.63.1.66 r2) : version X Window de `dosemu`.
  - **Engineering**
    - spice (2g6 r4) : programme de simulation de circuits électroniques ;
    - units (1.0 r5) : outil de conversion d'unités.
  - **Graphics**
    - ghostscript (3.33 r2) : interpréteur PostScript et PDF pour X Window, et pour bon nombre d'imprimantes ;
    - ghostscript-fonts (4.0 r1) : polices de caractères utilisées par `ghostscript` ;

  - giftrans (1.11.1 r4) : conversion et manipulation de fichiers au format gif ;
  - libgr-progs (2.0.9 r4) : bibliothèque de gestion des formats FBM, JPEG, PBM, PGM, PNM, PPM, REL, et TIF ;
  - netpbm (1mar1994 r6) : outils de manipulation et de transformation d'images ;
  - xfig (3.1.4 r6) : outil de création de graphiques vectoriels ;
  - zgv (2.7 r5) : outil d'affichage d'images.
- **Mail**
  - elm (2.4.25 r5) : gestionnaire de courrier électronique en mode texte très puissant. Gère en plus le format MIME.
  - exmh (1.6.7 r3) : interface graphique au système de courrier électronique `mh` ;
  - mailx (5.5.kw r6) : programme d'envoie de courrier électronique de base utilisé par de nombreux scripts ;
  - metamail (2.7 r6) : implémentation de MIME ;
  - mh (6.8.3 r11) : système de gestion de courrier électronique (gérant POP) ;
  - pine (3.95 r2) : client de courrier et de niouses ;
  - popclient (2.21 r6) : utilisé pour récupérer son courrier via un serveur POP.
- **Math**
  - bc (1.03 r6) : calculatrice en mode texte ;
  - gnuplot (3.5 r5) : logiciel d'affichage de courbes.
- **Networking**
  - lynx (2.5 r5) : arpenteur WWW en mode texte ;
  - ncftp (2.3.0 r4) : client `ftp` plus simple et plus puissant que `ftp` ;
  - tcpdump (3.0.2 r5) : utilisé pour déboguer une liaison réseau, ainsi que pour sécuriser un système.
- **News**
  - slrn (0.8.8.4 r1) : lecteur de news ;
  - tin (1.22 r5) : lecteur de news ;
  - trn (3.6 r8) : lecteur de news.
- **Productivity**
  - ical (2.0p2 r7) : calendrier et agenda graphique ;
- **Publishing**
  - groff (1.10 r6) : programme de formatage de texte. Utilisé par toutes les pages de manuel ;
  - groff-gxditview (1.10 r6) : programme permettant de consulter les documents `groff` sous X ;
  - linuxdoc-sgml (1.5 r5) : ensemble d'outils permettant de transformer du SGML en différents formats. Ce programme est utilisé pour formater tous les documents du LDP, dont ce guide.
  - lout (3.08 r1) : langage de formatage de texte ;
  - lout-doc (3.08 r1) : documentation concernant `lout` ;
  - texinfo (3.7 r5) : outils de formatage des documents au format `texinfo`.

- **Publishing/TeX**
  - bm2font (3.0 r6) : convertit des bitmaps en fontes TeX ;
  - tetex (0.3.4 r3) : version tetex de TeX ;
  - tetex-afm (0.3.4 r3) : polices postscripts ;
  - tetex-dvilj (0.3.4 r3) : convertisseurs pour HP-GL ;
  - tetex-dvips (0.3.4 r3) : convertisseur dvi vers PostScript ;
  - tetex-latex (0.3.4 r3) : paquetage des macros LaTeX ;
  - tetex-xdvi (0.3.4 r3) : programme `xdvi`.
- **Sound**
  - aumix (0.2 r4) : mixeur en mode texte ;
  - cdp (0.33 r5) : permet de jouer des CD audios en ligne de commande ou via une interface en mode texte ;
  - maplay (1.2 r4) : joue des fichiers au format MPEG-2 sur la carte son ;
  - playmidi (2.3 r6) : joue des fichiers Midi ;
  - playmidi-X11 (2.3 r6) : interface X Window pour `playmidi` ;
  - sox (11g r5) : programme de conversion de fichiers audio ;
  - tracker (4.3 r4) : permet de jouer des fichiers au format Amiga (MOD).

- **Base**
  - crontabs (1.3 r1) : fichier crontab de root ;
  - dev (2.3 r1) : fichiers périphériques du répertoire `/dev/` ;
  - etcskel (1.1 r1) : fichiers qui vont dans `/etc/skel` et qui sont utilisés lors de la création d'un nouveau compte utilisateur ;
  - filesystem (1.2 r1) : arborescence de base de la Red Hat ;
  - initscripts (2.73 r1) : scripts d'amorçage de la machine ;
  - logrotate (1.0 r2) : programme de rotation des fichiers de log ;
  - mailcap (1.0 r3) : paquetage `mailcap` ;
  - pam (0.50 r17) : outils pour les services d'identifications *Pluggable Authentication Modules* ;
  - pamconfig (0.50 r5) : fichiers de configuration pour `pam` ;
  - passwd (0.50 r2) : programme de changement de mots de passe, utilisant PAM ;
  - redhat-release (4.0 r1) : fichier de version de la RedHat ;
  - rootfiles (1.3 r1) : fichiers d'initialisation pour le super-utilisateur ;
  - setup (1.5 r1) : fichiers importants de configuration ;
  - termcap (9.12.6 r5) : fichier `/etc/termcap` pour la gestion des terminaux ;
  - **Kernel**
    - iBCS (2.0 r2) : modules permettant l'exécution de programmes au format IBCS (*Intel Binary Compatibility Standard*) ;

  - kernel (2.0.18 r5) : noyau Linux version 2.0.18 ;
  - kernel-headers (2.0.18 r5) : fichiers d'en-tête des sources ;
  - kernel-modules (2.0.18 r5) : modules chargeables ;
  - kernel-source (2.0.18 r5) : sources du noyau ;
  - umsdos_progs (0.9 r2) : utilitaires pour utiliser UMS-Dos.

- **Daemons**
  - at (2.9b r2) : lanceur différé de programmes ;
  - bdflush (1.5 r5) : vide les tampons du système ;
  - gpm (1.10 r5) : gestionnaire de souris en mode texte ;
  - pcmcia-cs (2.8.22 r5) : gestionnaires pour cartes pcmcia ;
  - procmail (3.10 r9) : utilisé pour (par exemple) filtrer son courier électronique ;
  - sendmail (8.7.5 r5) : programme d'envoie et de réception de courrier électronique ;
  - sendmail-cf (8.7.5 r5) : fichiers de configuration pour `sendmail` ;
  - sendmail-doc (8.7.5 r5) : documentation pour `sendmail` ;
  - sysklogd (1.3 r9) : programme de log des messages systèmes ;
  - SysVinit (2.64 r2) : programme de contrôle de l'amorçage de la machine, etc.
  - uucp (1.06.1 r6) : système de communication entre machine permettant le transfert de courrier, de news ou de fichiers ;
  - vixie-cron (3.0.1 r9) : ordonnanceur de programmes.

- **Development**
  - **Building**
    - autoconf (2.10 r6) : outil de configuration automatique de programmes en fonction de la plateforme ;
    - make (3.74 r8) : gestionnaire de projets et de compilation ;
    - pmake (1.0 r4) : version étendue de `make`.
  - **Debuggers**
    - ElectricFence (2.0.5 r4) : bibliothèque d'aide au débogage d'applications ;
    - gdb (4.16 r5) : débogueur ;
    - strace (3.1 r3) : affiche les appels systèmes utilisés par une application ;
    - xxgdb (1.0.8 r5) : interface X Window à `gdb`.
  - **Languages**
    - basic (1.20 r7) : interpréteur basic ;
    - bin86 (0.3 r3) : assembleur et éditeur de liens ;
    - gcc (2.7.2 r9) : compilateur C ;
    - gcc-c++ (2.7.2 r9) : compilateur C++ ;
    - gcc-objc (2.7.2 r9) compilateur objective C ;

  - p2c-devel (1.20 r7) : convertisseur Pascal vers C ;
  - python (1.3 r11) interpréteur du langage orienté objet python ;
  - umb-scheme (3.2 r1) : interpréteur scheme ;
  - xlispstat (3.44 r5) : implémentation du langage lisp pour X Window.
- **Languages/fortran**
  - f2c (19960205 r6) : convertisseur Fortran vers C.
- **Languages/Fortran**
  - fort77 (1.11 r1) : périphérique utilisé par `f2c`.
- **Languages/Tcl**
  - blt (1.9 r3) : extention pour Tcl/Tk ;
  - blt-devel (1.9 r3) : environnement de développement pour BLT ;
  - expect (5.19-5.20beta r3) : outil de gestion des connexions telnet, ftp, ... ;
  - tcl (7.5 r3) : langage inteprété TCL ;
  - tclx (7.5.2 r3) : extention du langage TCL ;
  - tix (4.0.5 r3) : extention pour Tk ;
  - tk (4.1 r3) : ensemble de widgets conçus pour fonctionner avec Tcl.
- **Libraries**
  - cracklib (2.5 r1) : vérifie les mots de passes ;
  - e2fsprogs-devel (1.04 r8) : bibliothèque et fichiers d'en-tête pour les programmes de gestion du système de fichiers ext2fs ;
  - gpm-devel (1.10 r5) : environnement de développement permettant de développer des applications utilisant `gpm` ;
  - libg++-devel (2.7.1.4 r4) : implémentation GNU des bibliothèques C++ ;
  - libgr-devel (2.0.9 r4) : utile pour développer des applications utilisant la bibliothèque `libgr` ;
  - ncurses-devel (1.9.9e r2) : fichiers d'en-tête et bibliothèques nécessaires pour développer avec `ncurses` ;
  - pythonlib (1.8 r1) : code utilisé par certains programmes Red Hat ;
  - rpm-devel (2.2.5 r1) : fichiers d'en-tête et bibliothèques nécessaires pour manipuler les fichiers `rpm` ;
  - slang-devel (0.99.33 r1) : bibliothèques et fichiers d'en-tête pour travailler avec le langage slang ;
  - typhoon (1.10.3 r4) : système de gestion de bases de données relationnelles ;
  - xpm-devel (3.4h r3) : bibliothèques et fichiers d'en-tête pour manipuler les fichiers xpm ;
  - xview-devel (3.2p1.pl2 r3) : utilisé pour développer des application utilisant `xview` ;
  - xview-devel-examples (3.2p1.pl2 r3) : exemples de programmes `xview` ;
  - zlib-devel (1.0.4 r1) : bibliothèque de *d*compression à la volée.
- **Libraries/Libc**
  - libc-debug (5.3.12 r8) : bibliothèque C compilée pour être utilisée lors de débogages ;

    - libc-devel (5.3.12 r8) : utilisé lors de développement de programmes qui utilisent la bibliothèque C ;
    - libc-profile (5.3.12 r8) : utilisé lorsque les programmes utilisent `gproff` ;
    - libc-static (5.3.12 r8) : version statique de la bibliothèque C.
  - **Tools**
    - binutils (2.7.0.2 r4) : ensemble de programmes nécessaires à la compilation ;
    - bison (1.25 r1) : analyseur grammatical GNU ;
    - byacc (1.9 r4) : analyseur grammatical ;
    - cdecl (2.5 r3) : programme permettant de convertir de l'anglais en déclarations de fonctions et vice versa ;
    - cproto (4.4 r4) : génération de prototypes de fonctions ;
    - flex (2.5.3 r3) : analyseur lexical ;
    - gencat (022591 r3) : programme de catalogues de messages ;
    - gettext (0.10 r5) : outils et bibliothèques de manipulation de messages ;
    - indent (1.9.1 r5) : programme d'indentation de sources C ;
    - xwpe (1.4.2 r9) : environnement de développement.
  - **Version Control**
    - cvs (1.8.1 r1) : gestionnaire de versions ;
    - rcs (5.7 r4) gestionnaire de versions.
- **Documentation**
  - faq (4.0 r1) : ensemble de *Frequently Asked Questions* concernant Linux ;
  - howto (4.0 r1) : howto sur Linux, version texte ;
  - howto-dvi (4.0 r1) : howto sur Linux, version dvi ;
  - howto-html (4.0 r1) : howto sur Linux, version html ;
  - howto-ps (4.0 r1) : howto sur Linux, version PostScript ;
  - howto-sgml (4.0 r1) : howto sur Linux, version sgml (format original) ;
  - indexhtml (4.0 r1) : page d'index Red Hat ;
  - ldp (4.0 r2) : contenu du *Linux Documentation Project* au format html ;
  - man-pages (1.12 r1) : pages de manuel (sections 1 à 8).
- **Extensions**
  - **Japanese**
    - kterm (6.1.0 r5) : il s'agit du *Kanji Terminal Emulator*.
- **Games**
  - abuse (1.10 r5) : jeu genre Super Mario ;
  - bsd-games (1.3 r6) : ensemble de jeux en mode texte ;

  - christminster (3 r2) : jeu d'aventure en mode texte ;
  - colour-yahtzee (1.0 r3) : version texte du jeu yahtzee ;
  - doom (1.8 r6) : vous ne connaissez pas ;-)?
  - fortune-mod (1.0 r3) : programme de génération aléatoire de messages, qui se met généralement au login ;
  - gnuchess (4.0.pl77 r1) : jeu d'échec GNU, mode texte ou X Window ;
  - koules (1.2 r2) : jeu d'arcade utilisant SvgaLib ;
  - koules-sound (1.2 r2) : sons du jeu `koules` ;
  - mysterious (1.0 r2) : jeu d'aventure en mode texte ;
  - pinfocom (3.0 r3) : interpréteur pour les jeux *Infocom* ;
  - scottfree (1.14 r2) : interpréteur pour les jeux *Scott-Adams* ;
  - trojka (1.1 r7) : jeu de blocs ;
  - vga_cardgames (1.3.1 r6) : jeux de cartes divers ;
  - vga_gamespack (1.3 r5) : jeux divers ;
  - vga_tetris (0.4 r3) : tétris ;
  - xkoules (1.2 r2) : interface X Window à `koules`.

- Libraries
  - aout-libs (1.4 r5) : vieilles bibliothèques ;
  - db (1.85 r10) : routines pour créer des bases de données très simples. Utilisé par Python, Perl, ...
  - db-devel (1.85 r10) : bibliothèques et fichiers d'en-tête pour manipuler des fichiers au format `db` ;
  - f2c-libs (19960205 r6) : bibliothèques partagées pour `f2c` ;
  - faces-devel (1.6.1 r5) : environnement de développement de `xface` ;
  - gdbm (1.7.3 r8) : bibliothèque de base de données ;
  - gdbm-devel (1.7.3 r8) : bibliothèques et fichiuers d'en-tête pour `gdbm` ;
  - ld.so (1.7.14 r4) : chargeur dynamique ;
  - libc (5.3.12 r8) : bibliothèques standards C ;
  - libelf (0.5.2 r4) : permet d'accéder au format ELF des exécutables ;
  - libg++ (2.7.1.4 r4) : implémentation GNU des bibliothèques standards C++ ;
  - libgr (2.0.9 r4) : bibliothèque de gestion des formats FBM, JPEG, PBM, PGM, PNM, PPM, REL, et TIFF ;
  - libpng (0.89c r1) : bibliothèque de gestion du format PNG ;
  - libpng-devel (0.89c r1) : fichiers d'en-tête et bibliothèques statiques pour utiliser la bibliothèque `png` ;
  - libtermcap (2.0.8 r2) : bibliothèque pour accéder à la base de données des terminaux ;
  - libtermcap-devel (2.0.8 r2) : bibliothèques de développement ;

- ncurses (1.9.9e r2) : bibliothèque d'affichage en mode texte ;
- newt (0.6 r1) : toolkit en mode texte de fenêtrage ;
- newt-devel (0.6 r1) : fichiers d'en-tête et bibliothèques ;
- p2c (1.20 r7) : convertisseur Pascal vers C ;
- readline (2.0 r10) : biblothèque de lecteure de ligne ;
- readline-devel (2.0 r10) : environnement de développement pour `readline` ;
- slang (0.99.33 r1) : interpréteur de langage ;
- svgalib (1.2.10 r2) : bibliothèque graphique en mode SVGA ;
- svgalib-devel (1.2.10 r2) : bibliothèques et fichiers d'en-tête pour développer des applications utilisant la bibliothèque `svgalib` ;
- zlib (1.0.4 r1) : bibliothèque de compression et de décompression à la volée.

- **Networking**
  - NetKit-B (0.08 r4) : fournis des clients et des serveurs pour les services standards Unix ;
  - pidentd (2.5.1 r5) : serveur d'identification ;
  - samba (1.9.16p2 r2) serveur SMB ;
  - ytalk (3.0.2 r4) : extension de `talk` ;
  - **Admin** :
    - anonftp (2.3 r1) : contient les fichiers nécessaires à l'installation d'un serveur ftp anonyme ;
    - net-tools (1.32.alpha r2) : outils de base nécessaire pour la configuration du réseau ;
    - nfs-server-clients (2.2beta16 r3) : clients NFS ;
    - tcp_wrappers (7.4 r3) : outil d'observation et de contrôle de connexions externes ;
  - **Daemons**
    - amd (920824upl102 r5) : démon automonteur (Berkeley) ;
    - apache (1.1.1 r3) : serveur Web ;
    - bind (4.9.4 r3) : serveur de nom (DNS) ;
    - bootp (2.4.3 r2) : serveur de requêtes bootp et DHCP ;
    - cmu-snmp (3.2 r2) : dérivé du *Simple Network Management Protocol* ;
    - gn (2.24 r2) : serveur gopher ;
    - imap (3.6.BETA r6) : serveur protocoles POP et IMAP ;
    - intimed (1.10 r3) : serveur de temps ;
    - mars_nwe (0.98 r4) : serveur de fichier et d'impression compatible NetWare ;
    - nfs-server (2.2beta16 r3) : serveur NFS ;
    - portmap (4.0 r3) : gestion des connexions RPC ;
    - ppp (2.2.0f r2) : démon PPP ;
    - wu-ftpd (2.4.2b11 r8) : démon ftp ;
    - yppasswd (0.8 r6) : implémentation de `passwd` pour les pages jaunes (NIS) ;

    - ypserv (1.0.4 r3) : serveur NIS (ou pages jaunes) ;
  - **News**
    - inews (1.4unoff4 r2) : utilisé par certains lecteurs de news pour poster ;
    - inn (1.4unoff4 r2) : serveur de news ;
  - **Utilities**
    - bind-utils (4.9.4 r3) : utilitaires d'accès aux services de nom ;
    - bootpc (050 r2) : client bootp pour Linux ;
    - cmu-snmp-devel (3.2 r2) : bibliothèque et fichiers d'en-tête pour CMU SNMP ;
    - cmu-snmp-utils (3.2 r2) : utilitaires pour CMU SNMP ;
    - dip (3.3.7o r8) : programme d'automatisation des connexions SLIP et PPP ;
    - fwhois (1.00 r5) : programme `whois` ;
    - ipxutils (1.0 r1) : programmes pour protocole IPX ;
    - ncpfs (2.0.5 r2) : programmes pour configurer et utiliser le système de fichiers ncpfs ;
    - rdate (10.10.94 r4) : programme de mise à jour de l'horloge de la machine par rapport à des machines distantes ;
    - rdist (1.0 r5) : programme de recopie de fichiers entre machines ;
    - statnet (2.00 r4) : indicateur de traffic réseau ;
    - traceroute (1.0.4.4bsd r2) : trace la route empruntée par des paquets ;
    - yp-clients (2.2 r5) : clients NIS.
- **Shells**
  - ash (0.2 r5) : interpréteur de commandes clone de celui de Berkeley ;
  - bash (1.14.7 r1) : interpréteur de commandes conforme à IEEE 1003.2 ;
  - csh (5.2.6 r5) : C-Shell Berley ;
  - mc (3.2.10 r1) : Midnight Commanders ;
  - pdksh (5.2.8 r1) : reimplémentation de `ksh` ;
  - tcsh (6.06 r8) : extension de `csh` ;
  - zsh (3.0pre6 r1) : interpréteur de commandes.
- **Utilities**
  - **Archiving**
    - cpio (2.4.2 r3) : utilitaires d'archivage de fichiers ;
    - gzip (1.2.4 r5) : programmes de compression de fichiers ;
    - lha (1.00 r4) : programme d'archivage et de compression de fichiers ;
    - ncompress (4.2.4 r6) : outil de compression compatible avec le `compress` original ;
    - tar (1.11.8 r8) : programme d'archivage de données ;
    - unarj (2.41a r3) : programme inverse de `arj` ;
    - unzip (5.12 r5) : décompression de fichiers `zip` ;

  - zip (2.1 r1) : compression de fichiers au format `zip`.
- **Console**
  - open (1.3 r5) : lance une commande sur une console virtuelle ;
  - vlock (1.0 r4) : vérouille le terminal courant, ou le système entier de consoles.
- **File**
  - file (3.20 r3) : programme d'identification de fichiers ;
  - fileutils (3.13 r2) : programmes de gestion de fichiers ;
  - findutils (4.1 r9) : programme de recherche de fichiers sur la machine ;
  - git (4.3.11 r4) : gestionnaire de fichiers en ASCII ;
  - macutils (2.0b3 r4) : ensemble d'outils pour manipuler les fichiers provenant d'un Mac ;
  - mtools (3.0 r3) : ensemble d'outils pour manipuler des systèmes de fichiers Ms-Dos ;
  - sharutils (4.2 r4) : utilitaires utilisés pour encoder des fichiers ;
  - smbfs (0.4 r1) : outils pour monter des systèmes de fichiers à partir de serveurs SMB ;
  - stat (1.5 r5) : affiche des informations sur le système de fichiers ;
  - symlinks (1.0 r5) : recherche d'éventuels problèmes avec les liens symboliques sur le système de fichiers ;
  - tree (1.0 r3) : portage Unix du programme Ms-Dos du même nom ;
  - which (1.0 r5) : indique si un programme se trouve dans votre PATH.
- **Printing**
  - mpage (2.3 r1): transforme plusieurs pages ASCII en un seul fichier PostScript.
- **System**
  - adduser (1.2 r1) : permet de rajouter un utilisateur à ton système ;
  - adjtimex (1.2 r4) : système de gestion de l'horloge ;
  - control-panel (2.5 r2) : gestionnaire de configuration Red Hat ;
  - cracklib-dicts (2.5 r1) : dictionnaires pour le programme `crack` ;
  - dump (0.3 r5) : outils de sauvegardes pour systèmes de fichiers ext2 ;
  - e2fsprogs (1.04 r8) : ensemble de programmes pour créer et maintenir les systèmes de fichiers ext2 ;
  - eject (1.3 r3) : programmes permettant d'éjecter les CD-ROM et les disquettes sous Sparc ;
  - ext2ed (0.1 r8) : permet de manipuler le système de fichier ext2. Uniquement pour les *hackers* ;
  - fstool (2.5 r1) : programme X de manipulation du fichier `/etc/fstab` ;
  - gcal (1.00 r1) : programme de calculatrice étendu ;
  - getty_ps (2.0.7h r3) : permettent d'accepter des login à partir de la console, terminaux ou modems ;
  - glint (2.1.4 r1) : interface graphique aux fichiers rpm ;
  - hdparm (3.1 r1) : utilitaire de configuration des disques dur ;
  - helptool (2.2 r1) : interface graphique de recherche ;
  - info (3.7 r5) : arpenteur pour fichiers *info* ;

- ipfwadm (2.3.0 r2) : outils d'administration pour firewall ;
- kbd (0.91 r8) : utilitaires pour modifier la police de caractères de la console, ainsi que le type de clavier utilisé ;
- kbdconfig (1.3 r1) : reconfigure la machine pour un nouveau type de clavier ;
- kernelcfg (0.3 r2) : interface graphique d'administration de la machine ;
- lilo (0.19 r1) : lanceur de Linux autorisant le multi-boot ;
- losetup (2.5l r1) : programme de configuration de *loopback device* ;
- lpr (0.10 r1) : outils d'impression ;
- MAKEDEV (2.2 r5) : fichiers `/dev/` ;
- man (1.4h r2) : pages de manuels ;
- mingetty (0.9.4 r1) : `getty` réduit uniquement pour la console virtuelle ;
- mkdosfs-ygg (0.3b r4) : permet de créer des systèmes de fichiers Ms-Dos ;
- mkinitrd (1.0 r1) : permet de créer un module chargeable pour le boot des machines ayant une carte SCSI ;
- mkisofs (1.03a r3) : paquetage utilisé pour créer des systèmes de fichiers iso9660 ;
- modemtool (1.1 r2) : outil de configuration graphique du modem ;
- modules (2.0.0 r4) : programme de chargement et de déchargement des modules chargeables ;
- mount (2.5l r1) : programme de montage des systèmes de fichiers ;
- mouseconfig (1.4 r1) : système de configuration en mode texte des souris ;
- mt-st (0.4 r2) : programme de manipulation des périphériques de sauvegardes ;
- netcfg (2.10 r1) : interface graphique d'administration réseau ;
- printtool (2.6 r1) : fournis une interface graphique pour configurer la file d'impression ;
- procinfo (0.9 r1) : paquetage permettant d'exploiter le contenu de `/proc` ;
- procps (1.01 r7) : paquetage d'outils permettant de connaître l'état de la machine ;
- psacct (6.2 r1) : système d'accounting de processus ;
- psmisc (11 r4) : affichage d'arborescence de processus ;
- quota (1.51 r3) : outils de gestion des quotas ;
- rpm (2.2.5 r1) : programme pour gérer les paquetages du même nom ;
- sh-utils (1.12 r9) : shell ;
- shadow-utils (960530 r6) : programmes d'utilisation des mots de passe shadow ;
- sliplogin (2.1.0 r2) : outils pour SLIP ;
- statserial (1.1 r7) : affiche la table des communications séries ;
- swatch (2.1 r4) : outil d'observation des fichiers de log ;
- taper (6.7.4 r1) : programme de sauvegarde ;
- time (1.7 r1) : programme affichant des statistiques sur les exécution des programmes ;
- timeconfig (1.4 r1) : outis de configuration de l'heure ;
- timetool (2.2 r1) : interface graphique pour `timeconfig` ;
- tksysv (0.91 r1) : outils graphiques pour manipuler les niveaux d'initalisation de la machine ;
- tmpwatch (1.1 r1) : permet de nettoyer certains répertoires ;

  - tunelp (1.3 r5) : configuration de l'imprimante ;
  - usercfg (2.3 r1) : outil graphique de gestion des utilisateurs ;
  - util-linux (2.5 r26) : ensemble d'outils nécessaires à Linux pour son administration ;
  - zoneinfo (96i r3) : outils et fichiers de configuration pour le fuseau horaire ainsi que pour les changement d'heures.
- **Terminal**
  - dialog (0.6 r6) : permet de construire des interfaces en mode texte ;
  - screen (3.7.1 r1) : permet d'avoir plusieurs fenêtre virtuelle sur le même terminal.
- **Text**
  - diffstat (1.25 r1) : statistiques sur la comparaison de fichiers ;
  - diffutils (2.7 r5) : outil de comparaison de fichiers ;
  - faces (1.6.1 r5) : transformation d'images en fichiers textes ;
  - faces-xface (1.6.1 r5) : outils pour gérer les en-têtes de courrier X-Faces ;
  - gawk (3.0.0 r5) : version GNU du programme `awk` ;
  - grep (2.0 r4) : programme de recherche de chaînes dans des fichiers ;
  - ispell (3.1.20 r3) : vérificateur d'orthographe ;
  - less (321 r3) : version GNU de `more` ;
  - locale (5.3.12 r8) : outils nécessaires à la gestion du multi-linguisme ;
  - m4 (1.4 r6) : expandeur de macro-commandes ;
  - mawk (1.2.2 r5) : version de `awk` ;
  - nenscript (1.13++ r5) : filtre d'impression ;
  - patch (2.1 r4) : permet de faire l'opération inverse de `diff` ;
  - perl (5.003 r4) : langage interprété ;
  - sed (2.05 r6) : programme de transformation de fichiers à la volée ;
  - textutils (1.19 r2) : outils de gestion de fichiers textes ;
  - words (2 r3) : dictionnaire anglais.

- **X11**
  - **Amusements**
    - multimedia (2.1 r2) : programmes pour jouer des CD audios ;
    - xearth (1.0 r4) : affichage d'un globe 3D en fond d'écran ;
    - xfishtank (2.0 r4) : un aquarium en fond d'écran ;
    - xsnow (1.40 r1) : chutes de neige en fond d'écran.
  - **Applications**
    - seyon (2.14c r6) : paquetage de télécommunication ;
    - xfm (1.3.2 r4) : gestionnaire de fichiers ;
    - xrn (8.02 r2) : lecteur de forum Usenet ;
    - xterm-color (1.1 r2) : `xterm` en couleur ;

- **Applications/Development**
  - xwpe-X11 (1.4.2 r9) : ensemble de programmes d'édition de texte.
- **Applications/Graphics**
  - ghostview (1.5 r5) : outil de visualisation de fichiers PostScript ;
  - ImageMagick (3.7 r5) : outils de visualisation et de manipulation d'images. Il est très puissant, rapide et surtout gère un bon nombre de formats d'image.
  - mxp (1.0 r4) : générateur de fractales de Mandelbrot ;
  - transfig (3.1.2 rb) : outil de conversion d'images ;
  - xanim (27063 r1) : programme d'animation graphique, donc QuickTime et FLiC ;
  - xloadimage (4.1 r5) : visualisateur d'images ;
  - xmorph (1996.07.12 r1) : programme de morphing ;
  - xmplay (1.0 r6) : permet de visualiser les animations MPEG ;
  - xpaint (2.4.4 r2) : logiciel de dessin bitmap ;
  - xv (3.10a r5) : permet de visualiser bon nombre de formats d'image.
- **Applications/Networking**
  - arena (0.98.beta3 r1) : arpenteur Wev HTML 3 ;
  - x3270 (3.1.0.7 r3) : émulateur de terminaux IBM 3270 ;
  - xgopher (1.3.3 r1) : outil graphique pour utiliser le service gopher.
- **Games**
  - acm (4.7 r4) : simulateur de vol ;
  - xzip (140 r3) : transforme les applications text en applications X.
- **Games/Strategy**
  - spider (1.0 r3) : solitaire ;
  - xboard (3.2.pl0 r5) : interface graphique à Gnu Chess ;
  - xdemineur (1.1 r4) : démineur ;
  - xgammon (0.98 r6) : backgammon ;
  - xpat2 (1.04 r2) : autre solitaire ;
  - xpilot (3.5.1 r2) : jeu réseau de combat... enfin bref, essayez ;-) !
  - xpuzzles (5.3.1 r1) : assortiement de puzzles.
- **Games/Video**
  - cxhextris (1.0 r4) : version couleur TeTris ;
  - flying (6.20 r2) : ensemble de petits jeux ;
  - paradise (2.3p19 r4) : jeu d'arcade ;
  - xbill (1.1 r3) : où comment empécher Bill Gates de conquérir le monde... ;
  - xbl (1.0f r4) : Tetris 3D ;
  - xboing (2.3 r3) : une version de Breakout ;
  - xchomp (1.0 r5) : jeu d'arcade ;
  - xevil (1.5 r2) : jeu d'aventure ;

  - xgalaga (1.6c r1) : clone de Galaga... un jeu d'arcade ;
  - xjewel (1.6 r4) : divers ;
  - xlander (1.2 r4) : le but est de poser une fusée ;
  - xtetris (2.6 r4) : Tetris ;
  - xtrojka (1.2.2 r1) : genre xjewel et tetris.
- **Libraries**
  - ImageMagick-devel (3.7 r5) : bibliothèques et fichiers d'en-tête pour créer des applications utilisant le code d'ImageMagick ;
  - nls (1.0 r2) : utilisé par certains vieux binaires X11R5 comme Netscape ;
  - Xaw3d (1.3 r3) : bibliothèque Xaw semblant être en 3 dimensions ;
  - Xaw3d-devel (1.3 r3) : bibliothèques statiques et fichiers d'en-tête ;
  - xpm (3.4h r3) : bibliothèque de manipulation de fichiers Xpm ;
  - xview (3.2p1.pl2 r3) : ensemble d'outil pour avoir le look OpenLook.
- **Utilities**
  - moonclock (1.0 r5) : affiche l'heure, ainsi que l'état de la lune ;
  - procps-X11 (1.01 r7) : outils de visualisation de l'état de la machine ;
  - rxvt (2.18 r2) : émulateur VT100 ;
  - Xconfigurator (2.4 r3) : outil de configuration X de RedHat ;
  - xdaliclock (2.07 r1) : horloge numérique ;
  - xlockmore (3.9 r4) : version améliorée de `xlock` ;
  - xmailbox (2.4 r3) : programme indiquant que du courrier vient d'arriver ;
  - xosview (1.4.1 r2) : statistique de fonctionnement de la machine et de trafic réseau ;
  - xscreensaver (1.26 r3) : économiseur d'écran ;
  - xsysinfo (1.5 r1) : programme de statistiques sur la machine ;
  - xtoolwait (0.3 r2) : divers ;
  - xwpick (2.20 r4) : fait une copie fichier d'une fenêtre.
- **Window Managers**
  - fvwm (1.24 r9) : gestionnaire de fenêtres ;
  - fvwm95 (2.0.42a r5) : gestionnaire de fenêtre simulant l'environnement Windows 95 ;
  - fvwm95-icons (2.0.42a r5) : icones pour `fvwm` ;
  - TheNextLevel (1.0 r1) : ensemble de bureaux pour `fvwm` ;
  - xview-clients (3.2p1.pl2 r3) : clients et outils pour OpenWindows.
- **XFree86**
  - X11R6-contrib (3.1.2 r6) : ensemble de programmes pour X ;
  - XFree86 (3.1.2 r11) : bibliothèques X ;
  - XFree86-devel (3.1.2 r11) : fichiers d'en-tête et bibliothèques statiques pour X ;
  - XFree86-fonts (3.1.2 r11) : polices de caractères pour X ;

- **XFree86/Servers**
  - XFree86-8514 (3.1.2 r11) : serveur X pour cartes ATI ;
  - XFree86-AGX (3.1.2 r11) : serveur X pour cartes AGX ;
  - XFree86-Mach32 (3.1.2 r11) : serveur X pour cartes Mach32 ;
  - XFree86-Mach64 (3.1.2 r11) : serveur X pour cartes Mach64 ;
  - XFree86-Mach8 (3.1.2 r11) : serveur X pour cartes Mach8 ;
  - XFree86-Mono (3.1.2 r11) : serveur X pour cartes monochromes ;
  - XFree86-P9000 (3.1.2 r11) : serveur X pour cartes P9000 ;
  - XFree86-S3 (3.1.2 r11) : serveur X pour cartes S3 ;
  - XFree86-SVGA (3.1.2 r11) : serveur X pour cartes Super VGA ;
  - XFree86-VGA16 (3.1.2 r11) : serveur X pour cartes VGA 16 couleurs ;
  - XFree86-W32 (3.1.2 r11) : serveur X pour cartes W32 ;
  - XFree86-Xvfb (3.1.2 r11) : serveur qui tourne dans une fenêtre X.

C'est terminé ! Cette description détaille la version originale de la RedHat 4.0 (dite Colgate). Il existe aujourd'hui de nombreux paquetages (dans le répertoire `upgrade`) permettant de mettre à jour correctement la machine avec une facilité déconcertante.

5.4.3 Utilisation du programme `rpm`

Le programme `rpm` est une véritable petite merveille qui permet d'installer, de supprimer et de mettre à jour les différents paquetages de ta machine, sans trop se préocupper des problèmes de version de bibliothèques, etc.

Je ne vais pas détailler toutes les options de `rpm` car il y en a vraiment trop (voir `rpm --help` pour cela, ou la documentation fournie par Red Hat). Globalement, `rpm` possède quatre modes :

- **interrogation** (*-q*). Cette option permet principalement de connaître le contenu d'une archive `rpm`, la liste des paquetages installés sur la machine, etc.

  Par exemple, pour savoir les paquetages installés sur ta machine, il suffit de faire :

  ```
  gandalf #  rpm -q -a
  mailcap-1.0-3
  rpm-devel-2.2.5-1
  groff-1.10-6
  ```

  Pour avoir plus de renseignement sur les différents paquetages installés (numéro de version, quelle distribution, ...,), il suffit de rajouter en plus l'option `-l` et `-i`.

  Par exemple :

  ```
  gandalf # rpm -q -i -l rpm-devel
  Name        : rpm-devel                    Distribution: Red Hat Linux Colgate
  Version     : 2.2.5                              Vendor: Red Hat Software
  ```

```
Release     : 1                             Build Date: Thu Sep 05 23:14:10 1996
Install date: Wed Nov 13 11:30:50 1996      Build Host: porky.redhat.com
Group       : Development/Libraries         Source RPM: rpm-2.2.5-1.src.rpm
Size        : 136142
Summary     : Header files and libraries for programs that manipulate rpm packages
Description :
The RPM packaging system includes a C library that makes it easy to
manipulate RPM packages and databases. It is intended to ease the
creation of graphical package managers and other tools that need
intimate knowledge of RPM packages.
/usr/include/rpm
/usr/include/rpm/dbindex.h
/usr/include/rpm/header.h
/usr/include/rpm/messages.h
/usr/include/rpm/rpmerr.h
/usr/include/rpm/rpmlib.h
/usr/lib/librpm.a
```

Ces options peuvent être utilisées sur un fichier `rpm` que tu souhaites installer pour savoir ce qu'il contient : il suffit de rajouter l'option `-p` :

```
gandalf # rpm -q -l -p giftrans-1.11.1-4.i386.rpm
/usr/bin/giftrans
/usr/man/man1/giftrans.1
```

Si tu rajoutes l'option `-i`, tu obiendras les renseignements sur le paquetage.

– **installation** (*-i*).

Une fois que tu as trouvé le bon fichier à installer, tu souhaites probablement l'installer. Pour cela, il suffit de faire :

```
gandalf# rpm -i howto-sgml-4.0-1.i386.rpm
```

– **destruction**.

Installer c'est bien... mais comment supprimer un paquetage devenu inutile? C'est tout simple :

```
gandalf# rpm -e howto-sgml-4.0-1
```

– **mise à jour**.

Enfin, pour terminer cette courte introduction à l'utilisation des fichiers `rpm`, nous allons voir la mise à jour. Tu as donc installé quelques paquetages fort interressant, et oh surprise, une nouvelle version du programme vient de sortir.

Tu récupères donc la nouvelle version. Normalement, il convient de détruire l'ancienne et d'installer la nouvelle. Grâce au système `rpm`, cela est effectué tout seul :

```
gandalf# rpm -U howto-sgml-4.0-2.i386.rpm
```

Il existe bon nombre d'autres options. Consultes la documentation pour plus de renseignements.

6 Installation

Après ces préliminaires, on passe aux choses sérieuses. Dans un premier temps, tu fais une sauvegarde de ton disque dur, c'est achtement plus prudent, et tu mets à côté de toi une disquette formatée.

Ensuite, tu éteins ta machine, tu mets la disquette de boot.

6.1 Installation initiale

Bon, tu es prêt? C'est parti. Tu allumes ta machine. **Linux** boote sur la disquette *boot*. Dans le cas de la Slackware, il te demande d'insérer une seconde disquette : la disquette *root*. Tu remplaces la disquette, puis tu appuies sur entrée.

Remarque à propos du message « BIOS32 » que certains peuvent voir lors du boot : ce message signifie que ton bus est PCI. Si tu as le message suivant : « BIOS32 extended no supported » signifie donc que ta machine n'est pas PCI. Il ne faut pas s'inquieter et utiliser malgré cela la disquette de boot standard. Si la machine ne démarre pas grâce à la disquette de boot recréer une disquette de boot grâce à un autre support. Essayer aussi de débrayer le cache processeur.

Avant de te lancer dans l'installation de ton système, voici quelques trucs. **Linux** utilise un procédé qui installe ton système d'une manière quasi automatique. Dans le cas de la RedHat, l'installation est vraiment toute simple et progressive. Il suffit de suivre les indications. Attention toutefois : le clavier est en qwerty dès le départ. Le passage en clavier français se fera plus tard.

6.2 Partitionnement du disque

Il va falloir dans un premier temps que tu effectues le formatage de ton disque. Tu vas créer au moins une partition **Linux** avec le gestionnaire de fichier *ext2fs*, et au moins une partition de swap (le swap est un système de mémoire virtuelle, c'est-à-dire que si tu n'as pas assez de mémoire pour faire fonctionner tel ou tel programme, il va alors utiliser l'espace disque comme de la mémoire... Cette explication est un peu simplifiée mais en gros c'est ça.)

Il est beaucoup plus facile de supprimer une partition de swap que d'en créer une après coup donc prévoir large. Si tu veux faire tourner X Window, 8 Mo de mémoire vive sont absolument nécessaires.

Les disques sous **Linux** sont des *devices*. Par exemple, le premier disque dur est `/dev/hdax`, le second `/dev/hdbx` pour un contrôleur IDE. Le x correspond au numéro de la partition du disque. Si tu as déjà un système MS-DOS installé, il se trouve sûrement sur `/dev/hda1`.

Linux va alors te guider dans l'installation. Il te permet d'installer ton système d'une manière complète d'un seul coup. Le seul problème, c'est que si jamais tu rates quelque chose, tu dois tout recommencer.

Si ton disque n'est pas partitionné, c'est le programme `fdisk` qui va te permettre de réaliser l'opération. Sa syntaxe est : `fdisk /dev/disque`. Dans le cas de la RedHat, il le lance tout seul.

Par exemple, si tu veux installer **Linux** sur ton second disque IDE, tapes `fdisk /dev/hdb`.

Les commandes de fdisk (dans l'ordre d'utilité) sont :

- m : la liste de toutes les commandes ;
- p : affiche les partitions ;
- n : ajouter une nouvelle partition ;
- d : détruire une partition ;
- t : changer le type de la partition ;
- q : **quitter sans sauvegarder** les changements ;
- w : **sauvegarder et quitter**.

Pour ajouter une partition, tu tapes n puis tu spécifies le numéro de la partition (principale ou étendue), le numéro du bloc de début et sa taille (une taille paire est conseillée). Ensuite, tu lui donnes son type (avec t) :

- 83 : **Linux**
- 82 : Swap.

Bon, tu sauvegardes (w).

Ta partition est créée. Il est toutefois possible que tu ais besoin de rebooter Linux pour qu'elle soit prise en compte par le noyau.

La RedHat s'occupe du formatage automatique des partitions. Pour les formater à main, fait : `mke2fs -cv /dev/hdax` où x correspond au numéro de la partition.

6.3 Installation avec la Slackware

Maintenant, tu lances alors le programme *setup*. C'est un programme en couleurs (ça dépend de la disquette Root que tu as choisie), avec un menu. Le formatage des partitions peut y être effectué sans problème.

Configure ta partition de swap et installe la série A sur la partition.

La configuration swap est très simple à effectuer : choisis l'option swap, et suis les instructions.

Pour installer la série A, choisis l'option `source`, et choisis la bonne option : disquettes par exemple. Ensuite, spécifie le device à utiliser pour accéder au lecteur A. On va maintenant spécifier les séries à installer. Tu as alors la liste qui s'affiche. Choisis la série.

Le programme d'installation te demande alors si tu veux lancer l'installation. Réponds oui. Puis, il te demande un mode de sélection. Choisis *Normal*. A partir de là, c'est à toi de jouer. Il va te demander d'insérer chacune des disquettes, puis va te donner la liste des programmes à installer. Bonne partie de mange-disquettes !

Il est aussi possible d'installer grâce à un CD ROM contenant une distribution. Si le lecteur n'est pas reconnu il n'est pas nécessaire de recopier tous les fichiers sur disquettes avant de procéder à l'installation. Il faut

recopier tous les fichiers de la distribution sur une partition MS-DOS puis relancer le système grâce aux disquettes boot/root, invoquer "setup" puis lui indiquer la partition MS-DOS où se trouvent les fichiers nécessaires.

Utilise les menus pour effectuer toutes ces opérations. **Linux** va te demander de créer une disquette de boot. C'est cette disquette qui va désormais te permettre de lancer le système.

Ta partition **Linux** formatée, tes partitions de swap générées, la série installée et ta disquette magique prête, tu n'as plus qu'à rebooter.

Donc, tu mets ta disquette fraîchement configurée par **Linux** dans le lecteur et tu appuies sur les touches *Ctrl-Alt-Del*.

Ne jamais éteindre brutalement la machine sous Linux !. En effet, **Linux** comme tout système Unix, utilise des caches disques : tu perdrais des informations en éteignant ta machine. La seule manière propre d'éteindre **Linux** est soit de faire un *Ctrl-Alt-Del*, soit l'une de ces trois commandes à lancer en tant que *root* : `halt, reboot, shutdown` (Regarde le man pour les différences).

Une fois rebootée, loge toi en *root*. Tu dois faire très attention maintenant. L'utilisateur *root* a tous les droits, dont celui de tout détruire. Cela arrive plus facilement que tu ne penses !

6.4 Installation avec la RedHat

L'installation avec la RedHat est très intuitive. Une fois le partitionnement du disque réalisé, il suffit de choisir le type d'installation souhaitée, ainsi que les paquetages que tu souhaites utiliser. Une fois ce choix réalisé, cette distribution va réaliser l'installation complète et d'une manière automatique.

Tu peux aller boire un café pendant que l'installation de ta machine se fait automatiquement.

7 Administration sous Linux

7.1 Clavier Français

Bien, commençons par la chose utile : le clavier français. Normalement, les distributions configurent ça tout seul... mais on ne sait jamais !

Une fois logé en *root*. Tape la commande suivante : `/usr/bin/loadkeys /usr/lib/kbd/keytables/fr.map`.

Maintenant, ton clavier est français ! Attention, cela reste temporaire. La solution la plus simple est de le rajouter dans le fichier `/etc/rc.local` avec la Slackware. Pour la RedHat, réaliser l'opération avec le panneau de configuration graphique.

7.2 LILO

LILO (**LI**nux **LO**ader) permet d'avoir un système multiple d'amorçage : Windows, Dos, Linux, etc. Ce programme va se loger dans le secteur de boot de ton disque pour te permettre de sélectionner la partition sur laquelle booter.

Si tu as installé *LILO* sur ton disque dur et que cela plante, boote avec une disquette MS-DOS et tape `FDISK /MBR`.

Le fichier de configuration LILO se trouve généralement dans `/etc/lilo.conf`. Les distributions permettent la génération d'un fichier d'une manière automatique.

Voici un exemple de fichier (le mien) :

```
boot = /dev/hda    # disque sur lequel on met le fichier lilo :

delay = 300         # temps d'attente avant le lancement du boot.
                    # permet d'avoir le temps de sélectionner la partition
                    # à amorcer.

vga = normal        # mode de l'écran à choisir
ramdisk = 0         # paranoia setting

# Linux : dernière version : le Pc boote ici par défaut
image = /vmlinuz    # Noyau Linux 1 (Le path pour y accéder)
  root = /dev/hda2
  append="no-hlt, aha1542=0x230"
  label = linux
  read-only

other = /dev/hda1   # Partition 1 :
  label = dos       # texte à taper pour booter dessus : dos
  table = /dev/hda  # disque : /dev/hda

image = /zimage    # Noyau Linux 2
  root = /dev/hda2  # Racine (Partition 2 de mon disque)
  append="aha1542=0x230" # append... voir explication plus bas
  label = old       # Texte a taper : old
  read-only         #
```

Plusieurs questions te viennent à l'esprit :

1. read-only : lorsque tu boot **Linux**, il va faire une vérification de la cohérence des informations qui se trouvent dessus. C'est une vérification des systèmes de fichiers. Pour réaliser cette opération, il est indispensable que la partition soit en lecture seule. En effet, vérifier un disque en cours d'utilisation est le meilleur moyen pour tout détruire. Cette vérification n'a pas lieu à tous les coups si tu éteins bien ta machine (appui sur les touches *Ctrl-Alt-Del* ou utilisation des commandes `halt`, `reboot` et `shudown`). Cette vérification n'est de toutes manière lancée que sur les partitions **Linux**. C'est le programme `fsck` qui effectue cette opération. Je te déconseille de le lancer « pour voir » : une utilisation erronée peut provoquer des pertes de données sur le disque !

2. pourquoi deux noyaux ? C'est assez simple : **Linux** change, bouge. Les noyaux sont modifiés assez

souvent. Par exemple, je peux amorcer actuellement ma machine soit avec un noyau stable (2.0.26), soit avec un noyau de développement (2.1.8).

3. *append* : Cette option est très importante : elle permet de donner un ou plusieurs paramètres au noyau Linux. Par exemple, ici je lui indique que ma carte SCSI se trouve à l'adresse 230. Dans le cas contraire, elle ne serait pas reconnue. Cette option est utilisée également si tu as plusieurs interfaces réseau, SCSI, etc.

Pour installer le fichier, tu exécutes `/sbin/lilo` et c'est tout !

En cas de problème avec LILO il faut d'abord essayer de supprimer l'option "compact" qui peut se trouver dans son fichier de configuration.

Ne pas oublier de relancer LILO après chaque modification de son fichier de configuration.

7.3 Montage automatique de partitions : /etc/fstab

Le fichier /etc/fstab contient toutes les informations concernant le montage de tes partitions.

Voici un exemple :

```
#/etc/fstab
# Device      Repertoire     type        options      frequence  passe
# Linux
/dev/hda2         /           ext2        defaults        5        1
/dev/hdb2         /usr2       ext2        defaults        5        2
/dev/sda2         /usr3       ext2        defaults        10       2
# MS-DOS
/dev/hda1         /dos        msdos       defaults        0        0
/dev/hdb1         /dos2       msdos       defaults        0        0

# Processus
none              /proc       proc        defaults        0        0
# Partition de swap
/dev/hda3         none        swap        defaults        0        0
# Fichier de swap
/usr2/swap_file   /usr2       swap        defaults        0        0
```

Voici une descriptions des six colonnes :

1. device (périphérique) de la partition. Dans le cas d'un fichier de swap, c'est le nom du fichier.
2. point de montage de la partition ;
3. type de la partition ;
4. options (tu peux spécifier une partition en lecture seule, etc).

5. fréquence correspond au nombre de jours entre deux traitements du fichier par la commande `dump`. Cette commande n'existe que pour ext2fs (c'est un portage de la version 4.4BSD) mais n'est pas encore incluse dans toutes les distributions.

6. Ordre de tests des partitions (fsck). Si tu mets 0, aucune vérification automatique n'est effectuée lors du démarrage. Les partitions situées sur un même disque seront testées d'une manière séquentielle mais si elles sont situées sur deux disques différents, c'est fait en parallèle.

Tu rajoutes ici les partitions dont tu as besoin.

Les volumes NFS profiteront des paramètres suivants : `rsize=8192,wsize=8192`, surtout si tu possèdes une carte rapide.

Pour autoriser un utilisateur à monter un volume il faut créer une ligne contenant l'option "user" Exemple (cas d'un CD-ROM SCSI) :

```
/dev/scd0       /mnt/cd iso9660 user,exec,dev,nosuid,ro,noauto
```

N'importe quel utilisateur pourra dès lors monter et démonter un CD (utiliser `mount /mnt/cd`, `umount /mnt/cd`) La page de man de mount, section 8 (`man 8 mount`) explique la signification des options possibles.

7.4 Montage manuel de partitions

Comme nous l'avons vu dans le paragraphe précédent, les partitions sont montées lors de l'amorçage grâce au fichier `/etc/fstab`. Tu peux toutefois les monter à la main. Il faut alors utiliser la commande `mount`. Par exemple, pour monter la disquette A:, tu fais : `mount -t msdos /dev/fd0 /mnt`.

Le `-t` indique le type du support. Parmis les plus courants, tu trouves

- ext2 : filesystem **Linux** ;
- msdos : disque(tte)s MS-DOS ;
- vfat : disque(tte)s Windows 95 ;
- iso9660 : Cd-ROM ;
- nfs : montage de partitions à travers réseau.

Bien évidement, il faut que ces types de filesystems aient été intégrés au noyau...

Le `/dev/fd0` est le nom du fichier spécial correspondant au périphérique. Le répertoire est le point de montage.

Pour démonter, tu tapes `umount /mnt`.

7.5 Mise en place des quota

7.5.1 Logiciels nécessaires

Le support des quota disque a été intégré dans le noyau Linux depuis la version 1.3.46. Il te faut donc utiliser un noyau postérieur pour pouvoir bénéficier des quota.

Tu as également besoin des outils de gestion des quota. On peut les trouver sur :

`ftp://ftp.funet.fi/pub/Linux/PEOPLE/Linus/subsystems/quota`

ou, plus près, sur :

`ftp://ftp.ibp.fr/pub/linux/kernel/sources/subsystems/quota`

L'archive à récupérer s'appelle `quota-1.51.tar.gz` et se trouve dans le sous-répertoire `utils`. Il y a également un sous-répertoire `doc` contenant des documentations (principalement les pages de manuel des outils) au format HTML.

7.5.2 Compilation des utilitaires

Après avoir extrait le contenu de l'archive `quota-1.51.tar.gz`, il te faut compiler ces utilitaires. Cela se fait en tapant tout simplement `make`. Dans certains cas (selon la version de bibliothèque C que tu utilises), il peut être nécessaire d'ajouter `-I.` à la variable `CFLAGS` définie dans le `Makefile`.

Si tu utilises le système de fichiers ext2 et que tu as installé les bibliothèques contenues dans la distribution e2fsprogs (utilitaires de gestion du système de fichiers ext2), tu peux inclure le support ext2fs dans les utilitaires des quota, afin d'accélérer leur exécution. Pour cela, édite le fichier `Makefile` afin que la variable `CFLAGS` contienne `-DEXT2_DIRECT` et que la variable `EXT2LIBS` contienne `-lext2fs -lcom_err` avant de compiler les programmes.

Après la compilation, installe les utilitaires par la commande `make install`. Attention, la commande `quota` est installée dans le répertoire `/usr/ucb` qui n'existe pas sur la plupart des installations Linux. Il te faut donc modifier le `Makefile` pour installer cette commande dans un autre répertoire (`/usr/bin` est un bon choix). Il peut également être nécessaire de définir la variable `LN` à la valeur `ln -sf`.

7.5.3 Activation du support des quota dans le noyau

Pour utiliser les quota, le noyau doit avoir été compilé avec le support des quota. pour cela, tu dois avoir un noyau dans lequel tu as répondu « Y » à la question « Quota support ». Attention : le support des quota n'est pour l'instant effectif que sur les systèmes de fichiers ext2.

7.5.4 Activation des quota sur les systèmes de fichiers

Il existe deux types de quota : les quota liés aux utilisateurs et les quota liés aux groupes d'utilisateur. Les premiers définissent le nombre maximal de fichiers et de blocs disque associés à chaque utilisateur, les seconds définissent le nombre maximal de fichiers et de blocs disque associés à chaque groupe d'utilisateurs. Les deux types de quota peuvent être activés séparément.

Pour activer les quota pour les utilisateurs, il faut indiquer l'option `usrquota` pour les systèmes de fichiers concernés dans `/etc/fstab`. Les quota concernant les groupes sont régis par l'option `grpquota`. Les fichiers de définition des quota s'appellent respectivement `quota.user` et `quota.group` et sont situés dans la racine de chaque système de fichiers concerné.

Il est possible de modifier les noms de fichiers de gestion des quota en utilisant la syntaxe `usrquota=nom_de_fichier` et `grpquota=nom_de_fichier`.

Voici un exemple de fichier `/etc/fstab`:

```
/dev/hda2    /         ext2   defaults,rw 0 1
/dev/hdb2    /home     ext2   defaults,rw,usrquota,grpquota 0 1
/dev/sda1    /usr/src  ext2   dzfaults,rw,usrquota 0 1
```

L'activation des quota est ensuite lancée par la commande `quotaon`. Pour les activer automatiquement lors du démarrage du système, il faut ajouter dans un fichier d'initialisation (situé généralement dans `/etc/rc.d`) les lignes:

```
# Enable quota
/usr/sbin/quotaon -avug
```

Il peut également être nécessaire de vérifier la cohérence des informations de gestion des quota après des arrêts intempestifs. Pour cela, il faut utiliser la commande `quotacheck`:

```
/usr/sbin/quotacheck -avug
```

Si les utilitaires ont été compilés avec `-DEXT2_DIRECT`, la commande `quotacheck` doit être relativement rapide. Dans le cas contraire, elle peut être très lente car elle doit explorer tous les répertoires du système de fichiers de manière hiérarchique. Il est, de toutes façons, conseillé de la lancer automatiquement lors du démarrage avant l'activation des quota par `quotaon`:

```
# Check quota
/usr/sbin/quotacheck -avug
```

7.5.5 Attribution d'un quota à un utilisateur

La commande `edquota` est utilisée pour affecter un quota à un utilisateur ou à un groupe d'utilisateurs. Sa syntaxe est `edquota -u utilisateur` ou `edquota -g groupe`. Cette commande lance un éditeur de texte contenant la définition des quota attribués à l'utilisateur ou au groupe et prend en compte leur nouvelle valeur lorsque le fichier est ré-écrit.

Pour chaque utilisateur ou groupe, il existe deux limitations: le nombre de fichiers et le nombre de blocs disque (exprimés en blocs de 1024 octets). Pour chacune, il existe deux limites:

1. la limite « douce »: lorsque cette limite est atteinte ou dépassée, un message d'avertissement est affiché lors de chaque nouvelle allocation de fichier ou de bloc;

2. la limite « dure » : lorsque cette limite est atteinte, il est impossible à l'utilisateur de créer de nouveau fichier ou d'allouer de nouveau bloc.

La limite « douce » se transforme en limite « dure » quand elle a été atteinte ou dépassée depuis un certain temps (sept jours par défaut).

7.5.6 Statistiques sur les quota

Tout utilisateur peut obtenir l'état des quota qui lui sont attribués (limites ainsi que le nombre de fichiers et de blocs qui lui sont alloués) grâce à la commande `quota`.

Le super-utilisateur peut obtenir les mêmes informations sur tout utilisateur ou groupe avec la même commande : `quota -u utilisateur` ou `quota -g groupe`. Il peut de plus utiliser la commande `repquota` pour obtenir une liste des quota associés à un ou plusieurs systèmes de fichiers.

7.5.7 Autres documentations sur les quota

L'implémentation des quota dans Linux est compatible avec l'implémentation originale dans 4.2BSD Unix. Toutes les documentations concernant les quota dans BSD, que ce soit au niveau des commandes ou au niveau de l'interface de programmation, sont donc utilisables.

Les pages de manuel des utilitaires sont assez claires et très instructives. Enfin, les documentations diffusées en format HTML forment une bonne introduction à la gestion des quota.

7.6 mtools

Plutôt que de monter tes partitions MS-DOS, je te conseille d'utiliser les mtools. Il s'agit de l'équivalent des commandes MS-DOS, mais avec un `m` avant !

Exemple : `mcopy toto.c a:` . Si tu veux copier un fichier texte pour qu'il soit exploitable sous MS-DOS, ou l'inverse, n'oublie pas de rajouter l'option `-t` : cela convertit le caractère de fin de ligne qui est différent entre les deux systèmes.

En principe, ils sont déjà à peu près configurés. Au cas où, tu dois modifier le fichier `/etc/mtools`.

```
drive a: file="/dev/fd0" exclusive
drive b: file="/dev/fd1" exclusive

# 1er disque Dur
drive c: file="/dev/hda1"

# 2nd disque Dur
drive d: file="/dev/sda1"

mtools_lower_case=1
```

Attention : le fichier a été modifié depuis la dernière version. C'est ici le dernier format qui est présenté car beaucoup plus clair et simple à utiliser. De plus, cette nouvelle version permet d'utiliser les noms longs (vfat de Windows 95).

7.7 Swap : création de la partition

Pour rajouter une partition de swap de 8 Mo, il faut d'abord créer la partition à l'aide du programme `fdisk`, puis soit tu passes par le programme *setup*, soit tu le fais à la main : `mkswap /dev/hda3 8192`

Tu rajoutes la ligne `/dev/hda3 swap swap defaults 0 0` dans `/etc/fstab`.

Enfin, pour que ça fonctionne, soit tu rebootes, soit tu fais un `swapon -a`.

7.8 Swap : fichier

Pour le fichier, c'est plus compliqué. Tu fais un

```
dd if=/dev/zero of=/usr2/swap_file bs=1024 count=8192
mkswap /usr2/swap_file 8192
```

Tu rajoutes dans /etc/fstab la ligne `/usr2/swap_file /usr2 swap defaults 0 0`.

Fais bien attention à mettre cette ligne après le montage de la partition /usr2. Sinon, ce n'est pas près de fonctionner.

Désormais, tu es obligé pour quitter **Linux** de faire un

```
swapoff -a
reboot           #Ou autre
```

En effet, si tu ne désactives pas le fichier d'échange, **Linux** ne va pas pouvoir démonter la partition, et donc, il fera un fsck dessus à chaque fois que tu relanceras ta machine. Il est aussi possible, avec certains systèmes, de placer la commande swapoff dans "/etc/rc.d/rc.K" afin d'automatiser cela.

7.9 Ajouter un utilisateur, un groupe

Si tu possèdes la RedHat, utilises le panneau de configuration : c'est vraiment très bien fait.

Rajouter un utilisateur se fait en deux temps et trois mouvements. Deux fichiers sont importants : `/etc/passwd` et `/etc/group`.

Une ligne du fichier passwd est composée de cette manière :

```
gandalf::501:100:Eric Dumas:/home/gandalf:/bin/bash
sgandalf::0:0:Super Eric Dumas:/:/bin/sh
```

Une ligne est composée de 7 colonnes :

1. Nom de l'utilisateur : 8 caractères max ;
2. Mot de passe chiffré ;
3. Numéro d'identificateur unique de l'utilisateur (uid). Rq : 0 = root ;
4. Numéro du groupe (gid) ;
5. Nom « officiel » de l'utilisateur ;
6. Home directory ;
7. Shell à utiliser.

Le fichier `/etc/group` contient la liste des groupes de la machine. Par exemple, le groupe 100 est :

```
users::100:games,gandalf
```

Pour rajouter un utilisateur *toto*, tu fais

1. tu le rajoutes dans `/etc/passwd` ;
2. si besoin est, tu crées le groupe dans lequel il va se trouver, en modifiant `/etc/group` ;
3. tu crées son home directory :

```
mkdir /home/toto
cp /etc/skel/* /home/toto
chown toto /home/toto
chgrp le_groupe_de_toto /home/toto
```

4. Enfin, tu lui donnes un mot de passe en tapant `passwd toto` .

Tu n'as plus qu'à essayer.

Je te conseille très fortement de travailler sous un utilisateur commun. En effet, il est très dangereux de travailler en root. Une fausse manipulation et tu dois tout réinstaller, voire perdre tes précieux fichiers personnels.

7.10 Lancements de programmes personnels lors du boot : /etc/rc.d/rc.local

C'est dans ce fichier que tu vas mettre tous les petits trucs personnels que tu veux lancer au boot. En voici une version :

```
#! /bin/sh
```

```
# du menage en v'la,...
echo "Cleaning /tmp..."
/bin/rm -fr /tmp
mkdir /tmp
chmod 1777 /tmp
# NOTE : ceci peut s'avérer néfaste car certains programmes lancés par
# les fichiers rc exécutés avant rc.local peuvent employer des fichiers de
# /tmp ...

# Vive l'azertyuiop :-)

echo "Loading french keyboard..."
/usr/bin/loadkeys /usr/lib/kbd/keytables/fr-latin1.map
```

Ce qui est important, c'est le clavier français. Si tu rajoutes cette ligne, ton clavier sera français dès le boot.

En cas de problème il faut modifier ainsi fr-latin1.map :

```
keycode   3 = eacute          two             asciitilde
keycode   8 = egrave          seven           grave
keycode  10 = ccedilla        nine            asciicircum
keycode  14 = Delete          BackSpace
```

En plus du clavier, je fais un petit nettoyage de `/tmp`. Ca peut toujours être utile.

7.11 Ordonnancement de travaux : la crontab

La crontab est une table qui contient des commandes qui doivent être lancées à intervalle régulier. Tu peux y mettre la sauvegarde journalière, etc.

Le format est assez simple :

```
# Faire une sauvegarde tous les  Lundi a 2 heures du matin
0  2  *    * 1 /home/gandalf/backup
# Rebooter la machine tous 1ers et 15 du mois a 4h15.
15 4  1,15 * * /sbin/shutdown -r +3
```

Les 5 premières colonnes sont :

1. Minute (0 à 59) ;
2. Heure (0 à 23) ;
3. Jour du mois (1 à 31) ;
4. Mois (1 à 12) ;
5. Jour de la semaine (0 à 6 : 0 = Dimanche, 1 = Lundi).

Ensuite vient la commande. Cette crontab est très utile si tu utilises UUCP (pour envoyer le courrier, récupérer les news, etc).

Tout utilisateur peut se créer une crontab grâce à la commande `crontab`.

7.12 Imprimer !

Maintenant que ta machine fonctionne, voyons comment imprimer. Les imprimantes sont gérées par le programme *lpc* et par le démon *lpd*.

Le démon doit être lancé au boot. Il est lancé soit dans `rc.M` soit dans `rc.inetd2`. Normalement, lpd est lancé au boot. (Tu dois le voir lorsqu'il lance tous les démons de ta machine. Si ce n'est pas le cas, il faut que tu décommentes quelques lignes dans l'un des fichiers cités).

Supposons que ton imprimante s'appelle *betty*. Dans un premier temps, dans le fichier `/etc/rc.d/rc.local`, rajoute

```
echo "Setting up printer betty"
/usr/sbin/lpc up betty
```

Passons maintenant aux choses sérieuses. Le fichier dans lequel va être décrit ton imprimante est le fichier `/etc/printcap`. Exemple :

```
betty|lp:lp=/dev/lp1:sd=/var/spool/lp1:sh:lf=/var/adm/lpd-errs:of=/etc/start-dj500:
```

La première colonne représente le nom de l'imprimante. J'ai mis betty et lp car de cette manière, c'est l'imprimante par défaut. Ensuite, la configuration se fait via des mots-clefs. Pour plus d'information, fais `man printcap`. Voici quand même une description de ceux utilisés.

- lp : device sur lequel est branché l'imprimante (LPT1 ici) ;
- sd : répertoire du spool d'impression ;
- sh : supprime la page d'entête ;
- lf : fichier d'erreurs ;
- of : programme de filtrage.

Comment ça marche? Bon, supposons que j'ai un fichier toto.ps. Je fais donc `lpr toto.ps` (ou `lpr -Pbetty toto.ps`). Le fichier va être déposé dans le répertoire `/var/spool/lp1`.

Le démon va ensuite envoyer le fichier sur le programme de filtrage. Ce système permet d'imprimer des fichiers ayant des formats assez variés.

Voici le fichier de filtrage (Pour HP-500,510,520,etc).

```
#!/bin/sh
# Script d'impression pour imprimante HP-500 Noir et Blanc
```

```
#
# Eric.Dumas@freenix.fr
#
# Version 2.0
#
# 5/01/95   (ED) : Ajout du format dvi ;
# 27/10/95  (ED) : Conversion des fichiers textes en fichier PostScript ;
# 01/11/95  (ED) : Un peu de ménage ;
# 12/08/96  (ED) : modification pour gs
# 04/12/96  (ED) : quelques corrections et rajouts.

TmpDir=/tmp
TmpFile=$TmpDir/deskjet.$$

# Utilisateur à prevenir en cas d'erreur
NOTIFY=lp-owner

# Programmes
CAT=/bin/cat
DVIPS=/usr/TeX/bin/dvips
PGS=/usr/bin/gs
AIIPS=/usr/local/bin/a2ps

# Chemins d'accès pour GS
GS_LIB=/usr/lib/ghostscript:/usr/lib/ghostscript/psfonts:/usr/lib/ghostscript/Type1:/usr/lib/ghostsc

# C'est partit
$CAT - > $TmpFile

echo -ne '\033E'

set -- `file $TmpFile`

shift

FileType=$*

# Transformation du PostScript en format Deskjet500
# résolution à 300x300 - format a4
GS="$PGS -I$GS_LIB -q -sDEVICE=djet500 -r300x300 -sPAPERSIZE=a4 -dNOPAUSE -sOutputFile=- - /usr/lib/

case $FileType in

       *DVI*) # Fichier DVI
```

```
          $DVIPS -t a4 $TmpFile \
            -f | $GS
          ;;

        *PostScript*)    # Impression de fichiers Postscript
          $CAT $TmpFile | $GS
          ;;

          *text*|*script*)
          $CAT $TmpFile | $AIIPS -nP -r -8 | $GS
          ;;

          *data*)
            echo -ne '\033&;k0G'  # C'est un et commercial sans ; !
            /bin/cat $TmpFile
           ;;

          *)
            echo "Deskjet: Unknow filetype $FileType" >> /dev/console
            echo "Deskjet: $TmpFile Unknow filetype $FileType" | mail $NOTIFY
          ;;
esac

/bin/rm -f $TmpFile
echo -ne '\033E'
```

Ce script permet d'imprimer aussi bien des fichiers texte que des fichiers Postscript, dvi. Si tu veux imprimer d'autres types de fichiers, il suffit de les rajouter ici. Merci de m'envoyer les modifications également : cela peut être bien que tout le monde en profite.

7.13 Timezone

Timezone est un système fort utile qui permet la gestion de changement d'horaires d'été et d'hiver. La France se trouve dans une zone horaire : MET (Medium European Time, soit GMT + 1).

La zone « MET DST » correspond à l'heure d'été activé (GMT + 2). Mieux vaut ne pas laisser tout cela reposer uniquement sur la variable TZ.

Tu te places dans le répertoire `/usr/lib/zoneinfo` et je te conseille de lire le fichier time.doc.

Les fichiers binaires placés dans `/usr/lib/zoneinfo` indiquent les règles de calcul de l'heure dans différentes zones du globe.

Tu copies le fichier « MET » sous le nom de `/usr/lib/zoneinfo/localtime`, puis tu fais un lien symbolique de ce fichier sur `/usr/lib/zoneinfo/posixrules`.

Commandes :

```
# cd /usr/lib/zoneinfo
# cp MET localtime
# ln -sf localtime posixrules
```

Utilise ensuite la commande `clock` pour mettre le système à l'heure. Deux possibilités :

- La machine est à l'heure GMT.

- La machine est à l'heure locale.

La première solution est préférable, mais MS-DOS ne gère pas correctement cette approche (:-)). L'horloge sera donc fausse sur ce système.

Par contre tous les changements d'heure, deux fois par an, seront pris en charge par **Linux**. Dans ce cas, tu dois rajouter dans `/etc/rc.d/rc.local`, la commande suivante : `clock -u -s`. **Linux** s'ajustera sur l'horloge sauvegardée.

Si la CMOS est à l'heure locale, la commande dans `/etc/rc.d/rc.local` devient `clock -s` et les changements d'heure ne seront automatiques que si la machine est allumée au moment des changements d'heure. Il faudra manuellement réécrire la nouvelle heure dans la CMOS par `clock -w`, ou mettre l'horloge à l'heure avec le setup.

La commande `date` permet de vérifier la validité de l'heure : elle renvoie heure et timezone `date -u` donne toujours l'heure en GMT :

```
plux:/home/rene $ date
Tue Sep 28 20:16:46 MET 1993
plux:/home/rene $ date -u
Tue Sep 28 19:16:49 GMT 1993
```

(En été, « MET DST » (Daylight Savings Time) serait indiqué).

7.14 Nouveaux changement d'heure

Il s'avère que cette année, la date de changement d'heure a été décalée. Pour que **Linux** ne se mélange pas les pinceaux, il est nécessaire de faire certaines modifications. La manipulation se base sur l'utilisation du programme `zic`, le *time zone compiler*.

Dans un fichier appellé `europe`, tu met :

```
Zone    MET     1:00      M-Eur     MET%s
Link    localtime MET
Rule    M-Eur   1986    max     -       Mar     lastSun 2:00s   1:00    " DST"
Rule    M-Eur   1986    1995    -       Sep     lastSun 2:00s   0       -
Rule    M-Eur   1996    max     -       Oct     lastSun 2:00s   0
```

Pour mettre tout en place, lancer la commande :

```
gandalf# zic europe
```

Pour vérifier que tout s'est bien passé, tu dois avoir le résultat suivant :

```
gandalf# zdump -v MET | grep 1996
MET  Mon Nov 11 12:29:02 1996 MET
MET  Sun Mar 31 00:59:59 1996 GMT = Sun Mar 31 01:59:59 1996 MET isdst=0
MET  Sun Mar 31 01:00:00 1996 GMT = Sun Mar 31 03:00:00 1996 MET DST isdst=1
MET  Sun Oct 27 00:59:59 1996 GMT = Sun Oct 27 02:59:59 1996 MET DST isdst=1
MET  Sun Oct 27 01:00:00 1996 GMT = Sun Oct 27 02:00:00 1996 MET isdst=0
```

7.15 locate

Lorsque tu installes le package GNU « find », en plus du programme de recherche se trouve un programme `locate`. Ce programme permet de retrouver le nom complet d'un fichier. Ce programme se trouve dans certaines distributions dans `/usr/lib/locate`.

Lancer, en tant que root, `/usr/lib/locate/updatedb`. Cela invoque un `find /` sur les disques montés et place les noms de tous les fichiers (sous forme hash) dans `/usr/lib/locate/find.codes`. `locate` permet ensuite de localiser l'un d'eux.

Remarque : lorsque tu lances ce programme, pense à démonter les partitions MS-DOS et ton CD-ROM !

Une autre solution assez agréable consiste à insérer dans la *crontab* la ligne `updatedb --prunepaths='/tmp /usr/tmp /var/tmp /proc /users /root /dos /mnt /var/adm /var/spool /var/catman /home'`. Cette commande réalise la mise à jour de l'arborescence sans les fichiers temporaires, l'arborescence utilisateur, une partition MS-DOS, etc.

Il est souhaitable d'exclure les répertoires "HOMES" des utilisateurs. En effet, ces partitions sont contiennent normalement de nombreux fichiers. Mais cela pose un grave problème de sécurité, car tout utilisateur peut sinon connaître le nom des fichiers du voisin.

7.16 Un second contrôleur IDE

Dans ce cas, il faut que tu crées les périphériques dans `/dev` . Lance *MAKEDEV.ide1* dans le répertoire `/usr/src/linux/drivers/block`. Ensuite, il te suffit de faire un `fdisk /dev/hdc` par exemple.

7.17 CD-ROM ATAPI

Bon, les CD-ROM en général ne posent pas trop de problèmes, sauf les ATAPI. En effet, le CD est géré sur un deuxième contrôleur disque. Procédons dans l'ordre :

1. tu recompiles le noyau avec l'option CD ATAPI IDE et le support du système de fichiers iso9660 ;

2. tu vas dans le répertoire `/usr/src/linux/drivers/block`. Il y a un script qui s'appelle *MAKEDEV.ide1*. Tu le lances. Ce script va créer les devices nécessaires pour ton second contrôleur disque.
3. Tu rebootes. Au niveau de LILO, tu tapes : **Linux** `hdd=cdrom`. Attention : tu es en qwerty ici.
4. tu crées un lien symbolique entre /dev/hdd (ou hdc) et /dev/cdrom. `ln -s /dev/hdd /dev/cdrom`.
5. tu crées le répertoire `/cdrom` et tu tapes : `mount -t iso9660 -o ro /dev/cdrom /cdrom`. Normalement, ton CD est monté !

En cas de problème il faut tenter de :

1. configurer le lecteur en mode EIDE "master" ;
2. laisser un CD chargé durant le boot ;
3. booter, attendre un moment puis rebooter à chaud (sans éteindre ni actionner reset : donc par control-alt-del) afin de laisser au lecteur le temps de s'ébrouer.

7.18 Mettre une console externe comme console

Certains utilisateurs peuvent voir mettre comme console un minitel ou un vieux vt100 qui ne passe pas sur la carte vidéo mais sur le port série. Il suffit de rajouter dans le fichier `lilo.conf` la ligne suivante :

```
append = "scon=0x03f8  serial=0,9600n8"
```

7.19 Deux cartes réseaux

Certaines machines possèdent deux (ou plus) cartes Ethernet. Pour qu'elles soient toutes les deux reconnues, il est nécessaire dans la configuration de Lilo :

```
append="ether=0,0,eth1"
```

7.20 Les fichiers d'initialisation des interpréteurs de commandes

Voici la description des fichiers qui sont chargés lors de la connexion d'un utilisateur. Le nom du ou des fichiers est spécifique à l'interpréteur de commandes utilisé.

Dans le cas de `bash`, c'est un peu compliqué. D'après la page de manuel de `bash`, lors d'une connexion interactive, il exécute :

- `/etc/profile` s'il existe ;
- `~/.bash_profile` s'il existe, sinon `~/.bash_login` s'il existe, sinon `~/.profile` s'il existe ;
- `~/.bashrc` s'il existe.

Dans le cas où `bash` est invoqué en tant que `sh`, il n'exécute que les fichiers `/etc/profile` et `~/.profile` s'ils existent.

Dans le cas de ksh, la page de manuel est moins précise mais une étude rapide du code source montre que :

- `/etc/profile` est exécuté s'il existe ;
- `~/.profile` est exécuté s'il existe, si le numéro d'utilisateur effectif est identique au numéro d'utilisateur réel et si le numéro de groupe effectif est égal au numéro de groupe réel ;
- `/etc/suid_profile` est exécuté s'il existe, si les numéros d'utilisateur effectif et réel diffèrent ou si les numéros de groupe effectif et réel diffèrent ;
- enfin, si la variable d'environnement `ENV` est positionnée, le fichier dont le nom est contenu dans `"$ENV` est exécuté s'il existe.

Enfin, pour être complet sur les interpréteurs de commandes compatibles avec le Bourne Shell, citons le cas de `zsh` qui exécute :

- `/etc/zshenv` s'il existe ;
- `~/.zshenv` (ou `"$ZDOTDIR/.zshenv` si ZDOTDIR est positionné) s'il existe ;
- `/etc/zprofile` s'il existe ;
- `~/.zprofile` (ou `"$ZDOTDIR/.zprofile`) s'il existe ;
- `/etc/zshrc` s'il existe ;
- `~/.zshrc` (ou `"$ZDOTDIR/.zshrc`) s'il existe.

Dans le cas de `csh` (enfin plutôt tcsh) :

- `/etc/csh.cshrc` est exécuté s'il existe ;
- `/etc/csh.login` est exécuté s'il existe ;
- `~/.tcshrc` est exécuté s'il existe, sinon `~/.cshrc` est exécuté s'il existe ;
- `~/.history` (ou le contenu de la variable histfile) est chargé en mémoire s'il existe ;
- `~/.login` est exécuté s'il existe ;
- `~/.cshdirs` (ou le contenu de la variable dirsfile) est chargé en mémoire s'il existe.

7.21 Les sauvegardes sous Linux

Comme tout système Unix, **Linux** fournit des outils standards pour réaliser les sauvegardes des disques. Principalement deux types d'outils sont actuellement utilisés.

7.21.1 dump et restore

Les outils `dump` et `restore` ont été adaptés pour **linux** par Rémy Card. Ils sont livrés dans les distributions.

Une fois les sources compilés et installés, l'utilisation de dump et restore est relativement simple. Pour effectuer la sauvegarde d'une partition `/dev/sda1` sur `/dev/rmt0`, il suffit de faire par exemple :

```
gandalf# dump 0sfu 3600 /dev/rmt0 /dev/sda1
gandalf# dump 0sfu ondee:/dev/rmt0 /dev/sda1
```

La deuxième commande permet de sauvegarder un disque sur un périphérique distant (par exemple situé ici sur la machine ondee). Les options de dump peuvent sembler complexes. En voici une courte description :

- 0 à 9 : niveau de sauvegarde. 0 correspond à une sauvegarde complète, alors que les autres niveaux n correspondent à la sauvegarde des fichiers qui ont été modifiés depuis la nième sauvegarde ;
- s : taille de la bande en pieds ;
- f : fichier - peut-être composé de `machine:fichier` ;
- u : écriture de la date et du niveau de sauvegarde dans le fichier `/etc/dumpdates`.

Il existe d'autres options. Pour plus renseignements, consulte les pages de manuel.

Il existe deux modes pour effectuer une restauration : en ligne de commande ou en mode dit « interactif ». Le deuxième mode est plus simple pour des restaurations partielles. Le premier est surtout utilisé pour des restaurations complètes.

Pour restaurer la bande en mode interactif il suffit de faire

```
gandalf# restore -if /dev/rmt0
gandalf# restore -if ondee:/dev/rmt0
```

Dans ce cas, un mini-interpréteur de commandes est lancé. Utilise la commande *help* pour plus de détails.

Pour restaurer une bande complètement, lancer :

```
gandalf# restore rf /dev/rmt0
```

Note importante : pour l'utilisation de dump et restore à travers un réseau (sauvegarde sur des périphériques distants), tu dois utiliser des fichiers `.rhosts`. Dans l'exemple de sauvegarde ci-dessus, la machine ondre doit avoir :

```
#cat ~root/.rhosts
gandalf
#
```

L'utilisation de périphériques distants nécessite également la présence du programme `rmt` sur la machine gérant les périphériques de sauvegarde. Ce programme est inclus dans la distribution source de dump pour Linux.

7.21.2 tar

A la différence de dump ou restore, tar permet de sauvegarder les fichiers désirés, d'exclure certains répertoires, etc. Il est à noter que le tar utilisé sous **Linux** est le tar GNU. Il possède certaines options particulières.

Pour connaître toutes les options possibles, je te conseille de faire `tar --help` . Une utilisation simple de tar peut être illustrée avec la sauvegarde d'une partition d'utilisateurs:

```
# tar cvf /dev/rmt0 /users | mail backup-user
```

La liste des fichiers sera ainsi envoyée à l'utilisateur *backup-user*.

Certains sites utilisent exclusivement `tar` pour effectuer leurs sauvegardes, c'est un choix.

7.22 Num Lock au démarrage

Pour configurer la touche Num Lock au démarrage (par exemple pour qu'elle soit positionnée), il suffit de mettre dans `/etc/rc.d/rc.local`

```
for tty in /dev/tty[1-9]*;
do
     setleds -D +num < $tty > /dev/null
done
```

7.23 Les fichiers core

Certains développeurs aiment exploiter le contenu des fichiers `core`. Or, sous **Linux**, ils sont désactivés pas défaut. Pour pouvoir les générer, il faut tapper `ulimit -c unlimited`.

7.24 Accounting - lastcomm

Le système d'accounting est implémenté dans **Linux** (mais bon, j'ai jamais testé car ça fait un peu Big Brother is Watching You). Il faut normalement compiler le paquetage acct-1.3.73.tar.gz et suivre les instructions qui sont livrées avec. Ca se trouve sur `ftp.ibp.fr:/pub/linux/sunsite/system/Admin/accounts`, et sur `ftp://iguana.hut.fi/pub/linux/Kernel/process_accounting/`. Attention: il est nécessaire d'avoir un noyau 1.3.* ou un 2.0.*.

Voici le résultat:

```
# lastcomm | more
Command          Flags User     Tty    PagFlt Time        Endtime
clear            -     merlin   ttyp2      85   0.00 secs Tue Aug  6 13:26:07
in.identd        -     root     __        100   0.00 secs Tue Aug  6 13:23:23
color-ls         -     merlin   ttyp2     121   0.01 secs Tue Aug  6 13:23:02
```

```
telnet          -     merlin   ttyp2     142   2.77 secs Tue Aug  6 13:23:01
Pnews           F     merlin   ttyp3      33   0.01 secs Tue Aug  6 13:22:15
sed             -     merlin   ttyp3     132   0.02 secs Tue Aug  6 13:22:15
Pnews           F     merlin   ttyp3      34   0.01 secs Tue Aug  6 13:22:15
sed             -     merlin   ttyp3     145   0.02 secs Tue Aug  6 13:22:15
cat             -     merlin   ttyp3      80   0.01 secs Tue Aug  6 13:22:13
Pnews           F     merlin   ttyp3      29   0.00 secs Tue Aug  6 13:22:13
```

Il fait faire attention car ce système a tendance à prendre beaucoup de place. La solution pour résoudre ce problème est de lancer le système d'accounting de cette manière :

```
# Lancement de l'accounting
accton /var/log/acct
accttrim -n 2000 /var/log/acct 2> /dev/null
```

8 X Window

Maintenant que tout fonctionne bien, on passe à un autre gros morceau : l'installation de X Window.

Pour pouvoir utiliser X Window, il faut posséder une configuration raisonnable (au moins 8Mo de mémoire). Dans le cas contraire, la machine sera très lente : elle va passer son temps à swapper.

Dans un premier temps, installe à l'aide ta distribution le serveur correspondant à ta carte graphique.

8.1 XF86Config

Le problème avec X Window, c'est de configurer le fichier. Il s'agit du fichier `XF86Config`. Attention : ce fichier est fort différent de l'ancienne version de XFree86 2. Il s'agit ici du serveur XFree86 3.2, basé sur X11R6. Avant de commencer à construire ton fichier, lances le programme `SuperProbe`. Dans mon cas, cela me permet d'obtenir les renseignements suivants (oui, je sais, c'est une vielle carte viédo... mais bon, on fait avec les moyens du bord !) :

```
First video: Super-VGA
        Chipset: Cirrus CL-GD5428
        Memory:  1024 Kbytes
        RAMDAC:  Cirrus Logic Built-in 15/16/24-bit DAC
                 (with 6-bit wide lookup tables (or in 6-bit mode))
```

Ensuite, le plus dur reste à faire : construire le fichier de configuration. Attention : dans certaines distributions, le fichier XF86config est placé dans le répertoire `/etc`, ce qui est une bonne idée dans le cas d'un réseau de stations Linux partageant `/usr` par NFS. Si tu utilises une ou plusieurs machines indépendantes, tu peux supprimer `/etc/XF86config` et utiliser `/usr/X11R6/lib/X11/XF86Config`.

Ce fichier se présente sous la forme de sections. En voici une description :

- Fichiers

```
Section "Files"

    RgbPath     "/usr/X11R6/lib/X11/rgb"

    FontPath    "/usr/X11R6/lib/X11/fonts/misc/"
    FontPath    "/usr/X11R6/lib/X11/fonts/Type1/"
    FontPath    "/usr/X11R6/lib/X11/fonts/Speedo/"
    FontPath    "/usr/X11R6/lib/X11/fonts/75dpi/"
    FontPath    "/usr/X11R6/lib/X11/fonts/100dpi/"
    FontPath    "/usr/X11R6/lib/X11/fonts/xtel/"
EndSection
```

  Dans cette section, tu n'as pas grand chose à modifier. Tu rajoutes ou enlèves les chemins des polices de caractères, et la liste des couleurs RGB.

- Options pour le serveur :

```
Section "ServerFlags"
EndSection
```

  Pour ma part, elle est vide...

- Clavier

```
Section "Keyboard"
    Protocol    "Standard"
#    AutoRepeat  500 5
    ServerNumLock

# Pour cela, voir plus bas
    LeftAlt     Meta
    RightAlt    ModeShift
    # RightCtl    Compose
    ScrollLock  ModeLock
```

- La souris :

```
Section "Pointer"
# pour le protocole, utilise MouseMan, MouseSystems, Logitech ou Microsoft
    Protocol    "MouseMan"

# Peripherique a utiliser
```

```
    Device      "/dev/mouse"
    BaudRate    1200

# Emulation 3 boutons : le click sur les 2 boutons
# simulent le bouton du milieu... pour les souris a deux boutons.
Emulate3Buttons

# ChordMiddle is an option for some 3-button Logitech mice
#    ChordMiddle
EndSection
```

– Le moniteur. Il s'agit de détailler les caractéristiques de ton moniteur. Voici celui qui occupe mon bureau :-). Les fichiers situés dans le répertoire `doc` contiennent les caractéristiques de tout un tas d'écrans. Il est fort possible que tu y trouves ton bonheur.

```
Section "Monitor"
    Identifier  "Sync3N"
    VendorName  "Samsung"
    ModelName   "3N"

    # Frequences :
    Bandwidth   25.2
    HorizSync   31.5-80
    VertRefresh 60-90

    # Les modes dans lesquels ton ecran peut bosser.
    Modeline "800x600"   40 800 840 968 1056 600 601 605 628
    Modeline "640x480"   25.175 640  664  760  800   480 491  493  525
    Modeline "1024x768i" 45 1024 1040 1216 1264 768 768 778 819 interlace
    Modeline "1024x768"  75 1024 1048 1184 1328 768 771 777 806
EndSection
```

– la carte graphique.

```
Section "Device"
    # Baratin
    Identifier  "CartePerso"
    VendorName  "Cirrus"

    # Nom de la carte
    BoardName   "clgd5428"

    #memoire video
    VideoRam    1024
```

```
    # nom du chip
    Chipset     "clgd5428"

    # clocks d'horloges.
    Clocks  25.23  28.32  41.16  36.08  31.50  39.99  45.08  49.87
    Clocks  64.98  72.16  75.00  80.01  85.23
EndSection
```

– Enfin, on réunit tout : la section écran :

```
Section "Screen"
    Driver      "svga"

    # References aux autres sections.
    Device      "CartePerso"
    Monitor     "Sync3N"

    # Le display utilise
    Subsection "Display"
        Depth       8             # 256 couleurs max

        # Modes par priorite : de gauche a droite
        Modes       "1024x768" "800x600" "640x480"
        ViewPort    0 0
        # Resolution virtuelle
        Virtual     1024 768
        option   "linear"
    EndSubsection
EndSection
```

Si ta carte te le permet, tu peux rajouter d'autres sous-sections pour le display :

```
Subsection "Display"
        Depth       32
        Modes       "640x480"
        ViewPort    0 0
        Virtual     800 600
EndSubsection
```

Dans ce cas, ton serveur fonctionnera en 16 millions de couleurs.

La configuration du serveur X est soit très rapide (marche tout de suite), soit très longue. Dans le deuxième cas, voici quelques conseils :

1. lis la documentation se trouvant dans `./doc`. Un bon nombre d'écrans et de cartes sont décrites avec les horloges à utiliser.

2. commence avec une résolution faible (640x480) : c'est plus simple ;
3. si tu as besoin des clocks de ta carte, lance `startx 2>/tmp/err`. Dans le fichier `/tmp/err` vont se trouver tous les messages du serveur, dont le chip de ta carte ainsi que ses clocks. La seule chose qui va te rester à configurer, c'est ton écran.
4. lis le « Video HOWTO », écrit par J.M. Vansteene et disponible sur `ftp.ibp.fr` dans le répertoire `/pub/linux/french/docs/HOWTO` ;
5. utilise le programme `xvidtune` : il permet « d'affiner » la mise au point de l'image (centrage, etc). Par exemple, j'avais un certain tremblement à l'image : avec ce programme, j'ai pu résoudre le problème.

Si jamais l'anglais te rebute, tu peux lire une version française de toute la documentation XFree86 3.2, au format HTML. Pour cela, va sur le serveur *http://www.kheops.com*.

8.2 Clavier français sous X : Xmodmap

On passe maintenant au clavier français sous X Window. Comme XFree a été conçu par des gens se servant de claviers américains (qwertyiop), la configuration du clavier peut sembler être assez affolante. Je te conseille de suivre ce qui suit...

Tout d'abord, tu dois avoir dans le fichier XF86Config les options suivantes :

```
LeftAlt     Meta
RightAlt    ModeShift
# RightCtl    Compose
ScrollLock  ModeLock
```

Ensuite, tu places le fichier suivant dans le répertoire `/usr/X11R6/lib/X11/xinit`. Il doit s'appeler `.Xmodmap`.

Le fichier qui suit a été fait par René Cougnenc. A l'heure actuelle, c'est celui dont je me sers... et je pense qu'il est difficile de faire mieux. Il fonctionne sur XFree86 3.2.

```
! clavier Français AZERTYUIOP pour XFree86 3.2
!
!
! Les accents circonflexes des principales voyelles sont obtenus avec
! la touche Mode_switch (ALT_GR).
!
! Les tremas sont obtenus par ALT_GR + SHIFT, sur 'e', 'i', 'o', et 'u'.
!
! Esperons qu'un jour on pourra obtenir le clavier Francais standard sur
! lequel tout le monde apprend a taper...
! ----------------------------------------------------------------------
!
keycode   9 = Escape
keycode  10 = ampersand 1
```

```
keycode  11 = eacute 2 asciitilde
keycode  12 = quotedbl 3 numbersign
keycode  13 = apostrophe 4 braceleft
keycode  14 = parenleft 5 bracketleft
keycode  15 = minus 6 bar
keycode  16 = egrave 7 grave
keycode  17 = underscore 8 backslash
keycode  18 = ccedilla 9 asciicircum
keycode  19 = agrave 0 at
keycode  20 = parenright degree bracketright
keycode  21 = equal plus braceright
keycode  22 = BackSpace
keycode  23 = Tab
keycode  24 = a A acircumflex
keycode  25 = Z
keycode  26 = e E ecircumflex ediaeresis
keycode  27 = R
keycode  28 = T
keycode  29 = Y
keycode  30 = u U ucircumflex udiaeresis
keycode  31 = i I icircumflex idiaeresis
keycode  32 = o O ocircumflex
keycode  33 = P
keycode  34 =  dead_circumflex dead_diaeresis
keycode  35 = dollar sterling
keycode  36 = Return
keycode  37 = Control_L
keycode  38 = Q
keycode  39 = S
keycode  40 = D
keycode  41 = F
keycode  42 = G
keycode  43 = H
keycode  44 = J
keycode  45 = K
keycode  46 = L
keycode  47 = M
keycode  48 = ugrave percent
keycode  49 = guillemotleft guillemotright
keycode  50 = Shift_L
keycode  51 = asterisk mu
keycode  52 = W
keycode  53 = X
keycode  54 = C
```

```
keycode  55 = V
keycode  56 = B
keycode  57 = N
keycode  58 = comma question
keycode  59 = semicolon period
keycode  60 = colon slash
keycode  61 = exclam slash
keycode  62 = Shift_R
keycode  63 = asterisk
keycode  64 = Alt_L
keycode  65 = space
keycode  66 = Caps_Lock
keycode  67 = F1
keycode  68 = F2
keycode  69 = F3
keycode  70 = F4
keycode  71 = F5
keycode  72 = F6
keycode  73 = F7
keycode  74 = F8
keycode  75 = F9
keycode  76 = F10
keycode  77 = Num_Lock
keycode  78 = Scroll_Lock
keycode  79 = 7
keycode  80 = 8
keycode  81 = 9
keycode  82 = KP_Subtract
keycode  83 = 4
keycode  84 = 5
keycode  85 = 6
keycode  86 = KP_Add
keycode  87 = 1
keycode  88 = 2
keycode  89 = 3
keycode  90 = 0
keycode  91 = period
keycode  92 = 0x1007ff00
keycode  93 =
keycode  94 = less greater
keycode  95 = F11
keycode  96 = F12
keycode  97 = Home
keycode  98 = Up
```

```
keycode  99 = Prior
keycode 100 = Left
keycode 101 = Begin
keycode 102 = Right
keycode 103 = End
keycode 104 = Down
keycode 105 = Next
keycode 106 = Insert
keycode 107 = Delete
keycode 108 = KP_Enter
keycode 109 = Control_R
keycode 110 = Pause
keycode 111 = Print
keycode 112 = KP_Divide
keycode 113 = Mode_switch
keycode 114 = Break

! Nouvelles valeurs pour le pave numerique a partir de XFree 3.1.1
!keycode 147 = Home
!keycode 148 = Up
!keycode 149 = Prior
!keycode 150 = Left
!keycode 152 = Right
!keycode 153 = End
!keycode 154 = Down
!keycode 155 = Next
!keycode 156 = Insert
!keycode 157 = Delete

!keycode  136 = 7
!keycode  137 = 8
!keycode  138 = 9
keycode  82 = KP_Subtract
!keycode  139 = 4
!keycode  140 = 5
!keycode  141 = 6
!keycode  86 = KP_Add
!keycode  142 = 1
!keycode  143 = 2
!keycode  144 = 3
!keycode  145 = 0
!keycode  146 = period
! ------------------------------------------------------------------------
```

Les caractères « et » sont obtenus respectivement en appuyant sur la touche où il y a un petit deux (en haut à gauche, sous "Echappement") pour « et Shift + deux pour ».

8.3 Clavier français sous X... 2ième possibilité

Rq: personnellement, j'utilise la première solution sans aucun problème... Si jamais tu as trop de problèmes, essaye celle-ci.

Cette solution nous vient de Michel Billaud. Bon, on part de la constatation que les touches accent-grave (ALT-4) et accent-aigu (ALT 7) sont pénibles à utiliser : si elles sont muettes il faut les doubler la plupart du temps, si elles ne le sont pas on ne peut pas accentuer les E A U etc.

Donc, il faut que ALT 4 et ALT 7 produisent de vrais accents aigus et apostrophe. Il nous faut quand même des touches muettes (en plus). Il va falloir modifier la « map » du clavier.

On va donc faire :

- ALT 4 = accent aigu en dur
- ALT 7 = accent grave en dur
- ALT 9 = circonflexe en dur
- ALT ^ = accent aigu
- ALT ù = accent grave

Dans le répertoire `/usr/lib/kbd/keytables.map`, copier `fr-lat1.map` sous un autre nom (`monclavier.map`). Lancer `showkey` pour voir le numéro des touches concernées dans l'ordre : 5 8 10 26 et 40. Puis éditer `monclavier.map` pour faire les modifications.

Pour tester, faire `loadkeys monclavier`. Lorsque tout semble fonctionner, modifier `/etc/rc.d/rc.keymap` pour qu'il charge *monclavier.map* et gribouiller le clavier avec un crayon, pas un feutre... :-).

On peut aussi mettre un « tilde muet » sur le 2 supérieur et un « tilde dur » (indispensable au shelliste) sur ALT-2.

8.4 xdm

Pour démarrer X Window directement au boot (`xdm`), tu peux modifier le fichier `/etc/inittab` en changeant la ligne contenant `initdefault`. Par exemple sur ma machine, il suffit de remplacer

```
id:5:initdefault:
#id:6:initdefault:
```

par

```
#id:5:initdefault:
id:6:initdefault:
```

Mais **attention**! Certaines distribution utilisent le niveau 6 pour réamorcer. Le plus simple est de regarder dans le répertoire `/etc/rc.d` si le fichier `rc.6` lance bien *xdm*.

Pour lancer `xdm` en 16 ou 32 bits, il faut modifier le fichier `/usr/X11/lib/X11/xdm` et mettre :

```
:0 local /usr/X11/bin/X -bpp 16 (ou 32)
```

8.5 Comment démarrer plusieurs sessions X ?

Afin de pouvoir démarrer plusieurs sessions X sur différentes consoles sans avoir à spécifier le « display » à utiliser, ajoutes à la fin du fichier startx :

```
CONSOLE=‘tty‘
CONSOLE=‘basename $CONSOLE‘
CONSOLE=‘echo $CONSOLE | sed s/tty//‘
CONSOLE=‘expr $CONSOLE - 1‘

echo "Starting X on display "$CONSOLE

exec xinit $clientargs -- $serverargs :$CONSOLE
```

8.6 Comment récupérer les erreurs sur la console?

Il arrive fréquemment que les erreurs se trouvent dans la fenêtre texte virtuelle à partir de laquelle tu as lancé le serveur. Une solution consiste à lancer le serveur de cette manière :

```
xinit > /dev/console 2>&1
```

Si tu utilises la Slackware, il suffit d'éditer le script `startx` et de remplacer la ligne

```
exec xinit $clientargs -- $serverargs
```

par

```
exec xinit $clientargs -- $serverargs >/dev/console 2>&1
```

8.7 XFree et XInput

A partir de la version XFree 3.1.2B, le serveur X de XFree86 supporte l'extension XInput. Cette extension permet d'avoir plusieurs périphériques d'entrées. Dans XFree86, il y a en a 3 pour l'instant :

- une tablette Wacom ou compatible avec le protocole wacom IV ;
- un écran tactile Elographics ;

– un joystick.

Le support pour ces périphériques est chargé dynamiquement dans le serveur à partir de la version 3.1.2D en ajoutant les lignes suivantes au fichier XF86Config:

```
Section "Module"
    Load "xf86Wacom.so"
    Load "xf86Elo.so"
    Load "xf86Jstk.so"
EndSection
```

Les modules sont cherchés dans un chemin d'accès prédéfini: /usr/X11R6/lib/modules ou bien suivant un chemin spécifié dans la section « Files »:

```
Section "Files"
    ModulePath  "/usr/X11R6/lib/,/usr/X11R6/lib/modules/"
    ModulePath  "/usr/X11R6/bin"
EndSection
```

Le ModulePath doit être specifié avant la section « Module » pour fonctionner.

Chaque périphérique possède sa propre sous-section de configuration. Consultez la page de manuel de XF86Config pour avoir le détail. Un péripherique n'est activé que s'il possède une sous-section. L'exemple suivant active un joystick et une tablette:

```
Section "XInput"
    SubSection "Joystick"
        Port                    "/dev/js0"
        Timeout                 20
        MinimumXPosition        28
        MaximumXPosition        1206
        MinimumYPosition        27
        MaximumYPosition        1239
        Delta                   20
    EndSubsection

    SubSection "WacomStylus"
        Port            "/dev/ttyS2"
    EndSubsection
EndSection
```

Ensuite le serveur peut émettre des évènements étendus ou bien basculer un périphérique en *core pointer*. Deux programmes existent pour manipuler les devices étendus:

– `xsetpointer` bascule un périphérique en core pointer ou liste les périphériques;

– `xsetmode` passe un périphérique en absolu ou en relatif. On doit changer le mode d'un périphérique avant qu'il ne soit basculé en core pointer.

8.8 Le fichier .Xauthority

Le fichier `.Xauthority` est généré par le programme `xauth`. Il s'agit d'un système d'authentification pour les applications graphiques. Cela permet d'éviter que d'autres personnes d'envoient des images, des fenêtres sur ton écran - mais également que des personnes puissent « voir » ce qu'il y a sur ton écran.

Le principe est de donner une clef d'indentification, en hexadécimal avec un nombre paire de caractères.

Tu lances `xauth` et tu fait :

```
add MaMachine:0 MIT-MAGIC-COOKIE-1 MonCode
add MaMachine/unix:0 MIT-MAGIC-COOKIE-1 MonCode
```

Pour la machine, c'est en fait « `hostname:NoDisplay` ».

Un fois lancé, ton serveur X interdit toute connexion, sauf si l'application :

– est exécuté sur une machine référencée ;

– possède le code.

Tu peux désactiver le système pour certaines machines avec un `xhost +toto` par exemple.

Rq: certaines versions obligent à lancer le serveur X avec la commande

```
xinit -- -auth $HOME/.Xauthority
```

.

8.9 Problème assez courant

Une question qui revient assez souvent : *je ne sais pas pourquoi sous XFree 86, de temps à autres je n'ai plus accès aux boutons et aux menus déroulant, sauf ceux des fenêtres système?* . La réponse est très simple : déverrouiller Scroll Lock (« arrêt défil ») !

9 Sécurité

9.1 Présentation

Cette partie est destinée aux administrateurs système de sites qui désirent installer Linux.

9.2 Problèmes de sécurité

La distribution Slackware contient certaines failles. Tu peux en consulter la liste à l'adresse suivante :

`http://bach.cis.temple.edu/pub/linux/linux-security/`

Linux est d'une manière générale très solide. En effet, dès qu'une faille est découverte, elle est immédiatement référencée et corrigée... C'est l'avantage d'avoir des sources publics. Toutefois, certaines failles importantes existent dans les distributions, et il est important de les corriger. Un document WEB propose une liste des problèmes :

`http://bach.cis.temple.edu/pub/linux/linux-security/Linux-Security-FAQ/`

9.3 Protection de la machine contre l'extérieur

Une solution pour éviter les connexions externes est d'utiliser TCP/Wrappers. Il est très fortement conseillé de le recompiler !

L'installation est assez intuitive. En bref, il te suffit d'indiquer le nom des machines autorisées dans le fichier `/etc/hosts.allow` et les machines interdites dans `/etc/hosts.deny`. On peut permettre l'envoi de courrier lorsqu'une machine tente de se connecter alors qu'elle est interdite en mettant par exemple dans le fichier `/etc/hosts.deny` :

```
wu.ftpd:        ALL: twist = /usr/sbin/real-daemon-dir/safe_finger -l @%h |
/bin/mail -s %d-%h root
```

(Sur une seule ligne :-)).

Si tu veux plus de détail, tu peux lire le document suivant : `ftp.win.tue.nl:/pub/security/tcp_wrapper.ps.Z`.

9.4 Quelques pointeurs

En France, le serveur `ftp.urec.fr` contient de nombreux utilitaires relatifs à la sécurité informatique.

Deux listes de diffusion sont consacrées à la diffusion d'informations liées à des problèmes de sécurité sous Linux : `linux-security@tarsier.cv.nrao.edu` et `linux-alert@tarsier.cv.nrao.edu`

10 Recompiler le noyau

10.1 Pentium

Linux gère sans aucun problème les machines basées sur Pentium. Pour optimiser un peu la compilation, il suffit de spécifier Pentium lors de la recompilation du noyau. Il existe toutefois un gcc spécialisé pour Pentium, qui est en cours de développement.

Il suffit de consulter la page `http://www.goof.com/pcg` pour plus de renseignements.

10.2 Compilation

Les nouveaux sources du noyau **Linux** peuvent être récupérés sur `ftp.ibp.fr` dans le répertoire `/pub/linux/kernel/sour` pour les versions dites « stables ». Il s'agit d'un miroir du site `ftp.funet.fi`.

Il existe un autre répertoire : `v2.1` : il s'agit de la version de développement. Je te déconseille de l'installer : il s'agit de noyaux en cours de développement; ils peuvent être stables mais il est également possible qu'ils soient complètement bogués. En les utilisant, tu prends le risque de perdre des données sur ta machine.

Tu installes donc les sources (normalement dans le répertoire `/usr/src/linux`). La compilation est très simple à réaliser mais tu dois suivre quelques points très scrupuleusement :

1. conserver à portée de la main une disquette permettant de démarrer le système et de monter la partition root, afin de pouvoir reprendre le contrôle en cas d'erreur. tester qu'elle permet bien de démarrer.

2. `make config` va te demander la configuration que tu désires. Certains points peuvent te sembler assez obscurs. Dans ce cas, ne les change pas.

   Configurons ensemble... Remarque importante : il s'agit du noyau 2.0.26... donc certains champs risquent ne pas exister dans d'autres versions.

   ```
   # make config
   rm -f include/asm
   ( cd include ; ln -sf asm-i386 asm
   /bin/sh scripts/Configure -d arch/i386/config.in
   ```

   Cela n'a que peu d'intérêt : il s'agit de positionner des liens symboliques, mais il ne faut pas oublier !

   ```
   *
   * Code maturity level options
   *
   Prompt for development and/or incomplete code/drivers (CONFIG_EXPERIMENTAL) [Y/n/?]
   *
   ```

   Cette option permet de spécifier si tu souhaites avoir certaines options particulières. Les options ainsi proposées peuvent entrainer l'inclusion de périphériques pas encore arrivé à maturité. Toutefois, si tu souhaites utiliser les modules chargeables avec `kerneld`, répond `Y`.

   ```
   *
   * Loadable module support
   *
   Enable loadable module support (CONFIG_MODULES) [Y/n/?]
   Set version information on all symbols for modules (CONFIG_MODVERSIONS) [N/y/?]
   Kernel daemon support (e.g. autoload of modules) (CONFIG_KERNELD) [Y/n/?]
   ```

   Cette partie est spécifique aux modules chargeables.

Pour répondre aux questions, il y a trois solutions : soit l'option par défaut (lettre en majuscule) est déjà adéquate, soit il suffit de la préciser. Si tu ne sais pas à quoi correspond cette option, tape ? : une brève description va alors apparaître. Dans le cas présent ci-dessus, je vais utiliser les modules chargeables avec le système de chargement automatique.

```
*
* General setup
*
Kernel math emulation (CONFIG_MATH_EMULATION) [N/y/?]
Networking support (CONFIG_NET) [Y/n/?]
Limit memory to low 16MB (CONFIG_MAX_16M) [N/y/?]
PCI bios support (CONFIG_PCI) [N/y/?]
System V IPC (CONFIG_SYSVIPC) [Y/n/?]
Kernel support for a.out binaries (CONFIG_BINFMT_AOUT) [Y/m/n/?]
Kernel support for ELF binaries (CONFIG_BINFMT_ELF) [Y/m/n/?]
Kernel support for JAVA binaries (CONFIG_BINFMT_JAVA) [N/y/m/?]
Compile kernel as ELF - if your GCC is ELF-GCC (CONFIG_KERNEL_ELF) [Y/n/?]
Processor type (386, 486, Pentium, PPro) [486]
  defined CONFIG_M486
```

La première option n'est nécessaire que si tu ne disposes pas d'un processeur arithmétique. Je te conseille très fortement de l'éviter. Si tu disposes d'une machine avec PCI et non 486, il te suffit de donner les caractéristiques de ta machine.

Du côté des formats exécutables, il est préférables d'activer les formats ELF et a.out.

```
*
* Floppy, IDE, and other block devices
*
Normal floppy disk support (CONFIG_BLK_DEV_FD) [M/n/y/?]
Enhanced IDE/MFM/RLL disk/cdrom/tape support (CONFIG_BLK_DEV_IDE) [Y/n/?]
```

Je demande ici a gérer mon lecteur de disquettes comme module chargeable, et d'une manière générale, j'active le support IDE.

```
*
* Please see Documentation/ide.txt for help/info on IDE drives
*
   Use old disk-only driver on primary interface (CONFIG_BLK_DEV_HD_IDE) [N/y/?]
   Include IDE/ATAPI CDROM support (CONFIG_BLK_DEV_IDECD) [N/y/?]
   Include IDE/ATAPI TAPE support (CONFIG_BLK_DEV_IDETAPE) [N/y/?]
   Support removable IDE interfaces (PCMCIA) (CONFIG_BLK_DEV_IDE_PCMCIA) [N/y/?]
   CMD640 chipset bugfix/support (CONFIG_BLK_DEV_CMD640) [N/y/?]
   Other IDE chipset support (CONFIG_IDE_CHIPSETS) [N/y/?]
```

Si tu disposes de matériel ATAPI, il faut que tu le précises ici. De même si tu as une interface PCMCIA. Il faut remarquer que Linux te propose certaines options particulières pour certains contrôleurs IDE connus pour être bogués. Si tu es dans cette situation, n'hésite pas à sélectionner l'option.

```
*
* Additional Block Devices
*
Loopback device support (CONFIG_BLK_DEV_LOOP) [N/y/m/?]
Multiple devices driver support (CONFIG_BLK_DEV_MD) [N/y/?]
RAM disk support (CONFIG_BLK_DEV_RAM) [N/y/m/?]
XT harddisk support (CONFIG_BLK_DEV_XD) [N/y/m/?]
```

A moins que tu saches exactement ce que tu fais, ne pas modifier.

Bien, on passe au réseau maintenant !

```
*
* Networking options
*
Network firewalls (CONFIG_FIREWALL) [N/y/?]
Network aliasing (CONFIG_NET_ALIAS) [N/y/?]
TCP/IP networking (CONFIG_INET) [Y/n/?]
IP: forwarding/gatewaying (CONFIG_IP_FORWARD) [N/y/?]
IP: multicasting (CONFIG_IP_MULTICAST) [N/y/?]
IP: accounting (CONFIG_IP_ACCT) [Y/n/?]
```

D'une manière générale, il est suggéré d'activer Tcp/Ip, même si tu n'est pas relié au réseau (certaines applications en ont besoin.). Le reste dépend de ta machine. Je passe sur les deux séries qui suivent pour le réseau.

```
*
* SCSI support
*
```

etc...etc : Pour le SCSI. Liste des cartes SCSI gérées... choisir la bonne (ou plusieurs).

```
*
* Network device support
*
Network device support (CONFIG_NETDEVICES) [Y/n/?]
Dummy net driver support (CONFIG_DUMMY) [M/n/y/?]
EQL (serial line load balancing) support (CONFIG_EQUALIZER) [N/y/m/?]
Frame relay DLCI support (EXPERIMENTAL) (CONFIG_DLCI) [N/y/m/?]
PLIP (parallel port) support (CONFIG_PLIP) [N/y/m/?]
PPP (point-to-point) support (CONFIG_PPP) [M/n/y/?]
```

Liste des possibilités réseau... ne pas oublier de positionner PPP ou SLIP si nécessaire.

Ensuite viennent les cartes elles-mêmes. Elles ne sont pas détaillées ici.

```
*
* ISDN subsystem
*
ISDN support (CONFIG_ISDN) [N/y/m/?]
```

Support des cartes RNIS. A moins d'en avoir une, ne pas activer.

```
*
* CD-ROM drivers (not for SCSI or IDE/ATAPI drives)
*
Support non-SCSI/IDE/ATAPI CDROM drives (CONFIG_CD_NO_IDESCSI) [N/y/?]
*
```

Pour les CD-ROM Sony CDU31A/CDU33A, Matsushita/Panasonic/Creative, etc. En bref, les CD branchés sur les cartes son par exemple.

```
*
* Filesystems
*
Quota support (CONFIG_QUOTA) [Y/n/?]
Mandatory lock support (CONFIG_LOCK_MANDATORY) [N/y/?]
Minix fs support (CONFIG_MINIX_FS) [M/n/y/?]
Extended fs support (CONFIG_EXT_FS) [N/y/m/?]
Second extended fs support (CONFIG_EXT2_FS) [Y/m/n/?]
xiafs filesystem support (CONFIG_XIA_FS) [N/y/m/?]
DOS FAT fs support (CONFIG_FAT_FS) [M/n/y/?]
MSDOS fs support (CONFIG_MSDOS_FS) [M/n/?]
VFAT (Windows-95) fs support (CONFIG_VFAT_FS) [M/n/?]
umsdos: Unix like fs on top of std MSDOS FAT fs (CONFIG_UMSDOS_FS) [N/m/?]
/proc filesystem support (CONFIG_PROC_FS) [Y/n/?]
NFS filesystem support (CONFIG_NFS_FS) [M/n/y/?]
SMB filesystem support (to mount WfW shares etc..) (CONFIG_SMB_FS) [N/y/m/?]
ISO9660 cdrom filesystem support (CONFIG_ISO9660_FS) [M/n/y/?]
OS/2 HPFS filesystem support (read only) (CONFIG_HPFS_FS) [N/y/m/?]
System V and Coherent filesystem support (CONFIG_SYSV_FS) [N/y/m/?]
Amiga FFS filesystem support (EXPERIMENTAL) (CONFIG_AFFS_FS) [N/y/m/?]
UFS filesystem support (read only) (CONFIG_UFS_FS) [N/y/m/?]
```

Il s'agit de configurer la liste des gestionnaires de fichiers gérés par Linux. Tu dois absolument avoir ext2fs. Il est très fortement suggéré d'y mettre également le support /proc.

SMB est utilisé pour accéder aux ressources partagées de MS-Windows For Workgroup. Désormais, il existe deux formats pour MS-DOS : le format standard et le format étendu qu'utilise MS-Windows 95 (VFAT). Si tu possèdes un CD-ROM, penses à inclure le support ISO9660.

Enfin, si tu désires utiliser les quota, penses à répondre Y à la première option.

```
*
* character devices
*
```

Support pour le port parallèle et pour diverses souris. Penses à activer l'imprimante si elle est connectée sur le port parallèle!

```
*
* Sound
*
```

Passons à la carte son! Je vais un peu limiter et me cantoner à la SoundBlaster 16. C'est ce que j'ai, et ça permet de donner un exemple. Seule les options qu'il convient de modifier sont présentes ici. En fait, Linux gère bien d'autres cartes son.

```
Sound card support (CONFIG_SOUND) [M/n/y/?]
Sound Blaster (SB, SBPro, SB16, clones) support (CONFIG_SB) [Y/n/?]
/dev/dsp and /dev/audio support (CONFIG_AUDIO) [Y/n/?]
MIDI interface support (CONFIG_MIDI) [Y/n/?]
FM synthesizer (YM3812/OPL-3) support (CONFIG_YM3812) [Y/n/?]
I/O base for SB Check from manual of the card (SBC_BASE) [260]
Sound Blaster IRQ Check from manual of the card (SBC_IRQ) [5]
Sound Blaster DMA 0, 1 or 3 (SBC_DMA) [1]
Sound Blaster 16 bit DMA (_REQUIRED_for SB16, Jazz16, SMW) 5, 6 or 7 (use 1 for 8 bit cards) (SB
MPU401 I/O base of SB16, Jazz16 and ES1688 Check from manual of the card (SB_MPU_BASE) [330]
SB MPU401 IRQ (Jazz16, SM Wave and ES1688) Use -1 with SB16 (SB_MPU_IRQ) [-1]
Audio DMA buffer size 4096, 16384, 32768 or 65536 (DSP_BUFFSIZE) [65536]

*
* Kernel hacking
*
Kernel profiling support (CONFIG_PROFILE) [N/y]
```

Il est très conseillé de laisser cette option à N.

3. `make dep` se charge de créer les dépendances.
4. `make clean` va nettoyer toute l'arborescence.

5. Lance `make zImage` pour compiler ton noyau. Tu peux aller boire un bon café (voire toute autre boisson dont la publicité est interdite...) : c'est long. Bien sûr si tu possède un P6 ou autre (temps de compilation environ 4 minutes, tu peux regarder)... mais avec mon DX2/66, la compilation prend environ 40 minutes. Cela dépend surtout de la quantité de mémoire vive disponible.

6. si tu as des modules, lances leur compilation par : `make modules` et installes-les par `make modules_install`. Ils seront copiés dans le répertoire `/lib/modules/2.0.26`.

7. Quand c'est terminé, tu vas dans `/usr/src/linux/arch/i386/boot`. Ton nouveau noyau flambant neuf est le fichier *zImage*, qui est déjà compacté. Copie le dans la racine : `/` , sous le nom attendu par lilo (généralement « vmlinuz »). Puis relancer `lilo` pour que le nouveau noyau soit pris en compte et reboote.

Remarque : je te conseille fortement de laisser ton ancien noyau dans un coin, et surtout de pouvoir booter dessus. Modifier pour cela la configuration de LILO mais ne rien retirer : ajouter en tête du fichier une section dérivant le nouveau noyau.

Tu rebootes et ça doit fonctionner.

11 Les fichiers spéciaux

Sous **Linux** et plus généralement sous Unix, les entrées-sorties sur un périphérique se font grâce à des fichiers situés dans le répertoire `/dev/`. Chacun d'entre eux est identifié par un nom (hda comme premier disque dur du contrôleur IDE, par exemple) mais surtout par deux numéros uniques : *majeur* et *mineur*. Ces fichiers dits *spéciaux* sont de deux sortes : *caractères* (terminal, imprimante, etc) ou *blocs* (disque par exemple).

Il est important de connaître ces fichiers. Les diverses distributions (Slackware, jurix, SLS, etc) créent ces fichiers. Or, certaines versions génèrent de mauvais fichiers (Slackware). De même, il est inutile de créer les fichiers nécessaires à l'utilisation des périphériques SCSI si l'on ne possède pas de carte.

Pour créer un fichier, il suffit de faire `mknod Nom_Fichier Type Major Minor`. D'autres options existent, consulte le manuel.

Un fichier de commandes appelé `MAKEDEV` est également disponible dans le répertoire `/dev`. Il permet de créer des fichiers spéciaux sans appeler directement `mknod`.

La liste de périphérique qui va suivre provient du document *LINUX ALLOCATED DEVICES* écrit par H. Peter Anvin `hpa@zytor.com`. Cette liste est fournie dans le répertoire de documentations des sources du noyau (`Documentation/devices.txt`) à partir de la version 1.3 du noyau.

Si tu veux développer un pilote de périphérique et ajouter un périphérique à la liste, contacte le, **après avoir lu le document original**. En effet la liste ci-dessous est quelque peu épurée.

Cette liste correspond à la version 2.0.26

- 0 : Périphérique Nul et sans nom.
- 1 car : Mémoire
  - 1 : /dev/mem : accès mémoire physique

  - 2 : /dev/kmem : accès mémoire virtuelle
  - 3 : /dev/null : périphérique nul
  - 4 : /dev/port
  - 5 : /dev/zero : source d'octets à 0
  - 6 : /dev/core : OBSOLETE (remplacé par /proc/kcore)
  - 7 : /dev/full : retourne ENOSPC lors d'une écriture
  - 8 : /dev/random : génération de nombres aléatoires
  - 9 : /dev/urandom : plus rapide, moins sûr.
- 1 bloc : disque virtuel :
  - 0 : /dev/ram0 : 1er disque
  - ...
  - 7 : /dev/ram7 : dernier disque
  - 250 : /dev/initrd : disque virtuel d'initialisation

  Les anciens noyaux possèdent /dev/ramdisk (1,1) à la place et `/dev/initrd` fait référence à un disque virtuel qui a été préalablement chargé par le *boot loader*.
- 2 car : Réservé pour les pseudo-tty
  - 0 : /dev/ptyp0 : 1er périphérique
  - ...
  - 255 : /dev/ptyef : dernier périphérique

  Les maîtres sont les *pty* et les esclaves sont les *tty*.
- 2 bloc : Lecteurs de disquettes
  - 0 : /dev/fd0 1er lecteur autodétecté
  - 1 : /dev/fd1 2nd
  - 2 : /dev/fd2 3ème
  - 3 : /dev/fd3 4ème
- 3 car : Réservé pour les pty esclaves
  - 0 : /dev/ttyp0 : 1er terminal esclave
  - ...
  - 256 : /dev/ttyef : dernier terminal esclave
- 3 bloc : 1er interface IDE/CD-ROM (pour MFM, RLL et IDE)
  - 0 : /dev/hda Disque maître
  - 64 : /dev/hdb Disque esclave (ou CD-ROM)

  - Pour les partitions, ajoutez au mineur le nombre :
    - 0 : /dev/hd? Tout le disque
    - 1 : /dev/hd?1 1ère partition principale
    - 2 : /dev/hd?2 2ème partition principale
    - 3 : /dev/hd?3 3ème partition principale
    - 4 : /dev/hd?4 4ème partition principale
    - 5 : /dev/hd?5 1ère partition logique
    - 6 : /dev/hd?6 2ème partition logique
    - 7 : /dev/hd?7 3ème partition logique
    - ...
    - 63 : /dev/hd?63 59ème partition logique
- 4 car : Périphériques TTY
  - 0 : /dev/console Périphérique de la console
  - 1 : /dev/tty1 1ère console virtuelle
  - ...
  - 63 : /dev/tty63 63ème console virtuelle
  - 64 : /dev/ttyS0 1er port série
  - ...
  - 127 : /dev/ttyS63 64ème port série
  - 128 : /dev/ptyp0 1er pseudo tty maître
  - ...
  - 191 : /dev/ptysf 64ème pseudo tty maître
  - 192 : /dev/ttyp0 1er pseudo tty esclave
  - ...
  - 255 : /dev/ttysf 64ème pseudo tty esclave
- 5 car : Autres périphériques TTY
  - 0 : /dev/tty périphérique TTY courant
  - 64 : /dev/cua0 Périphérique d'appel vers extérieur correspondant à ttyS0
  - ...
  - 127 : /dev/cua63 Périphérique d'appel vers extérieur correspondant à ttyS63
- 6 car : Périphériques pour imprimantes parallèles
  - 0 : /dev/lp0 1ère imprimante parallèle (0x3bc)
  - 1 : /dev/lp1 2nde imprimante parallèle (0x378)
  - 2 : /dev/lp2 3ème imprimante parallèle (0x278)

- 7 car : Contenu des consoles virtuelles (vcs)
  - 0 : /dev/vcs Contenu texte la console courante
  - 1 : /dev/vcs1 Contenu texte de tty1
  - ...
  - 63 : /dev/vcs63 Contenu texte de tty63
  - 128 : /dev/vcsa Attributs/Texte de la console courante
  - 129 : /dev/vcsa1 Attributs/Texte de tty1
  - ...
  - 191 : /dev/vcsa63 Attributs/Texte de tty63
- 7 bloc : périphériques dit *loopback*. Utilisés pour mounter des périphériques qui ne sont pas associés à des périphériques de type bloc.
  - 0 : /dev/loop0 : 1er périphérique
  - 1 : /dev/loop1 : second...
  - ...
- 8 bloc :Périphériques disques SCSI
  - 0 : /dev/sda 1er disque SCSI
  - 16 : /dev/sdb 2nd disque SCSI
  - 32 : /dev/sdc 3ème disque SCSI
  - ...
  - 240 : /dev/sdp 16ème disque SCSI
- 9 car : Périphériques lecteurs de bandes SCSI
  - 0 : /dev/st0 1er lecteur SCSI de bandes mode 0
  - 1 : /dev/st1 2nd lecteur SCSI de bandes mode 0
  - ...
  - 32 : /dev/st0l 1er lecteur SCSI de bandes mode 1
  - 33 : /dev/st1l 2nd lecteur SCSI de bandes mode 1
  - ...
  - 64 : /dev/st0m 1er lecteur SCSI de bandes mode 2
  - 65 : /dev/st1m 2nd lecteur SCSI de bandes mode 2
  - ...
  - 96 : /dev/st0a 1er lecteur SCSI de bandes mode 3
  - 97 : /dev/st1a 2nd lecteur SCSI de bandes mode 3
  - ...

  - 128 : /dev/nst0 1er lecteur SCSI de bandes mode 0,sans rembobinage
  - 129 : /dev/nst1 2nd lecteur SCSI de bandes mode 0, sans rembobinage
  - ...
  - 160 : /dev/st0l 1er lecteur SCSI de bandes mode 1,sans rembobinage
  - 161 : /dev/st1l 2nd lecteur SCSI de bandes mode 1,sans rembobinage
  - ...
  - 192 : /dev/st0m 1er lecteur SCSI de bandes mode 2,sans rembobinage
  - 193 : /dev/st1m 2nd lecteur SCSI de bandes mode 2,sans rembobinage
  - ...
  - 224 : /dev/st0a 1er lecteur SCSI de bandes mode 3,sans rembobinage
  - 225 : /dev/st1a 2nd lecteur SCSI de bandes mode 3,sans rembobinage
  - ...

- 9 bloc : Périphériques méta-disques*RAID*
  - 0 : /dev/md0 1er groupe de périphériques
  - 1 : /dev/md1 2nd groupe de périphériques
  - ...

- 10 car : Souris non-séries. Divers
  - 0 : /dev/logibm : souris Logitech
  - 1 : /dev/psaux : souris style PS/2
  - 2 : /dev/inportbm : souris Microsoft
  - 3 : /dev/atibm : souris ATI XL
  - 4 : /dev/jbm : J-mouse
  - 4 : /dev/amigamouse : souris amiga (68k/Amiga)
  - 5 : /dev/atarimouse : souris Atari
  - 6 : /dev/sunmouse : souris Sun
  - 128 : /dev/beep : beep !
  - 129 : /dev/modreq : requète de chargement de module
  - 130 : /dev/watchdog : port timer WatchDog
  - 131 : /dev/temperature : temperature interne de la machine
  - 132 : /dev/hwtrap : capture d'erreurs matérielles
  - 133 : /dev/exttrp : capture d'erreurs externes
  - 134 : /dev/apm_bios : BIOS *Advanced Power Management*
  - 135 : /dev/rtc : *Real Time Clock*
  - 136 : /dev/qcam0 : QuickCam sur lp0

    - 137 : /dev/qcam1 : QuickCam sur lp1
    - 138 : /dev/qcam2 : QuickCam sur lp2
    - 139 : /dev/openprom : SPARC OpenBoot PROM
- 11 car : clavier Sun (mode raw)
    - 0 : /dev/kbd
- 11 bloc : Périphériques CD-ROM SCSI
    - 0 : /dev/sr0 1er CD-ROM SCSI
    - 1 : /dev/sr1 2nd CD-ROM SCSI
    - ...
- 12 car : lecteurs de bandes QIC-02
    - 2 : /dev/ntpqic11 QIC-11, sans rembobinage
    - 3 : /dev/tpqic11 QIC-11, avec rembobinage
    - 4 : /dev/ntpqic24 QIC-24, sans rembobinage
    - 5 : /dev/tpqic24 QIC-24, avec rembobinage
    - 6 : /dev/ntpqic120 QIC-120, sans rembobinage
    - 7 : /dev/tpqic120 QIC-120, avec rembobinage
    - 8 : /dev/ntpqic150 QIC-150, sans rembobinage
    - 9 : /dev/tpqic150 QIC-150, avec rembobinage
- 12 bloc : Support CD-ROM MSCDEX
    - 0 : /dev/dos_cd0 1er CD-ROM MSCDEX
    - 1 : /dev/dos_cd1 2nd CD-ROM MSCDEX
    - ...
- 13 car : Haut parleur du PC
    - 0 : /dev/pcmixer Emulation de /dev/mixer
    - 1 : /dev/pcsp Emulation de /dev/dsp (8-bit)
    - 4 : /dev/pcaudio Emulation de /dev/audio
    - 5 : /dev/pcsp16 Emulation de /dev/dsp (16-bit)
- 13 bloc : Controleur 8-bit MFM/RLL/IDE
    - 0 : /dev/xda 1er disque XT
    - 64 : /dev/xdb 2nd disque XT
- 14 car : Carte son
    - 0 : /dev/mixer Contrôle du mixage

  - 1 : /dev/sequencer Séquenceur audio
  - 2 : /dev/midi00 1er port MIDI
  - 3 : /dev/dsp Audio digital
  - 4 : /dev/audio Audio compatible Sun
  - 6 : /dev/sndstat Etat de la carte son
  - 8 : /dev/sequencer2 Autre périphérique pour séquenceur
  - 16 : /dev/mixer1 contrôle du mixage Second pour 2ème carte son
  - 17 : /dev/patmgr0 Gestionnaire du séquenceur patch
  - 18 : /dev/midi01 2nd port MIDI
  - 19 : /dev/dsp1 Audio digital pour 2nde carte son
  - 20 : /dev/audio1 Audio compatible sun pour 2nde carte son
  - 33 : /dev/patmgr1 Gestionnaire du séquenceur patch pour 2nde carte son
  - 34 : /dev/midi02 3ème port MIDI
  - 50 : /dev/midi03 4ème port MIDI
- 14 bloc : Disque dur via BIOS
  - 0 : /dev/dos_hda 1er disque dur BIOS
  - 64 : /dev/dos_hdb 2nd disque dur BIOS
  - 128 : /dev/dos_hdc 3ème disque dur BIOS
  - 192 : /dev/dos_hdd 4ème disque dur BIOS
- 15 car : Joystick
  - 0 : /dev/js0 1er joystick
  - 1 : /dev/js1 2ème joystick
  - ...
  - 128 : /dev/djs0 1er joystick numérique
  - 129 : /dev/djs1 2nd joystick numérique
- bloc 15 : Sony CDU-31A/CDU-33A CD-ROM
  - 0 : /dev/sonycd Sony CDU-31a CD-ROM
- 16 car : Réservé pour les scanners non SCSI :
  - 0 : /dev/gs4500 : Genius 4500
- 16 bloc : CD-ROM GoldStar
  - 0 : /dev/gscd : CD-ROM GoldStar
- 17 car : Carte série *Chase*
  - 0 : /dev/ttyH0 1er port

  - 1 : /dev/ttyH1 2nd port
  - ...
- 17 bloc : CD-ROM de sauvegardes optiques
  - 0 : /dev/optcd CD-ROM de sauvegarde optiques
- 18 car : Carte série Chase
  - 0 : /dev/cuh0 Appel sortant correspondant à ttyH0
  - 1 : /dev/cuh1 Appel sortant correspondant à ttyH1
  - ...
- 18 bloc : CD-ROM Sanyo
  - 0 : /dev/sjcd CD-ROM Sanyo CD-ROM
- 19 car : Carte série Cyclades
  - 0 : /dev/ttyC0 1er port
  - ...
  - 31 : /dev/ttyC31 32ème port
- 19 bloc : Disque compressé « Double »
  - 0 : /dev/double0 1er disque compressé
  - ...
  - 7 : /dev/double7 8ième disque
  - 128 : /dev/cdouble0 miroir du 1er disque
  - ...
  - 135 : /dev/cdouble7 miroir du 8ième disque
- 20 car : carte série Cyclades
  - 0 : /dev/cub0 périphérique appelant correspondant à ttyC0
  - ...
  - 31 : /dev/cub31 périphérique appelant correspondant à ttyC31
- 20 bloc : CD-ROM Hitachi (en développement)
  - 0 : /dev/hitcd CD-ROM Hitachi
- 21 car : SCSI générique
  - 0 : /dev/sg0 1er périphérique
  - 1 : /dev/sg1 2nd périphérique
  - ...

- 22 car : Carte série Digiboard
  - 0 : /dev/ttyD0 1er port Digiboard
  - 1 : /dev/ttyD1 2ème port Digiboard
  - ...
- 22 bloc : Seconde interface IDE disque dur / CD-ROM
  - 0 : /dev/hdc maître (ou CD-ROM)
  - 64 : /dev/hdd serveur (ou CD-ROM)
- 23 car : Carte série Digiboard
  - 0 : /dev/cud0 périphérique appelant correspondant à ttyD0
  - 1 : /dev/cud1 périphérique appelant correspondant à ttyD1
  - ...
- 23 bloc : CD-ROM propriétaire Mitsumi
  - 0 : /dev/mcd CD-ROM Mitsumi
- 24 car : Carte série Stallion
  - 0 : /dev/ttyE0 Port 0 carte 0
  - 1 : /dev/ttyE1 Port 1 carte 0
  - ...
  - 64 : /dev/ttyE64 Port 0 carte 1
  - 65 : /dev/ttyE65 Port 1 carte 1
  - ...
  - 128 : /dev/ttyE128 Port 0 carte 2
  - 129 : /dev/ttyE129 Port 1 carte 2
  - ...
  - 192 : /dev/ttyE192 Port 0 carte 3
  - 193 : /dev/ttyE193 Port 1 carte 3
  - ...
- 24 bloc : CD-ROM Sony CDU-535
  - 0 : /dev/cdu535 CD-ROM Sony CDU-535
- 25 car : Stallion serial card - alternate devices
  - 0 : /dev/cue0 Périphérique appelant correspondant à ttyE0
  - 1 : /dev/cue1 Périphérique appelant correspondant à ttyE1
  - ...

    - 64 : /dev/cue64 Périphérique appelant correspondant à ttyE64
    - 65 : /dev/cue65 Périphérique appelant correspondant à ttyE65
    - ...
    - 128 : /dev/cue128 Périphérique appelant correspondant à ttyE128
    - 129 : /dev/cue129 Périphérique appelant correspondant à ttyE129
    - ...
    - 192 : /dev/cue192 Périphérique appelant correspondant à ttyE192
    - 193 : /dev/cue193 Périphérique appelant correspondant à ttyE193
    - ...
- 25 bloc : 1er CDROM Matsushita (Panasonic/SoundBlaster)
    - 0 : /dev/sbpcd0 Contrôleur 0 CD-ROM Panasonic unité 0
    - 1 : /dev/sbpcd1 Contrôleur 0 CD-ROM Panasonic unité 1
    - 2 : /dev/sbpcd2 Contrôleur 0 CD-ROM Panasonic unité 2
    - 3 : /dev/sbpcd3 Contrôleur 0 CD-ROM Panasonic unité 3
- 26 car : *Quanta WinVision Frame grabbers*
    - 0 : /dev/wvisfgrab Quanta WinVision
- 26 bloc : 2nd CDROM Matsushita (Panasonic/SoundBlaster)
    - 0 : /dev/sbpcd4 Contrôleur 1 CD-ROM Panasonic unité 0
    - 1 : /dev/sbpcd5 Contrôleur 1 CD-ROM Panasonic unité 1
    - 2 : /dev/sbpcd6 Contrôleur 1 CD-ROM Panasonic unité 2
    - 3 : /dev/sbpcd7 Contrôleur 1 CD-ROM Panasonic unité 3
- 27 car : lecteur de cartouches QIC-117
    - 0 : /dev/rft0 Unité 0, avec rembobinage
    - 1 : /dev/rft1 Unité 1, avec rembobinage
    - 2 : /dev/rft2 Unité 2, avec rembobinage
    - 3 : /dev/rft3 Unité 3, avec rembobinage
    - 4 : /dev/nrft0 Unité 0, sans rembobinage
    - 5 : /dev/nrft1 Unité 1, sans rembobinage
    - 6 : /dev/nrft2 Unité 2, sans rembobinage
    - 7 : /dev/nrft3 Unité 3, sans rembobinage
- 27 bloc : 3ème CDROM Matsushita (Panasonic/SoundBlaster)
    - 0 : /dev/sbpcd8 Contrôleur 2 CD-ROM Panasonic unité 0
    - 1 : /dev/sbpcd9 Contrôleur 2 CD-ROM Panasonic unité 1

  - 2 : /dev/sbpcd10 Contrôleur 2 CD-ROM Panasonic unité 2
  - 3 : /dev/sbpcd11 Contrôleur 2 CD-ROM Panasonic unité 3
- 28 car : Carte série Stallion (sauf Atari)
  - 0 : /dev/staliomem0 1ère carte mémoire I/O
  - 1 : /dev/staliomem1 2nd carte mémoire I/O
  - 2 : /dev/staliomem2 3ième carte mémoire I/O
  - 3 : /dev/staliomem3 4ième carte mémoire I/O
- 28 car : Imprimante laser SLM ACSI (68k/Atari)
  - 0 : /dev/slm0 1ère imprimante laser SLM
  - 1 : /dev/slm1 2nde imprimante laser SLM
  - ...
- 28 bloc : 4ème CDROM Matsushita (Panasonic/SoundBlaster) (sauf pour 68k/Atari)
  - 0 : /dev/sbpcd12 contrôleur 3 CD-ROM Panasonic unité 0
  - 1 : /dev/sbpcd13 contrôleur 3 CD-ROM Panasonic unité 1
  - 2 : /dev/sbpcd14 contrôleur 3 CD-ROM Panasonic unité 2
  - 3 : /dev/sbpcd15 contrôleur 3 CD-ROM Panasonic unité 3
- 28 bloc : disk ACSI (68k/Atari)
  - 0 : /dev/ada 1er disque ACSI
  - 16 : /dev/adb 2nd disque ACSI
  - 32 : /dev/adc 1er disque ACSI
  - ...
  - 240 : /dev/adp 16ième disque ACSI
- 29 car : Buffer Universal (68k et Sparc uniquement)
  - 0 : /dev/fb0 1er buffer
  - 1 : /dev/fb0autodetect
  - 24 : /dev/fb0user0
  - ...
  - 31 : /dev/fb0user7
  - 32 : /dev/fd1 2nd buffer
  - 33 : /dev/fb1autodetect
  - 56 : /dev/fb1user0
  - ...
  - 63 : /dev/fb1user7

- 29 bloc : CD-ROM Aztech/Orchid/Okano/Wearnes
  - 0 : /dev/aztcd CD-ROM Aztech
- 30 car : Périphériques compatibles iBCS-2
  - 0 : /dev/socksys Accès Socket
  - 1 : /dev/spx Interface locale X SVR3
  - 2 : /dev/inet/arp Accès réseau
  - 2 : /dev/inet/icmp Accès réseau
  - 2 : /dev/inet/ip Accès réseau
  - 2 : /dev/inet/udp Accès réseau
  - 2 : /dev/inet/tcp Accès réseau
- 30 bloc : CD-ROM Philips LMS CM-205 CD-ROM
  - 0 : /dev/cm205cd CD-ROM Philips LMS CM-205
- 31 car : MPU-401 MIDI
  - 0 : /dev/mpu401data MPU-401 port données
  - 1 : /dev/mpu401stat MPU-401 port état
- 31 bloc : Carte mémoire ROM/flash
  - 0 : /dev/rom0 1ère carte ROM (rw)
  - ...
  - 7 : /dev/rom7 8ème carte ROM (rw)
  - 8 : /dev/rrom0 1ère carte ROM (ro)
  - ...
  - 15 : /dev/rrom7 8ème carte ROM (ro)
  - 16 : /dev/flash0 1ère carte mémoire flash (rw)
  - ...
  - 23 : /dev/flash7 8ème carte mémoire flash (rw)
  - 24 : /dev/rflash0 1ère carte mémoire flash (ro)
  - ...
  - 31 : /dev/rflash7 8ème carte mémoire flash (ro)
- 32 car : Carte série Specialix
  - 0 : /dev/ttyX0 1er port
  - 1 : /dev/ttyX1 2nd port
  - ...

- 32 bloc : CD-ROM Philips LMS CM-206
  - 0 : /dev/cm206cd CD-ROM Philips LMS CM-206
- 33 car : Carte série Specialix
  - 0 : /dev/cux0 Périphérique appelant correspondant à ttyX0
  - 1 : /dev/cux1 Périphérique appelant correspondant à ttyX1
  - ...
- 33 bloc : 3ième carte controleur IDE
  - 0 : /dev/hde Maître
  - 64 : /dev/hdf Esclave
- 34 char : Z8530 HDLC
  - 0 : /dev/scc0 : 1er Z8530 - 1er port
  - 1 : /dev/scc1 : 1er Z8530 - 2nd port
  - 2 : /dev/scc2 : 2nd Z8530 - 1er port
  - 3 : /dev/scc3 : 2nd Z8530 - 2nd port
  - ...
- 34 bloc : 4ème interface IDE disque dur/CD-ROM
  - 0 : /dev/hdg : Maître
  - 64 : /dev/hdh : Esclave
- 35 car : tclmidi MIDI
  - 0 : /dev/midi0 1er port MIDI, timer via noyau
  - 1 : /dev/midi1 2nd port MIDI, timer via noyau
  - 2 : /dev/midi2 3ème port MIDI, timer via noyau
  - 3 : /dev/midi3 4ème port MIDI, timer via noyau
  - 64 : /dev/rmidi0 1er port MIDI, sans timer
  - 65 : /dev/rmidi1 2nd port MIDI, sans timer
  - 66 : /dev/rmidi2 3ème port MIDI, sans timer
  - 67 : /dev/rmidi3 4ème port MIDI, sans timer
  - 128 : /dev/smpte0 1er port MIDI, timer via SMPTE
  - 129 : /dev/smpte1 2ème port MIDI, timer via SMPTE
  - 130 : /dev/smpte2 3ème port MIDI, timer via SMPTE
  - 131 : /dev/smpte3 4ème port MIDI, timer via SMPTE
- 35 bloc : disque Modular RAM - fournit pour les vieux noyaux

- 36 car : Netlink
  - 0 : /dev/route : routage
  - 1 : /dev/skip : cache de sécurité
- 36 bloc : MCA ESDI disque dur
  - 0 : /dev/eda 1er
  - 64 : /dev/edb 2nd
  - ...
- 37 car : lecteurs de bande IDE
  - 0 : /dev/ht0 : 1er périphérique
  - 128 : /dev/nht0 : 1er périphérique, sans rembobinage
- 37 bloc : zorro II ramdisk
  - 0 : /dev/z2ram : Zorro II
- 38 car : cartes Myricom PCI Myrinet
  - 0 : /dev/mlanai0 : 1ère carte
  - 1 : /dev/mlanai2 : 2nde carte
  - ...
- 38 bloc : réservé pour Linux/AP+
- 39 car : ML-16P - carte expérimentale
  - 0 : /dev/ml16pa-a0 : 1ère carte, 1er canal analogique
  - 1 : /dev/ml16pa-a0 : 1ère carte, 2nd canal analogique
  - ...
  - 15 : /dev/ml16pa-a15 : 1ère carte, 16ème canal analogique
  - 16 : /dev/ml16pa-d : 1ère carte, lignes digitales
  - 17 : /dev/ml16pa-c0 : 1ère carte, 1er compteur/timer
  - 18 : /dev/ml16pa-c1 : 1ère carte, 2nd compteur/timer
  - 19 : /dev/ml16pa-c2 : 1ère carte, 3eme compteur/timer
  - 32 : /dev/ml16pb-a0 : 2ème carte, 1er canal analogique
  - 33 : /dev/ml16pb-a1 : 2ème carte, 2nd canal analogique
  - ...
  - 47 : /dev/ml16pb-a15 : 2ème carte, 16ème canal analogique
  - 48 : /dev/ml16pb-d : 2ème carte, lignes digitales
  - 49 : /dev/ml16pb-c0 : 2ème carte, 1er compteur/timer

    - 50 : /dev/ml16pb-c1 : 2ème carte, 2nd compteur/timer
    - 51 : /dev/ml16pb-c2 : 2ème carte, 3eme compteur/timer
    - ...
- 39 bloc : réservé pour Linux/AP+
- 40 char : Matrox Meteor
    - 0 : /dev/mmetfgrab Matrox Meteor
- 40 bloc : disque extractible Syquest EZ135 sur port parallèle
    - 0 : /dev/eza 1er disque sur port parallèle EZ135
- 41 : *Yet Another Micro Monitor*
    - 0 : /dev/yamm *Yet Another Micro Monitor*
- 42 : utilisé pour des démonstrations ou des exemples
- 43 car : modem virtuel isdn4linux
    - 0 : /dev/ttyI0 1er modem virtuel
    - ...
    - 63 : /dev/ttyI63 64ème modem virtuel
- 44 car : autres périphériques isdn4linux
    - 0 : /dev/cui0 1er périphérique d'appel
    - ...
    - 63 : /dev/cui63 64ème périphérique d'appel
- 45 car : périphérique BRI isdn4linux
    - 0 : /dev/isdn0 1er périphérique virtuel
    - ...
    - 63 : /dev/isdn63 64ième périphérique virtuel
    - 64 : /dev/isdnctrl0 1er canal de contrôle
    - ...
    - 127 : /dev/isdnctrl63 64ième canal de contrôle
    - 128 : /dev/ippp0 : 1er périphérique SyncPPP
    - ...
    - 191 : /dev/ippp63 : 64ième périphérique SyncPPP
    - 255 : /dev/isdninfo interface
- 46 car : carte série Rocketport
    - 0 : /dev/ttyR0 1er port

    - 1 : /dev/ttyR1 2nd port
    - ...
- 47 car : carte série Rocketport - autre périphériques
    - 0 : /dev/cur0 1er port d'appel
    - 1 : /dev/cur1 2nd
    - ...
- 48 car : carte série SDL RISCom
    - 0 : /dev/ttyL0 1ère carte RISCom
    - 1 : /dev/ttyL1 2ième carte RISCom
    - ...
- 49 car : carte série SDL RISCom - autre périphérique
    - 0 : /dev/cul0 1er périphérique d'appel
    - 1 : /dev/cul1 2ième périphérique d'appel
    - ...
- 50 car : réservé pour GLINT
- 51 car : modem radio Baycom
    - 0 : /dev/bc0 1er modem Baycom
    - 1 : /dev/bc1 2nd modem Baycom
    - ...
- 52 car : cartes ISDN DataComm/BRI Spellcaster
    - 0 : /dev/dcbri0 1ère carte DataComm
    - 1 : /dev/dcbri1 2nde carte DataComm
    - 2 : /dev/dcbri2 3ème carte DataComm
    - 3 : /dev/dcbri3 4ème carte DataComm
- 53 car interface BDM pour débogage de micro-contrôleurs MC683xx
    - 0 : /dev/pd_bdm0 interface PD BDM sur lp0
    - 1 : /dev/pd_bdm1 interface PD BDM sur lp1
    - 2 : /dev/pd_bdm2 interface PD BDM sur lp2
    - 4 : /dev/icd_bdm0 interface ICD BDM sur lp0
    - 5 : /dev/icd_bdm1 interface ICD BDM sur lp1
    - 6 : /dev/icd_bdm2 interface ICD BDM sur lp2

- 54 car Cartes séries Holter Electrocardiognosis Holter
  - 0 : /dev/holter0 1er port Holter
  - 1 : /dev/holter1 2nd port Holter
  - 2 : /dev/holter2 3ème port Holter
- 55-59 : non-utilisés
- 60-63 : usage local ou d'expérimentation
- 64-119 : non-utilisés
- 120-127 : usage local ou d'expérimentation
- 128-239 : non-utilisés
- 240-254 : usage local ou d'expérimentation
- 255 : Réservé

12 Communiquer

12.1 UUCP et modem

UUCP (Unix to Unix CoPy) te permet d'envoyer et de recevoir du mail, et de même pour les news.

Je te conseille de faire un lien entre `/dev/modem` et `/dev/cua1` (COM2), si c'est ton périphérique. Les répertoires utilisés par UUCP sont `/var/spool/uucp, /var/spool/uucpublic, /usr/lib/uucp` et `/var/lib/uucp`.

Bon, je suppose que tu as installé le package UUCP, elm et sendmail. Pour les news, je te conseille INN, comme serveur de news, et tin ou trn, comme lecteur de news.

Les fichiers UUCP décrits ici sont au format HDB (ou, je sais que Taylor est achtement plus simple... mais j'ai pas eu le temps de refaire cette partie qui date... !).

Bon, UUCP a besoin de plusieurs fichiers de configuration qui se trouvent dans `/var/lib/uucp/hdb_config`.

- Devices : Configure la ligne modem que tu vas utiliser : `ACU modem - 9600 hayes_usr`
- Dialers : Là, c'est le chat script pour lancer le modem. Il est possible que tu doives le modifier.

```
hayes_usr =,-,  "" \dATE0V1X1Q0\r\c OK\r ATDT\T\r\c CONNECT
```

- Permissions : Ce fichier contient tous les renseignements de permissions pour la connexion.

```
MACHINE=frmug LOGNAME=nuucp \
READ=/var/spool/uucppublic \
WRITE=/var/spool/uucppublic \
SENDFILES=yes REQUEST=yes
```

– Systems : Dans ce fichier, tu mets chacune des machines sur lesquelles tu veux te connecter.

```
remcomp any     ACU     9600    0042470611      ogin: nuucp
frmug   any     ACU     9600    0053176410       ogin: nuucp
frmug1   any     ACU     9600   0053176411       ogin: nuucp
```

Maintenant, ton système est configuré. Il n'y a plus qu'à tester. Lance ceci :

```
uucp remcomp\!\~/catalogue.Z ~/
```

Ta machine va se connecter sur remcomp pour récupérer le fichier `catalogue.Z`. Ce fichier sera déposé dans le répertoire `/var/spool/uucppublic`.

12.2 Appels entrants

Bon. Au point où l'on en est, tu peux appeler. Maintenant, on va configurer ta machine pour que l'on puisse t'appeler. Tout se joue dans le fichier `/etc/inittab`.

Tu dois décommenter une ligne par exemple :

```
s1:45:respawn:/sbin/agetty -h -t 60 9600 modem
|   |    |                  |    |    |     |--- Port a utiliser
|   |    |                  |    |    |--------- Vitesse du modem
|   |    |                  |    |--- Temps au bout du quel on racroche
|   |    |                  |-------- Active le controle de flux rts/cts
|   |    |-------------- A relancer lorsqu'il se termine
|   |------------------ Runlevels.
|----------------------- Identificateur
```

Donc, au prochain boot, un système de login est prêt sur ton port `/dev/modem` (ton port série sur lequel est connecté ton modem). Le dernier problème est de le configurer. En effet, ton modem doit être positionné pour la vitesse. En fait, ca dépend du modem. Si jamais tu peux positionner des switch, rend ton modem muet (`ATQ2`) et surtout, mets le en auto-reponse (`ATS0=1`).

Si tu n'as pas de switches, fait un `AT&W` pour sauvegarder les paramètres.

Pour vérifier la config utiliser `usr`lib/uucp/uuchk/.

Si tu veux d'autres renseignements, je te conseille de lire le HOWTO UUCP.

12.3 PPP

L'objet de ce paragraphe est l'installation d'une connexion PPP.

12.3.1 Introduction

PPP (Point to Point Protocol) permet d'établir une connexion IP sur une voie série de manière plus efficace que SLIP (Serial Line Internet Protocol). Une fois la connexion établie, tous les protocoles supérieurs à IP (FTP, telnet, NFS, http, X...) sont disponibles.

Pour établir une connexion PPP, il faut:

- un serveur supportant le protocole PPP
- un modem *au moins* V32 (9600 bauds)
- un soft de connexion à PPP
- un peu de patience

PPP est supporté par un grand nombre de systèmes (même MS-Windows) et on peut normalement se connecter d'un système à l'autre. Cependant, cette documentation ne traite que le cas de **Linux** !

12.3.2 Accès à un serveur PPP

Pour accéder à un serveur PPP installé, ton noyau doit être compilé avec le support TCP/IP. Pour cela, lors du make config il faut répondre comme suit :

```
Networking support (CONFIG_NET) [y] y
TCP/IP networking (CONFIG_INET) [y] y
IP forwarding/gatewaying (CONFIG_IP_FORWARD) [y] n
Network device support? (CONFIG_NETDEVICES) [y] y
PPP (point-to-point) support (CONFIG_PPP) [y] y
```

Tu dois en plus récupérer les sources des programmes d'accès à PPP (dans l'archive `ppp-2.1.2a.tar.gz`, voire b, c, ou d) de manière à installer les programmes suivants :

- `pppd` le programme d'établissement du protocole
- `chat` qui permet de dérouler le chat-script d'appel au site PPP

Tu peux par exemple installer `pppd` et `chat` dans le répertoire `/usr/sbin`.

Ensuite, tu dois configurer tes shell-scripts de boot de manière à configurer la couche TCP/IP (par exemple dans un fichier `/etc/rc.d/rc.net` qui sera lancé par `/etc/rc.d/rc.local`). En voici un exemple :

```
echo -n "/etc/rc.net: "
INETD=/sbin/inetd
PORTMAP=/sbin/portmap

# loopback
```

```
/sbin/ifconfig lo 127.0.0.1 up netmask 255.255.255.0
/sbin/route add 127.0.0.1 lo

# demarrage des demons

if [ -x $PORTMAP ]; then
        echo -n ", `basename $PORTMAP`"
        $PORTMAP
fi

if [ -x $INETD ]; then
        echo -n ", `basename $INETD`"
        $INETD
fi

echo ""
```

Il faut ensuite ajoute l'adresse IP de ta machine dans le fichier `/etc/hosts`:

```
     127.0.0.1          loopback localhost      # useful aliases
     199.103.124.170    pcpf pcpf.lectra.fr     # Mon beau PC en PPP
```

L'adresse IP de la machine correspond à celle affectée par l'administrateur du serveur PPP pour la connexion associée à un numéro de téléphone donné. Cela signifie que le copain qui se connecte 10 minutes après utilisera la même adresse IP que toi.

Enfin, tu dois configurer ton domaine dans le fichier `/etc/resolv.conf`:

```
domain lectra.fr
nameserver 192.1.2.1
```

La deuxième ligne est utilisée si tu veux utiliser un serveur de noms, bécane qui te renvoie l'adresse IP correspondant au nom de machine fourni. Si tu n'as pas de serveur de noms, tu ne pourras utiliser que des adresses numériques (199.103.124.x) ou bien tu devras mettre dans ton `/etc/hosts` toutes les machines auxquelles tu veux accéder avec leurs adresses IP...

Tu peux également mettre ton nom de domaine dans le fichier `/etc/ppp/options` qui doit exister même vide (si tu ne veux pas que `pppd` demande `/etc/ppp/options`, il faut le compiler avec l'option `-DREQ_SYSOPTIONS=0`).

Tu peux maintenant fébrilement tester ta connexion en utilisant un shell-script du style (attention : tout cela constitue une seule ligne) :

```
/usr/sbin/pppd connect '/usr/sbin/chat -v ABORT ERROR ABORT "NO CARRIER" \
ABORT BUSY "" ATB0 OK ATDTxxxxxxxx CONNECT "" ogin: ppp \
word: ppp0' /dev/modem 9600 -detach debug crtscts modem \
defaultroute 199.103.124.170:
```

La commande `chat` permet d'effectuer la connexion en appelant le numéro de téléphone de ton serveur. Attention de ne pas oublier les options de la deuxième ligne en particulier `modem` et `defaultroute`.

La spécification de la vitesse (9600) ainsi que le chat-script à utiliser (ABORT ERROR ABORT "NO CARRIER" ABORT BUSY "" ATB0 OK ATDTxxxxxxxx CONNECT) dépendent bien entendu de ton modem et de la configuration du serveur (voir également le login et le password).

L'adresse IP doit être celle déclarée en tant qu'adresse de ta machine en PPP dans `/etc/hosts` (tu peux également mettre le nom de ta machine en PPP).

Les informations de debug doivent sortir sur la console (si tu as configuré le syslog pour celà).

A partir de là, tu es connecté sur le réseau distant et tu peux faire par exemple:

```
$ ping 199.103.124.50
PING 199.103.124.50 (199.103.124.50): 56 data bytes
64 bytes from 199.103.124.50: icmp_seq=0 ttl=255 time=268 ms
64 bytes from 199.103.124.50: icmp_seq=1 ttl=255 time=247 ms
64 bytes from 199.103.124.50: icmp_seq=2 ttl=255 time=266 ms
```

pour tester ta connexion. Si le ping marche et que ton réseau est connecté à l'Internet, alors tu as l'Internet chez toi !

Si ça ne marche pas, tu peux tester en tapant `/sbin/ifconfig` et le résultat doit ressembler à ça :

```
lo        Link encap Local Loopback
          inet addr 127.0.0.1  Bcast 127.255.255.255  Mask 255.0.0.0
          UP LOOPBACK RUNNING  MTU 2000  Metric 1
          RX packets 0 errors 0 dropped 0 overrun 0
          TX packets 0 errors 0 dropped 0 overrun 0

ppp0      Link encap Serial Line IP
          inet addr 199.103.124.170  P-t-P 199.103.124.50  Mask 255.255.255.0
          UP POINTOPOINT RUNNING  MTU 1500  Metric 1
          RX packets 33 errors 0 dropped 0 overrun 0
          TX packets 42 errors 0 dropped 0 overrun 0
```

Surtout, tu dois avoir une ligne commençant par ppp0 sinon, no good. Tu peux également tester ton routage par `netstat -nr` qui doit donner un truc du style :

```
Kernel routing table
Destination     Gateway         Genmask         Flags Metric Ref Use    Iface
199.103.124.50  0.0.0.0         255.255.255.255 UH    0      0        6 ppp0
127.0.0.0       0.0.0.0         255.0.0.0       U     0      0        0 lo
0.0.0.0         199.103.124.50  0.0.0.0         UG    0      0     6298 ppp0
```

Avec une ligne 0.0.0.0 contenant l'adresse IP de ton serveur PPP. Si tu n'as pas cette ligne, il se peut que tu ais oublié l'option `defaultroute`.

12.3.3 Configuration d'un serveur PPP

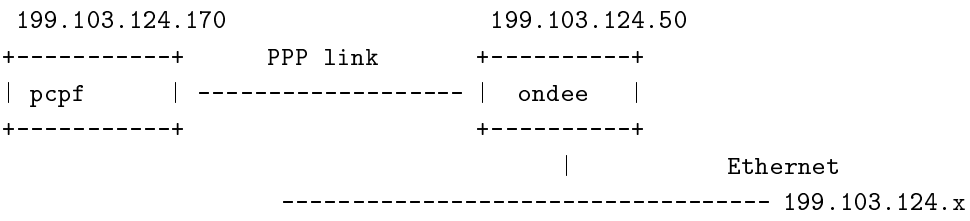
Pour cela, tu dois installer les mêmes logiciels que dans le chapitre précédent (configurer le noyau, installer `pppd`). En plus de ça, il faut que tu crées un login spécial pour les connexions PPP dans `/etc/passwd`, par exemple :

```
ppp::61:60:Connexion PPP:/:/usr/bin/ppp_login
```

Le script `/usr/bin/ppp_login` étant :

```
#!/bin/sh
# Connexion PPP --> adresse IP = 199.103.124.170
exec /usr/sbin/pppd -d -detach -ip modem proxyarp :199.103.124.170
```

L'option `proxyarp` permet à la machine connectée d'accèder à tout le réseau :



12.3.4 Bibliographie

Pour plus d'informations, voir le fichier README.**Linux** de la distribution `ppp-2.1.2a.tar.gz`.

Remarque : Ce fichier est dans la distribution standard de la Slackware.

12.4 Un réseau derrière une seule adresse

Une question commune est : « j'ai un accès Internet mais une seule adresse IP et je voudrais pourtant pouvoir connecter plusieurs machines ».

Le `NET2-HOWTO` répond à cette question mais il oublie une possibilité. Cette note a pour but d'expliquer les quatre façons de résoudre le problème ci-dessus.

- Obtenir plusieurs adresses. L'idéal serait sans doute d'avoir un vrai sous-réseau sur le site distant, qui pourrait être routé « normalement ». C'est la méthode la plus normale et la seule qui donnera à toutes les machines du réseau distant une vraie connectivité Internet.

  Mais si ton fournisseur d'accès est un opérateur commercial, il te faudra probablement payer *beaucoup* plus cher pour un tel accès (qui lui coûte effectivement plus). Si c'est une université ou association, elle n'aura peut-être pas envie de configurer un cas spécial pour toi. Il va donc falloir se contenter d'un des trois bricolages suivants, qui commencent tous par allouer aux ordinateurs supplémentaires une adresse privée, prise dans le RFC 1918 (qui a remplacé le fameux 1597).

- « IP masquerading ». C'est une traduction d'adresses dynamique fournie par le noyau de la machine Linux passerelle. Une idée simple et géniale. C'est peut-être la méthode la plus générale, car elle marche pour un grand nombre de services réseaux. Note quand même tout de suite que tous ne marcheront pas, notamment `ping` et beaucoup de services UDP. « IP masquerading » nécessite un noyau récent (2.* ou certains 1.3.* ou un patch du 1.2.*) et il est documenté sur `http://www.indyramp.com/masq/` ou `http://hwy401.com/achau/ipmasq/` (pas de encore de HOWTO).

- « Socks » qui est un relais générique, fonctionnant au niveau application (donc pas de modifications dans le noyau contrairement au « IP masquerading ». Il nécessite des programmes spéciaux (trouvés couramment sur les machines Unix) du côté client. Les documentations se trouvent en général dans les livres ou serveurs parlant de sécurité car on l'utilise en général dans ce contexte. C'est ainsi que le `Firewall-HOWTO` décrit cette solution (il est traduit en français, voir `http://www.freenix.fr/linux/HOWTO/`.)

- Relais applicatif : c'est ainsi qu'on nomme les programmes tournant au niveau application qui relaient les requêtes des clients, installés sur le réseau sans adresses IP légales. Outre le relayage qui permet à ces machines de sortir, ils fournissent typiquement une valeur ajoutée, par exemple en gardant en mémoire locale des informations fréquemment accédées. C'est à mon avis la solution la plus simple à mettre en oeuvre, elle marche sur tous les noyaux et même sur d'autres Unix que **Linux**.

  Certaines applications n'ont aucun problème à être ainsi relayées car elles ont toujours fonctionné comme ça : le courrier, les News, la synchronisation d'horloges, le service de noms, etc. D'autres se sont converties au relayage a posteriori et y ont gagne beaucoup (pour le Web, c'est le relayage qui permet de mettre en oeuvre les indispensables caches comme celui de Renater `http://cache.cnrs.fr/`). Ainsi, le gros avantage du relayage est de pouvoir apporter une valeur ajoutée : réécriture d'adresses et gestion de file d'attente centralisées pour le courrier, caches pour le Web. Le meilleur relais + cache Web à l'heure actuelle est Squid `http://www.nlanr.net/Squid/`.

  Le relayage a deux défauts : toutes les applications ne disposent pas d'un relais (on peut utiliser Socks pour combler les trous) et il faut installer un nouveau relais à chaque application. Le relayage est donc recommandé si on se contente des services de base : courrier, Web, News, etc.

12.5 Sendmail

La configuration du sendmail est toujours quelque chose de particulièrement pénible et difficile. Toutefois, certains outils facilitent grandement les choses. La Slackware est d'origine fournie avec certains fichiers, qui peuvent être une solution. Toutefois, il est plus sain de reinstaller le fichier de configuration en se basant sur les docuements qui suivent :

- Programmes :
  - sendmail : `ftp://ftp.ibp.fr/pub/unix/mail/sendmail/ucb` ;
  - kit Jussieu : `ftp://ftp.jussieu.fr/jussieu/sendmail/kit/` ;
  - UUCP : n'importe quel site miroir GNU, par exemple `ftp.pasteur.fr/pub/Gnu` ;
  - exmh : `http://www.smli.com/~bwelch/exmh/`.

- Documents :
  - excellente documentation dans le kit Jussieu, en français ;
  - FAQ FreeBSD notamment « How do I use sendmail for mail delivery with UUCP? » ;
  - `http://www.freebsd.org/FAQ/` ;
  - pourquoi utiliser UUCP : `http://web.pasteur.fr/other/computer/Linux/my-BBS/uucp<.txt` ;
  - la doc de Taylor-UUCP ;
  - Linux Network Administrator's Guide (voir plus bas).

12.6 Configuration du mail lors d'une connexion avec un provider

Les provideurs reçoivent les messages et les rendent accessibles via un serveur POP (en général mail.fournisseur.fr).
Il faut donc récupérer les messages reçus par ce serveur vers la machine linux dans /var/spool/mail/utilisateur.
Pour cela, tu peux utiliser le programme `gwpop` de Stéphane Bortzmeyer. On le trouve dans `ftp://ftp.pasteur.fr/pub/N`

Dans ce script, définir :

```
$mailhost = "mail.fournisseur.fr";
$deliver= "deliver NomUserLocal ";
```

Puis, une fois connecté, lancer via un script (histoire de ne pas avoir à tout retaper à chaque fois :

```
gwpop -p MotDePasseServeurPop UserServeurPop
```

Une fois que le courier est reçu, la question est de savoir comment en envoyer. Il faut recompiler `sendmail` sans DNS. Pour cela, définir `NAMED_BIND` à 0 dans `src/conf.h`. Les versions récentes de sendmail ont par défaut un comportement qui peut être génant. Les messages accentués sont automatiquement convertis en *Quoted Printable* si le serveur auquel il se connecte n'annonce pas qu'il supporte le protocole ESMTP 8bit. Si on trouve cette initiative génante, il faut définir MIME8TO7 à 0 dans `src/conf.h`.

Voici le fichier de règles qu'il faut indiquer via le kit sendmail Jussieu :

```
#!/bin/sh
#
# Regles pour le domaine lolonet
#
Domaine="MonNomdeDomaineAvecUnPoint"
AdressesInternes=RIEN
V8="o"
Aliases="/etc/aliases"
SendmailSt="/etc/sendmail.st"
SendmailHf="/etc/sendmail.hf"
Mqueue="/var/spool/mqueue"
```

et le fichier de config :

```
#!/bin/sh
Host="NomDeMaMachine"
AdressesLocales=TOUT_DOMAINE
RelaisExterieur="smtp.[mail.fournisseur.fr]"
ReecritureadressesLocales="fournisseur.fr"
RevAliases="hash -N /etc/revaliases"
MailerLocal='/usr/bin/deliver DFMPlmnrs deliver $u'
SansDNS=1
```

Il suffit ensuite de modifier le configurateur en décommentant la ligne suivante:

```
CPP="$CPP -traditional"
```

puis générer le fichier de config :

```
./configurateur regles.lolonet Machine.config >sendmail.cf.Machine
```

Si on a désactivé MIME8TO7 dans sendmail, commenter la ligne du sendmail.cf générée :

```
# O EightBitMode=mime
```

et copier le fichier dans `/etc`

Il est nécessaire de définir l'adresse IP du serveur SMTP dans `/etc/hosts`. Enfin, il faut réécrire l'adresse locale pour que les messages envoyés aient un champ `From` correspondant à l'adresse fourni par le fournisseur. Il suffit d'ajouter dans `/etc/revaliases` une ligne du style:

```
NomLocal:       NomFournisseur@fournisseur.fr
```

et ensuite regénérer la base avec :

```
/usr/sbin/sendmail -bi -oA/etc/revaliases
```

Voila ! Les mails sont stoqués dans `/var/spool/mqueue`. Lorsque tu es connecté par PPP et que tu souhaites envoyer tous tes mails, il suffit de faire

```
/usr/sbin/sendmail -q
```

13 ELF

13.1 Introduction

Depuis ses toutes premières versions, Linux supporte les binaires au format a.out. Récemment, le support pour les binaires au format ELF a été intégré dans le noyau et une distribution des bibliothèques partagées

au format ELF a été faite. Les avantages du format ELF par rapport au format a.out sont nombreux :

- compatibilité avec les systèmes Unix basés sur SVR4,
- plus grande facilité de création des bibliothèques partagées,
- édition de liens dynamique lors de l'exécution des programmes,
- support des fonctions `dlopen()`, `dlsym()` et `dlclose()`

Les dernières versions des distributions sont basées sur le format ELF. Ce qui suit n'a donc de sens que si tu ne possèdes pas une version récente et si tu veux mettre à jour toi même au fur à mesure. Attention : cela peut être long et périlleux.

13.2 Installation

La première version publique des utilitaires et des bibliothèques ELF a été diffusée le dimanche 21 mai 1995. La version à jour des bibliothèques ELF et des outils associés est disponible sur les sites FTP `tsx-11.mit.edu` et `sunsite.unc.edu` ainsi que sur leur miroir français `ftp.ibp.fr`, dans le répertoire `/pub/linux/GCC`.

Il est possible d'utiliser cette distribution pour mettre à jour ses bibliothèques. Toutefois, il est fortement recommandé d'utiliser une distribution binaire basée sur ELF (comme Slackware 3.1 ou Red Hat 4.0).

Enfin bon, si tu souhaites quand même faire la mise à jour toi-même, voici la liste des fichiers à récupérer :

- `binutils-2.7.0.3.bin.tar.gz` : les utilitaires (as, ld, ...),
- `binutils-2.7.0.3.tar.gz` : les sources des utilitaires (as, ld, ...),
- `gcc-2.7.2.1.bin.tar.gz` : le compilateur GCC 2.7.2.1,
- `ld.so-1.8.5.tar.gz` : l'éditeur de liens dynamique,
- `libc-5.4.17.bin.tar.gz` : les bibliothèques au format ELF,
- `libc-5.4.17.tar.gz` : les sources des bibliothèques,
- `libg++-2.7.2.1.bin.tar.gz` : la bibliothèque C++ au format ELF.

Il est impératif de suivre à la lettre les instructions qui suivent. En effet, il est très facile de rendre son système inutilisable lors de la mise à jour des bibliothèques, et c'est encore plus vrai lors de la transition entre les formats a.out et ELF.

La procédure de mise à jour est la suivante :

1. Installation de l'éditeur de liens dynamique

```
# cd /usr/src
# tar xvfz /.../ld.so-1.8.5.tar.gz
# cd ld.so-1.8.5
# sh instldso.sh
```

2. Sauvegarde des anciennes bibliothèques (au format a.out)

```
# mkdir -p /usr/i486-linuxaout/bin /usr/i486-linuxaout/lib
# echo "/usr/i486-linuxaout/lib" >> /etc/ld.so.conf
# ldconfig -v
# cd /usr/lib
# mv *.o *.a *.sa lib*.so.* /usr/i486-linuxaout/lib
# cd /lib
# mv *.o *.a *.sa lib*.so.* /usr/i486-linuxaout/lib
# ldconfig -v
# cd /usr
# mv include /usr/i486-linuxaout
```

3. Installation des nouvelles bibliothèques

```
# cd /
# tar xvfz /.../libc-5.4.17.bin.tar.gz
# tar xvfz /.../libg++-2.7.2.1.bin.tar.gz
# ldconfig -v
```

4. Installation des nouveaux utilitaires

```
# cd /
# tar xvfz /.../gcc-2.7.2.1.bin.tar.gz
# rm -rf /usr/lib/ldscripts
# rm -f /usr/bin/as /usr/bin/as.1.38.11.2 /usr/bin/as.2.21
# rm -f /usr/bin/ld /usr/bin/ld.1.91.4
# tar xvfz /.../binutils-2.7.0.3.bin.tar.gz
```

Voilà! Tu as maintenant un système dans lequel coexistent les bibliothèques aux formats a.out (sous `/usr/i486-linuxaout/lib`) et ELF (sous `/lib` et `/usr/lib`). Les anciens binaires doivent continuer à s'exécuter correctement.

Attention, si tu utilises des systèmes de fichiers différents pour `/` et `/usr`, tu risques d'avoir des problèmes au démarrage de la machine. En effet, les anciennes bibliothèques étant dans `/usr/i486-linuxaout/lib`, elles ne sont pas accessibles tant que `/usr` n'est pas monté, ce qui peut empêcher un démarrage normal du système. Si tu es dans ce cas, il faut impérativement que tu déplaces au moins l'image a.out de la bibliothèque C :

```
# cd /usr/i486-linuxaout/lib
# mv libc.so.* /lib
# ldconfig -v
```

13.3 Migration vers un système tout-ELF

Maintenant que tu as installé les utilitaires et les bibliothèques ELF, il ne te reste plus qu'à recompiler tous les programmes que tu utilises pour bénéficier de ce nouveau format d'exécutable. Il te faut donc retrouver

les sources de tous tes programmes et les compiler un par un (bon courage, ça peut prendre pas mal de temps :-). Tu peux également récupérer des binaires déjà compilés sur les sites FTP Linux, beaucoup sont désormais au format ELF.

Il est à noter que certains programmes nécessitent des modifications pour être compilables en ELF, c'est notamment le cas du noyau Linux (version 1.2.x) ainsi que de tous les logiciels contenant des bibliothèques partagées.

Sur le site `ftp.ibp.fr`, le répertoire `/pub/linux/ELF` contient des patches pour certains programmes.

13.4 Utilisation des utilitaires ELF

Les nouvelles versions du compilateur et des utilitaires génèrent des exécutables au format ELF, par défaut. Il est possible de créer des exécutables au format a.out en utilisant l'option `-b i486-linuxaout` lors de l'appel de gcc.

Il est désormais assez facile de construire des bibliothèques dynamiques. Les fichiers source doivent être compilés avec l'option `-fPIC` afin que le code généré puisse s'exécuter à n'importe quelle adresse (les bibliothèques dynamiques sont chargées à des adresses non prévisibles aussi le code doit être indépendant de son adresse physique). L'option `-shared` de gcc permet ensuite de générer une bibliothèque dynamique.

Par exemple, pour générer une bibliothèque `libtruc.so.1.0`, on utilisera les commandes suivantes :

```
$ gcc -c -fPIC *.c
$ gcc -shared -Wl,-soname,libtruc.so.1 -o libtruc.so.1.0 *.o
```

13.5 Sources d'information sur ELF

Quelques documents concernant ELF sont disponibles sur `ftp.ibp.fr` dans `/pub/linux/ELF/doc`. Il s'agit de :

- `ELF.doc.tar.gz` : la spécification du format ELF,
- `ELF.ps.gz` : un article, écrit par H.J. Lu, décrivant l'implémentation du format ELF dans Linux,
- `move_to_elf.gz` : une description de la transition entre les formats a.out et ELF.

Il est également possible d'en apprendre plus sur ELF par les URL suivants :

- `http://www.intac.com/~cully/elf.html`
- `http://www.sjc.ox.ac.uk/users/barlow/linux-gcc.html`

14 Catastrophe !

Ce chapitre est dédié aux moments de catastrophe, ou rien de marche, et ou l'on a envie de tout mettre par la fenêtre...

14.1 Impossible de booter !

Tu viens de ralumer ta machine, et tu n'arrives plus à booter. Dans ce cas, le mieux est d'utiliser les disquettes de boot (celles de la Slackware dans ce cas sont pas mal). Une fois la machine amorcée, tu montes ta partition racine et tu copie le noyau sur une nouvelle disquette en lançant : `cat /mnt/vmlinuz >/dev/fd0` par exemple. Normalement, ça devrait alors fonctionner si ton noyau est correct. Eventuellement, effectue un :

```
rdev /vmlinux /dev/hda2
```

pour spécifier au noyau quelle partition est la racine du système de fichiers.

14.2 Formattage accidentel d'une partition

Voila, tu as formatté par erreur une partition où se trouvait ton rapport de stage. Bien, pas de panique ! Inutile de te dire qu'il est impossible de faire un `unformat`.

La première solution de répcupération (très brutale) est de faire un `strings /dev/hda3 >Recup`. Il reste ensuite à parcourir le fichier et de récupérer ce que tu peux.

Une seconde solution est d'utiliser ce scrip Perl (voir *http://www-eleves.enst-bretagne.fr/ aubert/perl/*) :

```
# Auteur : aubert@titan.enst-bretagne.fr
#!/usr/local/bin/perl

$maxlines = 20;

@before = ();
$syntaxe = "Syntaxe: cgrep.pl terme_a_rechercher fichier_a_parcourir\n";

$terme = shift(@ARGV) || die $syntaxe;
$fichier = shift(@ARGV) || die $syntaxe;

open(F, $fichier) || die "Cannot read $fichier: $!\n";

# On remplit @before jusqu'a sa capacite maximales ($maxlines)
while (($_ = <F>) && (scalar(@before) < $maxlines))
{
  if (/$terme/o)
  {
    print @before;
    print $_;
    & print_next_lines;
  }
  push(@before, $_);
}
```

```
# Le tableau @before contient la bonne quantite d'elements, donc on
# passe maintenant dans une partie ou @before garde une taille constante
while (<F>)
{
  if (/$terme/o)
  {
    print @before;
    print $_;
    & print_next_lines;
  }
  push(@before, $_);
  shift(@before);
}

close(F);
exit 1;

sub print_next_lines
{
  for ($i = 0; $i < $maxlines; $i++)
  {
    print scalar(<F>);
  }
  exit 0;
}
```

14.3 Problèmes avec le disque dur : super bloc

Dans le cas où tu penses que le super-bloc est vérollé, dans ce cas, lances `e2fsck` avec les options suivantes : `e2fsck -b 8193 /dev/sda3`. Si ça ne marche toujours pas, faire augmenter le nombre de 8192 à chaque fois pour essayer de récupérer une copie du super-bloc en état.

Si ça ne fonctionne toujours pas, essayer de lancer la commande `mke2fs` avec l'option **-S**. Attention à ne pas l'oublier sinon tu reformates ta partition ! Cette option provoque la regénération des copies du superbloc et des descripteurs du système de fichiers, et n'écrase pas les fichiers existants.

14.4 Problèmes avec le disque dur : blocs défectueux

Si jamais ton disque a quelques problèmes (secteurs défectueux), tu peux récuperer `e2fsck` version 1.04 si tu ne l'as pas déjà. Ensuite,

- booter sur une disquette si le problème est à la racine ;
- lancer `badblocks >BLOCKS` ;

– lancer e2fsck -f -L BLOCKS -p -y.

14.5 fatal signal 11

Certaines machines ne pourront fonctionner convenablement et produiront alors de façon plus ou moins aléatoire un message (par exemple "fatal signal 11") puis stopperont brutalement l'exécution du programme actif. Cause la plus probable: problème du matériel (composants de mémoire trop lents, par exemple). Essayer tout d'abord de "débrayer" le cache processeur, grâce au "SETUP" de la machine. Puis diminuer le niveau de PIO de l'EIDE (ne pas dépasser PIO 3) Lire à ce propos le document http://einstein.et.tudelft.nl/ wolff/sig11

15 Divers

15.1 Comment se faire une disquette de Boot

Il peut être très utile d'avoir sur disquette un noyau pour booter dessus.

Pour faire cela, prend une disquette formatée et va à la racine de ton disque.

Identifie ton noyau. Il doit s'appeler `zImage` ou `vmlinuz`.

Tu le copies d'abord sur disquette:

```
cat /zImage > /dev/fd0
```

puis tu fais (exemple):

```
rdev zImage /dev/hda2
```

Le deuxième paramètre doit être le nom de la partition racine **Linux** de ton système.

Cette disquette te permet alors de booter (tester!).

15.2 Mon mot de passe, où est mon mot de passe?

En bref: tu as perdu ton mot de passe de root... Bon, il y a plusieurs solutions. La première:

1. Tu bootes avec les disquettes boot et root de ta distribution.
2. Tu te loges en root (là, pas de mot de passe).
3. Tu montes ta racine de ton disque dur: `mount -t ext2 /dev/hda1 /mnt`
4. Tu vas dans le répertoire `/mnt`. Il ne te reste plus qu'à éditer le fichier `/mnt/etc/passwd`, y supprimer le mot de passe de root: `root::0:0:root:/root:/bin/bash`.

Tu rebootes **Linux** normalement, et tu n'as plus besoin de mot de passe pour te loger en root.

Bon, passons à d'autres solutions : Rebooter le pc en mode dit *single-user*. Pour cela, lors du boot avec LILO, lui dire `LILO: linux single`. Un shell root va apparaître . Attention : le clavier est en qwerty et la partition en lecture seule. Pour y remédier :

```
loadkeys /usr/lib/kbd/keytables/fr.map
mount -w -n -o remount /
```

Soit tu utilises la commande `passwd`, soit tu édites le fichier `/etc/passwd`

15.3 Comment limiter le reboot en single user ?

Le problème du reboot en `single user`, c'est que n'importe qui peut alors réussir à passer sur la machine en root. Pour les entreprises ou les universités, cela pose un problème de sécurité. Certaines stations Unix permettent de mettre un mot de passe dans l'EPROM de la machine. Linux permet de demander le mot de passe root même en *single-user*.

Pour cela, tu dois récupérer les sources du programme `init` qui est lancé lors de l'amorçage du système. Au début du programme `init.c`, tu modifies, la définition de la constante *SOME_USER* pour qu'elle ait la valeur 2, tu recompiles `init`, et tu le ré-installes.

Cette première solution peut toutefois s'avérer être insuffisante car une personne peut toujours booter sur un autre périphérique (en utilisant l'option `root = MonLinux`).

En utilisant Lilo, pas de problème ! il suffit alors d'ajouter les lignes suivantes pour chacune des images dans le fichier `/etc/lilo.conf` :

```
password = le mot de passe en clair
restricted
```

(penser à mettre ce fichier en lecture seule pour le super-utilisateur et aucun droit pour les autres !)

Le boot normal de chaque image se passe sans problème et sans demander le mot de passe (important si l'on veut que la machine redémarre seule en cas de pépin : EDF, reboot à distance, ...) mais si l'on veut passer des paramètres au noyau lors du boot, lilo demande alors le mot de passe.

15.4 Fenêtres virtuelles

Linux te permet de travailler en même temps sur plusieurs fenêtres textes. Pour passer de l'une à l'autre, fais `Alt + Fn` ou `n` est le numéro de la fenêtre virtuelle. Tu peux configurer le nombre de fenêtres dans le fichier `/etc/inittab`.

Quand tu est sous X, la combinaison devient Ctrl + Alt + Fn. Il est alors possible de retourner sous X en accédant ainsi à la console qu'il emploie (la première "libre", c'est-à-dire non gérée par un getty).

15.5 Francisation

15.5.1 locale // nls

Les « locales » sont un système permettant de gérer le plus proprement possible les différences de langue et de style des utilisateurs. Par exemple, des francophones préfèrent écrire 3,14 plutôt que 3.14, ne notent pas les dates comme les Australiens (qui ne le font pas comme les États-Uniens), etc. Chaque utilisateur doit donc pouvoir choisir une locale qui lui convienne et les applications bien faites (rares sont celles qui gèrent complètement les locales mais nombreuses sont celles qui les utilisent au moins un peu) s'adaptent sans recompilation. En somme, il s'agit de mettre les préférences nationales en dehors de l'application, comme avec le système de ressources du Machinetoc.

Si tu es programmeur, tu peux consulter le locale-tutorial (`distributions/jurix/source/libc/nls` ou `sunsite/utils/nls`) ou le Mini-HOWTO sur les Locales. Si tu es simple utilisateur, tu peux connaitre les locales installées sur ton système avec `locale -a` (les noms semblent peu standardisés, tu auras des trucs du genre « fr_FR » ou « fr_FR.ISO8859-1 ») et en sélectionner une avec les variables d'environnement comme LANG ou LC_CTYPE (`man locale` donnera peut-être quelque chose).

Si tu es administrateur d'une machine, que les locales ne marchent pas comme tu veux, tu peux lire le Mini-HOWTO sur les Locales et/ou ce modeste texte.

Si ton Linux n'est pas venu avec une installation correcte des locales (je ne crois pas qu'il existe une bonne distribution de ce point de vue), tu as deux solutions. Je n'ai testé que la première et elle ne semble marcher qu'avec un noyau supérieur ou égal à 2.0 et une libc version 5.2.18 ou supérieure.

Première solution : compiler les programmes de la libc et utiliser le paquetage WG15-collection.

Pour cela, tu dois récupérer les sources de la libc. Il n'est pas nécessaire de la compiler en entier, juste les programmes de gestion des locales, qui permettent notamment de produire un fichier binaire à partir du source d'une locale. Pour résumer l'installation :

Récupérer libc (ici 5.2.18), par exemple dans `ftp://ftp.ibp.fr/pub/linux/GCC`.

```
(passer root)
cd /usr/src
mkdir libc-5.2.18
chown <USERNAME> libc-5.2.18
(quitter root)
cd libc-5.2.18
tar -xzvf /usr/tmp/libc-5.2.18.tar.gz
cd include
ln -s /usr/src/linux/include/asm .
ln -s /usr/src/linux/include/linux .
cd ../libc
```

Et là tu peux lire les instructions dans le README et faire « ./configure » (« make depend » est long et ne semble pas nécessaire si tu veux juste les locales)

Si tu ne veux pas compiler toute la libc, tu t'arrêtes là et :

```
cd locale
make SHARED= programs
(passer root)
mv localedef /usr/local/bin
mv locale /usr/local/bin
(quitter root)
```

Tu as alors les deux programmes importants. Tu peux maintenant générer des locales binaires à partir des sources.

Une bonne collection de sources se trouve dans le fichier WG15-collection.linux.tar.gz qu'on trouve par exemple en `ftp://ftp.ibp.fr/pub/linux/GCC` (il doit son nom à un « working group » Posix).

Dans ce fichier, une fois détaré, il y a une bonne documentation (Intro). Si tu ne veux pas la lire, ce qu'il faut faire est :

```
(passer root)
cd charmaps
mkdirhier /usr/share/nls/charmap
cp ISO_* /usr/share/nls/charmap
cd ../locales
mkdirhier /usr/share/nls/locale
cp POSIX ??_* /usr/share/nls/locale
mkdirhier /usr/share/locale
(la ligne suivante est à cause de ce qui semble être une bogue dans la
distribution)
localedef -c -i /usr/share/nls/locale/en_DK -f ISO_8859-1:1987 en_DK
(plein de message d'erreurs du genre "item ‘yesstr’ of category
‘LC_MESSAGES’ undefined" à ignorer)
localedef -c -i /usr/share/nls/locale/fr_FR -f ISO_8859-1:1987 fr_FR
(quitter root)
```

Tu peux tester avec `locale -a` que la nouvelle locale est désormais connue. C'est tout. Les utilisateurs peuvent désormais utiliser les variables d'environnement des locales comme `LANG` ou `LC_CTYPE`.

Deuxième méthode, que je n'ai pas réussi à faire fonctionner :

Récupérer `nslutils` (aujourd'hui `nlsutils-0.5.tar.gz`) par exemple (il semble nécessaire de récupérer des fichiers d'en-têtes comme localeinfo.h dans la libc).

```
make
make install
```

Pour tester ton installation des locales, tu peux te servir de Perl 5.003 (c'est la première version de Perl qui teste que l'installation est correcte et produit un message d'erreur autrement) ou ce programme :

```
#include <stdio.h>
```

```
#include <ctype.h>
#include <string.h>
#include <locale.h>
#include <sys/types.h>
#define STRING "bétö\b"

void main ()
{
    int i;
    u_char c;
    printf ("Setting locale: %s\n", setlocale(LC_ALL, ""));
    for (i=0; i<strlen(STRING); i++)  {
       c = STRING[i];
       printf ("%c (%d) is %s\n",
          c, (int)c, isprint((int)c)?"printable":"non-printable");
     }
}
```

Il ne doit pas afficher *Setting locale: (null)* et, avec la bonne locale française, doit trouver que tous les caractères, sauf le dernier, sont « printables ».

Certains outils GNU commencent à être francisés. Pour avoir cette possibilité, rajoute dans ton `.bashrc` les lignes suivantes:

```
export LC_CTYPE=ISO-8859-1
export LANG=fr
```

15.5.2 ispell

ispell est un correcteur orthographique, en français. Il est assez utilisé. Tu peux le récupérer sur `ftp.ibp.fr` dans `/pub/ispell`. Il est également possible de récupérer un dictionnaire français pour *ispell* sur le même site dans le répertoire `/pub/ispell/francais`.

15.5.3 Les accents sous bash

Rajoute dans le fichier `.inputrc` les lignes suivantes:

```
set meta-flag on
set convert-meta off
set output-meta on
```

15.5.4 less

Il est très désagréable de ne pas avoir les accents lorsque l'on fait un *less* ou un *more* d'un fichier. Pour que ce problème ne se pose pas, rajouter :

```
export LESSCHARSET=latin1
```

dans le fichier `/etc/profile`.

15.5.5 Formatage de textes

Pour *gs* et *xdvi*, il faut utiliser des options spéciales (sur ma machine, ce sont des alias). En effet, ils sont souvent configurés pour un papier de taille anglo-saxonne.

Ces options sont les suivantes :

```
gs -sPAPERSIZE=a4
xdvi -paper a4
ghostview -a4
```

Pour que *dvips* convertisse les documents dans un format papier a4, il faut spécifier dans le fichier `config.ps` (le chemin varie en fonction des versions de LaTeX) :

```
@ a4 210mm 297mm
@+ ! %%DocumentPaperSizes: a4
@+ %%PaperSize: a4
@+ %%BeginPaperSize: a4
@+ a4
```

15.5.6 elm

Pour que ton courrier soit en 8 bits (avec accents) sans avoir besoin de l'encapsuler via MIME ou autre, rajoute ces trois lignes dans ton fichier `~/.elm/elmrc` :

```
charset = iso-8859-1
displaycharset = iso-8859-1
textencoding = 8bit
```

15.6 TeX et les polices de caractères

Certains personnages indélicats se sont permis de bidouiller les polices de caractères `Computer Modern` sans en changer le nom ET CE CONTRAIREMENT A TOUS LES COPYRIGHTS de Knuth. Ceci est EXTREMEMENT grave parce que tout document typographie avec ces polices de caractères n'aura pas le meme look qu'avec les VRAIES `Computer Modern`. Or, malheureusement, ces polices de caractères fautives

ont été livrées sur plusieurs CDs Linux. Comme nous travaillons quasiment tous avec des CDs slackware, il convient de faire extremement attention.

C'est pourquoi que tu devrait vérifier ton système. Pour cela, c'est fastoche, un bon vieux terminal et deux minutes de temps :

- lancer tex
- Taper au prompt **

```
\setbox0=\hbox{ho}\showbox0
```

- Si TeX repond:

```
> \box0=
\hbox(6.94444+0.0)x10.55559
.\tenrm h
.\tenrm o
! OK.
<*> \setbox0=\hbox{ho}\showbox0
?
```

c'est gagne, tu as les bonnes polices de caractères (la valeur clef à regarder, c'est le 10.55559). Il suffit alors de repondre 'x' pour que TeX se termine.

Dans le cas contraire, ton système est vérollé. Il suffit de récupérer les mises à jour et le tour est joué.

15.7 Une carte son bien configurée

Lorsque l'on compile et que l'on se sert d'une carte son sous **Linux**, la grande question est de savoir si la carte est correctement configurée. Pour cela, il existe un fichier (`/dev/sndstat`) qui donne toutes les informations sur la configuration de la carte :

```
Sound Driver:3.5.4-960630 (Mon Nov 11 14:50:52 MET 1996 root,
Linux gandalf 2.0.25 #3 Mon Nov 11 11:44:54 MET 1996 i486)
Kernel: Linux gandalf 2.0.25 #3 Mon Nov 11 11:44:54 MET 1996 i486
Config options: 0

Installed drivers:
Type 1: OPL-2/OPL-3 FM
Type 2: Sound Blaster
Type 7: SB MPU-401

Card config:
Sound Blaster at 0x260 irq 5 drq 1,5
SB MPU-401 at 0x330 irq 5 drq 0
```

```
OPL-2/OPL-3 FM at 0x388 drq 0

Audio devices:
0: Sound Blaster 16 (4.11)

Synth devices:
0: Yamaha OPL-3

Midi devices:
0: Sound Blaster 16

Timers:
0: System clock

Mixers:
0: Sound Blaster
```

Pour tester la bonne marche de la carte son, il suffit d'envoyer quelques fichiers audio (.au) sur `/dev/audio`, etc.

15.8 Bug gcc

Certaines version de gcc ont un bug avec l'option `-O2`. Pour l'identifier, tester ce programme:

```
/* sr_bug.c
 *
 * This program tests for a gcc bug.
 * To compile this test program:  gcc -O2 sr_bug.c
 *
 * Sometimes gcc for Intel CPUs generates bad code at optimization
 * level 2.  The bad code is 'almost right' and stay hidden inside
 * a program that seems to work - including the Linux kernel.
 * The bug is very old and has been reported.  As of 19-Dec-95,
 * the bug has not been fixed.
 *
 * If you change this code you should test it, because even a
 * tiny change will hide this elusive bug.  If you think you
 * fixed the bug, please run the original test just to make
 * sure. You can find the original test below, after the #if 0.
 * I wrote this version of the test to be user-friendly, and
 * it may not be as solid as the original.
 *
 * Some people who know more than I do:
 * davis@space.mit.edu (John E. Davis)
```

```
 * anlauf@crunch.ikp.physik.th-darmstadt.de (Harald Anlauf)
 * craigs@iii2.iii.net (Craig Shrimpton)
 *
 * User-friendly version by Albert Cahalan
 *
 */

#include <stdio.h>

int gcc_sr_bug(void){
  static int Array[3]; /* must be static (or global) */
  unsigned int B = 3;  /* must be unsigned 32-bit */
  int i;
  for(i=0; i<B; i++) Array[i] = i - 3;
  for(i=0; i<B; i++) printf(" %d,%d", i, Array[i]);
  return !Array[1];
}

int main(){
  printf("Testing for gcc bug...");
  if(gcc_sr_bug()){
    printf("\n\nBad code! Your compiler generates bad output.\n\n");
    printf("Add -fno-strength-reduce to your gcc command line\n");
    printf("or put it into your gcc config file, such as in\n");
    printf("/usr/lib/gcc-lib/i486-linux/2.7.0/specs.\n");
    exit(1);
  }else{
    printf("\nOK, no problem.\n");
    exit(0);
  }
}
```

15.9 Emacs

Emacs est un éditeur de texte très puissant car cohérent et extensible mais consommateur de ressources (processeur et mémoire). La plupart de ceux qui commenceront à l'employer sérieusement, en explorant les documentations, ne pourront bientôt plus s'en passer.

Voici un fichier de configuration de base :

```
;;
;; Fichier .emacs: initialisation d'emacs
;;   Fichier de base : Guide du Rootard
;;
```

```
(display-time)                      ;; Pour avoir l'heure dans la barre d'etat
(setq display-time-24hr-format t)  ;; Format 24 heures

;; Nouveaux modes
(autoload 'c++-mode   "cplus-md" "C++ Editing Mode" t)
(autoload 'perl-mode   "perl-mode" "Perl Editing Mode" t)
(autoload 'c-mode "c-mode" "C Editing Mode" t)
; mieux vaudrait utiliser le "cc-mode"

(autoload 'jargon-mode "jargon" "Jargon Mode" t)

;; Auto-Mode Settings : positionne le mode selon l'extension
(setq auto-mode-alist
(append '(("\.c$"  . c-mode)    ;; utilise le mode C++ meme pour C
("\.h$"  . c-mode)
("\.C$"  . c++-mode)
("\.H$"  . c++-mode)
("\.cc$" . c++-mode)
("\.C$"  . c++-mode)
("\.pl$" . perl-mode)           ;; Perl
("/tmp/snd\.[0-9]*" . text-mode);; Text (pour le mail)
("[Rr][Ee][0-9]*" . text-mode)
("\.ada$"  . ada-mode)          ;; Ada
("\.spec$"  . ada-mode)
("\.body$"  . ada-mode)
("makefile$" . makefile-mode)  ;; Makefile
("Makefile$" . makefile-mode)
("Imakefile$" . makefile-mode))
auto-mode-alist))

# Remapes varies
(global-set-key "\eg" 'goto-line)       ;; ESC G = Goto line
(global-set-key "\eo" 'overwrite-mode)

(put 'eval-expression 'disabled nil)

;; Accents...
(standard-display-european 1)
(load-library "iso-syntax")

;; Sous X-Window, textes en couleurs (C/C++/Shell/Makefile,etc)
(cond (window-system
(setq hilit-mode-enable-list  '(not text-mode)
hilit-background-mode   'light
```

```
hilit-inhibit-hooks     nil
hilit-inhibit-rebinding nil)
(require 'hilit19)
))
(if (not (equal window-system ""))
(global-set-key "\C-?" 'delete-char))
))
```

Ce fichier peut être largement amélioré. Pour en savoir plus, consulte la documentation d'Emacs (appuyer sur "control-H" puis "i"), qui est bien faite mais un peu grosse.

Remarque: il est possible d'ajouter les lignes suivantes pour pouvoir se servir des accents lors de l'édition de fichiers en mode texte:

```
(set-input-mode (car (current-input-mode))
              (nth 1 (current-input-mode))
              0)
```

15.10 Vérouillage des ports série

L'utilisation des ports série pose un problème de verrouillage: il ne faut pas que deux processus concurrents accèdent à un port série donné en même temps. Mais d'autre part, si on a un port série sur lequel tourne un getty pour traiter des appels entrants, on voudrait bien quand même pouvoir utiliser le port série pour passer des appels sortants.

Pour résoudre ce problème, il y a deux approches. La première, c'est le verrouillage par le noyau. Elle utilise deux périphériques différents pour accéder à un port série: /dev/ttySn pour les appels entrant (le getty), et /dev/cuan pour les appels sortant. Quand le getty tente d'ouvrir /dev/ttySn, il est bloqué jusqu'à ce qu'un appel entrant soit reçu et que le modem le signale par la ligne DCD du port série.

Tant que le getty est bloqué dans l'appel système open, on peut encore utiliser le périphérique /dev/cuan pour les appels sortants (évidemment, le getty reste bloqué lorque DCD est activée pour l'appel sortant...)

D'autre part, une fois que /dev/cuan est ouvert, un processus d'une autre session ne peut pas l'ouvrir (si le système est corretement configuré – voir l'option session_lockout de setserial(8).) Cela évite que deux processus tentant indépendamment de passer un appel sortant se trouvent à écrire simultanément sur le port série.

La seconde technique est celle des "lock-files" (fichiers-verrous). Elle consiste, pour chaque processus qui utilise un port série, à écrire son PID dans un fichier spécial. De cette façon, quand un autre process veut utiliser le même port série, il vérifie la présence de ce fichier, le trouve, lit le PID qui y est écrit, vérifie que le processus correspondant existe, et sait donc que le port série est occupé et qu'il ne peut pas l'utiliser. Dans ce cas, on n'utilise que les /dev/cuan.

Chacune des deux approches a ses avantages et ses inconvénients. La méthode de verrouillage par le noyau nécessite d'avoir un modem en auto-réponse, et autorise l'ouvertue du port série par un processus qui veut passer un appel sortant entre le moment où le téléphone sonne et le moment où les modems on fini de négocier (ce qui peut prendre une dizaine de secondes.) La méthode des lock-files nécessite une configuration

homogène des logiciels utilisés (tous les logiciels doivent chercher et mettre le lock-file au même endroit, s'entendre sur le format du PID qui y est inscrit...) Elle nécessite aussi que le getty vérifie périodiquement l'existence du lock-file, et il peut donc "rater" un appel entrant entre le moment où un processus libère le port série et celui où il vérifie à nouveau la présence du lock-file. Il est aussi possible qu'un processus sortant arrive à créer le lock-file entre le moment où le getty détecte un appel et celui où il crée le lock-file pour cet appel.

Le choix d'une méthode est, finalement, une décision individuelle de chaque administrateur, sachant que l'approche "lock-file" est plus souple, mais moins élégante et plus délicate à mettre en oeuvre que le verrouillage par le noyau.

15.11 Minitel

Il est fort agréable de pouvoir se connecter sur le Minitel. Récupère le programme `xtel` de Pierre Ficheux. Il s'agit d'un émulateur Minitel achtement cool à utiliser.

Tu peux le trouver sur `ftp.ibp.fr` dans le répertoire `/pub/X11/videotext/xtel`.

15.12 Fax

Envoyer des documents par FAX est tout à fait possible. Pour cela, tu peux utiliser par exemple *Hylafax*.

Il se trouve sur le site `ftp.funet.fi` dans le répertoire `/pub/mirrors/sgi.com/sgi/fax/source`.

15.13 Modules Chargeables

Cette partie s'inspire allégrement de l'article que j'ai écris dans le journal « Les échos de Linux », au mois de Juillet 96.

15.13.1 Structure du noyau

Le noyau d'un système Unix peut être représenté sous la forme d'un objet monolythique. Toutefois, un tel objet possède l'inconvénient d'être gros et statique. A chaque fois que l'on désire rajouter un nouveau périphérique, il est nécessaire de recompiler le noyau. De plus, si l'on n'utilise certains gestionnaires particuliers que rarement, on est obligé de l'avoir dans le noyau, ce qui a tendance à consommer de la mémoire.

15.13.2 Qu'est-ce qu'un module chargeable ?

Les modules chargeables, permettent de joindre l'utile à l'agréable en ayant un noyau le plus petit que possible, charge à la demande ce dont il a besoin, d'une manière manuelle par le super utilisateur de la machine, soit d'une manière automatique. De cette manière, le gain de ressources est non négligeable.

La première question que l'on peut se poser, c'est : « Pourquoi deux techniques de chargement ? »

La première technique est manuelle : il faut charger ou décharger les modules à la main. La deuxième est automatique, grâce à l'utilisation d'un démon spécialisé qui est l'esclave du noyau et qui charge et décharge

les modules pour lui. En fait, la version 1.2 de Linux n'offrait que la possibilité d'un chargement manuel qui est limité au super-utilisateur de la machine et qui est assez lourd à manipuler. Au fil du développement de la version 2.0, un nouveau système implémenté par Bjorn Ekwall permit d'éffectuer un chargement dynamique et automatique des modules.

15.13.3 Compilation du noyau

Lors de la compilation du noyau, il est nécessaire de spécifier des options particlières pour activer l'utilisation des modules chargeables :

```
gandalf# make config
*
* Loadable module support
*
Enable loadable module support (CONFIG_MODULES) [Y/n/?]
Set version information on all symbols for modules (CONFIG_MODVERSIONS) [N/y/?]
Kernel daemon support (e.g. autoload of modules) (CONFIG_KERNELD) [Y/n/?]
```

Voici le détail de ces trois options :

- CONFIG_MODULES : active le mécanisme de modules chargeables. Sans cette options, ils ne fonctionneront pas sur ton système.
- CONFIG_MODVERSIONS : si tu laisses cette option à N (par défaut), tu devras recompiler les modules à chaque mise à jour du noyau de ta machine. Cela signifie que des modules compilés pour un noyau 2.0.n ne fonctionneront pas avec un module compilé pour 2.0.n+1. Si tu actives cette option, les modules pourront être lancés... toutefois, ce n'est pas vraiment conseillé.
- CONFIG_KERNELD : cette option active le chargement automatique des modules. Tu dois également activer les IPC Système V car le noyau et le démon communiquement via des files de messages.

Une fois fois configuré, il te suffit de lancer la compilation ainsi que l'installation :

```
gandalf# make dep ; make clean
gandalf# make zImage
gandalf# make modules ; make modules_install
```

Une fois ces opérations effectuées, les modules se trouvent alors dans le répertoire /lib/modules/x.y.z où x.y.z correspond au numéro de version du noyau. Il ne nous reste plus à voir que le chargement.

Note importante : tous les outils de manipulation des modules sont archivés dans un fichier `modules-2.0.0.tar.gz` dans le répertoire `v2.0`. Ils sont installés par défaut avec les distributions standards, mais il est important d'utiliser la bonne version.

15.13.4 Chargement : méthode manuelle

Le chargement manuel est basé sur trois commandes :

- `insmod` : insère un module dans le noyau ;
- `rmmod` : décharge un module, s'il plus aucun processus ne l'utilise ;
- `lsmod` : affiche la liste des modules chargés.

Leur utilisation oblige d'être en super-utilisateur. Voici un exemple d'utilisation :

```
gandalf# insmod nfs.o
gandalf# lsmod
Module:        #pages:  Used by:
nfs               12            4
gandalf# mount -t nfs /truc /mnt
gandalf# lsmod
Module:        #pages:  Used by:
nfs               12            5
gandalf# cd /mnt
...
gandalf# cd /
gandalf# umount /mnt
Module:        #pages:  Used by:
nfs               12            4
gandalf# ps axu | grep nfs
root      5535  0.0  0.0     0     0  q2 SW  17:15   0:00 (nfsiod)
root      5536  0.0  0.0     0     0  q2 SW  17:15   0:00 (nfsiod)
root      5537  0.0  0.0     0     0  q2 SW  17:15   0:00 (nfsiod)
root      5538  0.0  0.0     0     0  q2 SW  17:15   0:00 (nfsiod)
root      5557  0.0  0.4   864   300  q2 S   17:16   0:00 grep nfs
gandalf# kill -9 5535 5536 5537 5538
gandalf# lsmod
gandalf# rmmod nfs.o
```

Il est nécessaire de "killer" les 4 démons nfsiod car ils sont lancés dès que NFS est activé. Comme tu peux le voir, ces opérations deviennent relativement pénibles. C'est pour cette raison que le système de chargement automatique a été crée.

15.13.5 Chargement automatique : `kerneld`

Introduction Le système de chargement automatique de modules permet de réduire au minimum la taille de son noyau. Le principe de fonctionnement est particulièrement simple : un démon en mode utilisateur est à l'écoute des ordres du noyau (via une file de message de type IPC Système V). Lorsque un processus essaye

d'accéder à une ressource système (via un appel système open, etc...), le noyau envoie l'ordre de chargement du module à kerneld. Une fois le message reçu, kerneld exécute un modprobe pour charger les modules nécesaires: Schémas du chargement/déchargement des modules avec kerneld

Conseils pour la compilation du noyau Par contre, lors de la compilation du noyau, il est nécessaire d'y mettre au moins le support pour permettre l'amoçage de la machine et le montage de la racine de ton système de fichier (par exemple, support IDE + ext2fs). Tu peux y mettre tout le reste en module (carte son, systèmes de fichiers, carte SCSI, etc).

Mise en place **Cette partie de modification n'est valable qui si ta machine n'était pas pourvue de `kerneld`. Les nouvelles distributions effectuent une installation tout à fait correcte**.

Pour réaliser la mise en place du système de chargement de modules, il est nécessaire d'effectuer certaines modifications au niveau de ta configuration. En effet, il est nécessaire lors de l'amorçage de la machine de lancer le démon kerneld et de réaliser une espèce de liste des dépendances des modules: certains modules ne peuvent être lancés avant que d'autres ne le soient. Dans un premier temps, il faut créer le fichier `/etc/rc.d/rc.modules` dans lequel, tu y met:

```
# Modules

#
# Création d'un lien logique pour le noyau courant
#
#
/bin/rm -f /lib/modules/current
ln -sf /lib/modules/‘uname -r‘ /lib/modules/current
#
# Création des dépendances
if [ \! -r /lib/modules/current/modules.dep ]
then
        echo "Creating module dependencies"
        /sbin/depmod -a
fi
#
# Chargement des modules d'amorçage...
#
if [ -x /sbin/kerneld ]
then
        if find /lib/modules/boot -type f -o type l > /dev/null 2>&1
        then
                echo "Loading boot-time modules"
                /sbin/modprobe -a -t boot \*
        fi
else
```

```
        echo "Loading modules"
        /sbin/modprobe -a \*
fi
#
# Si vous possédez d'autres types de démons kerneld à lancer...
#
if [ -x /sbin/kdsound ]
then
        echo "Starting sound daemon"
        /sbin/kdsound &
fi
```

Cela permet de générer la dépendance de vos modules à chaque fois que tu amorces ta machine. Ensuite, dans le fichier `/etc/rd.d/rc.S` (peut dépendre de ta distribution...), il convient de rajouter :

```
# Start update.
/sbin/update &

# *** A RAJOUTER ***
# Lancement de kerneld le plus tôt possible, de telle manière
# que les modules de systèmes de fichiers puissent être chargés
if [ -x /sbin/kerneld ]
then
        echo "kerneld running"
        /sbin/kerneld
fi

...

# Un peu de ménage
cat /dev/null > /var/adm/utmp

# Lancement du script des modules
if [ -f /etc/rc.d/rc.modules ]; then
        /etc/rc.d/rc.modules
fi
```

Une fois ces modifications effectuées et la machine réamorcée, tout doit être en place. Si kerneld permet de charger automatiquement les modules, il permet également de les décharger au bout d'un certain temps de nom utilisation. Par défaut, si aucun processus n'accède au module pendant plus de 60 secondes, il est automatiquement déchargé. Il est possible de modifier cette valeur en rajoutant le paramètre `delay=Nb_Secondes` à kerneld où Nb_Secondes est le délai en nombre de secondes.

Le fichier /etc/conf.modules Il peut arriver qu'il soit nécessaire de configurer un dernier fichier : le fichier `/etc/conf.modules`. Ce fichier contient les chemins où se trouvent les modules devant être chargés et ensuite des alias pour les

modules. Si tu n'as pas ce fichier, tu peux le créer avec :

```
gandalf# /sbin/modprobe -c | grep -v '^path' >/etc/conf.modules
```

Il peut arriver que lors du premier amorçage tu obtiennes ce message :

```
Cannot locate module for net-pf-3
Cannot locate module for net-pf-4
Cannot locate module for net-pf-5
```

Pas de panique ! Ce message n'est pas méchant et pour ne plus l'avoir, rajouter dans le fichier `/etc/conf.modules` :

```
alias net-pf-3 off
alias net-pf-4 off
alias net-pf-5 off
```

Il peut arriver que certains périphérifériques aient besoin de certains paramètres particulier. Consultez le document Kernel HowTo.

Quelques références Tu peux consulter ces références pour plus d'information :

- *http://www.pi.se/blox/modules/index.html* : site officiel pour les archives des modules ;
- tout miroir des sources Linux ;
- Kerneld Mini Howto.

16 Pour aller plus loin

16.1 « Bien débuter sous Linux » (V 2.2.2)

Ce livre est la traduction du livre « **Linux** Installation and Getting Started » de Matt Welsh (`mdw@sunsite.unc.edu`), version 2.2.2. La traduction a été réalisée par René Cougnenc et elle est disponible sur `ftp.ibp.fr` dans `/pub/linux/french/books`.

16.2 Livres

Bien que **Linux** a pour principe d'être entièrement gratuit, de nombreux ouvrages sont vendus dans le commerce.

- « Le système **Linux** » par Matt Welsh & Lar Kaufman (618 pages, environ 280FF T.T.C.) chez O'Reilly. Traduction de René Cougnenc.
- « Administration Réseau sous **Linux** » par Olaf Kirch (336 pages, environ 220FF T.T.C.) chez O'Reilly. Traduction de René Cougnenc.
- « Linux Kernel Internals » par M. Beck, H. Böhme, M. Dziadza, U. Kunitz, R. Magnus et D. Verworner chez Addison-Wesley ;
- « Programmation Linux 2.0, API Système et Fonctionnement du noyau » par Rémy Card, Éric Dumas et Franck Mével chez Eyrolles.

A noter que tu peux récupérer les sources LaTeX de la traduction du deuxième ouvrage sur `ftp.ibp.fr` dans `/pub/linux/french/books`.

Le nombre de livres concernant Linux a eu tendance à exploser ces derniers temps. Tu peux consuter le site `http://excalibur.inp-fc.fr/anrtt/doc/index.html` qui essaye de présenter les différents livres disponibles.

16.3 Les HowTo's

La plupart des HowTo's ou mini HowTo sont désormais traduits. Ils sont disponibles sur `ftp.ibp.fr` dans le répertoire `/pub/linux/french/docs/HOWTO`. En voici la liste :

HowTo :

```
Boot-disk-HOWTO          BootPrompt-HOWTO   NIS-HOWTO
Net2-HOWTO               Busmouse-HOWTO     PCI-HOWTO
CDROM-HOWTO              PPP-FAQ            Commercial-HOWTO
DOSEMU-HOWTO             Printing-HOWTO     Distribution-HOWTO
ELF-HOWTO                SCSI-HOWTO         PPP-HOWTO
SCSI-Programming-HOWTO   Firewall-HOWTO     Ftape-HOWTO
Serial-HOWTO             GCC-HOWTO          Sound-HOWTO
HAM-HOWTO                Term-HOWTO         Tips-HOWTO
Hardware-HOWTO           UPS-HOWTO          IPX-HOWTO
UUCP-HOWTO               Installation-HOWTO Video-HOWTO
Java-HOWTO               XFree86-HOWTO      Kernel-HOWTO
Keyboard-HOWTO           MGR-HOWTO          Mail-HOWTO
Modems-HOWTO
```

Mini-HowTo :

```
Anon-ftp          NFS-Root          X-Terminal
```

```
Assembleur          Postgres            ZipIomega
Colour-ls           Proxy-ARP           IO-Port
Shadow-Password     Jaz                 Slip+ProxyARP
LILO                Term-Firewall       Large-IDE
Token-Ring          Man-Page            Win95+Win3x+Linux
```

Toutefois, il reste encore du travail ! Si jamais tu veux te joindre à l'équipe des traducteurs, tu peux t'abonner à la liste de traduction :

```
mail majordomo@linux.eu.org
subscribe traduc
```

17 Adresses

Voici quelques bonnes petites adresses :

- sites FTP
  - `ftp.ibp.fr` : miroir de *tsx-11.mit.edu*, de *sunsite.unc.edu* avec les dernières mises à jour du noyau, et surtout toute l'arborescence française. Il s'agit du site de référence en France concernant Linux (et sûrement pour le reste également... bravo Rémy !).
  - `ftp.ese-metz.fr` : miroir de `ftp.ibp.fr` ;
  - `ftp.loria.fr` (miroir de sunsite et d'ibp) ;
  - une liste plus complète est fournie dans la feuille d'information que tu peux consulter au début de ce guide.
- BBS : la liste des BBS peut être trouvée dans la feuille d'information au début de ce guide ;
- WWW :
  - `http://www.freenix.fr/linux` : le site officiel des howto en français. C'est ici que tu trouveras toutes les version html. C'est également le site officiel du Guide du ROOTard.
  - `http://www.loria.fr/linux` : miroir concernant les howto. Tu trouveras sur ce serveur d'autres informations, telle que tous les articles postés depuis plus d'un an dans f.c.o.l.
  - `http://www.alienor.fr/~dumas/Linux` : ma modeste page dédiée à **Linux**.
- Listes locales: Il en existe deux à ma connaissance :
  - toulouse :

    ```
    majordom@arlesienne.ensica.fr :
    subscribe linux-31
    ```

- bordeaux :

```
majordomo@rd.lectra.fr :
subscribe linux-33
```

- Linux Journal Français (Les echos de linux) : crée par Pierre Ficheux et moi-même, ce journal est diffusé par Web, mais les articles sont également disponibles sur des serveurs ftp : `http://echo-linux.alienor.fr`. Serveur ftp : `ftp.ibp.fr:/pub/linux/french/echo-linux`.

  Une mailing liste est accessible :

```
mail majordomo@rd.lectra.fr
subscribe echo-linux
```

Pour trouver de l'aide : dans un premier temps, essaye de trouver les renseignement dans ce guide. Ensuite, regarde les HowTo. Enfin, si tu sèches, vas jeter un coup d'oeil dans le newsgroup `fr.comp.os.linux`.

18 Contributions

Voici la liste des personnes ayant participé à la réalisation de ce document. Il est fort possible que j'en ai oublié... mes excuses ! Merci à tous.

- Ceux sans qui ce document ne serait pas ce qu'il est :
  - Stéphane Bortzmeyer (`bortzmeyer@pasteur.fr`)
  - Rémy Card (`Remy.Card@freenix.fr`)
  - René Cougnenc
  - Pierre Ficheux (`pierre@rd.lectra.fr`)
  - Nat Makarévitch (`nat@nataa.fr.eu.org`)
  - Julien Simon (`Julien.Simon@freenix.fr`)
- Contributions :
  - Jean-Michel Antoine (`Jean-Michel.Antoine@loria.fr`)
  - Michel Billaud (`billaud@labri.u-bordeaux.fr`)
  - Laurent Chemla (`chemla@brainstorm.eu.org`)
  - Laurent Frigault (`frigault@isicom.fr`)
  - Frédéric Lepied (`fred@sugix.frmug.fr.net`)
  - Stephane Marzloff (`100557.571@compuserve.com`)
  - Marc Merlin (`magic.metawire.com`)
  - Thomas Parmelan (`Thomas.Parmelan@darshiva.efrei.fr`)

- Thomas Quinot (thomas@cuivre.fdn.fr)
- David Robert (david@anrtt.inp-fc.fr)
- J.M. Vansteene (J.M.Vansteene@frcl.bull.fr)

19 Si tu veux participer à Linux

Linux est un système ouvert, qui évolue. De nombreux développements sont actuellement en cours, tels que *Wine* (**WIN**dows **E**mulator), le projet de documentation de Linux, l'internationalisation. Tous ces projets (et bien d'autres) sont réalisés par des gens de bonne volonté qui donnent un peu de leur temps. Cela va de la programmation, à la traduction de documents, etc.

Si jamais tu veux donner un peu de ton temps à **Linux**, n'hésite surtout pas à me contacter.

20 That's all folks

Voila, c'est fini.

Si certains points te semblent obscurs, etc, n'hésite pas à me contacter. Ce guide est amené à évoluer, mais il ne peut évoluer que grâce à **toi**, donc si as trouvé un truc achtement génial, envoie-le: (Eric.Dumas@freenix.fr).

Mont de Marsan, le 14 Décembre 1996.