Bien débuter sous Linux

Linux Installation and Getting Started, par Matt Welsh, Version 2.2.2, 11 Février 1995. Adaptation francaise par René Cougnenc, Version 2.2.2, 20 Février 1995.

Cet ouvrage est un guide d'installation et d'initiation à Linux, destiné aux nouveaux utilisateurs de ce système d'exploitation, qu'ils soient novices ou expérimentés. Vous y trouverez toutes les informations nécessaires pour obtenir Linux, l'installer et le configurer, une initiation pour les nouveaux venus à UNIX, et une introduction à l'administration système.

Il est réalisé de telle sorte qu'il soit suffisamment général, pour s'appliquer à n'importe quelle distribution du système Linux.

La distribution de ce document est libre, vous pouvez le copier et le redistribuer sous certaines conditions, que vous trouverez à la page xvi.

Table des matières

Avant-propos

« *You are in a maze of twisty little passages, all alike.* » [1]

Vous êtes face au plus au plus fascinant système jamais réalisé. Linux, le clone gratuit d'UNIX pour les ordinateurs personnels, réalisé par une équipe disparate composée d'experts UNIX, de programmeurs géniaux, et d'informaticiens de base. Le système en lui-même reflète parfaitement cet héritage pour le moins complexe ; et bien que son développement soit l'œuvre de bénévoles sans réelle organisation, il s'agit d'un véritable système d'exploitation 32 bits, puissant, rapide, fiable, et gratuit.

Mes propres aventures avec Linux ont commencé il y a plusieurs années, en tentant de comprendre comment installer l'unique « distribution » disponible à cette époque : deux simples disquettes réalisées par H.J. Lu. Je téléchargeai alors un bon nombre de fichiers et entrepris la lecture d'un grand nombre de notes d'installation peu explicites et désorganisées. Au bout du compte, je réussis à faire fonctionner ce système de base, tout marchait correctement ! Tout cela se passait bien avant que vous ne puissiez vous procurer Linux sur CD-ROM dans le monde entier ; en fait, c'était bien avant que le noyau connaisse ce type de support. C'était avant Xfree86, avant Emacs, avant qu'il n'y ait un support commercial, avant que Linux ne devienne le rival de MS-DOS, Microsoft Windows et OS/2 dans le monde des ordinateurs personnels.

Vous tenez entre les mains l'ouvrage qui vous guidera pas-à-pas dans le monde de Linux. J'espère qu'il vous aidera à utiliser ce que je considère comme le système d'exploitation le plus rapide et universel disponible pour votre ordinateur personnel. Mettre en route votre propre système Linux est une expérience très enrichissante et peut vous faire vivre des moments passionnants. Servez-vous une tasse de café, asseyez-vous, et lisez-donc la suite. Pendant que vous y êtes, servez m'en aussi une tasse ! Je suis plongé dans Linux depuis des jours et des des jours... [2]

---

1. « Vous êtes dans un labyrinthe plein de petits passages tordus, qui se ressemblent tous... » L'auteur fait allusion à l'un des plus anciens jeux informatiques : Zork.
2. Le traducteur veut bien un café aussi, s'il vous en reste ; et s'il était accompagné de quelques digestifs, il n'en serait que plus apprécié !

À qui s'adresse ce livre ?

Cet ouvrage est destiné à tout possesseur d'ordinateur personnel qui désire installer et utiliser Linux. Nous considérerons que vous avez des notions de base sur les micro-ordinateurs et systèmes d'exploitations comme MS-DOS. Aucune connaissance préalable de Linux ou d'UNIX n'est nécessaire.

Malgré cela, nous recommandons fortement aux personnes ne connaissant pas UNIX d'investir dans l'un des très bons livres disponibles sur le sujet. (Plusieurs d'entre eux sont indiqués dans l'annexe A). Ce guide contient des informations utiles aux nouveaux venus à Linux, mais pour obtenir des informations plus détaillées, vous devrez chercher ailleurs !

Organisation

Ce livre contient les chapitres suivants.

Le chapitre 1, *Introduction à Linux*, explique ce qu'est Linux, ce qu'il peut vous apporter, et ce qui sera nécessaire pour qu'il fonctionne chez vous. Il contient aussi quelques trucs utiles pour obtenir de l'aide et éviter de trop s'énerver.

Le chapitre 2, *Se procurer et installer Linux*, explique comment et où se procurer le système Linux, et comment l'installer; depuis le partitionnement du disque dur jusqu'à la mise en route. Il ne contient que des informations générales, et devra être complété le cas échéant par la documentation particulière à votre distribution de Linux.

Le chapitre 3, *Didacticiel Linux*, constitue une initiation à l'utilisation de Linux pour les nouveaux venus à UNIX. Si vous avez déjà une expérience sous quelque système UNIX que ce soit, le contenu de ce chapitre devrait vous sembler évident.

Le chapitre 4, *Administration système*, présente les aspects les plus importants de l'administration d'une machine UNIX, sous Linux. Il intéressera également les administrateurs désirant prendre connaissance des spécificités Linux en ce domaine.

Le chapitre 5, *Techniques avancées*, présente au lecteur un certain nombre de caractéristiques, applications ou extensions puissantes de Linux, comme par exemple X Window et TCP/IP. Un guide complet de configuration de XFree83-3.1 est inclus.

L'annexe A, *Sources d'informations sur Linux*, recense les moyens d'obtenir des informations sur le système et de se tenir au courant de son évolution, par Usenet, listes de diffusion, documents à télécharger, et publications.

L'annexe B, *Distributeurs de Linux*, est une liste réduite des principaux distributeurs vendant Linux par correspondance ou offrant divers autres services.

L'annexe C, *Didacticiel FTP et liste des sites*, est une initiation au téléchargement de Linux sur l'Internet par le protocole FTP, et contient une liste des principaux sites de diffusion de Linux.

L'annexe D, *Liste des BBS Linux*, propose une liste approximative de différents BBS du monde entier, proposant linux en téléchargement. Elle est très incomplète mais rendra service aux utilisateurs n'ayant aucun accès à l'Internet.

L'annexe E, Licence Publique Générale GNU, présente une copie de la « GPL » , qui protège Linux. Les utilisateurs doivent bien comprendre les termes de cette licence, dont le but est de soutenir et protéger la liberté des logiciels sans léser les auteurs, mais qui paraît assez confuse au premier abord.

Remerciements

Ce livre fut très long à réaliser, et a pu voir le jour grâce aux efforts de nombreuses personnes. En particulier, je voudrais remercier Larry Greenfield et Karl Fogel pour leur travail sur la première version du chapitre 3, et Lars Wirzenius pour son travail sur le chapitre 4. Merci à Michael K. Johnson pour son aide concernant le projet et les conventions LaTeX utilisées dans ce manuel, et à Ed Chi, qui eut la gentillesse de m'adresser une copie imprimée de ce livre pour l'édition. Un merci tout particulier à Melinda A. McBride de SSC, Inc., qui compléta les index des chapîtres 3, 4, et 5. Je voudrais aussi remercier Andy Oram, Lar Kaufman, et Bill Hahn de O'Reilly and Associates pour leur aide et l'intérêt qu'ils portent au Projet de Documentation Linux.

Merci également à *Linux Systems Labs, Morse Telecommunications, Yggdrasil Computing*, et bien d'autres pour leur support de ce projet matérialisé par la vente de ces ouvrages ou divers autres services.

Et bien entendu, un grand merci à tous les « linux-activists » , y compris (dans le désordre) Linus Torvalds, Donald Becker, Alan Cox, Remy Card, Ted T'so, H.J. Lu, Ross Biro, Drew Eckhardt, Ed Carp, Eric Youngdale, Fred van Kempen, Steven Tweedie, et bien d'autres, pour passer autant de temps et dépenser autant d'énergie pour réaliser Linux. Sans eux, ce livre n'existerait pas.

Enfin, merci aux myriades de lecteurs qui m'ont expédié commentaires et corrections. Nul besoin de correcteur orthographique, avec autant de lecteurs !

Matt Welsh
13 Janvier 1994

Notes du traducteur

Cet ouvrage constitue l'adaptation française de la seconde édition du livre « Linux Installation and Getting Started » , par Matt Welsh. Par rapport à la version originale, certaines informations rendues obsolètes ou erronées par la rapide évolution de Linux, ont été remises à jour; certaines précisions spécifiques à la France ont dû être ajoutées à divers endroits afin que ce manuel conserve le caractère pratique désiré par l'auteur.

La traduction d'un ouvrage technique n'est pas chose aisée, particulièrement dans le domaine informatique où sévit un jargon plus ou moins international, d'origine anglo-saxonne. La tâche est encore plus complexe dans le cas de Linux, lorsque le traducteur pratique le système de longue date et doit résister à la tentation de compléter ou préciser certains points omis par l'auteur, voire renchérir sur certains traits d'humour !

En ce qui concerne les termes employés, il a été décidé d'utiliser leur équivalent français dans toute la mesure du possible, en fonction de ce qui est généralement admis pour la rédaction de documents informatiques écrits : de nombreux manuels, livres, publications, traductions ont été consultés, et lorsqu'un doute subsistait le courrier électronique s'est avéré fort utile pour demander l'avis de spécialistes en la matière. Peu importe que dans la vie courante l'informaticien français « boote sa bécane » ; il doit écrire qu'il « amorce son ordinateur » . Il est certain que ce manuel, pour un habitué des systèmes UNIX rompu à la lecture de milliers de lignes de documentations en anglais, peut par moments être déroutant, voire prêter à rire. J'en suis parfaitement conscient : la première fois que j'ai eu à utiliser une machine UNIX tournant sous un système (commercial) entièrement francisé, je suis vite repassé en langue anglaise car je ne comprenais rien du tout de ce français là, par exemple le message d'erreur « Rupture du tuyau » me semblait moins parlant que le traditionnel « Broken pipe » . Hélas !, conserver les expressions originales, comme on le ferait dans la langue parlée, est irréaliste. Un texte ainsi truffé de mots étrangers est insoutenable, autant ne pas le traduire du tout.

Le respect du style de l'auteur n'est pas une mince affaire non plus, car ce qui semble élégant et clair en anglais peut donner une impression fort différente pour quelqu'un de culture francophone, ce qui explique certaines parties semblant un peu maladroites. Malgré tout, j'espère que ce livre reste aussi compréhensible que l'a voulu l'auteur. Ne perdez pas de vue que nous sommes tous bénévoles, et faisons de notre mieux pour le bien de la communauté Linux.

René Cougnenc
2 Juillet 1994

Crédits et mentions légales

Le groupe de documentation Linux (*Linux Documentation Project*) est une équipe bénévole de rédacteurs, correcteurs, et éditeurs qui travaillent sur un jeu de manuels officiels pour Linux. Le coordinateur de ce projet est Matt Welsh, aidé par Lars Wirzenius et Michael K. Johnson.

Ce manuel n'est qu'un seul des nombreux autres qui sont distribués par le Linux Documentation Project, comprenant un guide utilisateur de Linux, un guide d'administration système, et un guide du noyau Linux. Ces manuels sont tous disponibles sous leur forme originale, en source LaTeX ou au format Postscript sur des sites FTP anonyme comme par exemple `sunsite.unc.edu`, dans le répertoire `/pub/Linux/docs/LDP`.

Nous encourageons quiconque ayant un penchant pour l'écriture ou l'édition, à rejoindre notre équipe afin d'améliorer la documentation de Linux. Si vous avez un accès au courrier électronique, vous pouvez souscrire au canal `DOC` de la liste de diffusion `Linux-Activists`, en écrivant à

```
linux-activists-request@niksula.hut.fi
```

avec la ligne

```
X-Mn-Admin: join DOC
```

comme toute première ligne du corps du message.

N'hésitez pas à contacter l'auteur et le coordinateur de ce manuel si vous avez des questions, des cartes postales, de l'argent ou des idées. Matt Welsh peut être contacté par courier électronique à l'adresse `mdw@sunsite.unc.edu`, ou par courrier postal:

205 Gray Street
Wilson, N.C. 27896
U.S.A.

205 Gray Street NE, Wilson NC, 27893 USA
`mdw@sunsite.unc.edu`

Des exceptions à ces règles peuvent être autorisées pour l'enseignement : écrivez à Matt Welsh à l'adresse postale ci-dessus, ou bien par courrier électronique adressé à `mdw@sunsite.unc.edu`, et demandez ce qu'il en est. Ces restrictions sont destinées à nous protéger en tant qu'auteurs, et non pas pour vous gêner en tant qu'enseignants ou élèves.

L'auteur encourage les distributeurs de Linux à utiliser ce livre comme guide d'installation et d'initiation. Le copyright ci-dessus vous autorise à imprimer et distribuer des copies de ce manuel avec votre distribution du système. Vous devrez néanmoins ajouter un petit supplément décrivant l'installation de votre distribution, car ce guide ne contient que des informations très générales sur le sujet.

L'auteur tient à être tenu au courant des projets commerciaux de publication et distributions de cet ouvrage. Vous pourrez ainsi être assuré d'avoir une version à jour en fonction de chaque nouvelle version; ceci peut vous permettre de retarder légèrement votre publication si une révision est en cours de réalisation.

Si vous commercialisez ce livre, l'auteur appréciera toute rétribution, don, et/ou copies imprimées. Ce type de contribution montrera votre intérêt pour le logiciel libre et le projet de documentation Linux.

Tout le code source de *Bien débuter sous Linux* est placé sous la protection de la Licence Publique Générale GNU. Vous trouverez une copie de cette « GPL » dans l'annexe E page 255.

Conventions typographiques

Ces conventions devraient vous paraître évidentes, mais nous les expliquons tout de même ici pour éviter d'éventuels reproches.

Gras Utilisé pour signaler des **nouveaux concepts**, **AVERTISSEMENTS**, et **mots-clés** dans un langage.

italiques Utilisées pour *insister* sur un texte, et occasionnellement pour des citations, ou de courtes introductions précédant une section. Egalement utilisées pour indiquer les commandes que l'utilisateur est censé taper lors d'un dialogue avec la machine (voir plus bas).

⟨*onglets*⟩ Utilisées pour les **meta-variables** dans le texte, en particulier pour représenter une ligne de commande. Par exemple,

`ls -l` ⟨*bidule*⟩

où ⟨*bidule*⟩ serait un nom de fichier, comme `/bin/cp`.

`Télétype` Utilisé pour représenter toute interaction entre l'utilisateur et l'écran, comme dans

```
$ ls -l /bin/cp
-rwxr-xr-x  1 root    wheel    12104 Sep 25 15:53 /bin/cp
```

Aussi utilisé pour les exemples de code, que ce soit du langage C, du shell-script, de l'assembleur ou autres, et pour les fichiers texte, comme les fichiers de configuration. Ces exemples pourront éventuellement être encadrés pour clarifier la présentation.

$\boxed{\texttt{Touche}}$ Représente une touche à presser. Vous la verrez souvent sous cette forme:

Taper $\boxed{\text{entrée}}$ pour continuer.

◇ Un losange dans la marge signifie « danger » ou « attention. » Lisez très attentivement les paragraphes signalés de cette façon.

Chapitre 1

Introduction à Linux

LINUX est probablement la réalisation la plus importante parmi les graticiels depuis *Space War*, ou, plus récemment, Emacs. Linux est devenu un système d'exploitation pour l'industrie, l'éducation ou l'utilisation personnelle. Il n'est plus destiné aux *wizards* UNIX qui passent des heures devant leur écran (bien que nous puissions vous assurer qu'un bon nombre d'utilisateurs font partie de cette catégorie). Cet ouvrage vous aidera à obtenir le maximum de Linux.

Linux (se prononce en Finlande comme en français le plus naturellement du monde) est un clone du système d'exploitation UNIX, destiné aux ordinateurs équipés de processeurs Intel 80386 et 80486. Il supporte une large gamme de programmes, de TEX à X-Window, le compilateur C/C++ GNU et TCP/IP. C'est une implémentation multi-usages d'UNIX, distribuée gratuitement selon la licence générale GNU (voir annexe E).

Linux peut transformer n'importe quel PC 386 ou 486 en station de travail. Il vous apportera toute la puissance d'UNIX. Des sociétés commerciales installent des résaux entiers de machines tournant sous Linux, destinés à gérer des données financières ou médicales, des télécommunications, et bien d'autres choses. Dans le monde entier, des universités utilisent Linux pour l'étude des systèmes d'exploitation, des passionnés d'informatique utilisent Linux chez eux pour programmer, produire, etc...

Ce qui rend Linux si différent est qu'il est *gratuit*. Il est développé par un groupe de volontaires, au départ sur l'Internet, s'échangeant du code, notifiant les bogues, et corrigeant les problèmes selon un modèle de développement ouvert à tous. Tout le monde est invité à participer : il suffit de posséder de bonnes bases de programmation et de se passionner pour les entrailles d'un clone de système UNIX. Le livre que vous avez entre les mains est votre dépliant touristique.

1.1 À propos de ce livre

Cet ouvrage est un guide d'installation, et d'initiation pour les débutants, au système Linux. Son but est de permettre aux nouveaux utilisateurs d'utiliser Linux, en rassemblant le plus possible d'informations importantes en un seul volume. Au lieu de décrire de nombreux petits détails techniques susceptibles de changer au fil des versions, nous vous offrons ici une base solide qui vous permettra de découvrir le reste par vous même.

Linux n'est pas difficile à installer et à utiliser. Toutefois, comme avec toute implémentation d'UNIX, obtenir un système qui fonctionne correctement semble parfois relever de la sorcellerie... Nous espérons que ce guide vous permettra de découvrir et maîtriser Linux, et vous montrera la puissance et l'universalité de ce système d'exploitation.

Nous aborderons les sujets suivants :

- Qu'est-ce que Linux ? La philosophie et l'organisation de ce système d'exploitation unique, et ce qu'il peut vous apporter.
- Tous les détails nécessaires pour exploiter Linux, avec quelques suggestions sur la configuration matérielle recommandée pour un système complet.
- Comment obtenir et installer Linux. Il en existe de nombreuses distributions. Nous en présentons quelques unes, et expliquons comment installer le système à selon une méthode appliquable de manière générale.

  Cette version du livre contient étalement des instructions spécifiques à l'installation de la distribution Slackware de Linux.
- Une brève initiation à UNIX, pour les utilisateurs qui n'ont aucune expérience de ce système. Cette introduction devrait apporter suffisamment de connaissances de base aux novices pour qu'ils puissent utiliser UNIX sans être totalement perdus.
- Une introduction à l'administration système, sous Linux. Elle couvre les tâches les plus importantes que les nouveaux administrateurs Linux devront maîtriser, comme la création de nouveaux utilisateurs, la gestion des systèmes de fichiers, etc...
- Un aperçu des possibilités avancées de Linux, comme le système X Window, le réseau avec TCP/IP et SLIP, et la configuration du courrier électronique et des News Usenet.

Cet ouvrage s'adresse au possesseur de micro-ordinateur personnel désirant débuter sous Linux. Nous considérons qu'il n'a aucune expérience d'UNIX, mais partons du principe qu'il se référera éventuellement à d'autres ouvrages plus spécialisés s'il en ressent le besoin. Une liste de sources d'informations utiles est donnée dans l'annexe A. D'une manière générale, ce livre est fait pour être lu en ayant à côté un autre ouvrage sur les concepts de base d'UNIX.

1.2 Bref historique de Linux

UNIX est l'un des systèmes d'exploitation les plus populaires au monde, en raison du grand nombre d'architectures qu'il supporte. Il fut à l'origine développé en tant que système d'exploitation multitâches pour mini-ordinateurs et grands systèmes vers l'année 1970, mais a bien évolué depuis pour devenir l'un des systèmes les plus utilisés, en dépit de son interface parfois déroutante et de son manque de standardisation réelle.

La raison de la popularité d'UNIX ? Beaucoup de programmeurs le ressentent comme La Vérité ; Le Vrai Système d'Exploitation, presque de manière religieuse. D'où le développement de Linux, par un groupe évolutif de programmeurs UNIX désirant mettre les mains dans le cambouis et réaliser leur propre système.

Il existe des versions d'UNIX pour nombre d'ordinateurs, depuis les ordinateurs personnels jusqu'aux grosses machines comme le Cray Y-MP. La plupart de ses implémentations pour PC sont lourdes et onéreuses. A l'heure ou nous écrivons ces lignes, une version pour une seule machine de l'UNIX System V de AT&T pour i386 coûte environ US$1500.

Linux est une version d'UNIX gratuite et librement diffusable développée à l'origine par Linus Torvalds [1] à l'université de Helsinki, en Finlande. Linux a été développé avec l'aide de nombreux programmeurs et spécialistes UNIX, grâce au réseau mondial Internet, autorisant quiconque ayant suffisamment de connaissances à participer activement à l'évolution du système. Le noyau de Linux n'utilise aucun code en provenance de AT&T ou de quelque autre source propriétaire, et la plupart des programmes disponibles pour Linux est développée par le projet GNU à la Free Software Foundation à Cambrige, Massachusetts. Toutefois, des programmeurs du monde entier ont contribué à l'ensemble.

Linux était au départ un projet de loisirs de Linus Torvalds. Il fut inspiré de Minix, un petit système UNIX développé par Andy Tanenbaum, et les premières discussions à propos de Linux eurent lieu dans le forum USENET `comp.os.minix`. Ces discussions portaient principalement sur le développement d'un petit systeme UNIX académique pour les utilisateurs de MINIX qui désiraient mieux que cela.

Les prémices du développement de linux furent la maîtrise de la commutation de tâches en mode protégé sur le processeur 80386. Tout fut réalisé en assembleur. Linus écrit :

> « Après ça, tout coulait de source : encore de la programmation touffue, mais j'avais quelques périphériques, et le déboguage était plus

1. `torvalds@kruuna.helsinki.fi`.

> facile. C'est à ce stade que j'ai commencé à utiliser le langage C, ce qui a certainement accéléré le développement. C'est aussi à ce moment que j'ai commencé à prendre au sérieux mes idées mégalo de faire un "Minix meilleur que Minix". J'espérais un jour pouvoir recompiler `gcc` sous Linux... »
>
> « Deux mois pour le code de base, puis un peu plus jusqu'à ce que j'aie un pilote de disque dur (sérieusement bogué, mais par chance il fonctionnait sur ma machine), et un petit système de fichiers. C'est à cette époque que j'ai diffusé la version 0.01 [fin août 1991] : Ce n'était pas très beau, je n'avais pas de pilote de disquette, et ça ne pouvait pas faire grand chose. Je ne pense pas que quelqu'un ait un jour compilé cette version. Mais j'étais pris au jeu, et je ne voulais plus m'arrêter tant que je ne pouvais pas jeter Minix aux orties. »

Aucune annonce de Linux version 0.01 ne fut jamais faite. Ce n'était même pas exécutable ; l'archive ne contenait que les rudiments des sources du noyau, et considérait que vous aviez accès à un système Minix pour compiler Linux et jouer un peu avec.

Le 5 octobre 1991, Linus annonça la toute première version « officielle » de Linux, la version 0.02. A ce moment, Linux était capable d'éxéxuter `bash` (le Bourne Again Shell de GNU), mais pas grand chose d'autre. Encore une fois, c'était un système destiné aux hackers, focalisé sur le développement du noyau. Le support utilisateurs, la documentation, la distribution, ou autres, n'avaient jamais été évoqués. Aujourd'hui, la communauté Linux semble continuer à traiter ces choses là comme très secondaires en comparaison de la « vraie programmation » , le développement du noyau.

Linus écrit dans `comp.os.minix`,

> « Vous regrettez les beaux jours de Minix-1.1, lorsque les hommes étaient des hommes et écrivaient leurs propres pilotes de périphériques ? Vous manquez d'un superbe projet et vous languissez d'un système que vous pourriez modifier à votre convenance ? Vous êtes frustrés que tout fonctionne sous Minix ? Plus de nuits blanches passées à tenter de faire fonctionner un programme récalcitrant ? Alors ce message pourrait bien être pour vous. »
>
> « Comme signalé il y a un mois, je travaille actuellement sur une version libre et gratuite d'un système ressemblant à Minix pour les ordinateurs AT-386. J'ai finalement atteint un stade où il est utilisable (bien qu'il puisse ne pas l'être pour vous, selon ce que vous recherchez), et je compte soumettre les sources à une diffusion plus large. Il s'agit juste de la version 0.02...mais j'ai pu exécuter `bash`, `gcc`, `gnu-make`, `gnu-sed`, `compress`, etc. avec succès sous ce système. »

Après la version 0.03, Linus passa le numéro de version directement à 0.10, puisque de plus en plus de personnes commencèrent à travailler sur le système. Puis, après plusieurs autres révisions, Linus gonfla à nouveau le numéro pour sortir la version 0.95, afin de refléter son impression : Linux était prêt pour une version « officielle » très prochainement. (Généralement, un programme ne se voit attribuer le numéro de version 1.0 que lorsqu'il est théoriquement complet, ou sans bogue). Ceci se passait au mois de mars 1992. Presque un an et demi plus tard, fin décembre 1993, le noyau de Linux en était encore à la version 0.99.pl14... Approchant 1.0 de manière asymptotique. Certains pensaient que la version 1.0 ne verrait jamais le jour. Á l'heure où nous écrivons ces lignes, Linux en est à la version 1.1.94, et la sortie de la 1.2 est imminente.

Aujourd'hui, Linux est un clone d'UNIX complet, capable d'exécuter X Window, TCP/IP, Emacs, UUCP, le courrier électronique et les news Usenet, ou tout ce que vous voudrez. Pratiquement tout les programmes freewares importants ont été portés sous Linux, et on commence à voir apparaître des applications commerciales. Linux supporte beaucoup plus de périphériques que dans ses premières versions. Beaucoup de personnes ont effectué des tests de machines 80486 sous Linux et ont trouvé des performances comparables aux stations de travail de milieu de gamme de Sun Microsystems et Digital Equipment Corporation. Qui aurait pu imaginer qu'un jour, ce « petit » système d'exploitation serait devenu si grand ?

1.3 Caractéristiques du système

Linux offre la plupart des possibilités offertes par d'autres implémentations d'UNIX, plus quelques unes que l'on ne trouve nulle part ailleurs. Cette section fait un tour d'horizon des caractéristiques du noyau de Linux.

Linux est un système d'exploitation multi-tâches et multi-utilisateurs (exactement comme toute autre version d'UNIX). Cela signifie que plusieurs utilisateurs peuvent utiliser la même machine en même temps, tout en utilisant simultanément de nombreux programmes tous différents.

Le système Linux est compatible avec un certain nombre de standards UNIX (si tant est qu'UNIX ait des standards) au niveau code source, incluant à la fois IEEE POSIX.1, System V, et BSD. Il a été développé en ayant toujours en tête la portabilité au niveau source : par conséquent, vous trouverez sous Linux les caractéristiques les plus courantes que l'on rencontre sur diverses implémentations. Un très grand nombre d'applications UNIX gratuites disponibles sur l'Internet ou ailleurs se compilent sous Linux sans aucune modification. De plus, tout le code source du système, y compris le noyau, les pilotes, bibliothèques, programmes utilisateurs, et outils de développement, est gratuit et librement distribuable.

Linux comprend d'autres caractéristiques spécifiques comme le contrôle de processus POSIX (utilisé par des shells comme `csh` et `bash`), pseudo-terminaux (périphériques `pty`), et le support de claviers internationaux ou personnalisés modifiables dynamiquement. Linux supporte également des **consoles virtuelles**, qui vous permettent de commuter le clavier et l'écran entre plusieurs sessions de travail en mode texte. Les utilisateurs du programme « `screen` » seront très à l'aise avec l'implémentation des consoles virtuelles de Linux.

Le noyau étant capable d'émuler un co-processeur mathématique (387-FPU), les machines qui en sont dépourvues peuvent tout de même utiliser de manière totalement transparente des programmes qui utilisent ce co-processeur.

Linux possède différents types de systèmes de fichiers pour stocker les données. Certains, comme *ext2fs*, ont été développés spécialement pour Linux. D'autres types de systèmes de fichiers, comme Minix-1 et Xenix, sont aussi supportés. Le système de fichiers MS-DOS a été également implémenté, pour vous permettre d'accéder directement à vos disques MS-DOS depuis Linux, en montant une partition ou une disquette le plus naturellement du monde. Le type CD-ROM ISO 9660, qui permet de lire tous les formats standard de CD-ROM, est aussi supporté. Nous parlerons plus en détail des systèmes de fichiers dans les chapitres 2, et 4.

Linux fournit une implémentation complète du réseau TCP/IP. Ceci comprend les pilotes de périphériques pour beaucoup de cartes Ethernet populaires, SLIP (*Serial Line Internet Protocol*) et PPP (*Point to Point Protocol*), qui vous permettent d'accéder à un réseau TCP/IP depuis une connexion série, avec un modem par exemple) ; PLIP (*Parallel Line Internet Protocol*), permettant le réseau par l'interface imprimante parallèle ; NFS (*Network File System*), et bien d'autres choses. La totalité des clients et services TCP/IP est supportée, comme FTP, **`telnet`**, NNTP et SMTP par exemple. Nous parlerons du réseau dans le chapitre 5.

Le noyau de Linux utilise les possibilités du mode protégé des processeurs Intel 80386 et 80486. En particulier, Linux utilise les primitives de gestion mémoire et de tâches, ainsi que d'autres caractéristiques techniques de ces processeurs. Toute personne familière avec la programmation en mode protégé du 80386 sait que ce processeur fut réalisé pour les systèmes multi-tâches comme UNIX (ou, en l'occurrence, Multics). Linux exploite ceci avec bonheur.

Le noyau Linux supporte le chargement à la demande des exécutables. C'est à dire que seuls les segments d'un programme qui sont réellement utilisés sont lus et chargés en mémoire. Linux utilise aussi la technique de partage des pages mémoire avec copie à l'écriture, ce qui signifie que si plusieurs instances d'un même programme sont utilisées en même temps, elle partageront leurs pages de mémoire physique, ce qui réduit notablement la mémoire globalement utilisée.

Afin d'augmenter la mémoire disponible, Linux implémente également la pagination sur disque : Jusqu'à 256 Mégaoctets d' « espace de swap »[2] peuvent être alloués sur le disque dur. Lorsque le système a besoin de plus de mémoire physique, il transfère les pages inactives sur le disque, vous permettant ainsi de pouvoir utiliser de plus grosses applications ou de supporter plus d'utilisateurs à la fois. Mais

2. Le mot espace de swap est inapproprié ici : Les processus ne sont jamais swappés en entier, ce sont des pages mémoire qui le sont, individuellement. Bien sûr, dans beaucoup de cas des processus finiront par être entièrement swappés sur le disque, mais ce n'est pas forcément le cas à chaque fois.

attention, la zone de swap ne remplace pas la mémoire physique (RAM), c'est bien entendu beaucoup, beaucoup plus lent !

Le noyau utilise aussi une réserve de mémoire dynamique, commune aux programmes utilisateurs et au cache disque. De cette façon, Linux utilise la totalité de la mémoire RAM dont vous disposez pour le cache disque, celui-ci étant réduit ou augmenté en fonction des besoins en mémoire des programmes.

Les exécutables utilisent des bibliothèques partagées, ce qui signifie que les binaires partagent le code commun aux fonctions bibliothèque, si 50 programmes utilisent printf(), le code de printf() ne sera présent qu'une seule fois en mémoire. Ceci permet de réduire considérablement l'espace occupé par les programmes. Il existe aussi bien entendu la possibilité d'utiliser des bibliothèques statiques pour ceux qui ont besoin de déboguer un programme ou qui veulent conserver un exécutable « complet » , pouvant s'exécuter en l'absence des fichiers de bibliothèques partagées. Les bibliothèques partagées de Linux sont liées dynamiquement à l'exécution, autorisant le programmeur à remplacer certains modules des bibliothèques par ses propres routines s'il le désire.

Afin de faciliter le déboguage, le noyau de Linux génère des *core dumps* (sauvegarde d'image mémoire) à des fins d'analyse post-mortem des programmes. A l'aide d'un core-dump et d'un exécutable compilé en mode déboguage, il est possible de déterminer la cause de l'erreur qui a provoqué la fin prématurée du processus.

1.4 Applications

Dans cette section, nous allons vous présenter un certain nombre d'applications disponibles sous Linux, et discuter des travaux les plus couramment effectués à l'aide d'un ordinateur. Après tout, la partie la plus importante du système, c'est le grand nombre de programmes qui y sont disponibles. Le fait que l'essentiel de ces programmes soit gratuit et librement distribuable est encore plus impressionnant.

1.4.1 Commandes de base et utilitaires

Pratiquement chaque utilitaire que vous pouvez espérer trouver sur une implémentation standard d'UNIX a été porté sous Linux. Ceci comprend bien entendu les commandes de base comme `ls`, `awk`, `tr`, `sed`, `bc`, `more`, et ainsi de suite. Dites un nom au hasard, la commande existe sous Linux. Par conséquent vous pouvez espérer retrouver votre environnement de travail favori sous UNIX, reproduit à l'identique sur votre système Linux. Toutes les commandes standards et les utilitaires habituels sont là. (Les nouveaux utilisateurs peuvent se reporter au chapitre 3 pour une introduction à ces commandes de base UNIX).

De nombreux éditeurs de texte sont disponibles, dont `vi`, `ex`, `pico`, `jove`, aussi bien que GNU Emacs et ses variantes comme Lucid Emacs (avec ses extensions pour l'utilisation sous X Window) et `joe`. Quel que soit l'éditeur que vous ayez l'habitude d'utiliser sous UNIX, il y a de grandes chances pour que vous le retrouviez sous Linux.

Le choix d'un éditeur de texte est quelque chose d'intéressant. Beaucoup d'utilisateurs chevronnés d'UNIX utilisent toujours des éditeurs « simples » comme `vi` (en fait, l'auteur a écrit ce livre à l'aide de `vi` sous Linux, et le traducteur également). Toutefois `vi` a beaucoup de limitations, en raison de son âge, et des éditeurs plus modernes (et plus complexes) comme Emacs deviennent de plus en plus populaires.

Emacs supporte un interpréteur de macro-langage basé sur LISP, une syntaxe de commande très puissante, et bien d'autres extensions originales.

Il existe des paquetages de macros Emacs pour lire directement le courrier électronique et les news Usenet, éditer le contenu de sous-répertoires, et même engager une conversation psychothérapique utilisant l'intelligence artificielle. (Indispensable pour les programmeurs Linux à bouts de nerfs).

Il est important de noter que l'essentiel des utilitaires de base de Linux sont des programmes GNU. Ces utilitaires GNU comprennent des extensions très utiles aux commandes UNIX, par rapport aux versions standard de BSD ou AT&T. Par exemple, la version GNU de l'éditeur de texte `vi`, nommée `elvis`, comprend un macro-langage structuré différent de l'original implémenté par AT&T. Bien entendu les utilitaires GNU restent compatibles avec leurs équivalents BSD et System V. Beaucoup de gens considèrent les versions GNU de ces programmes comme bien supérieures aux versions originales et les compilent sur leurs systèmes UNIX commerciaux pour remplacer les commandes d'origine.

L'utilitaire le plus important pour beaucoup d'utilisateurs est l'interpréteur de commandes, que l'on appelle le **shell**. Le shell est un programme qui lit et exécute les commmandes tapées au clavier par l'utilisateur. De plus, plusieurs shells proposent des facilités supplémentaires comme le **contrôle de processus** (permettant à l'utilisateur de gérer plusieurs processus à la fois sur un même terminal), redirections des entrées et sorties, et un langage de commande destiné à écrire des **shell scripts**. Un shell-script est un fichier texte contenant un programme dans le langage compris par le shell, analogue à un « fichier batch » sous MS-DOS (mais en beaucoup plus puissant).

Différents types de shells sont disponibles sous Linux. La différence la plus importante entre les shells est leur langage de commande. Par exemple, le **C Shell** (`csh`) utilise un langage ressemblant au langage C. Le classique **Bourne Shell** utilise un tout autre style de langage de commande. Le choix personnel d'un shell est souvent basé sur le type de langage qu'il utilise. Le shell que vous utilisez définit pour partie votre environnement de travail sous Linux.

Peu importe le shell dont vous avez l'habitude, une de ses versions a probablement été portée sous Linux. Le plus populaire est le GNU Bourne Again Shell (`bash`), une variante du Bourne shell qui apporte énormément de possibilités supplémentaires, comme le contrôle de processus, la complétion automatique des noms de fichiers, un historique des commandes, une interface de type Emacs pour éditer la ligne de commande, et des extensions puissantes au langage de commande du Bourne shell standard. Un autre shell très populaire est `tcsh`, une version du C Shell offrant des possibilités supplémentaires similaires à celles trouvées dans `bash`. Citons d'autres shells comme `zsh`, un petit Bourne-shell ; le Korn shell (`ksh`) ; le `ash` de BSD ; et `rc`, le shell Plan 9.

Qu'est-ce qui est si important dans tout cela ? Linux vous donne une chance unique, celle d'ajuster exactement votre système à vos besoins. Par exemple, si vous êtes le seul utilisateur de votre machine, et que vous préférez utiliser exclusivement l'éditeur `vi` et le shell `bash`, il n'y a aucune raison pour que vous installiez d'autres éditeurs de texte ou d'autres shells. Le « sur mesures » est présent à tous les niveaux dans Linux.

1.4.2 Formatage de texte et traitement de texte

Pratiquement chaque utilisateur d'ordinateur a besoin d'outils de rédaction de documents. (Combien de passionnés d'informatique utilisant encore un stylo et du papier ? Pas beaucoup je parie...) Dans le monde PC, la norme est au *traitement de texte*: Cela sous-entend l'édition et la manipulation de texte (souvent dans un environnement « What-You-See-Is-What-You-Get ») et la réalisation de copies imprimées du document, comprenant des illustrations, des tableaux, et d'autres garnitures du genre.

Dans le monde UNIX, le *formatage de texte* est quelque chose de beaucoup plus commun, et c'est assez différent du concept classique du traitement de texte en bureautique par exemple.

Avec un formateur de texte, le texte est saisi par l'auteur en utilisant un « marquage » qui décrit comment le texte devra être présenté. Au lieu d'être obligé de taper le texte sous un environnement spécial de traitement de texte, le source peut être modifié avec n'importe quel éditeur comme `vi` ou Emacs. Une fois le texte source (incluant le marquage) saisi, l'utilisateur formate le texte avec un programme séparé, qui le convertit dans un format propre à l'impression.

Cette démarche est assez analogue à la programmation dans un langage comme C, suivi de la « compilation » du document dans une forme imprimée. Il y a beaucoup de formateurs de texte disponibles sous Linux. L'un deux est `groff`, la version GNU du classique formateur `nroff`, développé à l'origine aux Bell Laboratories et toujours utilisé sur la plupart des systèmes UNIX du monde entier. Un autre formateur de texte moderne fonctionnant parfaitement sous Linux est TEX, développé par Donald Knuth. Les dialectes de TEX, comme LATEX, sont également disponibles.

Les processeurs de texte comme TEX et `groff` diffèrent principalement dans la syntaxe et la puissance de leur langage de formatage. Le choix d'un formateur se fait aussi en fonction du nombre d'utilitaires disponibles satisfaisant vos besoins, et bien sûr du goût de chacun.

Par exemple, certains considèrent le langage de formatage de `groff` comme particulièrement obscur, alors ils utilisent TEX, qui est plus lisible pour un humain normalement constitué.

Toutefois, `groff` est capable de produire une sortie ASCII, lisible sur un terminal texte, alors que TEX est avant tout destiné à produire du code pilotant une imprimante. Bien sûr, il existe tout de même des programmes pemettant de traduire des documents TEX en ASCII, ou de convertir du code TEX en `groff`, par exemple.

`texinfo` est un autre formateur de texte, une extension à TEX utilisé pour documenter les programmes de la Free Software Foundation. `texinfo` est capable de produire aussi bien un document imprimé, qu'un document hypertexte « Info » à partir d'un unique fichier source. Les fichiers Info sont le format principal de documentation utilisé pour les programmes GNU comme Emacs.

Les processeurs de texte sont énormément utilisés dans la communauté informatique et scientifique pour produire des articles, thèses, magazines, et des livres (en fait, cet ouvrage a été réalisé avec LATEX). La possibilité de traiter le langage en un fichier purement texte ouvre la porte à de nombreuses extensions au processeur de texte lui-même. Comme les documents sources ne sont pas stockés dans un format complexe, lisible uniquement par un traitement de texte particulier, les programmeurs peuvent écrire des analyseurs et transcodeurs du language de formatage afin d'étendre le système à l'infini.

A quoi ressemble un tel langage de formatage? En règle générale, le code source contient essentiellement le texte lui-même, avec des « codes de contrôle » ici et là, destinés à produire un effet particulier comme un changement de fonte, un alignement de marges, une création de liste, etc...

A titre d'exemple, prenez le texte suivant:

> Monsieur Torvalds :
>
> Nous ne sommes pas d'accord sur vos projets concernant l'implémentation de *messages subliminaux* dans le code des contrôleurs de terminaux de **Linux**. Ceci pour trois raisons :
>
> 1. Ce type de message n'est pas seulement immoral, c'est une perte de temps ;
> 2. Il a été prouvé que ces « suggestions hypnotiques » sont sans effet sur un bon programmeur UNIX ;
> 3. Nous avons déjà implémenté par mesure de sécurité, des chocs électriques de haute tension dans le code de `/bin/login`.
>
> Nous espérons que vous allez reconsidérer cette idée.

Ce texte apparaîtrait dans le langage de formatage LATEX sous la forme suivante:

```
\begin{quote}
Monsieur Torvalds :

Nous ne sommes pas d'accord sur vos projets concernant l'implémentation de
{\em messages subliminaux \/} dans le code des contrôleurs de terminaux
de {\bf Linux}.
Ceci pour trois raisons :
\begin{enumerate}
\item Ce type de message n'est pas seulement immoral, c'est une perte
      de temps ;
\item Il a été prouvé que ces << suggestions hypnotiques >>  sont
      sans effet sur un bon programmeur UNIX ;
\item Nous avons déjà implémenté par mesure de sécurité, des
      chocs  électriques de haute tension dans le code de {\tt /bin/login}.
\end{enumerate}
Nous espérons que vous allez reconsidérer cette idée.
\end{quote}
```

L'auteur saisit le « source » ci-dessus en utilisant n'importe quel éditeur de texte, et génère la sortie formatée en passant ce source dans le programme LaTeX.

Au premier abord, le langage peut paraître assez obscur, mais en réalité il est très facile à apprendre. L'utilisation d'un système de formatage de texte force l'utilisation de standards typographiques de manière transparente lors de la rédaction.

Par exemple, toutes les listes d'énumération à l'intérieur d'un document auront la même allure, sauf si l'auteur en modifie la définition. Le but premier est d'autoriser l'auteur à se concentrer sur la rédaction du texte, plutôt que de passer son temps avec les conventions typographiques.

Les traitements de texte WYSIWYG sont attirants pour beaucoup de raisons; ils offrent une interface graphique puissante (et souvent complexe) pour l'édition du document. Malheureusement cette interface est limitée aux aspects de la maquette accessibles à l'utilisateur. Par exemple, beaucoup de traitements de texte offrent un « langage de formatage » spécial destiné à pouvoir produire des expressions complexes comme les formules mathématiques. Ceci est tout à fait identique au formatage de texte, mais sur une échelle beaucoup plus réduite.

L'intérêt du formateur de texte est qu'il vous permet de spécifier exactement ce que vous désirez. De plus, cette méthode vous permet de réaliser le code source avec n'importe quel éditeur, voire sur n'importe quel ordinateur équipé d'un système différent du vôtre, et ce source est très facile à convertir vers d'autres formats. La seule contrepartie à cette puissance et cette souplesse est le manque d'interface WYSIWYG.

Beaucoup d'utilisateurs de traitement de texte ont l'habitude de voir le texte formaté à peu près comme il apparaîtra une fois imprimé pendant qu'ils le saisissent. D'un autre côté, lors de l'écriture pour un formateur de texte, on ne se soucie généralement pas de l'apparence que prendra le travail une fois terminé. L'auteur apprend à apprécier la future présentation de son travail à partir des commandes qu'il utilise dans le code source.

De toutes façons il existe des programmes permettant de visualiser le document dans sa forme définitive avant l'impression, sur un écran graphique. Par exemple, le programme `xdvi` affiche un fichier généré par TeX sous l'environnement X Window, sous la forme exacte qu'il aura après impression. D'autres applications comme `xfig`, fournissent une interface WYSIWYG pour le dessin d'illustrations ou de shémas, qui seront ensuite convertis dans le langage de formatage utilisé pour être inclus dans le document.

Il faut reconnaître que les processeurs de texte comme `nroff` existent depuis bien plus longtemps que les traitements de texte de bureautique. Et beaucoup de personnes continuent à préférer le formatage de texte, parce qu'il est beaucoup plus souple, universel et indépendant de tout environnement graphique ou système d'exploitation. Quoi qu'il en soit, le traitement de texte `idoc` est disponible sous Linux, pour les inconditionnels, en attendant de voir fleurir des traitements de texte commerciaux utilisables sous notre système favori. Si vous ne voulez vraiment pas abandonner le traitement de texte au profit du formatage de texte, vous pouvez toujours utiliser MS-DOS, ou quelque autre système d'exploitation, en alternance avec Linux. Toutefois, vous devez savoir que la plupart des publications techniques ou scientifiques que vous avez coutume de dévorer, sont réalisées avec des formateurs de texte, et que TeX existe aussi en version MS-DOS et y est très utilisé.

Il y a d'autres utilitaires relatifs au formatage de texte disponibles sous Linux. Le puissant système METAFONT utilisé pour construire des fontes pour TeX, est inclus avec le portage Linux de TeX. D'autres programmes comprennent `ispell`, un correcteur orthographique interactif ; `makeindex`, utilisé pour générer des index dans les documents LaTeX ; aussi bien que de nombreux paquetages de macros `groff` et TeX destinés à mettre en forme différents types de documents et textes mathématiques. Des programmes de conversion de sources TeX ou `groff` vers des myriades d'autres formats sont disponibles.

1.4.3 Langages et outils de programmation

Linux offre un environnement de développement UNIX complet, comprenant toutes les bibliothèques standard, outils de programmation, compilateurs, débogueurs, que vous êtes en droit de trouver sur tout système UNIX. Dans le monde du développement UNIX, la programmation est généralement faite en C ou C++. Le compilateur C et C++ standard sous Linux est le GNU `gcc`, qui est un compilateur moderne et performant offrant de multiples options. Il est également capable de compiler du code C++ (y compris AT&T 3.0) aussi bien que du code Objective-C, un autre dialecte orienté objet du langage C.

À côté de C et C++, beaucoup d'autres langages compilés ou interprétés ont été portés sous Linux, comme Smalltalk, FORTRAN, Pascal, LISP, Scheme, et Ada (si vous êtes assez masochiste pour programmer en Ada, nous ne vous en empêcherons pas). En plus de tout cela, différents assembleurs pour le mode protégé 80386 ou le mode réel sont disponibles, de même que les langages favoris des experts UNIX comme Perl (le langage script qui élimine tous les langages scripts) et Tcl/Tk (un interpréteur de commandes permettant de développer de petites applications X Window).

Le débogueur `gdb` est bien sûr disponible, il vous permet de faire tourner un programme pas à pas pour détecter les bogues, ou examiner la raison d'un plantage a l'aide d'un core-dump. `gprof`, un profiler, vous donnera des statistiques sur les performances de votre programme, vous montrant à quels endroits votre code perd le plus de temps à l'exécution. L'éditeur de texte Emacs vous offrira un environnement interactif de développement pour différents langages de programmation. Parmi les nombreux autres outils on peut citer bien sûr le GNU `make` et `imake`, permettant de gérer la compilation de gros projets, et RCS, un système de contrôle de version de fichiers source pour les gros projets demandant plusieurs développeurs.

Linux utilise des bibliothèques partagées chargées dynamiquement, ce qui réduit considérablement la taille des exécutables puisque les fonctions bibliothèques sont incluses uniquement au moment du chargement en mémoire lors de l'exécution. Ces bibliothèques DLL permettent aussi au programmeur de remplacer les fonctions qu'il désire par son propre code. Par exemple, si un développeur a besoin d'écrire sa propre version de la fonction standard `malloc()`, l'éditeur de liens utilisera la fonction qu'il a spécialement réalisée plutôt que celle trouvée dans les bibliothèques Linux.

Linux est idéal pour le développement d'applications UNIX. Il offre un environnement de programmation moderne, avec tous les gadgets possibles. Différents standards comme POSIX.1 sont supportés, permettant aux programmes écrits sous Linux d'être très facilement portés vers d'autres systèmes. Les développeurs UNIX professionnels et les administrateurs système peuvent utiliser Linux pour développer des applications chez eux, et ensuite transférer le programme à leur travail sur d'autres systèmes UNIX. Ceci permet non seulement de substantielles économies d'argent, mais également de travailler avec un confort maximum, tranquillement chez vous, et sur un système où vous êtes le seul utilisateur[3]. Les étudiants peuvent utiliser Linux pour apprendre la programmation UNIX et explorer d'autres aspects du système, comme l'architecture du noyau.

Avec Linux, vous avez non seulement accès au jeu complet de bibliothèques standards et d'outils de programmation, mais aussi à l'intégralité du code source du noyau et des bibliothèques. (Imaginez un peu toutes les bêtises que vous allez pouvoir faire avec tout ça !)

3. L'auteur utilise son système Linux pour développer et tester des applications X Window chez lui, qui sont ensuite compilées sur des stations de travail partout ailleurs.

1.4.4 Le système X Window

Le système X Window est l'interface graphique standard des systèmes UNIX. C'est un environnement très puissant supportant de nombreuses applications. Avec X Window, l'utilisateur peut avoir plusieurs terminaux graphiques simultanément à l'écran, chacun contenant un programme différent ou une session utilisateur différente. La souris est très utilisée sous l'interface X, bien qu'elle puisse ne pas être indispensable.

Beaucoup d'applications pour X Window ont été écrites, y compris des jeux, des utilitaires graphiques, des outils de programmation et de documentation, et bien d'autres. Linux associé à X Window transforme votre machine une en excellente station de travail graphique. Avec un réseau TCP/IP, vous pouvez même afficher des applications graphiques tournant sur d'autres machines sur votre propre écran, comme avec n'importe quel autre système utilisant X Window.

Le système X Window fut à l'origine développé au MIT, et il est diffusable librement. Toutefois, beaucoup de constructeurs ont réalisé des versions propriétaires des programmes originaux X Window. La version de X Window disponible pour Linux est connue sous le nom de XFree86, un portage de X11R5 (et maintenant X11R6) de diffusion libre, adapté spécialement aux systèmes UNIX basés sur les processeurs 80386, comme Linux. XFree86 supporte une très large gamme de cartes vidéo, comprenant le VGA, Super VGA, et un certain nombre de cartes vidéo comportant des fonctions accélératrices. Il s'agit d'une distribution complète du système X Window, incluant le serveur X, les applications et utilitaires, les bibliothèques de programmation et la documentation.

Les applications X standard comprennent `xterm` (un émulateur de terminal utilisé pour la plupart des applications textes à l'intérieur d'une fenêtre X) ; `xdm` (le gestionnaire de sessions, qui s'occupe des logins) ; `xclock` (une petite horloge graphique) ; `xman` (un lecteur graphique des pages de manuel Linux), et beaucoup d'autres. Le nombre d'applications X disponibles pour Linux est trop important pour les citer toutes ici ; la distribution de base de XFree86 comprend les applications « standard » que l'on trouve dans la distribution originale du MIT. Bien d'autres sont disponibles séparément, et en théorie n'importe quelle application écrite pour le système X Window peut être compilée sans aucun problème sous Linux.

L'apparence de l'interface graphique de X Window est contrôlée dans une large mesure par le **gestionnaire de fenêtres** (window-manager). Ce programme convivial a la charge de placer les fenêtres à l'écran, d'offrir à l'utilisateur la possibilité de les déplacer, changer leur taille, les icônifier, décorer leurs pourtours, etc... La distribution standard de XFree86 comprend `twm`, le gestionnaire de fenêtres classique du MIT, mais bien d'autres gestionnaires de fenêtres sont disponibles, comme `olvwm` (Open Look Virtual Window Manager), par exemple. L'un des plus populaires sous Linux (et de plus en plus sur d'autres systèmes) est `fvwm`. Il s'agit d'un petit gestionnaire de fenêtres, utilisant moitié moins de mémoire que `twm`, qui offre une interface 3-D et un bureau virtuel. `fvwm` peut se paramétrer de nombreuses façons et autorise des fonctions avancées aussi bien par la souris que directement au clavier. Beaucoup de distributions de Linux fournissent `fvwm` comme gestionnaire de fenêtres standard.

La distribution de XFree86 contient le nécessaire de programmation pour les développeurs désirant écrire des applications X Window. Différents ensembles de widgets sont disponibles, comme Athena, Open Look, et Xaw3D. Toutes les fontes standards, bitmaps, pages de manuel, et documentation sont là. PEX (une interface de programmation graphique 3-D) est aussi supporté.

Beaucoup de programmeurs sous X Window utilisent le widget propriétaire Motif pour développer leurs applications. Plusieurs sociétés vendent des licences mono ou multi-utilisateurs de versions binaires de Motif pour Linux. Mais Motif étant lui-même relativement cher, très peu d'utilisateurs Linux l'utilisent. Toutefois, les binaires liés avec les bibliothèques Motif statiques sont librement diffusables. Mais si vous écrivez un programme utilisant Motif et voulez le distribuer librement, vous devez fournir un binaire statique afin que les utilisateurs ne possédant pas Motif puissent malgré tout utiliser le programme.

Les inconvénients majeurs de X Window sont la configuration matérielle nécessaire et la quantité de mémoire utilisée. Un 386 équipé de 4 mégaoctets de mémoire est capable d'exécuter X, mais il faudra au minimum 8 mégaoctets pour l'utiliser de manière confortable. Un processeur plus rapide est également utile, mais la quantité de mémoire RAM est ce qui compte le plus. Ensuite, pour obtenir un affichage vraiment rapide, une carte vidéo comportant des fonctions accélératrices (comme les cartes local-bus équipées de chips S3) est fortement recommandée. Avec une configuration suffisante, des performances dépassant 140.000 xstones ont été mesurées sous Linux et XFree86. Avec une configuration correcte, vous vous rendrez compte que X Window sous Linux est au moins aussi rapide, sinon plus, que sur d'autres stations de travail UNIX.

Nous verrons comment installer et utiliser X dans le chapitre 5.

1.4.5 Réseau

Communiquer avec le reste du monde vous intéresse ? Oui ? Non ? Peut-être ? Linux supporte les deux protocoles de base des systèmes UNIX: **TCP/IP** et **UUCP**. TCP/IP (Transmission Control Protocol/Internet Protocol, pour les amateurs d'acronymes) est l'ensemble de protocoles qui permet aux ordinateurs du monde entier de communiquer entre eux sur un unique réseau connu sous le nom d'Internet. Avec Linux, TCP/IP, et une connexion au réseau, vous pouvez communiquer avec des utilisateurs ou des machines à travers l'Internet par courrier électronique, news Usenet, transférer des fichiers à l'aide de FTP, et bien plus. Il y a déjà de nombreux systèmes Linux connectés sur l'Internet.

La plupart des réseaux TCP/IP utilisent Ethernet pour le transport physique des données. Linux supporte la plupart des cartes Ethernet populaires pour le PC, vous ne devriez pas avoir de problèmes pour vous connecter à un réseau.

Cependant, comme tout le monde ne possède pas un réseau local à domicile, Linux supporte également **SLIP** (Serial Line Internet Protocol) et **PPP** (Point do Point Protocol), qui permettent de vous connecter à l'Internet (ou tout autre réseau TCP/IP) par modem. Vous aurez besoin d'un accès à une machine serveur SLIP ou PPP, connectée au réseau. Un grand nombre d'entreprises ou d'universités proposent de tels services[4]. En fait, si votre système Linux possède à la fois une connexion Ethernet et un modem, vous pouvez le configurer en tant que serveur de ce type pour d'autres machines.

NFS (Network File System) permet de partager des systèmes de fichiers entiers avec d'autres machines par le réseau. FTP (File Transfer Protocol) permet de transférer des fichiers entre des machines, mêmes équipées de systèmes d'exploitation non compatibles. Vous disposez également de **`sendmail`**, qui gère l'envoi et la réception de courrier électronique par le protocole SMTP ; des serveurs de news Usenet supportant NNTP comme C-News et INN vous permettront de vous connecter à des machines distantes et d'y exécuter des commandes ; **`finger`** vous donnera des informations sur d'autres utilisateurs sur l'Internet. Il y a littéralement des tonnes d'applications basées sur TCP/IP.

Toute la gamme des lecteurs de courrier ou de news Usenet est disponible sous Linux, citons par exemple **`elm`**, **`pine`**, **`rn`**, **`nn`**, et **`tin`**.

Si vous avez une bonne expérience d'applications TCP/IP sur d'autres systèmes UNIX, Linux vous sera très familier. Le système comporte les sockets BSD standard, pratiquement tout programme qui utilise TCP/IP peut être compilé sous Linux. Le serveur X supporte également TCP/IP bien entendu, vous permettant d'utiliser d'autre machines pour lancer certains programmes, qui s'afficheront néanmoins sur votre écran par le biais du réseau, ou inversement.

Dans le chapitre 5 nous parlerons de la configuration et de la mise en route de TCP/IP, ainsi que de SLIP, pour Linux.

UUCP (UNIX-to-UNIX Copy) est un mécanisme plus ancien destiné à transférer des fichiers, du courrier électronique et des news Usenet entre machines UNIX. Traditionnellement, les machines UUCP se connectent entre elles par téléphone à l'aide d'un modem, mais UUCP est aussi capable de transporter des données sur une liaison TCP/IP. Si vous ne pouvez avoir accès à un réseau TCP/IP, vous pouvez configurer votre système de façon qu'il recoive et envoie des fichiers et le courrier électronique par UUCP. Voir le chapitre 5 pour plus d'informations.

4. En France hélas c'est beaucoup plus rare et bien plus cher, mais cette situation est en train d'évoluer à l'heure où nous écrivons ces lignes.

1.4.6 Télécommunications et programmes de BBS

Si vous possédez un modem, vous voudrez sans doute communiquer avec d'autres machines en utilisant l'un des paquetages de télécommunications disponibles sous Linux. Beaucoup utilisent des programmes de communications pour se connecter à leur BBS (Bulletin Board Systems) favoris, aussi bien qu'à des services commerciaux comme Prodigy, CompuServe, et America On-Line[5]. D'autres utilisent leur modem pour se connecter à des systèmes UNIX à leur travail ou à leur école. Vous pouvez même, si votre modem le permet, utiliser Linux pour recevoir ou expédier des télécopies (FAX). Les programmes de communications disponibles sous Linux ressemblent beaucoup à ceux que vous pouvez trouver sous MS-DOS ou d'autres systèmes d'exploitation. Quiconque a déjà utilisé ce type de programme saura immédiatement utiliser l'un de ceux qu'il trouvera sous Linux.

L'un des programmes de communications les plus populaires s'appelle Seyon, c'est un programme graphique fonctionnant sous X Window qui offre une interface ergonomique, des protocoles de transferts incorporés, et qui s'utilise à la souris très simplement. Vous trouverez bien entendu l'inévitable Kermit, puis `pcomm` et `minicom`. Ces derniers fonctionnent en mode texte et sont semblables à ce que l'on peut trouver sous MS-DOS.

Si vous n'avez pas accès à un serveur SLIP ou PPP (voir la section précédente), vous pourrez utiliser `term` pour multiplexer votre ligne série. `term` vous permettra d'ouvrir plusieurs sessions simultanées sur la ligne du modem vers une machine distante. `term` permet aussi de rediriger des connexions de clients X Window vers votre serveur X local, à travers le modem, ce qui vous permet d'afficher chez vous des applications X distantes, même sans accès IP. KA9Q est un programme similaire.

Être opérateur de son propre BBS (bulletin board system) est le passe-temps favori de beaucoup d'utilisateurs de micro-ordinateurs. Une bonne gamme de programmes de BBS est disponible sous Linux, la plupart ayant bien plus de possibilités que ce que l'on peut trouver sous d'autres systèmes d'exploitation. Avec une ligne de téléphone, un modem, et Linux, vous pouvez transformer votre machine en BBS, offrant un accès à votre système au monde entier. Citons par exemple d'excellents programmes de BBS comme XBBS et UniBoard.

La plupart des programmes de BBS habituels confinent l'utilisateur dans un menu où seulement quelques fonctions ou applications spécifiquement prévues dans le programme sont disponibles. Une alternative à ce type d'accès réduit peut être l'accès total à UNIX, qui autorise les utilisateurs à utiliser votre machine directement sous shell. Ceci réclame un travail de maintenance et d'administration

5. En France, nous avons le Minitel. Si votre modem est capable de supporter le mode V23 indispensable pour ce type de communications, il existe sous Linux d'excellentes émulations Minitel graphiques fonctionnant sous X Window, vous pourrez donc continuer à utiliser les services de l'annuaire électronique d'un simple clic de souris, ou ne pas perdre le contact avec votre messagerie rose...

système bien plus conséquent pour l'opérateur de la machine, bien entendu; mais ce type d'accès est très apprécié. Avec TCP/IP ou simplement UUCP vous pouvez également fournir courrier électronique et news Usenet à vos utilisateurs.

Si vous ne pouvez vraiment pas accéder au réseau par TCP/IP ou simplement par UUCP, Linux vous permettra malgré tout de communiquer avec un certain nombre de réseaux de BBS comme FidoNet, avec lesquels vous pouvez tout de même échanger du courrier et accéder à divers forums par la ligne téléphonique. Vous trouverez plus d'informations sur ce sujet dans le chapitre 5.

1.4.7 Interfaçage avec MS-DOS

Il existe différents utilitaires destinés à accéder (tant bien que mal) au monde MS-DOS depuis Linux. L'application la plus connue dans ce domaine est DOSEMU, l'Emulateur MS-DOS, qui permet d'exécuter un certain nombre d'applications MS-DOS directement depuis Linux. Bien que Linux et MS-DOS soient totalement différents, le mode protégé du microprocesseur 80386 permet à certaines tâches de se comporter comme si elles tournaient sur un 8086, comme le fait MS-DOS.

L'émulateur MS-DOS est toujours en cours de développement mais permet déjà d'exécuter beaucoup de programmes bien connus. Il faut comprendre que malgré tout, certaines applications MS-DOS utilisant des caractéristiques ésotériques de ce système ne pourront jamais fonctionner sous émulateur, puisque ce n'est justement qu'un émulateur. Par exemple, il sera impossible de faire tourner les programmes utilisant le mode protégé du 80386, comme Microsoft Windows, (en version 386 bien sûr).

Citons une liste non exhaustive d'applications qui ne posent aucun problème à l'émulateur MS-DOS: 4DOS (interpréteur de commandes), Foxpro 2.0, Harvard Graphics, MathCad, Stacker 3.1, Turbo Assembler, Turbo C/C++, Turbo Pascal, Microsoft Windows 3.0 (en mode *réel*), et WordPerfect 5.1. Les commandes standards MS-DOS et les utilitaires les plus connus (comme `PKZIP`) fonctionnent également très bien.

Cet émulateur MS-DOS est surtout destiné à fournir une solution aux personnes qui ont besoin de MS-DOS de temps à autres pour certaines applications, mais qui utilisent Linux pour tout le reste. Il n'est pas et ne sera jamais une implémentation complète destinée à remplacer MS-DOS. Bien sûr s'il ne vous satisfait pas, vous avez toujours la solution de lancer le vrai système MS-DOS original, le programme LILO vous permettra de choisir au lancement de la machine entre tous les systèmes d'exploitation installés sur votre ordinateur.

Linux possède une interface très simple pour transférer des fichiers entre Linux et MS-DOS : vous pouvez monter une partition ou une disquette MS-DOS sous Linux, comme n'importe quelle autre partition, et accéder aux fichiers qui s'y trouvent comme avec n'importe quels autres fichiers.

Un projet connu sous le nom de **WINE** est en cours de développement à l'heure où nous écrivons ces lignes, il s'agit d'un émulateur Microsoft Windows pour le système X Window sous Linux. Lorsqu'il sera terminé, les utilisateurs pourront exécuter des applications MS-Windows directement sous Linux. C'est un programme un peu similaire à l'émulateur WABI, propriété de Sun Microsystems.

Nous parlerons des outils relatifs à MS-DOS dans le chapitre 5.

1.4.8 Autres applications

Une quantité phénoménale de programmes capables de faire tout et n'importe quoi fonctionne sous Linux, comme on peut s'y attendre avec un système d'exploitation aussi universel. L'orientation première de Linux est d'être un système pour les micro-ordinateurs domestiques, mais ceci est en train d'évoluer très rapidement. Des applications professionnelles et scientifiques arrivent, et des éditeurs commencent à proposer des logiciels commerciaux.

Plusieurs programmes de gestion de base de données relationnelles sont disponibles sous Linux, dont Postgres, Ingres, et Mbase. Ce sont des applications de bases de données client/serveur complètes, similaires à ce que l'on trouve sous forme commerciale sous d'autres plate-formes UNIX. `/rdb`, une base de donnée commerciale, est également disponible.

Les programmes scientifiques que l'on peut rencontrer comprennent FELT, `gnuplot`, Octave (similaire à Matlab), `xspread`, `xfractint`, `xlispstat`, Spice, Khoros, etc...

Bien sûr, il y a beaucoup plus d'applications de ce type qui ont été (ou peuvent être) portées sous Linux. Quel que soit votre domaine d'activités, le portage d'applications UNIX sous Linux est la plupart du temps très simple, tout se compile pratiquement sans modification.

Comme tout système d'exploitation, Linux possède son lot de jeux. Ils comprennent les jeux classiques en mode texte comme Nethack et Moria; MUDs (donjon et dragons multi-utilisateurs), DikuMUD et inyMUD; aussi bien que des jeux graphiques sous X Window comme `xtetris`, `netrek`, `Xboard` (la version X11 de `gnuchess`), et `xboing` qui pourra utiliser votre carte sonore si vous en possédez une. Le célèbre jeu d'arcade `Doom` a également été porté sous Linux. C'est vrai, les jeux sous UNIX n'ont jamais atteint la perfection de certains jeux d'arcade que l'on trouve ailleurs, mais certains développements en cours pourraient bien vous apporter des surprises dans les mois qui viennent...!

Pour les audiophiles, Linux supporte différentes cartes sonores et leurs outils associés comme CDplayer (pour piloter un lecteur de CD-ROM), des séquenceurs et éditeurs MIDI, et des éditeurs de son numérique.

Vous cherchez une application particulière? La « Linux Software Map » , décrite en annexe A, contient une liste de nombreux paquetages qui ont été écrits ou portés sous Linux. Bien que cette liste soit loin d'être complète, elle constitue déjà une base solide pour la recherche de programmes. Vous pouvez également lors de vos recherches fouiller les listes d'`INDEX` des sites FTP Linux, si vous avez un accès Internet.

Si vraiment vous ne trouvez pas ce que vous cherchez, vous pouvez toujours tenter de porter une application en provenance d'un autre système que Linux. La plupart des programmes en diffusion libre prévus pour UNIX se compileront sous Linux le plus facilement du monde; et si cela ne marche pas, vous pouvez écrire l'application vous-même. Si l'application que vous recherchez est commerciale, il peut exister un « clône » gratuit quelque part. Ou bien, vous pouvez insister auprès de l'éditeur pour qu'il réalise une version Linux sous forme binaire afin de la commercialiser. Plusieurs particuliers ont déjà agi ainsi, et ont reçu des réponses des plus positives aux plus négatives. Bonne chance...

1.5 À propos du Copyright de Linux

Linux est protégé par ce qui est connu sous le nom de la *General Public License* GNU, ou *GPL*. La GPL fut développée pour le projet GNU par la Free Software Foundation. Elle garantit un certain nombre de choses concernant la distribution et la modification du « free software » . « Free » se réfère à la liberté, pas seulement au prix. La GPL a toujours été mal interprétée, et nous espérons que cette mise au point vous permettra de mieux comprendre les tenants et aboutissants de la distribution de Linux sous licence GNU. Une copie complète de la GPL est donnée dans l'annexe E.

Au départ, Linus Torvalds avait diffusé Linux selon des modalités plus restrictives que la GPL, qui permettait de distribuer le système librement et de le modifier, mais interdisait de demander de l'argent pour sa distribution et son utilisation. De son côté, la GPL autorise quiconque le désire à vendre ses programmes et à gagner de l'argent avec, mais impose également que tout le monde puisse parallèlement distribuer la même chose comme il l'entend, sans restrictions.

Tout d'abord, il faut bien préciser que le « free software » protégé par la GPL ne relève *pas* du domaine public. Un programme en domaine public est un programme qui n'appartient à personne, et par conséquent tout le monde en est propriétaire. Un programme protégé par la GPL, à l'opposé, appartient à son ou ses auteurs. Cela signifie que le programme est protégé par les lois internationales en vigueur pour cette discipline, et que l'auteur existe vis à vis de la loi. Le fait que le programme puisse être distribué librement et gratuitement ne signifie pas qu'il appartienne au domaine public.

Un programme sous la GPL n'est pas non plus un « shareware » (partagiciel), un « partagiciel » est propriété de son auteur, mais l'auteur demande aux utilisateurs réguliers de son programme une rétribution volontaire. Les programmes distribués selon la GPL peuvent par contre être diffusés et utilisés totalement gratuitement.

La GPL autorise aussi les utilisateurs à modifier les programmes et à en distribuer leur propre version. Toutefois, tout travail dérivé d'un programme soumis à la GPL doit également être obligatoirement diffusé selon cette licence. En d'autres termes, une entreprise ne peut pas récupérer Linux, le modifier, et le revendre sous une licence plus restrictive. Tout programme dérivé de Linux doit être diffusé selon la GPL.

La GPL permet de distribuer et d'utiliser les programmes avec une gratuité totale. Mais elle permet aussi de faire payer pour cela, et même de faire des bénéfices ainsi. Toutefois, la vente de programmes soumis à la GPL n'autorise aucune restriction de licence à l'achat ; si vous achetez un programme diffusé selon la GPL, quel que soit le vendeur, vous avez le droit de le diffuser gratuitement par la suite, ou de le revendre le prix que vous voudrez.

Cela peut paraître contradictoire, au premier abord. Pourquoi vouloir demander de l'argent pour un programme que tout le monde, selon la GPL, peut obtenir sans dépenser un sou ? Par exemple, disons qu'une compagnie a décidé de rassembler une grande quantité de programmes et de les distribuer sous forme de CD-ROM. Cette compagnie voudra amortir ses frais de fabrication, et probablement faire quelques bénéfices sur la vente de leur produit. Ce cas de figure est autorisé par la GPL.

Les organisations vendant des programmes libres doivent se soumettre à certaines restrictions précisées dans la GPL. Tout d'abord, elle ne peuvent pas réduire les droits des utilisateurs qui ont acheté les programmes. Cela signifie que que si vous achetez un CD-ROM de programmes soumis à la GPL, vous pouvez les copier autant que vous voulez et redistribuer ce CD-ROM sans contrepartie financière. En second lieu, les distributeurs doivent préciser clairement aux utilisateurs que les programmes sont couverts par la GPL. Troisièmement, ces distributeurs sont dans l'obligation de fournir, gratuitement, tout le code source des programmes qu'ils distribuent. Ceci permet à toute personne achetant un tel programme de pouvoir lui apporter toutes les modifications qu'il désire.

Autoriser une entreprise à distribuer et vendre des programmes libres est une très bonne chose, quoi que certains puissent en penser. Tout le monde n'a pas accès à l'Internet pour télécharger ces programmes sans bourse délier. La GPL permet aux sociétés de vendre ces programmes aux personnes pour qui un accès Internet coûterait beaucoup trop cher. Par exemple, beaucoup de petits distributeurs vendent Linux sur disquettes, bandes ou CD-ROM par correspondance, et gagnent de l'argent ainsi. Les développeurs de Linux ne verront jamais un centime de cet argent ; lorsqu'un programmeur décide de diffuser son œuvre selon la GPL il est parfaitement d'accord sur ce point. En d'autres termes, Linus sait que certaines sociétés gagnent beaucoup d'argent en vendant Linux, mais qu'il ne touchera jamais aucun pourcentage sur ces bénéfices.

Dans le monde du « free software » , l'argent n'a aucune importance. Le but premier est de développer et diffuser de superbes programmes et de permettre à tout le monde de les obtenir et de les utiliser.

Nous verrons dans la section suivante comment ceci s'applique dans le cas du développement de Linux.

1.6 Conception et philosophie de Linux

Lorsque de nouveaux utilisateurs découvrent Linux, ils se font souvent de fausses idées sur le système. Linux est unique, et il est important de comprendre sa philosophie et la manière dont il est conçu pour l'utiliser efficacement. Trêve de plaisanteries. Même si vous êtes un vieux routard du monde UNIX, ce qui va suivre va certainement vous intéresser.

Dans les multinationales développant des systèmes UNIX, la totalité du système est développé selon des règles très rigoureuses permettant de maîtriser la qualité des programmes, de la documentation, de recenser et corriger les bogues, et ainsi de suite ; les développeurs n'ont pas le droit de rajouter des améliorations ou de changer des portions de code selon leurs envies ; chaque programmeur se voit assigner une ou plusieurs parties du code du système, et lui et uniquement lui pourra avoir l'autorisation de modifier certaines sections une fois que les modifications seront dûment testées et validées selon des processus techniques et administratifs longs et complexes.

Le département qualité utilise des tests de non-régression sur chaque nouvelle version du système, et signale tout problème. Les développeurs ont la responsabilité de corriger ces problèmes comme il leur est indiqué ; de complexes méthodes d'analyse statistiques sont employées pour s'assurer qu'un certain pourcentage de bogues est corrigé avant chaque nouvelle version, et que le nouveau système passe avec succès un certain nombre de tests critiques.

Pour tout dire, le processus de développement et la maintenance d'un système UNIX commercial sont très compliqués. La compagnie doit avoir la preuve que la prochaine version du système est réellement prête à être diffusée. Le développement d'un UNIX commercial est un énorme travail, employant souvent des centaines (voire des milliers) de programmeurs, testeurs, documentalistes, et personnel administratif. Bien sûr cela diffère selon le cas, mais vous avez une idée de ce que ce peut être.

Avec Linux, vous pouvez oublier complètement le concept de développement organisé, contrôle de sources, rapport de bogues structurés, analyses statistiques, ou autres. Linux est, et sera probablement toujours, un système de « hackers » [6].

Linux a vu le jour grâce aux efforts d'un groupe de volontaires répartis aux quatre

6. Ce que nous désignons par « hacker » (terme très courant, qui n'a pas d'équivalent français), est un programmeur passionné et extrêmement compétent, qui maîtrise comme par don les ordinateurs et en fait pratiquement ce qu'il désire, pour la bonne cause. Il n'y a aucune notion péjorative dans ce terme, au contraire c'est plutôt honorifique; hélas beaucoup confondent les termes « hacker » et « cracker » , le second désignant effectivement une personne compétente mais ayant des intentions malhonnêtes.

coins du monde, reliés entre eux par l'Internet. Grâce à ce réseau, quiconque ayant suffisament de connaissances et de temps peut aider au développement et au déboguage de Linux, écrire des applications, porter des programmes, rédiger de la documentation, ou aider les utilisateurs perdus. Il n'existe pas une organisation, une unique entité responsable du développement du système. Pour sa plus grande part, la communauté Linux communique par diverses listes de diffusion et de forums Usenet. Un certain nombres de conventions se sont installées d'elles-mêmes au fil du développement ; par exemple celui qui désire voir son code inclus dans le noyau Linux « officiel » doit l'envoyer par courrier électronique à Linus Torvalds, qui le testera et l'incorporera au noyau (tant qu'il ne provoque pas d'effets de bords ou ne va pas à l'encontre de la philosophie du système, il y a beaucoup de chances pour qu'il l'accepte).

Le système en lui-même est conçu pour être très ouvert à toutes sortes d'améliorations ou possibilités. Bien que récemment, le nombre de nouvelles possibilités et de changements de conception de certaines parties du code ait énormément diminué, la règle générale veut qu'une nouvelle version du noyau apparaisse régulièrement au bout de quelques mois, (et à certaines époques bien plus souvent encore). Bien sûr cela est très variable, tout dépend du nombre de bogues à corriger, du nombre de critiques des utilisateurs testant les pré-versions du code, et du nombre d'heures de sommeil en retard de Linus. Disons tout de suite que tous les bogues n'ont pas été corrigés et que certains problèmes ne sont pas éliminés à chaque version. Mais tant que le système parait fonctionner suffisamment bien, que les bogues sont mineurs et ne provoquent pas de situations critiques, il est considéré comme « stable » et de nouvelles versions peuvent alors être diffusées. Le but du développement Linux n'est pas de fournir à tout prix des versions totalement parfaites, sans aucun bogues, mais de développer une implémentation d'UNIX libre et gratuite. Linux est avant tout fait *pour* les développeurs.

Quiconque ayant écrit une nouvelle application ou rajouté une nouvelle possibilité au système, le diffuse généralement comme version « alpha » , qui correspond à une étape où les personnes courageuses et ayant du temps libre testent le code pour trouver ses points faibles. Du fait que la communauté Linux dépend en grande partie de l'Internet, ces versions alpha sont généralement mises à disposition sur un ou plusieurs sites FTP (voir annexe C), et un message annonçant leur disponibilité posté dans l'un des forums Usenet consacrés à Linux. Les utilisateurs récupérant et testant ces programmes peuvent contacter l'auteur directement par courrier électronique pour lui faire part des problèmes éventuellement rencontrés.

Après ce stade alpha initial, et une fois que le programme a été amélioré et corrigé, le code arrive au stade « bêta » , dans lequel il est en général considéré comme stable, mais encore incomplet. (Par exemple, il fonctionne bien et sans erreurs mais ne contient pas encore tout le superflu). Sinon il peut passer directement au stade « final » , où il est considéré complet, stable et utilisable. S'il s'agit de code appartenant au noyau, le développeur peut alors demander à Linus de l'inclure dans le noyau standard, ou comme option de configuration supplémentaire.

Dites vous bien que ce ne sont que des conventions, et pas des règles. Certains sont tellement sûrs de la qualité de leur code qu'ils n'ont pas besoin de passer par les stades alpha et bêta-test. C'est toujours au développeur de prendre ce genre de décision.

Vous êtes sans doute perplexe et vous demandez comment une équipe aussi peu structurée de volontaires pourrait bien arriver à programmer et déboguer un système UNIX complet. Au risque de vous étonner, c'est l'une des méthodes de développement les plus efficaces et motivantes qui ait jamais été utilisée. La totalité du noyau de Linux a été écrite *en partant de zéro* sans emprunter une seule ligne de code de tout autre système. Beaucoup de volontaires ont travaillé spontanément des nuits entières pour porter tous les programmes libres qu'ils trouvaient, écrire les bibliothèques, développer des systèmes de fichiers originaux, ou faire marcher les cartes d'extension du PC dont ils disposaient avec Linux.

Linux est généralement diffusé sous forme d'une *distribution*, qui consiste en un ensemble de programmes regroupés de manière cohérente, formant après installation un système complet. Il serait très difficile pour beaucoup d'utilisateur de construire un système complet en partant du noyau, des sources des utilitaires, commandes, applications, installant tous ces programmes un par un manuellement.[7] Encore une fois, précisons qu'il n'existe pas de distribution standard de Linux, mais un certain nombre, chacune ayant ses avantages et inconvénients. Nous verrons cela plus en détail dans la section 2.1.

Bien que ces distributions permettent l'installation d'un système complet, vous aurez tout de même besoin de quelques connaissances UNIX pour obtenir une installation parfaite vous donnant toute satisfaction. De même qu'il n'existe pas de programme sans bogue, il n'y a probablement pas de distribution de Linux sans erreur ou omission.

Administrer un système UNIX n'est pas une tâche aisée, qu'il soit gratuit ou commercial. Si vous désirez entretenir votre système sérieusement, notez bien que cela vous demandera beaucoup d'efforts et d'attention : ceci est vrai pour *tous* les systèmes UNIX, et Linux ne fait pas exception à la règle. En raison de la diversité de la communauté Linux et des ressources tellement variées que tant de programmes peuvent attendre, il est impossible de prévoir à l'avance toutes les manipulations que vous aurez à faire et vous mâcher le travail dans tous les cas de figure.

1.6.1 Conseils aux débutants sous UNIX

Installer et utiliser votre propre système Linux ne demande pas de connaissances approfondies d'UNIX. En fait, beaucoup de novices en la matière y parviennent

7. Note du traducteur: C'est pourtant ce que les tout premiers utilisateurs de Linux furent obligés de faire, avant que ces distributions n'existent, c'était un des charmes de Linux que les nouveaux ne connaîtront jamais...

tous les jours. C'est une expérience très enrichissante, mais vous devez savoir qu'elle peut devenir très frustrante pour certains. Si vous avez de la chance, vous arriverez à faire fonctionner Linux sans aucune notion préalable d'UNIX. Toutefois, dès que vous voudrez l'utiliser efficacement, et installer de nouveaux programmes, un minimum de connaissance d'UNIX sera impératif.

Heureusement, vous progresserez vite grâce à Linux ; et ce livre vous apprendra toutes les bases nécessaires. Le chapitre 3 est un didacticiel concernant UNIX et le chapitre 4 contient suffisament d'informations pour que vous puissez commencer à administrer correctement votre système. Lisez ces chapitres avant de commencer toute installation, cela vous évitera de commettre certaines erreurs.

Personne ne peut devenir administrateur système UNIX en une nuit. Aucune version d'UNIX n'est faite pour tourner toute seule sans aucune maintenance. Vous devez vous préparer à ce qui vous attend. Sinon, si vous ne connaissez pas du tout UNIX, vous serez rapidement frustré et déçu par le système.

1.6.2 Conseils aux gourous UNIX

Même ceux qui ont des années d'expérience en programmation et administration système UNIX peuvent avoir besoin d'aide pour installer Linux. Il y a toujours certains aspects d'UNIX avec lesquels on n'est pas trop familiarisé. Pour commencer, Linux n'est pas un système UNIX commercial. Il ne cherche pas à suivre les mêmes standards que les autres systèmes UNIX que vous avez utilisé. Pour être plus précis, bien que la stabilité soit un facteur très important dans le développement de Linux, ce n'est pas le *seul* facteur qui entre en ligne de compte.

Les fonctionnalités offertes, par exemple, sont peut être plus importantes. Dans de nombreux cas, du nouveau code se verra incorporé dans le noyau même s'il est encore bogué et incomplet. Simplement parce qu'il est beaucoup plus important de diffuser du code que des milliers d'utilisateurs vont pouvoir utiliser et tester, que d'attendre des mois qu'une version soit « complète » . Par exemple, WINE (l'émulateur de Microsoft Windows pour Linux) avait une version alpha-test « officielle » bien avant qu'il ne commence à fonctionner. De cette façon, toute la communauté Linux a la possibilité de travailler sur le code, de le tester, voire de le développer, et ceux pour qui cet alpha-code est « suffisament bon » pour leurs besoins peuvent commencer à l'utiliser. Les vendeurs de systèmes UNIX commerciaux utilisent rarement, voire jamais, ce genre de méthodes.

Si vous êtes administrateur UNIX depuis plus de dix ans, et avez utilisé pratiquement toutes les versions commerciales du système possibles et imaginables, il vous faudra sans doute un peu de temps pour vous habituer à Linux. Le système est très moderne et dynamique. De nouvelles versions du noyau arrivent plusieurs fois par an, et même plusieurs fois par mois de temps à autres. Les programmes et applications sont constamment mises à jour. Votre système peut être une fois totalement à jour, et quelques jours plus tard être considéré comme un Linux de l'âge de pierre.

Avec autant d'activité, comment pouvez-vous espérez suivre le mouvement ? Le mieux est de faire des mises à jour incrémentales ; changez uniquement les parties du système qui *ont besoin* d'être mises à jour, et seulement lorsque vous pensez qu'une mise à jour est vraiment nécessaire. Si vous n'utilisez jamais Emacs, il n'y a aucune raison d'installer toutes les nouvelles versions de cet éditeur qui arrivent chaque mois. Mieux encore ; même si vous êtes un utilisateur forcené d'Emacs, il n'y a en général pas d'intérêt à changer la version que vous utilisez, sauf si il vous manque une possibilité qui est offerte dans la version suivante. Il n'y a aucun intérêt à vouloir passer son temps à être toujours équipé des toutes dernières versions de tous les programmes qui composent Linux.

Nous espérons que Linux correspondra à ce que vous attendez d'un système UNIX personnel, et dépassera même vos espérances. Au plus profond de Linux se trouve l'esprit de la programmation libre, offerte à tous, en constant développement pour le bonheur de tous, loin des considérations bassement matérielles. La communauté Linux favorise l'expansion du système plutôt que sa stabilité, et c'est un concept assez difficile à appréhender pour beaucoup de gens, particulièrement ceux qui sont habitués au monde des UNIX commerciaux. Vous ne pouvez pas demander à Linux qu'il soit parfait, rien n'est jamais parfait dans le monde du logiciel public. Néanmoins, nous pensons que Linux est réellement aussi complet, utile et fiable que n'importe quelle autre implémentation d'UNIX.

1.7 Différences entre Linux et d'autres systèmes

Il est important de comprendre les différences entre Linux et d'autres systèmes d'exploitation comme MS-DOS, OS/2, et d'autres implémentations d'UNIX pour ordinateurs personnels. Tout d'abord, il doit être clair dans l'esprit de tous que Linux peut coexister sans problèmes avec d'autres systèmes sur la même machine : Autrement dit, la même machine peut tourner sous MS/DOS et OS/2 à certains moments, ou bien sous Linux à d'autres moments sans avoir à réinstaller quoi que ce soit. Nous verrons qu'il est même possible d'échanger des données entre les différents systèmes d'exploitation.

1.7.1 Pourquoi utiliser Linux ?

Pourquoi utiliser Linux au lieu d'un système d'exploitation commercial réputé, bien testé, bien documenté ? Nous pourrions vous donner une centaine de raisons. L'une des plus importantes, toutefois, est que Linux est un excellent choix en tant que système UNIX personnel. Si vous êtes un programmeur système UNIX, pourquoi utiliser MS-DOS chez vous ? Linux vous permettra de développer et tester vos programmes UNIX sur votre PC, y compris des programmes de base de données ou des applications X-Window. Si vous êtes étudiant, il y a beaucoup de chances pour que les systèmes informatiques de votre université tournent sous UNIX. Avec Linux, vous pourrez avoir votre propre système UNIX et le configurer selon vos besoins. Installer et utiliser Linux est aussi une excellente méthode pour apprendre si vous n'avez pas accès à d'autres machines UNIX.

Mais ne nous y trompons pas : Linux n'est pas uniquement destiné à un usage domestique. Il est suffisament fiable et complet pour assurer de lourdes tâches. Beaucoup d'entreprises (particulièrement les petites) remplacent petit à petit leur environnement de travail, basé sur des stations de travail fonctionnant sous UNIX, par des machines tournant sous Linux. Les universitaires trouvent Linux idéal pour l'enseignement des systèmes d'exploitation. Certains éditeurs d'applications commerciales commencent à réaliser l'intérêt qu'un système d'exploitation gratuit peut avoir.

Les sections qui suivent vont décrire les différences les plus importantes entre Linux et d'autres systèmes d'exploitation. Nous espérons que vous en déduirez que Linux peut correspondre à vos besoins, ou (au moins) améliorer votre environnement informatique actuel. Rappelez-vous que le meilleur moyen de se faire une idée sur Linux est de l'essayer ; vous n'avez même pas besoin d'installer une version complète pour cela, nous vous expliquerons comment dans le chapitre 2.

1.7.2 Linux contre MS-DOS

Il n'est pas rare d'utiliser à la fois Linux et MS-DOS sur le même ordinateur. Beaucoup d'utilisateurs de Linux ont besoin de MS-DOS pour des applications comme le traitement de texte. Bien que Linux fournisse des applications équivalentes (par exemple, TEX, avec lequel ce livre est réalisé), il y a diverses raisons pour lesquelles certains utilisateurs ne peuvent les utiliser. Si votre thèse a été saisie sous WordPerfect pour MS-DOS, vous ne pourrez pas la convertir facilement en TEX ou quelque autre format. Il existe beaucoup d'applications commerciales pour MS-DOS qui ne sont pas disponibles sous Linux, et il n'y a aucune raison pour que vous ne puissiez pas utiliser les deux.

Comme vous le savez sans doute déjà, MS-DOS n'utilise qu'une petite partie des possibilités des processeurs 80386 et 80486. A l'opposé, Linux tourne exclusivement dans le mode protégé de ces processeurs, et exploite toutes leurs possibilités. Vous pouvez directement accéder à toute la mémoire dont vous disposez (et même au-delà, grâce à la mémoire virtuelle). Linux offre une interface UNIX complète qui n'existe pas sous MS-DOS ; le développement ou le portage d'applications UNIX sous Linux est très simple, alors que sous MS-DOS vous êtes limité à un tout petit sous-ensemble des fonctionnalités nécessaires. Parce que Linux est un véritable système UNIX, vous n'avez pas ces limitations.

Nous pourrions débattre encore sur le sujet pendant des pages et des pages. Disons simplement pour clore le sujet que Linux et MS-DOS sont deux choses totalement différentes. MS-DOS est bon marché (comparé à d'autres systèmes d'exploitation commerciaux), et fait partie de l'histoire de l'ordinateur IBM-PC. Aucun autre système d'exploitation pour Intel n'a atteint le niveau de popularité de MS-DOS ; en grande partie parce que le prix de ces autres systèmes était inabordable pour un particulier. Très peu de gens peuvent se permettre de dépenser plus de $1000, juste pour le système d'exploitation de leur ordinateur personnel. Linux étant gratuit, vous avez enfin l'occasion de choisir.

Nous allons vous permettre de vous faire votre propre opinion sur Linux et MS-DOS, en fonction de ce que vous attendez et de vos réels besoins. Linux n'est pas pour tout le monde. Si vous avez toujours désiré utiliser un système UNIX chez vous, sans payer le prix généralement élevé des autres implémentations pour PC de ce système, Linux pourrait bien être ce que vous recherchiez depuis toujours...

Il existe des outils vous permettant de communiquer avec MS-DOS. Par exemple, il est très facile d'accéder à des fichiers MS-DOS depuis Linux. Il existe aussi un émulateur MS-DOS, qui permet d'exécuter beaucoup d'applications populaires. Un émulateur Windows est actuellement en cours de développement.

1.7.3 Linux contre les autres systèmes

Un certain nombre d'autres systèmes d'exploitation performants sont en train d'apparaitre dans le monde du PC. En particulier, OS/2 d'IBM et Windows NT de Microsoft deviennent très populaires, car de plus en plus d'utilisateurs abandonnent MS-DOS.

OS/2 et Windows NT sont tous deux des systèmes d'exploitation multi-tâches, un peu comme Linux. Techniquement, OS/2, Windows NT, et Linux sont assez similaires : ils ont grosso-modo les mêmes fonctionnalités en matière d'interface, réseau, sécurité... Toutefois la réelle différence entre Linux et Les Autres Types est que Linux est une version d'UNIX, et par conséquent bénéficie des contributions de la très importante communauté UNIX.

Qu'est-ce qui rend UNIX si important ? Ce n'est pas seulement le système d'exploitation le plus populaire pour les machines multi-utilisateurs, mais c'est aussi la base du monde des graticiels ou programmes en domaine public. Si vous avez accès à l'Internet, presque tous les programmes qui y sont disponibles sont écrits pour UNIX. (L'Internet en lui-même est largement constitué de réseaux basés sur UNIX.)

Il existe beaucoup d'implémentations d'UNIX, provenant d'un grand nombre d'éditeurs, il n'y a pas une unique organisation responsable de sa distribution. Un mouvement de standardisation est largement entamé dans la communauté UNIX, sous la forme de systèmes ouverts, mais aucune sociéte n'a le monopole de cette standardisation. C'est pourquoi n'importe quel éditeur (ou, du coup, n'importe quel programmeur) peut implémenter ces standards dans sa propre version d'UNIX.

OS/2 et Windows NT, en revanche, sont des systèmes propriétaires. L'interface et le fonctionnement sont contrôlés par leur seul propriétaire, qui est le seul à en posséder les droits. (N'espérez pas voir une version indépendante et gratuite d'OS/2 de sitôt.) En un sens, ce type d'organisation est bénéfique : il définit un standard strict pour le programmeur et l' interface utilisateur, au lieu de ce que l'on trouve dans la communauté des systèmes ouverts. OS/2 est OS/2 où que vous vous trouviez, et il en va de même pour Windows NT.

Toutefois, l'interface d'UNIX est en constante évolution. Plusieurs organisations tentent de standardiser le modèle de développement, mais la tâche est difficile. Linux, en particulier, est en majeure partie conforme au standard POSIX.1 pour l'interface de programmation. Petit à petit, Linux adhèrera de plus en plus à d'autres standards de ce type, mais la standardisation n'est pas le but premier dans la communauté des développeurs de Linux.

1.7.4 Autres implémentations d'UNIX

Il existe plusieurs autres implémentations d'UNIX pour les 80386 et 80486. L'architecture 80386 offre tout ce qui est nécessaire pour réaliser UNIX, et un certain nombre d'éditeurs ont profité de cet avantage.

Côté caractéristiques, les autres implémentations d'UNIX pour le PC sont pratiquement semblables à Linux. Vous verrez que pratiquement toutes les versions commerciales d'UNIX supportent grosso-modo les mêmes programmes, possèdent les mêmes possibilité réseau, et que les environnements de programmation sont les mêmes. Malgré tout, il y a d'énormes différences entre Linux et les versions commerciales d'UNIX.

En premier lieu, Linux supporte une configuration matérielle différente des implémentations commerciales. En règle générale, Linux supporte les périphériques les plus populaires, mais ceci est toujours limité par le matériel auquel les développeurs peuvent avoir accès. Les UNIX commerciaux supportent souvent une gamme de matériel plus étendue, mais Linux n'est en réalité pas très loin derrière. Nous décrirons la configuration matérielle nécessaire à Linux dans la section 1.8.

En second lieu, les versions commerciales d'UNIX sont fournies avec une documentation très complète, ainsi qu'un service après vente. A l'opposé, l'essentiel de la documentation de Linux est limitée aux documents disponibles sur l'Internet, et quelques livres comme celui-ci. Dans la section 1.9, nous verrons quels sont les documents et informations relatifs à Linux.

Pour ce qui est de la stabilité et de la fiabilité de Linux, de nombreux utilisateurs ont vérifié que Linux est au moins aussi stable que ses équivalents commerciaux. Linux est toujours en développement, et parfois certaines choses sont momentanément moins fiables mais les problèmes se règlent très rapidement.

Le facteur le plus important pour beaucoup d'utilisateurs, est le prix. Linux est gratuit, si vous avez accès à l'Internet (ou tout autre système informatique) vous pouvez le télécharger. Si toutefois vous n'avez pas accès à un site diffusant Linux ainsi, vous aurez besoin d'acheter l'une des distributions vendues par correspondance sur disquettes, bandes magnétiques, ou CD-ROM (Voir annexe B).

Bien entendu, vous pouvez recopier Linux chez un ami qui a déjà installé le système, ou partager le coût des distributions postales avec quelqu'un d'autre. Si vous avez l'intention d'installer Linux sur un grand nombre de machines, vous n'aurez besoin que d'une seule copie : Il n'y a aucune notion de licence limitée à un nombre d'utilisateurs sous Linux.

La valeur des UNIX commerciaux ne doit pas être diminuée pour autant : le prix payé comprend aussi la documentation, le support clientèle, et l'assurance d'une certaine qualité. Ce sont des facteurs très importants pour de grandes sociétés, mais l'utilisateur d'un ordinateur personnel peut ne pas avoir besoin de tout cela. Quoi qu'il en soit, beaucoup d'entreprises ou d'universités trouvent que l'utilisation de Linux dans un laboratoire équipé de PC bon marché, est préférable à l'achat de versions commerciales d'UNIX et de coûteuses stations de travail. Linux peut offrir les mêmes fonctionnalités qu'une station de travail, sur une architecture PC, pour un prix nettement inférieur.

Un exemple en « vraie grandeur » de l'utilisation de Linux dans la communauté informatique : Des systèmes Linux ont traversé les mers pour être utilisés dans le Pacifique Nord, gérant les télécommunications et l'analyse de données sur un navire de recherche océanographique. Des systèmes Linux sont utilisés dans des laboratoires de recherche en Antarctique. Plus près de nous, plusieurs hôpitaux aux USA utilisent Linux pour gérer les dossiers des malades. Il se comporte de façon aussi fiable, et est aussi souple et utile, que les autres implémentations d'UNIX.

Il exite d'autres implémentations gratuites ou très bon marché d'UNIX pour les processeurs Intel. L'une des plus connues est 386BSD (souvent connue sous le nom « NetBSD »), un portage de l'UNIX de Berkeley pour le 386. 386BSD est comparable à Linux sur beaucoup de points, mais lequel est le « meilleur » dépend essentiellement de vos besoins et de ce que vous désirez. La seule distinction importante que l'on puisse faire, est que Linux est développé de manière ouverte (n'importe quel volontaire peut aider au développement), alors que 386BSD est développé par une équipe fermée de programmeurs. Pour cette raison, de grandes différences de philosophie et de structure existent entre les deux projets. Les buts de chacun sont entièrement différents : pour Linux, c'est le développement d'un système UNIX complet totalement original, sans emprunter de code déjà existant, (et de trouver du bonheur dans ce jeu), alors que le but de 386BSD est en grande partie de modifier le code déjà existant de BSD pour le faire fonctionner sur 386.

NetBSD est un autre portage de la distribution NET/2 de BSD sur un certain nombre d'architectures, y compris le 386. Son développement s'effectue de manière plus ouverte, et le système est très comparable à 386BSD, qu'il tend à remplacer de plus en plus.

Un autre projet à signaler est HURD, réalisé par la Free Software Foundation pour développer et distribuer une libre implémentation d'UNIX pour de nombreuses architectures diverses. Contactez la FSF (l'adresse est donnée dans l'annexe E) pour plus d'informations sur ce projet. A l'heure où nous écrivons ces lignes, HURD en est encore au tout début de son développement.

D'autres versions bon marché d'UNIX existent également, comme Coherent (disponible pour un prix voisin de $99) et Minix (un clone UNIX académique mais utile, utilisé comme plate-forme de développement des toutes premières versions de Linux). Quelques unes de ces implémentations sont surtout d'intérêt éducatif, alors que d'autres sont des systèmes complets pour une réelle utilisation en production. Il n'est pas besoin de dire que, malgré tout, beaucoup de personnes sont en train de migrer vers Linux.

1.8 Configuration matérielle nécessaire

Vous devez maintenant être convaincu que Linux est merveilleux, qu'il pourra faire pour vous des choses fantastiques. Toutefois, avant de vous précipiter sur la procédure d'installation, vous devez être au courant des besoins matériels de Linux et de ses limitations.

N'oubliez pas que Linux a été développé par ses utilisateurs. Cela signifie que pour l'essentiel, l'équipement que Linux sait utiliser correspond seulement à celui que ses développeurs et utilisateurs possèdent, ou celui auquel ils ont pu avoir accès à un moment donné. Du coup, Linux fonctionne sur l'essentiel des ordinateurs 80386/80486 et leurs périphériques les plus populaires (en fait, Linux supporte bien plus de matériel que certaines implémentations commerciales d'UNIX). Malgré tout, quelques périphériques complexes et ésotériques hors de prix ne sont pas encore supportés. Cela change de jour en jour, donc si votre super carte d'extension n'est pas encore reconnue par le système, il y a des chances pour que son support soit en cours de développement ou soit prévu un jour ou l'autre.

Une autre entrave au support de périphériques sous Linux, est que beaucoup de constructeurs ont décidé de garder secrète la documentation nécessaire pour faire fonctionner leurs produits. Le résultat est que les développeurs bénévoles de Linux ne peuvent purement et simplement pas écrire de pilotes pour ces périphériques (s'ils y arrivaient, ces programmes deviendraient la propriété du constructeur qui fabrique le produit, ce qui violerait la GPL). Les constructeurs qui agissent ainsi écrivent leurs propres pilotes pour des systèmes d'exploitation comme MS-DOS et Microsoft Windows ; l'utilisateur final (c'est vous, qui avez payé leur carte) n'est pas censé savoir comment ça marche. Manque de chances, cela ne permet pas aux développeurs d'exploiter ces périphériques, qui ne seront donc jamais supportés par aucun programme du domaine public.

On ne peut pas faire grand chose contre cette situation. Dans certains cas, des programmeurs ont essayé de deviner comment certaines interfaces fonctionnaient.

Dans d'autres cas ils ont travaillé avec le constructeur en question sur le sujet et tenté d'obtenir des informations, avec différents degrés de succès.

Dans les sections qui suivent, nous essaierons de voir les besoins matériels de Linux. Le document *Hardware HOWTO* contient une liste plus détaillée du matériel supporté par Linux (voir la section 1.9), bien que fatalement très incomplète en raison de l'évolution constante du système.

Attention: Une bonne quantité de périphériques est toujours en cours d'intégration dans les pilotes Linux. Certaines distributions peuvent ou peuvent ne pas supporter telle ou telle caractéristique. Cette section recense essentiellement ce que Linux supporte depuis un certain temps et qui est réputé stable. En cas de doute, consultez la documentation de la distribution particulière de Linux que vous utilisez (voir section 2.1 pour plus d'informations sur les distributions Linux). Il y a de grandes chances pour que le système Linux que vous possédez supporte beaucoup plus de choses que celles qui sont citées ici.

1.8.1 Carte mère et microprocesseur

Linux fonctionne actuellement sur les machines équipées des microprocesseurs Intel 80386, 80486 ou Pentium. Cela comprend toutes les variations sur ce type de CPU, comme les 386SX, 486SX, 486DX, et 486DX2. Linux fonctionne également sur le processeur Pentium d'Intel. Les « clones » comme les processeurs Cyrix et AMD fonctionnent aussi avec Linux.

Si vous possédez un 80386 ou 80486SX, vous pouvez aussi utiliser un co-processeur mathématique, bien qu'il ne soit pas indispensable (le noyau de Linux peut l'émuler). Tous les couplages FPU sont supportés comme IIT, Cyrix FasMath, et coprocesseurs Intel.

La carte mère doit utiliser une architecture ISA ou EISA. Ces termes définissent comment le système s'interface avec les périphériques et les autres composants sur le bus. La plupart des systèmes vendus actuellement sont soit à bus ISA, soit à bus EISA. Le bus MicroChannel d'IBM (bus MCA), rencontré sur certaines machines de marque IBM comme le PS/2, n'est pas supporté.

Les systèmes utilisant une architecture à bus local (pour des accès disque et vidéo plus rapides) sont également supportés ; nous vous suggérons d'avoir une architecture à bus local standard comme le VESA Local Bus (« VLB »).

1.8.2 Mémoire nécessaire

Linux ne nécessite que très peu de mémoire pour fonctionner, en comparaison d'autres systèmes d'exploitation modernes. Vous devez avoir au strict minimum 2 Mégaoctets de RAM ; toutefois il est fortement suggéré de posséder au moins 4 Mo. Plus vous aurez de mémoire RAM, plus votre système fonctionnera rapidement.

Linux supporte bien entendu le mode d'adressage 32 bits des processeurs 386/486, en d'autres termes il sait utiliser automatiquement toute la mémoire RAM dont vous pouvez disposer.

Linux tournera correctement avec seulement 4 mégaoctets de RAM, même avec des programmes comme X Window, Emacs, ou autres. Malgré tout, posséder plus de mémoire est au moins aussi important que d'avoir un processeur plus rapide. 8 mégaoctets sont plus que suffisants pour un usage personnel ; 16 mégaoctets ou plus peuvent être parfois nécessaires si vous compter charger énormément le système.

La plupart des utilisateurs de Linux dédient une petite fraction de leur disque dur comme zone de swap, qui est utilisée comme mémoire virtuelle. Même si vous disposez de beaucoup de mémoire RAM dans votre machine, vous pourriez avoir besoin d'utiliser un peu de swap. Bien que le swap ne puisse pas remplacer de la mémoire RAM rapide, cela peut permettre à votre système d'exécuter des applications plus importantes en stockant les portions inactives du code sur le disque dur. La quantité d'espace de swap que vous devez allouer dépend de plusieurs facteurs ; nous reviendrons sur ce point dans la section 2.2.3.

1.8.3 Contrôleurs de disques durs

Vous n'avez pas forcément besoin d'un disque dur pour exécuter Linux ; vous pouvez lancer un système minimum contenu entièrement sur disquette. Mais ceci est plutôt anecdotique, c'est bien entendu extrêmement lent et beaucoup trop limité ; de toute manière de nos jours tout le monde possède un disque dur. Vous devez posséder un contrôleur standard AT (16 bits). Il est possible en dépannage d'utiliser un vieux contrôleur de type XT (8 bits) ; cela dit la plupart des contrôleurs disques utilisés de nos jours sur les PC sont au standard AT. Linux supportera tous les contrôleurs MFM, RLL et IDE. La plupart (mais pas tous) des contrôleurs ESDI fonctionneront, pourvu qu'ils proposent l'émulation ST506.

La règle générale à retenir pour les contrôleurs de disques non SCSI et les contrôleurs de disquettes, est que si vous pouvez accéder au disque depuis MS-DOS ou un autre système d'exploitation, Linux saura le faire également.

Linux reconnait également un certain nombre de contrôleurs de disque durs SCSI parmi les plus populaires. Toutefois le support SCSI est un peu plus limité en raison de la très large gamme de matériel disponible. Les contrôleurs SCSI supportés comprennent (au minimum) l'Adaptec AHA1542B, AHA1542C, AHA1522, AHA1742A (BIOS version 1.34), AHA1740 (contrôleur SCSI-2, BIOS 1.34); Future Domain 1680, TMC-850, TMC-950; Seagate ST-02; UltraStor SCSI; Western Digital WD7000FASST, et de nouveaux pilotes pour de nouvelles cartes voient le jour régulièrement. Les clones basés sur ces cartes fonctionnent également.

1.8.4 Espace disque dur requis

Bien entendu, pour installer Linux, il vous faudra un peu de place libre sur votre disque dur. Si votre machine est équipée de plusieurs disques durs, vous pourrez allouer de l'espace pour Linux sur chacun d'entre eux si nécessaire.

La *quantité* d'espace disque dont vous aurez besoin dépend dans une large mesure de vos besoins et des programmes que vous comptez installer. Linux est relativement petit ; il est possible de faire tourner un système complet dans 10 ou 20 Mégaoctets d'espace disque. Mais si vous voulez avoir de quoi travailler confortablement, prévoir l'avenir, et si vous comptez utiliser de grosses applications comme X Window, il vous faudra bien plus de place disque. Si vous avez l'intention d'autoriser plusieurs utilisateurs à travailler sur la machine, vous aurez besoin de prévoir suffisament de place pour que chacun puisse travailler et stocker ses fichiers.

De plus, à moins que vous ayez une grande quantité de mémoire RAM (plus de 16 Mo), vous devrez probablement allouer une zone de swap, qui sera utilisée comme mémoire RAM virtuelle. Les détails à ce propos sont expliqués dans la section 2.2.3.

Avec chaque distribution particulière de Linux est généralement fournie un peu de littérature qui devrait vous aider à déterminer la place disque dont vous aurez besoin en fonction des programmes que vous allez installer. Vous pouvez utiliser un système minimal dans moins de 20 Mo ; un système complet avec tous les gadgets possibles dans moins de 80 Mo ; et un très gros système avec de l'espace pour de nombreux utilisateurs et de futures extensions dans 100-150 Mégaoctets. Encore une fois, ce ne sont que des approximations destinées à vous donner une idée des besoins de Linux ; la quantité d'espace disque nécessaire dépend de vos besoins et de ce que vous comptez faire avec Linux.

1.8.5 Écran et carte vidéo

Linux peut utiliser toutes les cartes vidéo standards rencontrées sur le PC : Hercules, CGA, EGA, VGA, Super VGA, IBM monochrome sur leur moniteur associé, pour l'affichage en mode texte utilisé par défaut. En règle générale, si le couple carte vidéo/moniteur fonctionne sur un autre système d'exploitation comme MS-DOS, alors il fonctionnera également très bien sous Linux.

Les environnements graphiques comme X Window ont par contre des besoins particuliers ; plutôt que d'en donner le détail ici, nous vous renvoyons à la section 5.1.1. Disons simplement ici que pour que X11 fonctionne sur votre machine, elle devra être équipée de l'une des cartes vidéo décrites dans cette partie.

1.8.6 Périphériques divers

Les sections précédentes décrivent le matériel qui est indispensable pour pouvoir utiliser Linux. Beaucoup d'utilisateurs possèdent un certain nombre de périphériques supplémentaires, comme les lecteurs de bandes magnétiques ou de CD-ROM, cartes sonores, souris, et souvent voudraient savoir s'ils fonctionneront sous Linux, ou non. Nous allons voir ça.

1.8.6.1 Souris et périphériques similaires

Vous n'utiliserez de souris pratiquement que sous une interface graphique comme X Window. Néanmoins, plusieurs applications Linux non graphiques font usage de la souris.

Linux supporte toutes les souris standard rencontrées sur PC, comme Logitech, MM series,Mouseman, Microsoft (deux boutons) et Mouse Systems (3 boutons). Linux reconnaît aussi les souris bus Microsoft, Logitech,et ATIXL. L'interface souris type PS/2 est également reconnue.

Tous les autres périphériques du même type comme les trackballs devraient fonctionner sans problèmes.

1.8.6.2 Lecteurs de CD-ROM

Pratiquement tous les lecteurs de CD-ROM sérieux utilisent l'interface SCSI. A partir du moment ou vous possédez un contrôleur SCSI supporté par Linux, votre CD-ROM fonctionnera. Certains lecteurs de CD-ROM bon marché peuvent ne pas fonctionner sur le bus SCSI, mais utiliser une interface particulière. Linux en reconnaît un certain nombre, consultez la documentation de votre distribution particulière pour vérifier ceux qui y sont supportés.

Linux utilise le sytème de fichiers standard ISO-9660 pour les CD-ROM.

1.8.6.3 Lecteurs de bandes magnétiques

Il existe différents types de lecteurs de cartouches magnétiques sur le marché. La plupart utilise l'interface SCSI, et par conséquent seront tous supportés par Linux sans complications. Echanger des données par cartouches magnétiques entre d'autres machines Unix (Sun par exemple) et Linux ne pose aucun problème. Toutefois, certains utilisateurs de PC possèdent des lecteurs ésotériques économiques non standard, qui n'utilisent pas l'interface SCSI. Certains d'entre eux sont supportés, là encore vérifiez dans votre distribution les pilotes dont vous disposez.

1.8.6.4 Imprimantes

Linux peut utiliser toute la gamme des imprimantes fonctionnant sur le port parallèle de la machine. Si vous pouvez utiliser votre imprimante sous MS/DOS, vous pourrez de la même façon l'utiliser sous Linux. L'impression sous Linux se fait de manière standard à l'aide de `lp` et `lpr`, vous permettant également l'impression à travers le réseau si vous en êtes équipé.

1.8.6.5 Modems

Tout comme pour le support imprimante, Linux peut utiliser toute la gamme de modems existants fonctionnant sur un port série du PC ; qu'il soit interne ou externe. Si votre modem est accessible par MS/DOS, il fonctionnera sous Linux sans difficulté. Il existe une très grande collection de programmes de communications sous Linux.

1.8.7 Cartes Ethernet

Beaucoup de cartes Ethernet parmi les plus populaires fonctionnent sous Linux. Entre autres :

- 3com 3c503, 3c503/16, 3c509
- Novell NE1000, NE2000
- Western Digital WD8003, WD8013
- Hewlett Packard HP27245, HP27247, HP27250
- D-Link DE-600

Les clones suivants fonctionnent également :

- LANNET LEC-45
- Alta Combo
- Artisoft LANtastic AE-2
- Asante Etherpak 2001/2003,
- D-Link Ethernet II
- LTC E-NET/16 P/N 8300-200-002
- Network Solutions HE-203,
- SVEC 4 Dimension Ethernet

– 4-Dimension FD0490 EtherBoard 16

Les clones compatibles avec les modèles ci-dessus devraient aussi fonctionner. D'autres pilotes voient le jour régulièrement et cette liste n'est bien sûr pas à jour.

1.9 Sources d'informations sur Linux

Comme vous l'avez probablement deviné, il y a beaucoup de sources d'informations à propos de Linux, en dehors de ce guide. En particulier, vous trouverez un grand nombre de livres, non spécifiques à Linux mais orientés sur UNIX en général, qui seront très utiles, particulièrement aux personnes n'ayant aucune expérience préalable d'UNIX. Si vous êtes nouveau dans le monde UNIX, nous vous suggérons absolument de prendre le temps de parcourir l'un de ces ouvrages avant de tenter de vous attaquer à Linux. En particulier, le livre *Learning the UNIX Operating System*, par Grace Todino et John Strang, est un très bon début.

Un bon nombre des sources d'informations citées plus bas sont disponibles en ligne sous une quelconque forme électronique. C'est à dire que vous devez avoir accès à un réseau comme l'Internet, Usenet, ou Fidonet, pour pouvoir accéder à ces informations. Si vous ne pouvez vraiment pas faire autrement, vous pouvez bien entendu demander à quelqu'un de vous envoyer des copies imprimées de ces documents.[8]

1.9.1 Documentation en ligne

Si vous avez un accès Internet, vous pourrez trouver beaucoup de documentation sur Linux disponible par FTP anonyme sur des sites du monde entier. Si vous n'avez pas directement accès à l'Internet, vous pouvez tout de même récupérer ces documents: ils sont diffusés sur bien d'autres réseaux, comme FidoNet et CompuServe. Si vous pouvez poster et recevoir du courrier électronique, vous pouvez également recevoir ces fichiers en utilisant un service `ftpmail`, voir en annexe C comment fonctionnent les serveurs `ftpmail`. Vous pourrez vous procurer aussi des distributions vendues sur CD-ROM qui contiennent tous les documents cités ici.

Il existe beaucoup de serveurs FTP disposant de Linux et des documents associés. Une liste des plus connus est donnée en annexe C. Afin de réduire le trafic international et les coûts induits, vous devez toujours utiliser le site qui est topologiquement le plus près de chez vous.

8. Cela dit, il faut vivre avec son temps. Pratiquement tout le monde possède le téléphone de nos jours, et le prix des modems performants est devenu du même ordre de grandeur que n'importe quel périphérique bon marché pour ordinateurs personnels. Un utilisateur possédant un micro-ordinateur moderne à domicile devrait avoir le réflexe de l'équiper d'un modem, tout comme il pense à l'équiper d'une souris ou d'un clavier!

L'annexe A contient une liste de quelques documents Linux qui sont disponibles en FTP anonyme. Les noms de fichiers peuvent différer, selon le site archive utilisé; la plupart des serveurs classent les documents relatifs à Linux dans un sous-répertoire `docs` de leur répertoire d'archivage Linux. Par exemple, sur le site FTP français `ftp.ibp.fr`, les fichiers Linux se trouvent dans `/pub/linux`, et les documentations dans `/pub/linux/docs`.

Citons quelques exemples de documents Linux disponibles en ligne, comme la *FAQ Linux*, (Frequently Asked Questions), une collection des questions les plus souvent posées concernant Linux; les documents Linux appelés *HOWTO* (Comment faire pour...) qui décrivent chacun un aspect particulier du système, comme l' *Installation HOWTO*, le *Printing HOWTO*, (concernant les imprimantes), et l'*Ethernet HOWTO* (pour les cartes réseau) ; ainsi que la META-FAQ Linux, une liste d'autres sources d'informations sur Linux disponibles sur l'Internet.

La plupart de ces documents sont également postés régulièrement dans l'un ou plus des forums Usenet dédiés à Linux; voir la section 1.9.4 un peu plus loin.

1.9.2 Linux sur le *the World Wide Web*

La page de sommaire des documentations Linux est disponible, pour les utilisateurs du Web, à l'URL :

```
http://sunsite.unc.edu/mdw/linux.html
```

Vous y trouverez de nombreux documents (dont les *HOWTO*) au format HTML, ainsi que des pointeurs vers d'autres sites concernant Linux.

1.9.3 Livres et autres publications

Pour l'instant, il y a encore très peu d'ouvrages traitant de Linux. L'essentiel de ce qui est disponible provient du Projet de Documentation Linux, un groupe de volontaires travaillant grâce aux possibilités offertes par l'Internet, sur un jeu complet de « manuels » pour Linux. Ces manuels sont semblables à la documentation généralement disponibles pour les versions commerciales d'UNIX: Ils traitent de l'ensemble du système, depuis l'installation jusqu'au développement du noyau.

Les manuels du projet de documentation Linux sont disponibles par FTP anonyme sur l'Internet, aussi bien que par commande postale auprès de différents fournisseurs. Il existe également une publication mensuelle, *Linux Journal*, distribué dans le monde entier ; vous trouverez dans l'annexe A les renseignements nécessaires pour vous y abonner.

Il n'y a pas beaucoup de littérature concernant Linux pour l'instant. Toutefois, il existe un très grand nombre de livres traitant d'UNIX en général qui s'appliquent parfaitement à Linux; tant qu'il s'agit d'utilisation ou de programmation, Linux offre peu de différences avec les autres implémentations d'UNIX. En résumé, pratiquement tout ce que vous voulez apprendre concernant l'utilisation du système et la programmation sous Linux peut être trouvé dans n'importe quel livre traitant de ces sujets sous UNIX. En fait, ce présent ouvrage est destiné à être complété par l'abondante littérature sur UNIX disponible en bibliothèque; ici nous ne présentons que les détails les plus spécifiques à Linux et nous espérons que vous approfondirez vos connaissances à l'aide d'autres publications traitant des systèmes UNIX.

Armé de quelques bons livres sur UNIX, ainsi que de celui que vous tenez entre les mains, vous devriez être capable de tout maîtriser. Vous trouverez une liste des lectures recommandées aussi bien pour les nouveaux venus à UNIX que pour les experts, en annexe A.

Il existe également un magazine mensuel consacré entièrement à Linux, le *Linux Journal*. Il est distribué dans le monde entier, et c'est un excellent moyen de se tenir au courant de ce qui se passe dans la communauté Linux, particulièrement si vous ne pouvez accéder à Usenet (voir plus bas). Vous trouverez en annexe A les moyens de vous abonner au *Linux Journal*.

1.9.4 Forums USENET

USENET est un système de forums électroniques répartis mondialement contenant un bon nombre de « newsgroups »[9] dédiés aux discussions concernant Linux. Le plus gros du développement de Linux s'est fait grâce à l'Internet et Usenet, il n'est donc pas surprenant de trouver une certaine quantité de forums dédiés à Linux.

Le tout premier newsgroup créé pour Linux fut `alt.os.linux`, et a été mis en place afin de déplacer les discussions sur Linux qui se tenaient dans `comp.os.minix` et différentes listes de diffusion vers un lieu plus approprié. Très rapidement, le trafic dans `alt.os.linux` est devenu si important qu'un groupe officiel dans la hiérarchie `comp` devint nécessaire; un vote fut organisé en février 1992 et le groupe `comp.os.linux` vit le jour.

`comp.os.linux` devint vite l'un des forums les plus populaires de Usenet, en tout cas le plus populaire de la hiérarchie `comp.os`. En décembre 1992, il fut décidé d'organiser un vote pour découper ce groupe en plusieurs forums plus spécialisés. Plus de 2000 personnes participèrent au vote concernant cette réorganisation de `comp.os.linux`, réalisant l'un des plus gros scrutins jamais vus sur Usenet. (Usenet a beau être mondial, en règle générale très peu de gens prennent la peine de voter. 200 ou 300 participants sont souvent tout ce que l'on obtient). Le trafic étant de plus en plus important, une nouvelle réorganisation a eu lieu en décembre 1994,

9. Le mot consacré dans le monde entier, quelle que soit la langue, pour désigner une zone particulière de discussion sur Usenet est newsgroup. En Français nous utiliserons indifféramment newsgroup, groupe, forum, groupe de discussion, voire conférence ou continuum, pour faire plaisir au ministre de la culture tout en conservant un style d'écriture suffisament clair et compréhensible.

afin de répartir les discussions dans de nouveaux groupes plus appropriés aux besoins des utilisateurs.

Si vous n'avez pas accès directement à Usenet, mais pouvez poster et recevoir du courrier électronique, il existe des passerelles vous permettant de recevoir ces groupes dans votre boîte aux lettres. A l'heure ou nous écrivons ces lignes, comptez une moyenne de 300 à 500 Kilooctets par jour pour l'ensemble des forums Linux.

`comp.os.linux.announce`
: `comp.os.linux.announce` est un groupe modéré destiné aux annonces importantes concernant Linux (comme les rapports de bogues, les modifications importantes, les nouvelles versions). Si vous n'avez pas le temps de lire les groupes Linux, lisez au moins celui-là, son trafic est faible. Souvent, les annonces postées dans ce groupe ne le sont pas ailleurs. Vous y trouverez également beaucoup des documents décrits dans la section précédente et listés dans l'annexe A, postés régulièrement tous les mois ou toutes les semaines selon le cas.

  Les messages destinés à ce groupe doivent être approuvés par ses modérateurs, actuellement Matt Welsh et Lars Wirzenius. Si vous voulez soumettre un article dans ce groupe selon les manipulations normales (en utilisant un lecteur de news standard), le programme dirigera automatiquement votre texte par courrier aux modérateurs pour approbation. Toutefois si vous utilisez un mauvais programme (comme ceux que certains peuvent hélas utiliser ailleurs que sous Unix), il se peut qu'il ne sache pas opérer ainsi et que vous deviez tout faire manuellement; dans ce cas l'adresse à utiliser est `linux-announce@tc.cornell.edu`.

`comp.os.linux.answers`
: C'est un groupe modéré destiné à la publication régulière des divers documents « HOWTO » et autres annonces périodiques ayant trait à la documentation, qui n'entrent pas dans le cadre du groupe prédédent.

  Tous les autres newsgroups cités ci-dessous ne sont pas modérés.

`comp.os.linux.misc`
: C'est le plus populaire de tous les groupes Linux. Il est destiné aux questions et réponses concernant l'utilisation, la configuration, ou tout autre problème général concernant Linux. Si vous êtes perdus, vous pouvez demander de l'aide dans ce forum, et vous recevrez probablement beaucoup de réponses de personnes qui vous dépanneront. Néanmoins, il est fortement recommendé de lire toute la documentation dont vous disposez avant de poser une question dans ce groupe.

`comp.os.linux.setup`
: Ce forum est dédié aux questions et discussions concernant l'installation et l'administration de Linux. Toute discussion concernant l'administration système sous Linux est la bienvenue dans ce groupe.

`comp.os.linux.development.system`
: Ce groupe a été créé pour les discussions concernant le développement du système Linux. Tout ce qui concerne le développement du noyau ou qui lui est intimement lié doit être discuté dans ce forum. Par exemple, si vous écrivez un pilote de périphérique et que vous ayez besoin d'éclaircissements sur certains aspects de cette programmation, c'est l'endroit idéal pour obtenir les renseignements nécessaires. Il est aussi destiné aux discussions concernant le futur de Linux, les options et orientations à choisir, comme nous le décrivons dans la section « 1.6. »

`comp.os.linux.development.apps`
: Destiné aux discussions concernant le développement ou le portage d'applications sous Linux. Les programmeurs réalisant des applications auront tout intérêt à fréquenter ce groupe.

`comp.os.linux.hardware`
: Tout ce qui concerne la compatibilité de Linux avec le matériel, périphériques, doit être traité dans ce groupe. Consultez-le avant tout achat important, ou pour obtenir des renseignements sur la configuration particulière de tel ou tel périphérique pour son fonctionnement sous Linux.

`comp.os.linux.networking`
: Dans ce groupe, ont lieu les discussions ayant trait au réseau sous Linux; aussi bien à propos de l'implémentation courante de TCP/IP que des différents programmes et serveurs réseau disponibles.

`comp.os.linux.advocacy`
: Ici prennent place toutes les discussions qui ne concernent aucun des groupes précédents. En particulier, les querelles de clocher (comme la sempiternelle polémique « Linux contre Window NT ») doivent se tenir uniquement dans ce forum. Toute discussion non technique ou générale à propos de Linux doit rester dans `comp.os.linux.misc`.

`fr.comp.os.linux`
: Tous les groupes que nous venons de voir sont internationaux et bien entendu, en langue anglaise. Le forum `fr.comp.os.linux`

est particulier, c'est un groupe francophone qui fait partie de la hiérarchie `fr.*` de Usenet. Chaque pays peut en effet créer des hiérarchies locales dont la diffusion est limitée, afin de pouvoir utiliser sa langue maternelle. La hiérarchie `fr.*` est diffusée dans l'essentiel des pays francophones, comme le Québec, la Belgique, la Suisse Romande, et bien entendu la France. On peut dans `fr.comp.os.linux` discuter de Linux en français, et qui plus est utiliser les caractères accentués, pour peu que l'on respecte la norme ISO-8859-1, dite ISO Latin-1, qui est celle utilisée sur la plupart des ordinateurs et systèmes supportant les caractères 8bits (à l'exception de MS-DOS).

Ce groupe `fr.comp.os.linux` est vraiment particulier. En effet, Usenet n'étant pas très répandu en France, les discussions sur Linux commencèrent en 1992 sur les BBS (Bulletin Board Systems), et une conférence répartie sur l'essentiel des BBS Français fut créée à l'initiative de deux « pionniers de Linux » en France, René Cougnenc et Nat Makarévitch. Petit à petit cette conférence prit de l'ampleur, Linux devint populaire, et il fut décidé de relier par une passerelle Usenet/BBS cette conférence très active à un nouveau newsgroup créé pour la circonstance, `fr.comp.os.linux`. Depuis, Linux réunit en France par l'intermédiaire de ce forum, deux mondes aussi différents que celui de Usenet et des BBS, dans la plus parfaite harmonie; ce qui mérite d'être signalé car ce type d'association a souvent tendance à causer des polémiques de part et d'autres.

Vous devez noter que le newsgroup `comp.os.linux`, qui était l'unique groupe Linux autrefois, a été remplacé par la nouvelle hiérarchie que nous venons de voir. Si vous avez accès à un groupe nommé `comp.os.linux`, et à aucun autre, vous devez absolument prévenir votre administrateur système car c'est une erreur d'administration, il doit créer les nouvaux groupes dans la hiérarchie, et interdire l'usage de `comp.os.linux`, seul, qui n'existe plus depuis le début de l'année 1993...! De même, depuis le mois de mars 1995, les groupes `comp.os.linux.help`, `comp.os.linux.development` et `comp.os.linux.admin` n'existent plus.

1.9.5 Listes de diffusion Internet

Si vous avez un accès à Internet ou simplement au courrier électronique, vous pouvez souscrire à un certain nombre de listes de diffusion (mailing-lists) même si vous ne pouvez accéder aux newsgroups Usenet. Notez bien que si vous n'êtes pas directement relié à l'Internet, vous pouvez recevoir ces listes de diffusion tant que vous pouvez échanger du courrier électronique par un moyen quelconque (par exemple UUCP, FidoNet, CompuServe, et bien d'autres services).

La liste de diffusion nommée « Linux Activists » est en premier lieu destinée aux développeurs de Linux et aux personnes désireuses de les aider. C'est une liste de diffusion « multi-canaux » , et vous pouvez souscrire à un ou plusieurs « canaux » en fonction de vos intérêts particuliers. Quelques uns des nombreux canaux disponibles sont: `NORMAL`, pour les discussions générales; `KERNEL`, consacré au développement du noyau; `GCC`, pour ce qui est relatif au compilateur `gcc` et aux bibliothèques; `NET`, pour ce qui concerne le code réseau TCP/IP; `DOC`, pour la rédaction et la distribution de la documentation Linux; et beaucoup d'autres encore.

Pour plus d'informations à propos de la liste de diffusion Linux-Activists, envoyez un courrier électronique à l'adresse

```
linux-activists@niksula.hut.fi
```

Vous recevrez alors une liste des canaux disponibles, et les instructions nécessaires pour souscrire ou vous désabonner aux canaux particuliers qui vous intéressent.

Quelques autres listes de diffusions très spécifiques à propos de Linux existent également. La meilleure façon de les trouver est de lire les annonces faites dans les forums Usenet consacrés à Linux, ou bien de lire la récapitulation des listes de diffusion publiques disponibles sur tous sujets, postée régulièrement dans le groupe Usenet `news.answers`.

1.10 Obtenir de l'aide

Vous aurez à coup sûr besoin d'un peu d'assistance lors de vos aventures dans le monde de Linux. Même le spécialiste UNIX le plus expérimenté pourra de temps à autres tomber dans un piège dû à une caractéristique ou un défaut de Linux, et il est bon de savoir comment et où, obtenir de l'aide lorsque vous en aurez besoin.

La première méthode pour trouver de l'aide concernant Linux est de fréquenter les newsgroups Usenet et les mailing-lists dédiés à Linux, comme il est expliqué dans la section 1.9. Si vous ne pouvez avoir accès à ces sources d'informations, vous pourrez sans doute trouver des discussions sur Linux à peu près comparables, sur d'autres services comme les BBS locaux, CompuServe, etc...

Un certain nombre d'entreprises proposent un support commercial de Linux. Moyennant une souscription financière, vous pourrez téléphoner à ces consultants pour trouver de l'assistance. L'annexe B contient une liste de services commerciaux consacrés à Linux. Mais si vous avez accès à Usenet ou au courrier électronique, vous trouverez probablement autant d'aide gratuite par ce biais.

Afin d'être sûr de trouver l'aide dont vous avez besoin, et d'améliorer sans cesse vos connaissances, ayez toujours en tête les suggestions suivantes :

Consultez toute la documentation disponible... avant tout ! La première chose à faire lorsque vous rencontrez un problème est de consulter les différentes sources d'informations décrites dans la section 1.9 et l'annexe A. Ces documents ont demandé beaucoup de travail, et ont été réalisés pour des personnes comme vous, qui ont besoin d'aide concernant Linux. Même les livres écrits pour UNIX en général sont applicables à Linux, et ils vous apprendront beaucoup de choses. Il y a de fortes chances que vous trouviez quelque part dans toute cette documentation, la solution aux problèmes auxquels vous êtes confrontés, aussi impossible que cela puisse paraître.

Si vous avez accès aux news Usenet ou à une liste de diffusion Linux, surtout *lisez* attentivement les informations qui s'y trouvent avant de poser une question à tout le monde. Très souvent, la solution à des problèmes courants n'est pas facile à trouver dans la documentation, mais ils sont fréquemment traités dans les newsgroups relatifs à Linux. Si vous ne faites que poser des questions dans ces groupes, sans jamais lire ce qui s'y passe, attendez vous à des réactions désagréables de la part des autres utilisateurs.

Apprenez à vous débrouiller par vous même. Il est préférable dans beaucoup de cas de faire l'effort de rechercher soi-même la cause d'un problème avant de le soumettre à la communauté. Après tout, vous saviez à quoi vous attendre lorsque vous avez décidé d'installer Linux la toute première fois ! Rappelez-vous que Linux est l'un des rares systèmes fourni avec ses sources, et il vous permet donc de corriger vous même les bogues rencontrés. Ce n'est pas un programme commercial et il n'est pas destiné à le devenir. Programmer un peu ne vous tuera pas. En fait, de cette façon vous apprendrez beaucoup sur le système, et peut être qu'ainsi vous deviendrez un jour un « gourou » Linux. Ne comptez pas obtenir un système Linux complet, parfait, conforme en tous points à ce que vous désirez, sans y mettre un petit peu du vôtre.

Restez calme. Il faut éviter à tout prix de s'énerver et d'en arriver à insulter le système. Taper sur l'ordinateur ne vous servira à rien. Les auteurs, par expérience, conseillent de passer ses nerfs sur tout objet peu fragile et déjà inanimé, l'idéal étant un « punching-ball » . Linux évoluant de plus en plus et les distributions devenant meilleures avec le temps, nous espérons qu'à terme ces problèmes disparaîtront. En attendant, lorsque cela vous arrive, asseyez-vous, faites une pause, buvez un verre, et reprenez vos expérimentations lorsque vous vous sentirez parfaitement détendu.

Retenez-vous de poser des questions trop spontanément. De nombreuses personnes font l'erreur d'écrire pour demander de l'aide prématurément. Lorsqu'un problème survient, *ne vous ruez pas* sur le premier terminal venu pour exposer vos ennuis dans l'un des forums publics Linux. Très souvent vous vous rendrez compte de votre erreur dans les cinq minutes qui suivent l'envoi de votre message, et vous vous verrez obligé de poster un rectificatif et vous pourriez bien avoir l'air ridicule... Avant d'écrire au monde entier, vérifiez bien quel est votre problème et s'il ne s'agit pas d'une erreur de votre part. Est-ce-que votre système semble se mettre en marche lorsque vous l'allumez ? Peut être que l'ordinateur est débranché, simplement.

Si vous demandez de l'aide, faites le correctement. Si vous ne vous en sortez vraiment pas, vous posterez alors un message dans l'un des forums publics dédiés à Linux, sur Usenet par exemple. Rappelez-vous que les personnes qui liront votre appel au secours ne sont pas à votre disposition. Par conséquent il est important de rester courtois, et aussi informatif que possible.

Comment faire ? Tout d'abord, donnez le plus d'informations appropriées sur le système que vous possédez, et décrivez clairement votre problème. Poster un simple message comme « Je n'arrive pas à expédier un courrier » n'a aucun sens si vous n'indiquez pas quels programmes vous utilisez, leur version, la façon dont ils sont configurés, ce que vous désiriez faire avec et ce que vous obtenez en réalité. Sans faire une liste de plusieurs pages, il est souvent utile de préciser quelle version de Linux vous utilisez, quelle est votre configuration matérielle, le problème peut très bien être relatif à une certaine carte d'extension et être déjà connu.

Ensuite, n'oubliez pas que vous devez tenter (ne serait-ce qu'un tout petit peu) de résoudre vos ennuis par vous même avant tout cela. Si, par exemple, vous n'avez jamais essayé de configurer le courrier électronique dans votre vie, et que vous commencez par demander aux utilisateurs de Usenet comment faire, vous commettez une grave erreur. Il y a un certain nombre de documents disponibles (voir section 1.9) destiné à vous initier aux tâches d'administration les plus courantes sous Linux. Vous devez toujours essayer d'aller le plus loin possible tout seul, et ne demander de l'aide que lorsque vraiment vous êtes totalement dépassé.

Notez bien également que les personnes qui vont lire votre message, peuvent être parfois exaspérés de voir sempiternellement la même question posée; lisez bien auparavant ce qui s'est passé dans le forum où vous écrivez, très souvent la solution à la question qui vous préoccupe a été donnée une dizaine de fois les semaines précédentes; un rapide coup d'œil vous aurait permis de la découvrir.

Enfin, lors de l'utilisation de forums électroniques, restez aussi poli que possible et n'entrez pas dans le jeu des stupides querelles d'opinion que vous pourrez, hélas, trop souvent rencontrer. Cela fera économiser du temps à tout le monde, et de l'argent à beaucoup de lecteurs qui payent très cher pour avoir accès à ces sources d'informations. Le réseau est le moyen idéal pour obtenir de l'aide sur Linux, mais il est très important de savoir comment l'utiliser *efficacement*.

Chapitre 2

Se procurer et installer Linux

Nous décrirons dans ce chapitre comment se procurer l'une des différentes distributions toutes faites de Linux, et comment installer la distribution que vous avez choisie.

Comme nous l'avons déjà signalé, il n'existe pas une seule et unique distribution « officielle » de Linux, il y a en fait beaucoup de distributions indépendantes, chacune réalisée dans un but différent. Ces distributions sont disponibles par FTP anonyme sur l'Internet, sur de nombreux BBS du monde entier, et par commande postale auprès de certains distributeurs, sur support disquette, bande magnétique, ou CD-ROM.

Nous ne présenterons qu'un aperçu général des procédures d'installation. Chaque distribution contient ses propres directives d'installation, mais armé des concepts qui vous sont présentés ici vous devriez pouvoir vous débrouiller facilement. L'annexe A recense différentes sources d'informations concernant l'installation au cas où vous seriez totalement perdu.

Cette version du livre contient des sections supplémentaires relatives à la distribution Slackware de Linux.

2.1 Distributions de Linux

Puisque Linux est libre, aucune organisation ou entité unique n'est responsable de la distribution d'une version « officielle » . Par conséquent tout le monde est libre d'assembler tous les programmes composant un système Linux complet et de le diffuser, tant que les conditions requises par la GPL sont observées. La conséquence de cette situation, est qu'il existe beaucoup de distributions différentes de Linux, disponibles par FTP anonyme ou par correspondance.

Vous devez vous décider sur l'une ou l'autre de ces distributions, suivant vos besoins; elles ne se ressemblent pas toutes. Certaines permettent l'installation d'un système plus que complet, d'autres un peu moins, et quelques unes sont de « petites » distributions contenant le minimum nécessaire, permettant d'installer Linux dans très peu d'espace disque. Beaucoup de distributions ne contiennent que ce qui compose réellement un système UNIX, et considèrent que vous devrez installer vous même le reste des grosses applications que vous désirerez utiliser, comme par exemple le système X-Window. (Nous vous montrerons comment faire dans le chapitre 4).

Le document Linux *Distribution HOWTO* (voir l'annexe A) contient une liste des différentes distributions disponibles aussi bien gratuitement sur l'Internet, qu'en achat par correspondance. Vous trouverez également dans l'annexex B une liste de certains distributeurs vendant le système de cette manière. Si vous avez acheté ce livre sous forme imprimée, vous devriez pouvoir vous procurer une version de Linux au même endroit.

Comment choisir ? Si vous avez accès aux news Usenet ou à quelque autre système de forums équivalents, vous pouvez demander en public les opinions personnelles de personnes ayant déjà installé Linux. Mieux encore, si vous connaissez quelqu'un qui a installé Linux, demandez-lui des conseils. Différents facteurs entrent en ligne de compte lors du choix d'une distribution binaire de Linux, chaque utilisateur a ses besoins et opinions propres. Actuellement, la plupart des distributions les plus populaires contiennent à peu près la même chose, votre choix peut donc tout aussi bien être totalement arbitraire.

Cette version de ce livre contient des informations sur l'installation des célèbres distributions « Slackware » et « Slackware Pro » de Linux.

2.1.1 Se procurer Linux par l'Internet

Si vous disposez d'un accès Internet, la façon la plus simple de se procurer Linux est par FTP anonyme.[1] L'annexe C recense un certain nombre de sites FTP qui archivent le système Linux. L'un deux, situé en France, est **ftp.ibp.fr** et propose les différentes distributions binaires de Linux dans le répertoire

```
/pub/linux/distributions
```

Beaucoup de distributions se présentent sous forme de fichiers images de disquettes. C'est-à-dire que vous trouvez un certain nombre de fichiers, chacun d'eux étant l'image binaire d'une disquette complète. Afin de reconstituer cette disquette à

1. Si vous n'avez pas d'accès direct au réseau mais que vous pouvez recevoir du courrier électronique en grosse quantité, vous pouvez obtenir Linux par un service **ftpmail**; voir l'annexe C pour plus de détails.

partir du fichier, vous pouvez utiliser le programme `RAWRITE.EXE` sous MS-DOS; ce programme copie indépendamment du format de la disquette, bloc par bloc, le contenu du fichier. [2]

`RAWRITE.EXE` est bien entendu disponible sur les divers sites FTP diffusant Linux.

Par conséquent dans la plupart des cas, il vous suffit de télécharger le jeu de fichiers images, et utiliser `RAWRITE.EXE` sur chaque image pour reconstituer un jeu complet de disquettes d'installation Linux. Vous amorcez la machine sur la disquette en général appelée « disquette de boot » Et vous êtes parés. Le système est en général installé directement depuis les disquettes, toutefois certaines distributions autorisent l'installation depuis une partition MS-DOS du disque dur ou par un réseau TCP/IP. La documentation de chaque distribution décrit les méthodes d'installation proposées.

D'autres distributions s'installent à partir d'un jeu de disquettes qui sont au format de MS-DOS. Par exemple, la distribution Slackware de Linux ne demande que deux disquettes spéciales réalisée par `RAWRITE.EXE`: la disquette d'amorçage, et le système de fichiers racine (« boot et root »). Le reste des fichiers sont recopiés sur des disquettes MS-DOS ordinaires à l'aide de la commande `COPY` de ce système. La procédure d'installation saura lire ces disquettes, ce qui évite d'avoir à utiliser maintes fois le très lent programme `RAWRITE.EXE`; néanmoins cela vous oblige à avoir accès à un système MS/DOS ou une machine sachant créer de telles disquettes.

Chaque distribution de Linux disponible en FTP anonyme contient (en principe) un fichier nommé `README`, c'est un fichier texte qui décrit comment télécharger et préparer les disquettes pour l'installation. Assurez-vous d'avoir bien lu toute la documentation mise à votre disposition; le guide que vous lisez en ce moment n'est là que pour vous donner des indications générales.

Lorsque vous téléchargez les fichiers, assurez-vous bien d'utiliser le mode *binary* (binaire) pour le transfert des données. Ce mode est mis en service par la commande « `binary` » sur la plupart des clients FTP.

Consultez la section 2.1.4 pour savoir comment vous procurer la distribution Slackware sur l'Internet.

2.1.2 Se procurer Linux depuis d'autres sites informatiques

Si vous avez accès à d'autres réseaux d'ordinateurs comme CompuServe ou Prodigy, il doit bien y avoir moyen d'y télécharger Linux. De plus, de nombreux bulletin board systems (BBS) de part le monde proposent le téléchargement de

2. Si vous avez accès à une station de travail UNIX équipée d'un lecteur de disquettes, vous pouvez aussi utiliser la commande standard `dd` pour copier le fichier image directement sur la disquette. Une commande comme « `dd of=/dev/rfd0 if=bidule bs=18k` » écrira le contenu du fichier `bidule` sur le lecteur disquette d'une station Sun par exemple. Consultez vos gourous UNIX locaux pour plus d'informations sur l'utilisation de la commande `dd`.

Linux. Une liste de ces BBS est proposée dans l'annexe D. Ces sites ne proposent pas toutes les distributions de Linux existantes, et il est bon de vérifier que les fichiers proposés sont suffisament récents et complets.

2.1.3 Se procurer Linux par correspondance

Si vous n'avez accès ni à l'Internet ni à un simple BBS, un certain nombre de distributions sont vendues par correspondance sur support disquettes, bande, ou CD-ROM. Vous trouverez une liste de quelques uns de ces distributeurs dans l'annexe B. Beaucoup d'entre eux acceptent les cartes de crédit, vous pouvez donc commander chez eux, même si vous n'êtes pas aux États-Unis ou au Canada.

Linux est libre et totalement gratuit, mais la GPL autorise les distributeurs à demander une rétribution quelconque. Par conséquent, commander Linux par correspondance vous coûtera de l'argent. Regardez bien les tarifs qui vous seront proposés, s'ils vous paraissent excessifs par rapport au service rendu, évitez de commander. En général le prix varie entre 90 et 700 francs selon le type de distribution et de service. Bien sûr, si vous connaissez quelqu'un qui a déjà acheté une distribution, vous êtes libre de l'emprunter et de la recopier autant de fois qu'il vous plaira. Les distributeurs de Linux n'ont pas le droit de restreindre la licence de distribution GNU, sous aucune forme que ce soit. Linux est libre et gratuit.

2.1.4 La distribution Slackware

La distribution Slackware est l'une des distributions de Linux les plus connues, elle est très à jour et contient une grande quantité de programmes supplémentaires, dont X Window, TeX, et bien d'autres. Elle est constituée d'un certain nombre de « paquetages disquettes », chacun contenant un type particulier de programme (par exemple, le paquetage `d` contient les outils de développement comme le compilateur `gcc`, le paquetage `x` contient le système X Window, etc). Vous pouvez choisir d'installer n'importe quel ensemble, à tout moment. L'installation est facile, et la distribution peut être obtenue aussi bien par téléchargement sur l'Internet ou différents serveurs, que sur support CD-ROM auprès de nombreux distributeurs dont une liste non exhaustive est donnée dans l'annexe B.

La version décrite ici est la 2.1.0, du 30 octobre 1994. L'installation des versions ultérieures devrait être très similaire, la plupart des informations données ici seront donc utiles assez longtemps.

2.1.4.1 Les paquetages

Malheureusement, Slackware ne maintient pas une liste complète de la place disque requise par chaque paquetage. Vous aurez besoin au minimum de 7 mégaoctets pour installer juste la série de disquettes A; une grossière approximation pourrait être 2 ou 2,5 Mo par disquette.

Sond disponibles les paquetages suivants :

`A` Le système de base. Suffisant pour obtenir un système qui fonctionne, et avoir `elvis` et des programmes de communications. Basé sur le noyau Linux 1.0.9 et la nouvelle arborescence de fichiers standard (FSSTND).

Ces diques sont prévus pour tenir sur des disquettes 1.2Mo, le reste de la Slackware demande par contre du 1.44Mo. Si vous n'avez qu'un lecteur de 1.2Mo, vous pouvez tout de même installer un système de base, et télécharger les autres paquetages dont vous avez besoin, pour les installer directement depuis votre disque dur.

`AP` Différentes applications et améliorations, comme les pages de manuel, `groff`, `ispell` (versions GNU et internationales), `term`, `joe`, `jove`, `ghostscript`, `sc`, `bc`, ainsi que les patches de gestion des quota.

`D` Environnement de développement. GCC/G++/Objective C 2.5.8, `make` (GNU et BSD), `byacc` et GNU `bison`, `flex`, les librairies C 4.5.26 , `gdb`, sources du noyau, `SVGAlib`, `ncurses`, `clisp`, `f2c`, `p2c`, `m4`, `perl`, `rcs`.

`E` GNU Emacs 19.25.

`F` Une collection de FAQ (*foire aux questions*) et documentations diverses.

`I` Pages au format Info des programmes GNU. Documentation lisible par `info` ou Emacs.

`N` Réseau. TCP/IP, UUCP, `mailx`, `dip`, `deliver`, `elm`, `pine`, `smail`, `cnews`, `nn`, `tin`, `trn`.

`OOP` Programmation Orientée Objet. GNU Smalltalk 1.1.1, et l'interface Smalltalk pour X. (STIX)

`Q` Sources des versions ALPHA des noyaux en cours de développement, la mise à jour se fait le plus souvent possible.

`TCL` Tcl, Tk, TclX, blt, itcl.

`Y` Jeux. La collection BSD, et Tetris en version terminal texte.

`X` Le système de base XFree86 3.1, avec `libXpm`, `fvwm` 1.20, et `xlock` en supplément.

XAP Applications X: X11 `ghostscript`, `libgr13`, `seyon`, `workman`, `xfilemanager`, `xv` 3.01, GNU `chess` et `xboard`, `xfm` 1.2, `ghostview`, et divers jeux sous X-Window.

XD Programmation X11. Bibliothèques X11 , kit de reconstruction du serveur X, support PEX.

XV Xview 3.2 release 5. Bibliothèques , et les gestionnaires de fenêtres avec ou sans bureau virtuel *Open Look*.

IV Bibliothèques Interviews, comprenant les fichiers, ainsi que `doc`, `idraw` et applications. Tout cela est extrêmement lent sur ma machine, mais il peut être intéressant d'y jeter un oeil.

T Le système de formatage de textes TeX et LaTeX (avec lequel ce livre est réalisé, soit dit en passant).

Vous devez posséder l'ensemble "`a`", le reste est optionnel. Nous vous suggérons d'installer au moins les paquetages `a`, `ap`, and `d` , ainsi que le `x` si vous comptez utiliser X Window.

2.1.4.2 Téléchargement sur l'Internet

La distribution Slackware de Linux peut être récupérée sur de nombreux sites FTP du monde entier. L'annexe C en recense un certain nombre; nous vous suggérons de récupérer Linux depuis un site proche de chez vous, pour réduire le trafic sur les lignes internationales (en France: `ftp.ibp.fr`). Cela dit, les deux sites de référence sont `sunsite.unc.edu` and `tsx-11.mit.edu`.

The Slackware release may be found at least on the following sites:

- `ftp.ibp.fr:/pub/linux/distributions/slackware`
- `sunsite.unc.edu:/pub/Linux/distributions/slackware`
- `tsx-11.mit.edu:/pub/linux/packages/slackware`
- `ftp.cdrom.com:/pub/linux/slackware`

`ftp.cdrom.com` est le site officiel de Slackware. `ftp.ibp.fr` reçoit un miroir journalier de ce site.

2.1.4.2.1 Téléchargement des fichiers Vous devrez télécharger les fichiers suivants, depuis le site FTP de votre choix. Vérifiez bien que vous utilisez le mode de transfert binaire dans le protocole FTP ! Consultez l'annexe C si vous comprenez pas cette dernière phrase.

- Les différents fichiers **README**, ainsi que le fichier **SLACKWARE_FAQ**. Lisez absolument ces fichiers avant d'essayer d'installer Linux, afin d'être informé de toute mise à jour ou changements intervenus dans ce document.

- Un fichier image de disquette d'amorce. Il s'agit d'un fichier que vous écrirez sur une disquette pour recréer la disquette d'amorçage Slackware. Si vous avez un lecteur de disquettes 1.44 Mo (3.5"), regardez dans le répertoire `bootdsks.144`. Si vous ne possédez qu'un lecteur 1.2 (5.25"), allez dans le répertoire `bootdsks.12`.

  Vous aurez besoin de l'un des fichiers suivants. (Le fichier READ.ME dans ces répertoires contient de une liste descriptive à jour) :

  - `bare.gz`. C'est une disquette de d'amorce qui ne contient que les pilotes disques IDE. (Pas de SCSI, CDROM ou support réseau). Utilisez cela si vous avez un disque dur IDE et ne comptez pas installer Linux à travers le réseau, ou depuis un CD-ROM.
  - `xt.gz`. Le même que le précédent, avec en plus un pilote pour les anciens contrôleurs disques durs de type PC-XT.
  - `cd.gz`. Contient des pilotes disque dur IDE et CD-ROM non SCSI. Utilisez celui-ci si vous comptez installer depuis un CD-ROM qui n'est pas SCSI. (Utile uniquement si vous avez acheté Slackware sur un CD/ROM particulier).
  - `cdscsi.gz`. Contient des pilotes IDE et disque dur SCSI, ainsi que pour des CD-ROMS non SCSI.
  - `scsi.gz`. Contient des pilotes IDE et SCSI. (Donc CD/ROMS SCSI uniquement).
  - `net.gz`. Contient des pilotes IDE, et le support réseau TCP/IP. Utilisez celui-ci si vous allez installer Linux par réseau en utilisant NFS.
  - `scsinet.gz`. Contient les pilotes : disques IDE et SCSI, SCSI CD-ROM, et réseau TCP/IP. Il est conseillé de tenter d'utiliser celui-ci en premier, essayez `bare.gz` ou un autre s'il ne marche pas pour vous.

  Vous n'aurez besoin que d'**un seul** de ces fichiers images, en fonction de la configuration de votre ordinateur.

  Le problème est que certains pilotes entrent parfois en conflit les uns avec les autres, et au lieu de passer son temps à tenter de régler un problème de configuration sur votre machine, il est plus simple d'utiliser une disquette de d'amorce qui ne contienne que certains pilotes. La plupart des utilisateurs peuvent toutefois utiliser sans problèmes `scsi.gz` ou `bare.gz`.

- Un fichier image de disquette « racine ». Il s'agit d'un fichier que vous écrivez sur une disquette pour recréer la disquette d'installation Slackware. Comme avec l'image d'amorçage, regardez dans `rootdsks.144` ou `rootdsks.12` en fonction de votre lecteur de disquettes.

  Vous aurez besoin de l'un des fichiers suivants :

  - `color144.gz`. Une disquette d'installation utilisant des menus en couleurs, pour lecteur de disquettes 1.44 Mo. La plupart des utilisateurs pourront utiliser cette version.
  - `umsds144.gz`. Une version du disque `color144`, pour l'installation du système de fichiers UMSDOS, qui permet d'installer Linux directement sur une partition MS-DOS existante. Cette méthode n'est pas présentée en détail pour l'instant, disons simplement qu'au prix d'une légère dégradation des performances des accès disques, elle permet d'installer Linux sans repartitionner votre disque dur si celui-ci est déjà entièrement réservé à MS-DOS.
  - `tty14.gz`. Disquette d'installation utilisant un mode ligne de commandes pour lecteurs de disquettes 1.44 Mo. Mieux vaut utiliser la version `color144.gz`, mais quelques personnes ont signalé avoir eu des problèmes sur leur machine. Si `color144.gz` ne fonctionne pas pour vous, essayez `tty144.gz` à la place. La procédure d'installation est différente et plus ancienne, mais devrait fonctionner si `color144.gz` marche mal.
  - `colrlite.gz`. Une disquette d'installation utilisant des menux en couleurs, pour lecteur de disquettes 1.2 Mo. Quelques programmes ont été supprimés pour que tout tienne sur ce type de support, mais ça devrait suffire si vous ne possédez qu'un lecteur de ce type.
  - `tty12.gz`. L'installation en mode ligne de commandes pour lecteur de disquettes 1.2 Mo. Utilisez ce fichier si vous n'avez qu'un lecteur 1.2 Mo et que `colrlite.gz` ne fonctionne pas pour vous.
  - `umsds12.gz`. Une version du disque `colrlite` destiné au système de fichiers UMSDOS. Voir `umsds144.gz`, plus haut.

  Encore une fois, vous n'avez besoin que d'*un seul* de ces fichiers, en fonction de votre configuration.

- `GZIP.EXE`. Il s'agit d'une version exécutable sous MS/DOS, de `gzip`, le programme de compression de données utilisé pour compacter les images des disquettes d'amorce et de système de fichiers racine. (L'extension `.gz` indique cela).

- `RAWRITE.EXE`. Il s'agit d'un programme exécutable sous MS/DOS qui écrit le contenu d'un fichier directement sur une disquette, sans s'occuper de la façon dont elle est formatée. (La disquette est considérée comme un grand fichier continu). Vous utiliserez `RAWRITE.EXE` pour recréer les disquettes d'amorçage et de système de fichiers racine.

Vous n'aurez besoin de `RAWRITE.EXE` et `GZIP.EXE` que si vous comptez recréer les disquettes « boot » et « root » à partir d'un ordinateur tournant sous MS/DOS. Si vous avez accès à une machine utilisant UNIX, une station de travail par exemple, contenant un lecteur de disquettes, vous pourrez directement créer ces disquettes dessus en utilisant simplement la commande `dd`. Si vous ne connaissez pas Unix, demandez de l'aide aux gourous autour de vous.

- Les fichiers des répertoires `slakware/a1`, `slakware/a2`, `slakware/a3` et `slakware/a4`. Ces fichiers constituent le paquetage « `a` » de la distribution Slackware. Ils sont indispensables. Plus tard, vous copierez ces fichiers sur des disquettes formatées au standard MS/DOS pour les installer (ou bien vous pourrez les installer directement depuis votre disque dur). Toutefois, lorsque vous téléchargerez ces fichiers, conservez-les dans des répertoires séparés. Ne mélangez pas les fichiers `a1` avec les fichiers `a2`, par exemple.

  Assurez-vous de transférer les noms de fichiers correctement; c'est-à-dire, sous FTP, utilisez la commande « `mget *` » au lieu de « `mget *.*` ».

- Les fichiers des répertoires `ap1`, `ap2`, etc., en fonction des paquetages que vous voulez installer. Par exemple, si vous installez la série « `x` », récupérez les fichiers dans les répertoires `x1` jusqu'à `x5`. Comme précédemment, assurez-vous de conserver ces fichiers dans des répertoires séparés lors du téléchargement.

2.1.4.3 Slackware sur CD-ROM

La distribution Slackware est aussi disponible sur CD-ROM. La plupart des disques contiennent une copie des fichiers tels qu'ils apparaissent sur les sites d'archive FTP, c'est-à-dire exactement comme nous venons de le décrire. Par conséquent, si vous disposez d'un CD-ROM contenant cette distribution, vous êtes en possession de tout le nécessaire.

Il vous faudra créer les disquettes d'amorçage et racine à partir des fichiers du CD-ROM. Consultez la section 2.1.4.2.1, ci-dessus.

Tout d'abord, vous devrez choisir quelles versions de ces fichiers utiliser. Ensuite, suivez les instructions qui vont suivre pour réaliser ces disquettes.

2.1.4.4 Méthodes d'installation

Slackware propose différents moyens pour installer les programmes. La méthode la plus populaire auprès des novices est l'installation à partir d'une partition MS-

DOS de votre disque dur, ou bien d'installer depuis une série de disquettes MS-DOS créées à partir des fichiers que vous avez téléchargés.

Si vous disposez d'un CD-ROM, vous pouvez procéder à l'installation directement depuis ce support. La distribution *Slackware Pro*, de *Morse Telecommunications*, vous permet même d'utiliser une partie du système depuis le CD-ROM, ce qui peut vous permettre d'économiser beaucoup de place sur votre disque dur ; le seul inconvénient étant l'extrême lenteur de ce type de support.

Vous pouvez également installer Slackware depuis un système de fichiers monté par NFS, à travers un réseau TCP/IP. Toutefois cette méthode déborde du cadre de ce document. Si vous avez besoin d'aide pour cela, demandez aux gourous Unix autour de vous la manière de paramétrer votre système pour monter une partition NFS depuis laquelle vous pourrez installer Slackware. (Pour cela, vous aurez besoin d'une autre machine sur le réseau, contenant une distribution Slackware exportable par NFS).

2.1.4.4.1 Création des disquette « boot » et « root »

Vous devez réaliser des disquettes à partir des fichiers images bootdisk et rootdisk que vous avez téléchargés (ou que vous avez sur CD-ROM), quel que soit le type d'installation que vous utiliserez ensuite.

Sur un système MS-DOS, vous devez décompacter les fichiers images en utilisant `GZIP.EXE`. Par exemple, si vous utilisez le fichier `bare.gz` , tapez la commande MS-DOS :

```
C:\>GZIP -D BARE.GZ
```

qui décompactera `bare.gz` en vous donnant le fichier `bare`. Vous devez décompacter l'image rootdisk de la même manière. Par exemple, si vous utiliser le fichier `color144.gz`, tapez la commande:

```
C:\>GZIP -D COLOR144.GZ
```

qui décompactera le fichier en vous donnant `color144`.

Ensuite, vous devez avoir deux disquettes *haute-densité* formatées sous MS-DOS . (Elles doivent être du même type, ne mélangez pas 1.44 Mo et 1.2 Mo). Vous utiliserez `RAWRITE.EXE` pour recréer ces disquettes à partir des fichiers image.

Tapez la commande :

```
C:\>RAWRITE
```

Répondez aux questions que le programme vous posera concernant le fichier à écrire, le lecteur de disquettes à utiliser (par exemple `A:`). `RAWRITE` copiera le fichier, bloc par bloc, directement sur la disquette. Utilisez également `RAWRITE` pour l'image de "root" (par exemple `COLOR144`). Lorsque vous aurez fini, vous aurez deux disquettes: l'une étant la disquette d'amorçage, l'autre contenant le système de fichiers racine, d'installation. Notez que ces deux disquettes finales ne sont pas lisibles par MS-DOS (elles sont au « format Linux », en quelque sorte).

Assurez-vous d'utiliser des disquettes neuves, sans aucune erreur. Les disquettes ne doivent contenir aucun mauvais secteur, elles doivent être parfaites.

Notez bien qu'il n'est pas nécessaire de posséder MS-DOS pour installer la distribution Slackware. Toutefois, utiliser MS-DOS est plus simple pour les débutants, lors de la réalisation des disquettes; et si vous possédez une partition MS/DOS sur votre disque dur il sera plus rapide d'effectuer l'installation à partir de celle-ci.

Il n'est pas non plus indispensable d'utiliser `GZIP.EXE` et `RAWRITE.EXE` sous MS-DOS pour créer les deux disquettes de départ, non plus. Vous pouvez utiliser les commandes `gzip` et `dd` sur un système Unix pour réaliser la même chose. (Pour cela, il vous faudra une station de travail UNIX possédant un lecteur de disquettes, bien sûr). Par exemple, sur une station Sun dont le lecteur de disquettes correspond au device `/dev/rfd0`, vous pouvez utiliser les commandes:

```
$ gunzip bare.gz
$ dd if=bare of=/dev/rfd0 obs=18k
```

Vous devez sur certaines stations (Sun par exemple) préciser la taille de bloc correcte (l'argument `obs`) sinon l'opération échouera. En cas de problèmes, consultez la page de manuel de `dd`, qui est très instructive.

2.1.4.4.2 Préparation du disque dur pour l'installation Si vous avez l'intention d'installer la distribution Slackware directement depuis le disque dur (ce qui est beaucoup plus rapide et plus fiable qu'à partir des disquettes), vous aurez besoin d'une partition MS-DOS sur le système où vous comptez installer Linux. (Il y a fort à parier que MS-DOS se trouve déjà sur cet ordinateur.)

Note : Si vous comptez installer depuis une partition MS-DOS, cette partition ne doit PAS être compressée par DoubleSpace, Stacker, ou tout autre utilitaire DOS de ce type. Linux est incapable de lire de telles partitions MS-DOS. (Vous pourrez y accéder par le biais de l'émulateur MS/DOS plus tard, mais c'est impossible lors de l'installation du système).

Pour préparer votre installation à partir du disque dur, créez simplement un répertoire pour y stocker les archives de la distribution, par exemple,

```
C:\>MKDIR SLACKWAR
```

créera le répertoire `C:\SLACKWAR` pour contenir les fichiers Slackware. Dans ce répertoire, vous devrez créer les sous-répertoires `A1`, `A2`, etc..., pour chaque disquette des paquetages que vous avez téléchargés. Tous les fichiers de la disquette `A1` doivent aller dans le répertoire `SLACKWAR\A1`, et ainsi de suite.

2.1.4.4.3 Préparation pour l'installation à partir de disquettes Si vous désirez installer la distribution Slackware à partir de disquettes, vous aurez besoin d'une disquette vierge, formatée MS-DOS, pour chaque disquette de la distribution téléchargée. Ces disquettes doivent être bien sûr haute-densité.

Les disquettes `A` (disquettes `A1` jusqu'à `A4`) peuvent être indifféremment au format 3.5" ou 5.25". En revanche, le reste des disquettes doit être absolument en 3.5". Donc, si vous ne disposez que d'un lecteur de disquettes de 5.25", vous devrez emprunter un lecteur de 3.5" afin d'installer les paquetages autres que `A`. (Ou bien, vous pouvez installer depuis le disque dur, comme expliqué dans la section précédente.)

Pour réaliser les disquettes, copiez tout simplement les fichiers de chaque sous-répertoire Slackware sur une disquette formatée MS-DOS, à l'aide de la commande MS-DOS `COPY`. Par exemple:

```
C:\>COPY A1\*.* A:
```

Copiera le contenu du disque `A1` sur la disquette contenue dans le lecteur `A:`. Vous devrez répéter cette opération pour chaque disquette téléchargée.

Vous ne *devez pas* modifier ou décompacter les fichiers des disquettes, vous devez juste les copier tels quels. La procédure d'installation de Slackware s'occupera de cette opération elle-même le moment venu.

2.1.4.4.4 Préparation pour l'installation depuis un CD-ROM Si votre distribution se trouve sur CD-ROM, une fois que les deux disquettes de départ sont réalisées, tout est prêt ; Linux pourra être installé directement depuis les fichiers se trouvant sur ce disque.

2.2 Préparatifs avant installation

Après vous être procuré une distribution binaire de Linux, vous devez vous préparer à l'installation du système. Le plus important dans cette étape est le repartitionnement de votre disque dur pour allouer de l'espace pour Linux. Ceci demande un peu de réflexion, particulièrement si vous utilisez déjà votre ordinateur sous un autre système d'exploitation que vous désirez conserver parallèlement. Nous allons voir dans les sections suivantes comment préparer proprement le terrain.

2.2.1 Aperçu de la procédure d'Installation

Bien que chaque distribution de Linux soit différente, la méthode générale utilisée pour installer le système se passe comme suit :

1. **Repartitionnement de votre disque dur.** Si vous avez un autre système d'exploitation déjà installé, qui occupe toute la place disque disponible, vous allez devoir repartitionner le(s) disque(s) de manière à pouvoir allouer de l'espace pour Linux. Cette opération est décrite dans la section 2.2.4, un peu plus loin.

2. **Amorçage de l'ordinateur sur le support d'installation.** Chaque distribution de Linux possède un support d'installation, en général une « disquette d'amorçage » (disquette de boot) qui est utilisée pour démarrer l'ordinateur sous Linux et installer le système. Ce support d'installation vous offrira soit un menu d'installation qui vous guidera pas à pas, soit la possibilité d'installer tout vous même manuellement.

3. **Création des partitions Linux.** Après avoir repartitionné pour pouvoir installer Linux, vous créez les partitions désirées sur cet espace disque. Ceci se fait à l'aide de la commande `fdisk` de Linux, décrit dans la section 2.3.3.

4. **Création des systèmes de fichiers et de la zone de swap.** A ce stade, vous initialiserez un ou plusieurs *systèmes de fichiers*, utilisés pour stocker vos fichiers, sur les nouvelles partitions que vous avez prévues. De plus, si vous compter utiliser une zone de swap (ce qui est très conseillé), vous utiliserez une de ces partitions pour cet usage. Ces opérations sont décrites dans les sections 2.3.4 et 2.3.5.

5. **Installation du système sur les nouveaux systèmes de fichiers.** Finalement, vous installerez les fichiers Linux. Après cela, si tout va bien vous serez tiré d'affaire. La section 2.3.6 décrit l'installation, et si tout va mal vous pourrez vous reporter à la section 2.5 qui vous permettra sûrement de rétablir la situation.

Beaucoup de distributions de Linux offrent un programme d'installation qui automatise un certain nombre de tâches. Gardez donc à l'esprit lors de la lecture de ce chapitre que certaines étapes décrites peuvent être parfois réalisées automatiquement pour vous, selon la distribution que vous utilisez.

La Slackware, décrite dans ce manuel, ne nécessitera par exemple qu'un repartionnement du disque à l'aide de `fdisk`, puis le programme `setup` vous permettra d'effectuer aisément l'installation proprement dite.

Conseil important: Lors de la préparation à l'installation de Linux, le meilleur conseil que nous puissions vous donner est de *prendre des notes* pendant toute la procédure. Notez tout ce que vous faites, tout ce que vous tapez, et tout ce qui s'affiche qui semble anormal. L'idée est simple : si quelque chose se passe mal, vous pourrez retracer chaque étape pour trouver où est le problème. Installer Linux n'est pas difficile du tout, mais il y a beaucoup de petits détails à ne pas oublier. Si vous avez bien tout noté, vous pourrez également essayer une méthode différente en cas de problème. De plus, garder une trace de votre expérience vous sera très utile si vous devez demander de l'aide à quelqu'un, en postant par exemple un message dans un forum Usenet. Et peut être qu'un jour vous serez fier de montrer ce carnet de notes à vos petits-enfants.[3] [4]

2.2.2 Partitions - concepts

En général, les disques durs sont divisés en *partitions*, chaque partition étant dédiée à un unique système d'exploitation. Vous pouvez avoir par exemple sur un disque dur, plusieurs partitions, l'une contenant MS-DOS, une autre OS/2, et une troisième Linux.

Si vous avez déjà un système sur votre machine, vous pouvez avoir besoin de modifier la taille de ces partitions de manière à libérer un peu d'espace disque pour y loger Linux. Vous pourrez alors créer une ou plusieurs partitions sur l'espace ainsi libéré pour y placer le système Linux et sa zone de swap. Nous appelons ce procédé *repartitionnement*.

Beaucoup de machines MS-DOS utilisent une seule partition prenant toute la place disponible sur le disque dur. Pour MS-DOS, cette partition est connue en tant que disque `C:`. Si vous avez plus d'une partition, MS-DOS les appellera `D:`, `E:`, et ainsi de suite. En un sens, chaque partition est vue comme un disque dur séparé.

Sur le tout premier secteur du disque dur se trouve le **secteur maître d'amorçage** (master boot record). Ce secteur d'amorçage, comme son nom l'indique, est utilisé pour démarrer le système. La table de partitions contient des informations concernant la localisation et la taille de vos partitions sur l'espace global du disque.

Il existe trois types de partitions : **primaires**, **étendue**, et **logiques**. Les plus couramment utilisées sont les partitions primaires. Mais en raison de la limitation en taille de la table de partitions, il ne peut y avoir que quatre partitions primaires sur un disque dur donné.

La méthode pour contourner cette limite de quatre partitions est d'utiliser une partition étendue. Une partition étendue ne contient pas de données réellement : elle agit comme un « emballage » des partitions logiques. Par conséquent, vous pourrez créer une grande partition étendue, utilisant tout le disque dur, et à l'intérieur créer beaucoup de petites partitions logiques. Cependant vous ne pouvez

3. L'auteur avoue honteusement qu'il a gardé des notes de toutes ses tribulations sous Linux depuis ses premières expériences avec le système. Ce carnet prend maintenant la poussière sur son étagère.

4. Ce n'est pas le cas du traducteur. Il a tout dans la tête, et beaucoup moins de poussière chez lui...

avoir qu'une seule partition étendue par disque dur.

2.2.3 Partitions nécessaires

Avant d'apprendre à repartitionner vos disques, vous devez avoir une idée de l'espace que vous devrez allouer à Linux. Nous verrons comment créer ces partitions plus tard, dans la section 2.3.3.

Sur les systèmes UNIX, les fichiers sont stockés sur un **système de fichiers**, qui est essentiellement une portion du disque dur (ou tout autre support comme un CD-ROM ou une disquette) formaté pour contenir des fichiers. Chaque système de fichiers est associé à une branche particulière de l'arborescence générale des répertoires ; par exemple il peut y avoir un système de fichiers contenant tous les fichiers du répertoire `/usr`, un autre pour `/tmp`, et ainsi de suite. Le **système de fichiers racine** (root file system) est le système de fichiers primaire, qui correspond au répertoire le plus haut, `/`.

Sous Linux, chaque système de fichiers réside sur une partition séparée du disque dur. Si vous désirez par exemple deux systèmes de fichiers séparés pour `/` et `/usr`, vous aurez besoin de deux partitions pour les contenir.

Avant d'installer Linux, vous devrez préparer des systèmes de fichiers qui recevront le système et toutes les données. Vous devez avoir au minimum un système de fichiers (la racine), et par conséquent une partition, dédiée à Linux. Beaucoup d'utilisateurs choisissent lors de leur apprentissage d'UNIX de stocker l'ensemble de leurs fichiers sur le système de fichiers principal, ce qui est beaucoup plus facile à gérer que de posséder beaucoup de partitions différentes pour un débutant.

Mais vous pouvez bien sûr créer de multiples systèmes de fichiers pour Linux si vous le désirez ; par exemple vous pourriez vouloir utiliser des partitions différentes pour `/usr` et `/home`. Les lecteurs ayant une certaine expérience de l'administration UNIX sauront exploiter cette possibilité. Nous discuterons de cette technique dans le chapitre 4.

Pourquoi utiliser plusieurs systèmes de fichiers ? La raison la plus courament invoquée est la sécurité. Si pour une raison quelconque, l'un de vos systèmes de fichiers est endommagé, les autres seront généralement épargnés puisque situés sur des partitions différentes. Si vous mettez tous vos fichiers sur un seul système de fichiers racine, en cas de corruption vous pourriez perdre jusqu'à la totalité du système d'exploitation. Ce cas est toutefois assez rare ; et si vous faites des sauvegardes régulières vous devriez être à l'abri de ce genre d'ennui [5] [6].

Une autre raison d'utiliser de multiples systèmes de fichiers est de diviser l'espace de stockage entre plusieurs disques durs. Si vous avez, disons, 40 Mégaoctets libres

5. L'auteur utilise une unique partition de 200 mégaoctets pour tous ses fichiers Linux et n'a jamais eu d'accident pour l'instant.

6. Le traducteur par contre possède 900 mégaoctets de fichiers Linux répartis sur quatre disques durs scindés en sept partitions. Cela lui a sauvé la mise plusieurs fois.

sur un disque dur, et 50 Mo sur un autre, vous pouvez par exemple créer la racine sur le premier disque et placer `/usr` sur la partition de 50 Mo située sur l'autre disque. Il n'est pas pour l'instant possible à un système de fichiers de s'étaler sur plusieurs disques ; si votre espace disque dur est fragmenté en plusieurs disques durs ou partitions, vous devrez créer autant de systèmes de fichiers pour utiliser cet espace.

En résumé, Linux nécessite au moins une partition, pour la racine. Si vous désirez créer plusieurs systèmes de fichiers, vous aurez besoin de partitions séparées pour chacun d'eux. Quelques distributions de Linux créent automatiquement des partitions et des systèmes de fichiers pour vous, vous n'aurez donc peut être pas à vous soucier de ces problèmes.

Une autre chose importante à considérer lorsque vous planifiez le partitionnement de vos disques, est la zone de swap. Si vous voulez utiliser un espace de swap sous Linux (c'est conseillé), vous avez deux options. La première est d'utiliser un *fichier de swap* qui se trouvera tout simplement sur l'un de vos systèmes de fichiers. Vous créerez ce fichier spécial, utilisé comme mémoire RAM virtuelle, après avoir installé le système. La seconde option est de créer une partition totalement dédiée à cette zone de swap. La plupart des gens utilisent cette dernière solution, pour des raisons de sécurité et de performances.

Une zone de swap (fichier ou partition) peut faire jusqu'à 16 Mégaoctets. Si vous avez besoin de plus de 16 Mo de swap, vous pouvez créer jusqu'à huit zones de swap différentes. Si par exemple vous désirez pouvoir disposer de 32 Mo de swap, vous pouvez créer deux partitions de swap d'une taille de 16 Mo chacune.

La création d'une partition de swap est décrite dans la section 2.3.4, et la réalisation d'un fichier de swap dans le chapitre 4.

Par conséquent, vous créerez en général au minimum deux partitions pour Linux ; une pour le système de fichiers racine et l'autre utilisée pour la zone de swap. Il y a bien entendu, beaucoup de variations possibles sur ce thème, mais c'est la configuration minimale. Vous n'êtes pas obligé d'utiliser une zone de swap sous Linux, mais si vous possédez moins de 16 Mégaoctets de mémoire RAM nous vous conseillons de le faire absolument.

Bien sûr, vous devez avoir une idée de *la quantité d'espace* demandée par ces partitions. La taille de vos systèmes de fichiers Linux (contenant le système par lui-même) dépend grandement des paquetages que vous comptez installer et de la distribution que vous utilisez. La documentation fournie avec votre distribution vous donnera une approximation de l'espace disque nécessaire. Un petit système Linux peut se contenter de 20 mégaoctets, voire moins ; un système moyen tiendra dans 80 ou 100 Mo, et il n'y a pas de limite supérieure. N'oubliez pas qu'en plus de l'espace nécessaire pour le système lui-même, vous aurez besoin de place supplémentaire pour les répertoires des utilisateurs, et qu'il faut penser à l'avenir...

La taille de votre partition de swap dépend de la quantité de mémoire virtuelle dont vous avez besoin. Une règle grossière et arbitraire est de prendre le double de la quantité de mémoire physique RAM qui équipe la machine ; par exemple si vous n'avez que 4 Mo de RAM, une zone de swap de 8 Mo devrait faire l'affaire. Bien sûr c'est très largement approximatif, en réalité la quantité de swap nécessaire dépend des programmes que vous utiliserez. Si vous disposez de beaucoup de mémoire RAM (disons plus de 16 Mo) vous pourrez sans doute vous passer de swap, ou ne prévoir qu'une petite zone par précaution.

Note importante : Beaucoup de cartes contrôleurs SCSI sont incapables d'amorcer un système d'exploitation depuis des partitions utilisant des cylindres supérieurs au numéro 1 023. Par conséquent, lorsque vous prévoyez vos partitions Linux, pensez éventuellement à ne pas créer de partition dans la gamme de cylindres >1 024 pour le système de fichiers racine. (Celui qui contiendra le noyau du système). Cet avertissement peut sembler prématuré, mais il vaut mieux prendre le maximum de précautions.

Pour tous les autres systèmes de fichiers, une partition de plus de 1 024 cylindres ne pose aucun problème à Linux bien entendu.

Si vous devez absolument utiliser une partition de plus de 1 024 cylindres pour la racine, vous pourrez malgré tout toujours lancer Linux depuis une disquette d'amorçage. Ce n'est pas si catastrophique, après tout cela ne prend que quelques secondes de plus, c'est simplement beaucoup moins élégant.

2.2.4 Repartitionner vos disques durs

Nous verrons dans cette section comment changer la taille de vos partitions actuelles (si vous en avez déjà) afin de faire de la place pour Linux. Si vous installez le système sur un disque « vierge » , vous pouvez sauter cette section pour aller directement section 2.3, un peu plus loin.

La méthode usuelle pour modifier la taille d'une partition existante est de la supprimer (donc de détruire toutes les données qui s'y trouvent) et de la recréer avec une taille différente. Avant de repartitionner vos disques durs, *faites une sauvegarde complète.* Une fois les partitions recréées, vous pourrez réinstaller vos fichiers depuis cette sauvegarde. Toutefois, il existe plusieurs programmes disponibles pour MS-DOS qui sont capables de modifier la taille des partitions de manière non destructive. L'un de ces programmes est connu sous le nom de « `FIPS` » et peut être trouvé sur de nombreux sites FTP diffusant Linux.

Notez bien également qu'en réduisant la taille de vos partitions, vous pourriez ne plus avoir suffisament de place pour restaurer tous vos fichiers. Organisez-vous, donc, et faites le ménage.

Le programme utilisé pour partitionner les disques durs s'appelle **`fdisk`**. Chaque système d'exploitation possède une commande analogue, par exemple sous MS-DOS, la commande est invoquée en tapant `FDISK`. Consultez la documentation du système que vous utilisez, quel qu'il soit, à ce sujet. Nous ne parlerons ici que du cas le plus courant, c'est à dire modifier la taille d'une partition MS-DOS à l'aide de sa commande `FDISK`, mais ces informations devraient pouvoir facilement être extrapolées sous n'importe quel autre système d'exploitation.

Surtout consultez la documentation de vos systèmes d'exploitation avant de repartitionner vos disques durs. Cette section donne une vision générale de la procédure ; il y a beaucoup de subtilités que nous omettrons. Vous pouvez perdre toutes vos données et tous vos programmes si vous ne faites pas les choses correctement.

Attention : Ne modifiez pas ou ne créez pas de partitions pour d'autres systèmes d'exploitation (y compris Linux) avec la commande `FDISK` de MS-DOS. Vous devez uniquement utiliser la version de `fdisk` fournie avec chaque système d'exploitation pour travailler ses propres partitions. Certaines incompatibilités pourraient êtres fatales. Plus tard, dans la section 2.3.3, nous verrons comment créer des partitions Linux, mais pour l'instant ce qui nous intéresse est de changer la taille de celles qui sont sur votre disque dur.

Admettons que vous n'ayez qu'un seul disque dur dans votre ordinateur, actuellement entièrement dédié à MS-DOS. De plus, ce disque ne contient qu'une seule partition, couramment connue sous le nom de « `C:` » . Puisque la méthode de repartitionnement que nous allons utiliser va détruire toutes les données présentes sur le disque, vous devez préparer une disquette système MS-DOS qui contient tout ce qui est nécessaire pour amorcer et faire tourner la machine sous MS-DOS, lancer la commande `FDISK`, et restaurer vos fichiers depuis votre sauvegarde par la suite.

La plupart du temps les disquettes d'installation originales de MS-DOS sont idéales pour ça. Si malgré tout vous devez réaliser votre propre disquette système, formatez une disquette vierge par la commande

```
FORMAT /s A:
```

Copiez dessus tous les utilitaires MS-DOS nécessaires (souvent tout cela se trouve dans le répertoire `\DOS` de votre disque), et n'oubliez pas les deux programmes indispensables `FORMAT.COM` et `FDISK.EXE`. Vous devriez maintenant pouvoir amorcer la machine avec cette disquette, et lancer la commande :

```
FDISK C:
```

pour appeler le programme `FDISK`.

L'utilisation de `FDISK` est très conviviale, mais consultez votre documentation MS-DOS pour les détails. Lorsque vous lancez `FDISK`, utilisez l'option du menu qui affiche l'état de la table de partition, et *notez exactement* ce qui s'affiche. Il est très important de garder une trace de votre configuration de départ au cas où vous voudriez abandonner l'installation de Linux et restaurer votre système à l'identique.

Pour supprimer une partition existante, utilisez l'option « `Delete an MS-DOS Partition or Logical DOS Drive` » du menu de `FDISK`.[7] Précisez le type de partition que vous voulez supprimer (primaire, étendue, ou logique) ainsi que son numéro. Vérifiez bien tous les avertissements...et pouf !

Pour créér une nouvelle (et plus petite) partition MS-DOS, choisissez simplement le menu « `Create an MS-DOS Partition or Logical DOS Drive` » de la commande `FDISK`. Précisez le type (primaire, étendue, logique), et la taille désirée. `FDISK` devrait créer la partition et vous êtes alors paré.

Après avoir utilisé `FDISK` ainsi, vous devez sortir du programme et reformater toute nouvelle partition. Par exemple, si vous avez modifié la taille de la première partition DOS sur votre disque (`C:`) vous devez taper la commande :

```
FORMAT /s C:
```

Puis vous pourrez réinstaller vos programmes depuis la sauvegarde.

2.3 Installation de Linux

Après avoir fait de la place sur votre disque dur pour recevoir Linux, vous êtes prêt à installer le système. Voici un bref aperçu de la procédure :

- Amorcer l'ordinateur depuis la disquette d'installation ;
- Lancer `fdisk` sous Linux pour créer les partitions Linux ;
- Utiliser `mke2fs` et `mkswap` pour créer les systèmes de fichiers Linux et la zone de swap ;
- Installer le système ;
- Enfin, soit installer le programme `LILO` sur votre disque dur, soit créer une disquette d'amorçage pour pouvoir lancer votre nouveau système Linux.

Comme nous l'avons déjà dit, l'une (ou plusieurs) de ces étapes peut être automatisée dans la procédure d'installation, selon la distribution que vous utilisez. Consultez la documentation fournie avec pour obtenir les instructions qui lui sont spécifiques.

7. Il est possible que vous disposiez d'une version française de MS-DOS auquel cas ce message sera fort différent. Le traducteur avoue n'avoir jamais possédé une telle version de MS-DOS...

2.3.1 Démarrer Linux

La première étape consiste à amorcer l'ordinateur depuis le support d'installation. Dans la plupart des cas il s'agit d'une « disquette de boot » qui contient un système Linux très réduit. Lors du démarrage, un menu d'installation ou quelque chose d'équivalent vous sera présenté, qui vous mènera étape par étape à un système Linux installé, en état de fonctionner. Sur d'autres distributions, vous n'aurez juste qu'une invite de login lors du lancement de cette disquette. Dans ce cas, vous devez généralement entrer sur le compte `root` ou `install` et la procédure d'installation commencera.

La documentation fournie avec votre distribution vous expliquera exactement quelles sont les manipulations nécessaires au démarrage de Linux depuis le support d'installation fourni.

Si vous installez la distribution Slackware de Linux il vous suffira d'amorcer la machine sur la disquette réalisée dans la section précédente.

Beaucoup fournissent une disquette d'amorçage qui permet de donner des paramètres relatifs au matériel pour forcer la détection de certains des périphériques. Par exemple, si votre contrôleur SCSI n'est pas détecté, vous devrez relancer la machine et spécifier ses paramètres (comme l'adresse d'entrée/sortie et la ligne d'IRQ) à l'invite d'amorçage qui vous sera proposée.

De même, les ordinateurs IBM PS/1, ThinkPad et ValuePoint ne stockent pas la géométrie du disque en mémoire CMOS, vous devrez donc l'indiquer au démarrage.

Cette invite d'amorçage est souvent affichée automatiquement, c'est le cas dans la distribution Slackware par exemple. D'autres fois, vous devrez la demander explicitement en appuyant sur les touches Shift ou Control pendant l'amorçage. Vous devriez voir alors :

```
boot:
```

Ainsi que d'autres messages éventuels.

Essayez alors de ne pas préciser de paramètres, tapez la touche Entrée. Regardez les messages que le système affiche à l'initialisation. Si vous avez un contrôleur SCSI, vous devriez voir s'afficher la liste des périphériques connectés. Si au contraire vous voyez :

```
SCSI: 0 hosts
```

C'est que votre contrôleur n'a pas été détecté, et vous devrez employer la procédure que nous allons décrire bientôt.

Le système détaille également les partitions trouvées. Si rien ne s'affiche ou si les informations sont incorrectes, vous devrez également forcer la détection.

D'un autre côté, si tout se passe bien et que vos périphériques semblent parfaitement reconnus, vous pouvez aller directement à la section suivante, « 2.3.2 ».

Pour forcer la détection du matériel, vous devez entrer les paramètres appropriés à l'invite d'amorçage, selon la syntaxe suivante :

`ramdisk` ⟨*paramètres...*⟩

Il y a un certain nombre de paramètres possibles ; voici les plus courants

`hd=`⟨*cylindres*⟩`,`⟨*têtes*⟩`,`⟨*secteurs*⟩
: Spécifient la géométrie du disque dur. Nécessaire pour les machines comme IBM PS/1, ValuePoint et ThinkPad. Par exemple, si votre disque a 683 cylindres, 16 têtes et 32 secteurs par piste, mettez :

```
ramdisk hd=683,16,32
```

`tmc8xx=`⟨*adresse-mémoire*⟩`,`⟨*irq*⟩
: Spécifie l'adresse te l'IRQ pour une carte SCSI sans bios Future Domain TMC-8xx. Par exemple :

```
ramdisk tmc8xx=0xca000,5
```

Le préfixe `0x` doit être utilisé pour toutes les valeurs données en hexadécimal. Ceci est vrai pour toutes les options suivantes.

`st0x=`⟨*adresse-mémoire*⟩`,`⟨*irq*⟩
: Spécifie l'adresse et l'IRQ pour un contrôleur sans bios Seagate ST02.

`t128=`⟨*adresse-mémoire*⟩`,`⟨*irq*⟩
: Spécifie l'adresse et l'IRQ pour un contrôleur sans bios Trantor T128B.

`ncr5380=`⟨*port*⟩`,`⟨*irq*⟩`,`⟨*dma*⟩
: Spécifie le port, l'IRQ et le canal DMA pour un contrôleur NCR5380.

`aha152x=`⟨*port*⟩`,`⟨*irq*⟩`,`⟨*scsi_id*⟩`,1`
: Spécifie le port, l'IRQ et l'id SCSI pour les un contrôleurs sans bios AIC-6260. Ils comprennent les Adaptec 1510, 152x, et les contrôleurs SCSI Soundblaster.

À chaque fois, vous devez préciser **`ramdisk`** suivi des paramètres dont vous avez besoin (Ce « ramdisk » est indispensable pendant l'installation).

Si vous avez des questions à propos de ces options d'amorçage, lisez le document linux « SCSI HOWTO », qui devrait être disponible sur n'importe quel site diffusant Linux, ainsi que le « CD-ROM HOWTO ». Ces documents sont toujours à jour et comprennent de nombreux détails.

2.3.2 Disques et partitions sous Linux

La plupart des distributions vous demandent de créer les partitions Linux manuellement à l'aide du programme `fdisk`. Certaines peuvent éventuellement réaliser l'opération automatiquement pour vous; quoi qu'il en soit vous devez posséder quelques notions de base sur le partitionnement et le nom des périphériques mis en jeu.

Les disques durs et leurs partitions sous Linux (et sous UNIX en général) portent des noms très différents de leurs équivalents sous d'autres systèmes d'exploitation. Sous MS-DOS, les lecteurs de disquette s'appellent `A:` et `B:`, et les partitions du ou des disques durs se nomment `C:`, `D:`, et ainsi de suite. Sous Linux ces conventions diffèrent totalement.

Les **pilotes de périphériques**, situés dans le répertoire `/dev`, servent à communiquer avec les périphériques présents dans votre machine (comme les disques durs, souris, etc...). Par exemple si vous disposez d'une souris, vous y accédez par l'intermédiaire du pilote `/dev/mouse`. Les lecteurs de disquettes, disques durs et partitions individuelles se voient attribuer chacun leur propre pilote. Pour l'instant, la seule chose importante pour vous est de bien comprendre comment les différents pilotes s'appellent pour être capable de les utiliser correctement.

Le tableau 2.1 récapitule les différents noms des divers pilotes de périphériques.

Quelques remarques à propos de ce tableau. Notez bien que `/dev/fd0` correspond au premier lecteur de disquettes (`A:` sous MS-DOS) et que `/dev/fd1` représente le second lecteur (`B:`).

Faites également attention aux disques SCSI, ils ont un nom différent des disques ordinaires. Les disques IDE, MFM, ou RLL sont accessibles par les pilotes `/dev/hda`, `/dev/hdb`, et ainsi de suite, et les partitions de ces disques sont gérées par `/dev/hda1`, `/dev/hda2`, etc..., alors que les disques SCSI se nomment `/dev/sda`, `/dev/sdb`, ..., avec des partitions accessibles par exemple par `/dev/sda1` et `/dev/sda2`. Vous pouvez mémoriser cela en vous disant que la lettre 's' signifie SCSI.

Voici un exemple. Disons que vous possédez un unique disque dur IDE, comportant 3 partitions primaires. Les deux premières sont destinées à MS-DOS, et la troisième est une partition étendue contenant deux partitions logiques, toutes deux dédiées à Linux. Les pilotes se référant à ces partitions seraient :

Première partition MS-DOS (`C:`)	`/dev/hda1`
Seconde partition MS-DOS (`D:`)	`/dev/hda2`
Partition étendue	`/dev/hda3`
Premiere partition logique Linux	`/dev/hda5`
Seconde partition logique Linux	`/dev/hda6`

Pilote	Nom
Premier lecteur de disquettes (`A:`)	`/dev/fd0`
Second lecteur de disquettes (`B:`)	`/dev/fd1`
Premier disque dur (totalité du disque)	`/dev/hda`
Premier disque dur, partition primaire 1	`/dev/hda1`
Premier disque dur, partition primaire 2	`/dev/hda2`
Premier disque dur, partition primaire 3	`/dev/hda3`
Premier disque dur, partition primaire 4	`/dev/hda4`
Premier disque dur, partition logique 1	`/dev/hda5`
Premier disque dur, partition logique 2	`/dev/hda6`
⋮	
Second disque dur (totalité du disque)	`/dev/hdb`
Second disque dur, partition primaire 1	`/dev/hdb1`
⋮	
Premier disque dur SCSI (totalité du disque)	`/dev/sda`
Premier disque dur SCSI, partition primaire 1	`/dev/sda1`
⋮	
Second disque dur SCSI (totalité du disque)	`/dev/sdb`
Second disque dur SCSI, partition primaire 1	`/dev/sdb1`
⋮	

TAB. 2.1 - *Nom des partitions sous Linux*

Notez que `/dev/hda4` n'est pas cité, il correspondrait à la quatrième partition primaire, que nous n'utilisons pas dans cet exemple. Les partitions logiques sont nommées incrémentalement à partir de `/dev/hda5`.

2.3.3 Création des partitions Linux

Maintenant, vous allez créer vos partitions Linux à l'aide de la commande **`fdisk`**. Comme nous l'avons vu dans la section 2.2.3, vous aurez besoin d'au moins une partition pour installer le système, et d'une autre pour la zone de swap.

Après avoir amorcé la machine sur le support d'installation, lancez **`fdisk`** en tapant :

`fdisk` ⟨*disque*⟩

où ⟨*disque*⟩ est le nom du pilote du périphérique sur lequel vous comptez ajouter vos partitions (voir le tableau 2.1). Par exemple, si vous voulez utiliser **`fdisk`** sur le premier disque dur SCSI de votre machine, utilisez la commande **`fdisk`** `/dev/sda`. Si vous ne précisez aucun argument, le défaut est de prendre `/dev/hda` (le premier disque IDE).

Si vous comptez créer des partitions Linux sur plusieurs disques durs, lancez **fdisk** successivement sur chacun d'eux.

```
# fdisk /dev/hda

Command (m for help):
```

Ici, **fdisk** attend une commande de votre part. Vous pouvez taper **m** pour obtenir une liste des options possibles.

```
Command (m for help): m
Command action
a toggle a bootable flag
d delete a partition
l list known partition types
m print this menu
n add a new partition
p print the partition table
q quit without saving changes
t change a partition's system id
u change display/entry units
v verify the partition table
w write table to disk and exit
x extra functionality (experts only)

Command (m for help):
```

La commande **n** sert à créer une nouvelle partition. Vous n'avez pas besoin de vous occuper de la plupart des autres commandes. Pour quitter **fdisk** sans enregistrer les modifications, utilisez la commande **q**. Pour quitter **fdisk** et enregistrer la nouvelle table de partitions sur le disque, utilisez la commande **w**.

La première chose que vous devez faire est de demander l'affichage de votre table de partitions courante, et noter précieusement ces informations pour plus tard, par sécurité. Utilisez la commande **p** pour cela.

```
Command (m for help): p
Disk /dev/hda: 16 heads, 38 sectors, 683 cylinders
Units = cylinders of 608 * 512 bytes

   Device Boot  Begin   Start     End  Blocks   Id  System
/dev/hda1   *       1       1     203   61693    6  DOS 16-bit >=32M


Command (m for help):
```

Dans cet exemple, nous avons une unique partition MS-DOS sur **/dev/hda1**, qui comporte 61 693 blocs (environ 60 Mégaoctets)[8]. Cette partition commence au cylindre numéro 1, et se termine au cylindre 203. Nous avons un total de 683

8. Sous Linux, un bloc fait 1 024 octets

cylindres sur ce disque, il reste donc 480 cylindres disponibles pour créer des partitions Linux.

Utilisez la commande `n` pour créer une nouvelle partition. Dans cet exemple, nous allons créer deux partitions primaires (`/dev/hda2` et `/dev/hda3`) pour Linux.

```
Command (m for help): n
Command action
e extended
p primary partition (1-4)
p
```

Ici, `fdisk` demande le type de partition à créer : primaire ou étendue. Dans notre exemple, nous ne créons que des partitions primaires, nous choisissons donc `p`.

```
Partition number (1-4):
```

`fdisk` va alors nous demander le numéro de la partition à créer ; comme la partition 1 est déjà utilisée, notre première partition Linux portera le numéro 2.

```
Partition number (1-4): 2
First cylinder (204-683):
```

Nous entrons maintenant le cylindre de départ de la partition voulue. Puisque les cylindres 204 à 683 sont inutilisés, nous prendrons le premier disponible (numéro 204). Il est inutile de gâcher de la place entre les partitions.

```
First cylinder (204-683): 204
Last cylinder or +size or +sizeM or +sizeK (204-683):
```

`fdisk` demande maintenant la taille désirée pour cette partition. Nous pouvons lui indiquer soit par le numéro du cylindre de fin, ou par une taille en octets, Kilooctets ou Mégaoctets. Puisque nous désirons une partition de 80 Mo, nous allons donc lui indiquer le plus simplement du monde `+80M`. Lorsque l'on indique la taille de cette façon, `fdisk` arrondit toujours au nombre de cylindres le plus proche.

```
Last cylinder or +size or +sizeM or +sizeK (204-683): +80M
Warning: Linux cannot currently use 33090 sectors of this partition
```

Si vous voyez un avertissement de ce genre, n'en tenez aucun compte. Le programme `fdisk` affiche ce message car il est assez ancien, et au tout début de Linux, les partitions de plus de 64 Mégaoctets n'étaient pas supportées. Ce n'est bien sûr plus le cas depuis longtemps.

Nous sommes maintenant prêts à créer notre seconde partition Linux. Pour les besoins de la démonstration, nous lui donnerons une taille de 10 Mégaoctets.

```
Command (m for help): n
Command action
e extended
p primary partition (1-4)
p
Partition number (1-4): 3
First cylinder (474-683): 474
Last cylinder or +size or +sizeM or +sizeK (474-683): +10M
```

Enfin, nous affichons la nouvelle table de partitions pour vérifier. Là encore, notez bien toutes ces informations quelque part, en particulier la taille en blocs de vos nouvelles partitions. Vous aurez besoin de vous en souvenir plus tard, lors de la création des systèmes de fichiers. Vérifiez également qu'aucune partition n'empiète sur une autre par erreur.

```
Command (m for help): p
Disk /dev/hda: 16 heads, 38 sectors, 683 cylinders
Units = cylinders of 608 * 512 bytes

   Device Boot  Begin   Start    End  Blocks   Id  System
/dev/hda1   *       1       1    203   61693    6  DOS 16-bit >=32M
/dev/hda2         204     204    473   82080   81  Linux/MINIX
/dev/hda3         474     474    507   10336   81  Linux/MINIX
```

Comme vous le voyez, `/dev/hda2` est maintenant une partition de 82 080 blocs (ce qui correspond environ à 80 Mo), et `/dev/hda3` fait 10 336 blocs (à peu près 10 Mo).

Notez que beaucoup de distributions (comme Slackware) nécessitent que vous utilisiez la commande **t** de **fdisk** pour changer le type de la partition de swap, généralement il s'agit de la valeur 82. La commande **L** vous affichera une liste des types de partitions connus, choisissez la valeur prévue pour le type « Linux swap » . De cette façon, la procédure d'installation saura détecter automatiquement vos partitions de swap en lisant cette valeur.

Dans l'exemple ci-dessus, les cylindres restants sur le disque (de 508 à 683) restent inutilisés. Vous pouvez avoir besoin de laisser de la place libre sur le disque, pour créer des partitions supplémentaires un autre jour, que ce soit pour Linux ou tout autre système d'exploitation.

Enfin, nous utilisons la commande **w** pour écrire physiquement les modifications sur le disque et sortir de **fdisk**.

```
Command (m for help): w
#
```

Rappelez-vous bien qu'aucune des modifications que vous faites lors de l'utilisation de `fdisk` n'a d'effet tant que vous n'avez pas utilisé la commande `w` pour l'écrire réellement. Vous pouvez donc expérimenter autant qu'il vous plaira différentes partitions, et quitter le programme à tout moment en toute sécurité par la commande `q`. Souvenez-vous également que vous ne devez pas toucher aux partitions appartenant à d'autres systèmes que Linux avec le programme `fdisk` de Linux.

N'oubliez pas qu'il est possible que vous ne puissiez pas lancer le système depuis une partition comportant plus de 1024 cylindres. Mais que dans ce cas il sera toujours possible d'utiliser une disquette pour le démarrage.

Quelques distributions vous demandent de réinitialiser le système après avoir utilisé `fdisk`. Les versions récentes de `fdisk` mettent automatiquement à jour les informations à la fois sur le disque et dans le noyau, et il est donc inutile de relancer Linux. Par sécurité, vous pouvez le faire quand même, mais à l'heure où vous lisez ces lignes vous ne devriez pas être en possession de versions aussi dépassées.

2.3.4 Création de la partition de swap

Si vous comptez utiliser une partition de swap pour disposer de mémoire virtuelle, il est maintenant temps de le faire.[9] Nous traiterons le cas des fichiers de swap dans le chapitre 4 au cas où vous ne voudriez pas utiliser une partition individuelle pour cet usage.

Un certain nombre de distributions nécessitent que vous prépariez et activiez une partition de swap avant d'installer le système, car si vous avez trop peu de mémoire RAM, la procédure d'installation pourrait manquer de mémoire et échouer.

La distribution Slackware est de celles-ci, si vous ne possédez que 4 Mo de mémoire, ou moins. Sinon, sa procédure d'installation vous autorisera à préparer la zone de swap automatiquement. En cas de doute, faites ce que nous allons décrire, ça ne peut pas faire de mal.

La commande utilisée pour préparer une partition de swap se nomme **mkswap**, et elle s'utilise ainsi :

`mkswap -c` ⟨*partition*⟩ ⟨*taille*⟩

où ⟨*partition*⟩ est le nom de votre partition de swap, et ⟨*taille*⟩ correspond au nombre de blocs que contient cette partition.[10] Par exemple, si votre partition de swap est `/dev/hda3` et comporte 10 336 blocs, utilisez la commande :

```
# mkswap -c /dev/hda3 10336
```

9. Encore une fois, quelques distributions de Linux s'occuperont automatiquement de cela pour vous, ou vous proposeront un menu pour cela.

10. Il s'agit de la taille en blocs telle que l'affiche `fdisk` en utilisant la commande `p` du menu. Sous Linux, un bloc fait 1 024 octets.

L'option `-c` demande à `mkswap` de tester les mauvais secteurs présents sur la partition lors de la création de la zone de swap.

Si vous utilisez plusieurs partitions de swap, vous devrez exécuter la commande `mkswap` appropriée sur chacune d'elles.

Après avoir ainsi formaté la zone de swap, il vous faudra la mettre en service pour que le système l'utilise. En général, le système le fait automatiquement lors de son démarrage. Mais comme vous n'avez pas encore installé tout le système, vous devez pour l'instant le faire manuellement.

La commande pour mettre la zone de swap en service est `swapon`, et s'utilise ainsi :

```
swapon ⟨partition⟩
```

Dans notre exemple, pour mettre en service la partition de swap `/dev/hda3`, nous utiliserons la commande

```
# swapon /dev/hda3
```

2.3.5 Création des systèmes de fichiers

Avant de pouvoir utiliser vos partitions Linux pour y stocker des fichiers, vous devez y créer des **systèmes de fichiers**. Initialiser un système de fichiers est analogue au formatage d'une partition sous MS-DOS. Nous avons décrit brièvement les systèmes de fichiers dans la section 2.2.3.

Il existe plusieurs types de systèmes de fichiers pour Linux. Chacun possède ses propres format et caractéristiques (comme la taille du plus grand fichier, des noms de fichiers, et d'autres paramètres). Linux supporte également plusieurs systèmes de fichiers « étrangers » , comme ceux de MS-DOS par exemple, afin de faciliter l'échange de données avec d'autres systèmes d'exploitation.

Le type le plus couramment utilisé sous Linux est le **Second Extended Filesystem**, plus connu sous le nom de *ext2fs*. C'est l'un des plus souples et des plus performants; il permet des noms de fichiers jusqu'à 256 caractères et des systèmes de fichiers jusqu'à 4 téraoctets. Nous verrons dans le chapitre 4 les différents systèmes de fichiers disponibles pour Linux. Mais pour l'instant, faites donc comme tout le monde et utilisez *ext2fs*.

Si vous installez la distribution Slackware, les systèmes de fichiers sont automatiquement créés par la procédure d'installation qui sera décrite dans la section suivante. Si, toutefois, vous désirez les réaliser manuellement, conformez-vous aux instructions qui vont suivre.

Pour créer un système de fichiers *ext2fs*, utilisez la commande

```
mke2fs -c ⟨partition⟩ ⟨taille⟩
```

où ⟨*partition*⟩ est le nom de la partition, et ⟨*taille*⟩ la taille de cette partition en nombre de blocs. Par exemple pour créer un système de fichiers de 82 080 blocs sur la partition `/dev/hda2`, utilisez la commande

```
# mke2fs -c /dev/hda2 82080
```

Si vous utilisez plusieurs systèmes de fichiers pour Linux, vous devrez utiliser `mke2fs` sur chacun d'eux.

Si vous avez rencontré des problèmes lors de cette opération, reportez-vous à la section 2.5 en fin de chapitre.

2.3.6 Installation du système

Finalement, vous allez installer le système sur votre machine. Chaque distribution possède un mécanisme différent pour cette opération. Beaucoup contiennent un programme d'installation qui vous guidera étape par étape pour arriver au but. Avec d'autres, vous devrez **monter** vos systèmes de fichiers dans un certain sous-répertoire (comme `/mnt`) et copier les fichiers du système manuellement. Sur les distributions CD-ROM, il peut vous être proposé d'installer une portion du système sur votre disque dur, et d'utiliser le reste directement depuis le CD-ROM.

Certaines distributions proposent plusieurs possibilités pour installer Linux. Par exemple, vous pourrez installer depuis une partition MS-DOS de votre disque dur, pour éviter l'utilisation de disquettes; ou bien installer à travers un réseau TCP/IP via FTP ou NFS. Lisez la documentation de votre distribution pour les détails.

Par exemple, la distribution Slackware ne nécessique que la création des partitions à l'aide de `fdisk`, d'une zone de swap optionnelle avec `mkswap` et `swapon` (si vous n'avez que 4 Mo de RAM, ou moins), puis le lancement du programme `setup`. Celui-ci vous proposera des menus très simples et conviviaux vous permettant une installation aisée de tout le système. Nous allons voir cela.

2.3.6.1 La procédure `setup` de la distribution Slackware

Si vous installez la distribution Slackware, après avoir créé les partitions (et l'éventuelle zone de swap), tapez la commande:

```
# setup
```

Il vous sera alors présenté un menu très explicite qui vous permettra de procéder à l'installation en sélectionnant les différentes étapes.

La description qui va suivre correspond à ce que l'on trouve sur les disques `color144` et `colrlite`; les autres disquettes racines peuvent différer quelque peu.

Ce menu `setup` propose les options suivantes, que l'on peut sélectionner en se positionnant par les touches de déplacement puis en pressant `enter` ou `spacebar` :

`Help` Visualisation du fichier d'aide de cette procédure.

`Keymap` Cette option permet de spécifier l'assignation des touches de votre clavier, s'il n'est pas américain. Il vous sera présenté une liste dans laquelle vous choisirez celui correspondant à votre équipement.

`Quick` Vous permet de choisir entre les modes d'installation « rapide » (*quick*) et « verbeux » (*verbose*). C'est ce dernier qui est la valeur par défaut, et c'est celui que nous recommendons dans tous les cas (sauf si vous avez déjà installé la Slackware une bonne dizaine de fois, et que vous savez ce que vous faites).

`Make tags` Permet aux experts de créer leur propre installation pour y incorporer des paquetage sélectionnés ; vous n'êtes en principe pas concerné si vous lisez ces lignes.

`Addswap` C'est la première option que la plupart des utilisateurs devront sélectionner pour installer la distribution Slackware. Elle vous proposera une liste de des partitions de swap disponibles (celles dont le type positionné par `fdisk` est « Linux swap »). Après avoir choisi celles que vous désirez exploiter, il vous sera proposé d'exécuter `mkswap` pour les formater.

Si vous avez déjà fait `mkswap` et `swapon` (comme décrit dans la section 2.3.4), vous ne devrez *surtout pas* autoriser `setup` à le refaire sur les mêmes partitions.

Même si vous avez déjà exécuté `mkswap` et `swapon`, il est nécessaire d'utiliser cette option `Addswap` du menu : les partitions seront alors prises en compte lorsque l'installation du système sera complète.

◇ Attention ! La création d'une zone de swap sur une partition détruit irrémédiablement toutes les données qu'elle contenait. Vérifiez bien que vous ne vous trompez pas de partition avant qu'il ne soit trop tard !

Si vous avez sélectionné cette option du menu, celles qui vont suivre vous seront automatiquement proposées. C'est la procédure à suivre, en général.

`Target` Permet de spécifier les partitions sur lesquelles Linux doit être installé. Une liste vous sera proposée (celles dont l'identificateur a été positionné sur « Linux native » par `fdisk`, et il vous sera demandé le nom de la partition racine, comme par exemple `/dev/hda2`. Ensuite, vous devrez choisir un type de système de fichiers ; nous vous suggérons d'utiliser `ext2fs`, comme décrit dans la section 2.3.5. Cela créera un système de fichiers sur la partition indiquée ; ce qui est analogue au « formatage » sous MS-DOS.

Cette procédure se répétera pour chaque partition que vous avez décidé d'attribuer à Linux. Par exemple, si vous avez créé une partition séparée pour le répertoire `/usr` (voir section 2.2.3), vous pourrez l'indiquer, choisir le répertoire dans lequel elle sera montée, etc.

◊ Attention ! La création d'un système de fichiers sur une partition détruit irrémédiablement toutes les données qu'elle contenait. Vérifiez bien que vous ne vous trompez pas de partition avant qu'il ne soit trop tard !

Même si vous avez déjà créé vos systèmes de fichiers par `mke2fs`, vous devez utiliser cette option `Target` afin de spécifier où Linux doit être installé.

`Source` Permet de spécifier où se trouvent les fichiers de la distribution qui doivent être installés : disquette, disque dur, CD-ROM, etc.

Si vous installez depuis le disque dur, il vous sera demandé sur quelles partitions se trouvent les fichiers Slackware, ainsi que le type du système de fichiers. Par exemple, si vous avez mis votre distribution sur une partition DOS, entrez le nom de cette partition (comme par exemple `/dev/hda1`), et choisissez le type `MS-DOS FAT`. Il vous sera ensuite demandé dans quel répertoire de cette partition résident les fichiers. Par exemple s'ils sont dans le répertoire `C:\SLACK` de votre disque DOS, entrez ceci :

```
/slack
```

Notez que vous devez utiliser le caractère de séparation d'Unix (`/`) et non pas celui de DOS, malgré tout.

Si vous installez depuis un CD-ROM, il vous sera demandé le type de périphérique CD-ROM utilisé, et le répertoire dans lequel se trouvent les fichiers Slackware. Dans beaucoup de cas, ce sera `/slakware`, mais cela dépend de la version.

Si vous installez la distribution *Slackware Professional*[11], deux répertoires du CD-ROM sont utilisés : `slakware` pour le système standard, qui s'installe sur votre disque dur, et `slackpro` qui est celui contenant le système fonctionnant directement depuis le CD-ROM. Ce dernier peut faire économiser beaucoup de place sur le disque dur, mais l'accès à ce type de média étant beaucoup plus lent, certaines applications se traîneront lamentablement. Si

11. Il s'agit d'une version de la distribution Slackware vendue par *Morse Telecommunications*.

vous avez suffisament de place sur votre disque dur, nous vous conseillons d'y installer le système, de manière traditionnelle.

Lors de l'installation depuis un disque dur ou un CD-ROM, Slackware peut vous annoncer une erreur de montage (`mount` *error*). C'est en général l'indication qu'il y a un problème d'accès au périphérique concerné. Consultez la section 2.5.3 si vous êtes confronté à ce problème.

`Disk sets` Cette option vous permet de choisir les paquetages que vous désirez installer. Vous devez sélectionner au minimum la série `A`. Utilisez les touches fléchées du clavier pour circuler dans la liste; l'appui sur la barre d'espacement sélectionne ou déselectionne le paquetage. Pressez `Entrée` lorsque vous aurez sélectionné tout ce dont vous avez besoin.

Notez que la sélection d'un ensemble ne signifie pas que tous les programmes qu'il contient seront installés ; il vous sera demandé confirmation pour chaque application repérée comme « optional » (facultative) ou « recommended » (recommandée).

`Install` Après avoir saisi tous les paramètres ci-dessus, tout est prêt pour l'installation de Linux. D'abord le programme vous demandera quel type de dialogue vous désirez pendant l'installation; vous devrez utiliser le mode « normal » (sauf si vous êtes un expert et avez modifié les fichiers d'installation).

Le programme d'installation installera chaque paquetage. Pour chaque module, une boîte de dialogue s'affichera, décrivant brièvement le paquetage. Si vous ne connaissez pas bien UNIX, beaucoup de ces descriptions n'auront pas grande signification pour vous, mais peu importe. Les modules qui sont absolument nécessaires seront installés automatiquement. Pour tous ceux qui sont optionnels, le programme vous donnera le choix entre installer ou sauter les fichiers correspondants. (Si vous ne voulez pas installer un paquetage maintenant, vous pourrez toujours le faire plus tard en utilisant la commande `setup`).

Pendant l'installation, les erreurs les plus courantes que vous risquez de rencontrer son l'absence d'un fichier, ou une erreur de lecture sur une disquette. La première peut signifier que le fichier est endommagé ou incomplet, la seconde que c'est la disquette qui est en mauvais état. Toute disquette provoquant ces erreurs doit être remplacée ; consultez la section 2.5.3. savoir quoi faire.

Vous pouvez aussi rencontrer des erreurs de lecture sur un CD-ROM. Assurez-vous alors qu'il est propre, ne comporte aucune trace de doigts ou rayures, etc.

`Configure` Ce menu permet la configuration du système, une fois qu'il a été installé. C'est l'objet de la section suivante.

2.3.7 Création de la disquette d'amorce ou installation de LILO

Chaque distribution fournit un moyen de lancer votre nouveau système Linux fraîchement installé. Dans beaucoup de cas, la procédure d'installation créera une « disquette de boot » , contenant un noyau Linux configuré pour utiliser votre tout nouveau système de fichiers principal (root file system). Vous démarrerez l'ordinateur sur cette disquette afin de lancer Linux, et une fois le système lancé, vous pourrez la retirer, Linux fonctionnera depuis votre disque dur. Dans d'autres distributions, cette disquette d'amorçage est directement la disquette d'installation.

Beaucoup de distributions vous proposent d'installer **LILO** sur votre disque dur. LILO est un programme qui s'installe sur le secteur d'amorce de votre disque dur. Il est capable de lancer plusieurs systèmes d'exploitation, y compris MS-DOS et Linux, et vous permet de choisir l'un d'eux lors du démarrage de l'ordinateur.

Dans le cas de la distribution Slackware, l'option `Configure` du menu de `setup` vous permettra de créer une disquette d'amorçage, ou bien d'installer LILO. Elle vous permettra aussi de spécifier votre modem, votre souris, et la zone horaire du pays où vous habitez.

LILO a besoin d'une bonne quantité d'informations sur votre disque dur pour pouvoir fonctionner correctement. Par exemple, quelle partition contient tel système d'exploitation, comment lancer chaque système particulier, etc... Lors de l'installation de LILO, beaucoup de distributions tentent de « deviner » quels seront les bons paramètres pour votre configuration. Bien que cela se produise rarement, l'installation automatique de LILO peut échouer, et laisser le secteur d'amorce de votre disque dur dans un état indéterminé, (mais sans jamais abîmer les données présentes sur les partitions, c'est impossible). En particulier, si vous utilisez le Boot Manager OS/2, vous ne *devez pas* installer LILO selon la procédure automatique ; il y a des instructions spéciales pour ce cas particulier, que nous décrirons plus tard.

En règle générale, il vaut mieux utiliser une disquette d'amorce, jusqu'à ce que vous soyez capable d'installer LILO vous même, manuellement. Si vous avez vraiment confiance ou n'avez rien à perdre, vous pouvez toutefois laisser votre procédure d'installation installer LILO automatiquement.

Dans le chapitre 4, nous détaillerons la façon de configurer et installer LILO selon votre configuration particulière.

Si tout s'est bien passé, félicitations ! Vous venez d'installer Linux sur votre ordinateur personnel. Servez-vous un verre, vous l'avez bien mérité.

Au cas où vous ayez rencontré des problèmes, la section suivante décrira les points les plus critiques de l'installation de Linux, et comment corriger les éventuels problèmes.

2.3.8 Procédures d'installation additionnelles

Quelques distributions de Linux contiennent un certain nombre de procédures d'installation supplémentaires, vous permettant de configurer différents paquetages comme le réseau TCP/IP, X Window, ou autres. Si vous disposez de ces options de configuration lors de l'installation, vous pouvez auparavant vous informer grâce à ce guide sur la manière de configurer ces applications. Sinon, laissez ces programmes supplémentaires de côté jusqu'à ce que vous en ayez besoin et en compreniez parfaitement leur configuration.

C'est à vous de voir; si quelque chose va mal, continuez pour voir ce qui se passe. Il est probable que tout ce que vous pourriez incorrectement configurer, puisse se corriger un peu plus tard (touchez du bois.)

2.4 Après l'installation

Après l'installation du système, il ne devrait plus vous rester grand chose à faire avant de pouvoir réellement l'utiliser. Dans la plupart des cas vous devriez pouvoir relancer la machine, entrer sous le compte `root`, et commencer à explorer Linux. (Chaque distribution utilise une méthode différente pour cela, suivez bien les instructions qui vous seront données.)

À ce stade, il paraît judicieux d'expliquer comment arrêter le système proprement. Vous ne devez jamais arrêter un ordinateur tournant sous UNIX brusquement, simplement en coupant l'alimentation électrique ou en utilisant le bouton de remise à zéro. Comme la plupart des systèmes UNIX, Linux « cache » les écritures disques en mémoire. Par conséquent, si vous ne stoppez pas « proprement » le système, vous pouvez corrompre les fichiers présents sur vos disques durs, Linux n'ayant pas eu le temps d'écrire les données résidant encore en mémoire. Le résultat est imprévisible.

La méthode la plus simple pour arrêter le système est d'utiliser la commande `shutdown`. Par exemple, pour arrêter immédiatement et relancer le système juste après, utiliser (en tant que `root`) la commande suivante:

```
# shutdown -r now
```

Ceci redémarrera proprement votre système. La page de manuel de `shutdown` décrit les autres paramètres possibles.

Notez toutefois que beaucoup de distributions Linux ne fournissent pas la commande `shutdown` sur le support d'installation, et qu'elle ne sera disponible que lors du fonctionnement normal de Linux, après avoir, pour la toute première fois suivant l'installation, relancé le système autrement.

Maintenant que vous pouvez explorer et utiliser Linux, plusieurs choses quant à la configuration seront sous-entendues, vous devrez dans la mesure du possible vous y conformer. La première est de créer un compte utilisateur pour vous-même (et pour toute autre personne à qui vous autoriserez l'accès). La création de comptes utilisateurs est expliquée dans la section 4.4. Généralement, il vous suffit, en tant que `root`, d'utiliser la commande `adduser` (que l'on trouve parfois aussi sous le nom de `useradd`) qui vous permettra de créer de nouveaux comptes de manière interactive.

Si vous avez plus d'un système de fichiers pour Linux, ou si vous utilisez une partition de swap, vous pourrez avoir besoin d'éditer le fichier `/etc/fstab` afin qu'il corresponde à votre configuration et que ces systèmes de fichiers soient disponibles automatiquement après chaque démarrage du système. (Par exemple, si vous utilisez une partition différente pour `/usr`, et qu'aucun des fichiers qui devraient s'y trouver n'apparaisse, vous devrez manuellement monter cette partition. Le fichier `/etc/fstab` permet d'éviter cette situation). La section 4.8 décrit tout cela. Notez que la distribution Slackware configure automatiquement vos systèmes de fichiers et votre zone de swap lors de la procédure d'installation.

2.5 En cas de problème

Pratiquement tout le monde se retrouve confronté à un problème quelconque, si minime soit-t-il, lors de la toute première installation de Linux. La plupart du temps, l'erreur est causée par une simple confusion. Mais parfois ce peut être plus sérieux, comme une omission de la part de l'un des développeurs, ou un bogue.

Cette section décrira quelques uns des problèmes les plus fréquemment rencontrés lors de l'installation, et leurs solutions. Si votre installation semble s'être très bien passée, mais que vous avez malgré tout noté quelques avertissements ou messages d'erreurs pendant l'opération, vous en trouverez les explications dans ce qui suit.

2.5.1 Problèmes d'amorçage du support d'installation

Au moment d'amorcer l'ordinateur sur le support d'installation pour la première fois, vous pouvez rencontrer un certain nombre de problèmes. En voici une liste. Notez que ces problèmes ne sont pas relatifs au démarrage du système une fois installé, reportez-vous à la section 2.5.4 pour ce type d'ennui.

- **Erreur disquette lors de l'amorçage.**

La cause la plus courante de ce type d'erreur est une disquette d'amorçage en mauvais état. Soit le support est physiquement endommagé, auquel cas il vous faudra recréer la disquette en utilisant un support *neuf*, ou bien les données présentes sur le support sont mauvaises, et dans ce cas vous devez vérifier l'intégrité des fichiers que vous avez téléchargés, et la qualité de la copie sur le support. Dans bien des cas, recréer la disquette d'amorçage suffira à régler le problème.

Si cette disquette provient d'un achat par correspondance, contactez votre distributeur et faites-vous remplacer le produit après avoir bien vérifié que le problème venait de là. Ou bien téléchargez juste le fichier nécessaire et recréez-la vous-même, si vous êtres pressé.

– **Le système « plante » pendant ou après l'amorçage.**

Après l'amorcage du support d'installation, vous devez voir s'afficher un certain nombre de messages provenant du noyau, indiquant les périphériques détectés et comment ce noyau est configuré. Ensuite, il vous sera généralement proposé une invite de « login » , vous permettant de procéder au reste de l'installation (quelques distributions vous lancent directement dans un programme d'installation). Le système peut paraître « planter » pendant certaines de ces étapes. Soyez patient, le chargement de Linux depuis une disquette est très lent. Dans bien des cas le système n'a pas planté du tout, mais prend son temps. Vérifiez bien qu'il n'y a pas eu d'activité système ou disque pendant au moins plusieurs minutes avant de décréter que le noyau est planté.

1. Après avoir démarré depuis `LILO`, le système doit charger le noyau depuis la disquette. Cela peut prendre du temps; vous saurez que tout se passe bien tant que le témoin d'activité du lecteur est allumé.
2. Pendant l'amorçage, le système peut tenter de détecter les périphériques SCSI. Si vous n'avez aucun périphérique SCSI, Linux semblera inactif pendant une bonne vingtaine de secondes; ceci se produit généralement après que la ligne

   ```
   lp_init: lp1 exists (0), using polling driver
   ```

   soit apparue à l'écran.
3. Une fois le système démarré, le contrôle est transféré aux fichiers de démarrage qui sont situés sur la disquette. Enfin, vous verrez apparaître une invite « login » ou serez propulsé dans une procédure d'installation. Si vous voyez une invite du type

   ```
   Linux login:
   ```

   vous devez alors entrer dans le système (généralement sous `root` ou encore `install` selon la distribution). Après avoir entré ce nom d'utilisateur, le système peut encore sembler arrêté pendant 20 secondes ou plus pendant que tout se charge depuis la disquette, qui, rappelons-le, est lente. Là aussi, regardez la petite lumière sur le lecteur.

Il est toutefois possible que le système « plante » vraiment lors de l'amorçage, ce qui peut être dû à plusieurs choses. Tout d'abord, vous pouvez ne pas posséder suffisamment de RAM pour l'installation de la distribution que vous avez choisie. (Voyez un peu plus loin comment remédier à cette situation en invalidant le disque virtuel).

La cause de beaucoup de plantages système est une incompatibilité matérielle. La section 1.8 dans le premier chapitre présente un résumé du matériel supporté par Linux. Même si vos périphériques sont tous supportés de longue date par Linux, vous pouvez rencontrer des problèmes d'incompatibilité entre plusieurs d'entre eux, qui font planter le système. Voyez la section 2.5.2 un peu plus loin à ce sujet.

- **Le système indique un manque de mémoire lors des tentatives d'amorçage ou d'installation.**

  Ceci est en rapport direct avec la quantité de mémoire RAM installée dans votre machine. Sur les ordinateurs ne comportant que 4Mo de RAM ou moins, vous pouvez avec certaines distributions manquer de mémoire au moment du chargement du support d'installation, car elles peuvent utiliser un disque virtuel en mémoire (« ramdisk ») dans le but d'améliorer la vitesse du système lors de l'installation.

  La solution à ce problème est d'invalider cette option. Chaque version possède une méthode différente pour cela ; reportez-vous à la documentation de la distribution que vous utilisez.

  Il se peut que le message « out of memory » ne s'affiche pas mais que ce soit tout de même la cause du plantage.

  Souvenez-vous que Linux a besoin d'au moins 2 mégaoctets de RAM pour fonctionner ; et que quelques distributions demandent au minimum 4 Mo ou plus pour cela.

- **Le système affiche une erreur comme « `permission denied` » ou « `file not found` » lors du démarrage.**

  C'est souvent l'indication que votre support d'installation est corrompu. Nous avons déjà vu comment remédier à ce problème en début de section.

- **Le message « `VFS: Unable to mount root` » s'affiche après le chargement du noyau.**

  Ce message d'erreur signifie que le système de fichiers racine (qui se trouve sur le support d'installation lui-même) n'a pas pu être trouvé. Soit le support est en mauvais état, soit vous ne démarrez pas le système correctement.

  Par exemple, beaucoup de distributions sur CD-ROM demandent que le CD-ROM soit présent dans le lecteur au moment de l'amorçage. Vérifiez bien que le lecteur est allumé, et qu'il y a une activité. Il est aussi possible que votre lecteur de CD-ROM ne soit pas reconnu; voyez la section 2.5.2 pour plus d'information.

Si vous êtes certain que vous lancez tout correctement, alors c'est que le support est vraiment corrompu. C'est un cas très rare, et les problèmes se seraient manifestés bien avant. Vérifiez bien tout encore une fois.

2.5.2 Problèmes matériels

Les problèmes les plus courants rencontrés lors de l'installation ou de l'utilisation de Linux sont des incompatibilités matérielles. Même si tous vos périphériques sont supportés par Linux, une erreur de configuration ou un conflit entre deux cartes d'extension peut quelquefois provoquer d'étranges réactions; certains périphériques peuvent être incorrectement détectés ou la machine être incapable de démarrer.

Il est très important d'isoler ces problèmes si vous suspectez qu'ils sont la cause de vos ennuis. Les sections qui suivent décrivent quelques problèmes fréquemment rencontrés et la manière de les résoudre.

2.5.2.1 Isoler les problèmes matériels

Si vous vous trouvez devant un problème que vous pensez relatif à la configuration matérielle de votre machine, la première chose à faire est de tenter de l'isoler. Cela signifie tenter d'éliminer toutes les variables possibles, généralement en ouvrant la machine, et, en ôtant petit à petit tous les éléments susceptibles de provoquer un conflit jusqu'à la découverte de l'élément responsable.

Ce n'est pas aussi effrayant qu'il y paraît. En pratique vous enlèverez tous les éléments non indispensables au démarrage de la machine, puis les réinstallerez un par un. Déshabillez bien le système pour commencer; même des cartes d'extensions anodines comme une carte série ou un contrôleur de souris peuvent paralyser un ordinateur.

Par exemple, admettons que le système « plante » lors de la détection de la carte Ethernet. Vous pourriez penser qu'il y a un conflit ou un problème avec la carte réseau, donc. La meilleure façon de le savoir est d'ôter cette carte et de tenter de relancer le système. Si tout marche, vous savez alors que (a) la carte Ethernet n'est pas supportée par Linux (voir section 1.8 pour une liste de cartes compatibles) ou (b) il y a un conflit d'adresse ou de ligne d'interruption entre cette carte et une autre.

« Addresse ou ligne d'IRQ ? » Que signifie ce jargon obscur ? Tous les périphériques de votre machine utilisent une *IRQ*, ou *interrupt request line*, (ligne de demande d'interruption), pour indiquer au système qu'ils ont besoin qu'on leur prête attention à un moment donné. Vous pouvez imaginer une IRQ comme un cordon de sonnette que le périphérique tirerait chaque fois qu'il a quelque chose à signaler au reste de l'ordinateur. Si plus d'un périphérique tire sur la même ficelle, l'ordinateur ne saura pas déterminer lequel demande un service. Blocage assuré.

Par conséquent, assurez-vous que tous les périphériques installés dans votre machine utilisent une ligne d'IRQ unique. En général cela se règle par des cavaliers sur les cartes; lisez la documentation de chaque carte pour plus de détails. Quelques périphériques n'ont pas forcément besoin d'utiliser une IRQ, mais il est suggéré de les configurer pour qu'ils en utilisent dans la mesure du possible (les contrôleurs SCSI Seagate ST01 et ST02 en sont un excellent exemple).

Dans certains cas, le noyau qui vous est fourni sur le support d'installation est configuré pour utiliser une certaine IRQ pour certains périphériques. Par exemple sur certaines distributions, le noyau est préconfiguré pour utiliser IRQ 5 pour le contrôleur TMC-950 SCSI, le contrôleur CD-ROM Mitsumi, et le contrôleur de souris BUS. Si vous voulez utiliser deux de ces périphériques à la fois (ou plus), vous devrez d'abord installer Linux avec un seul de ces périphériques validé dans la machine, puis recompiler le noyau de manière à indiquer de nouvelles IRQ pour chacun d'eux. (Reportez vous au chapitre 4 pour la recompilation du noyau).

Une autre source de conflits matériels peut être les canaux DMA (accès direct mémoire), les adresses d'entrées/sorties, et les adresses de mémoire partagée. Tous ces termes décrivent des mécanismes par lesquels le système communique avec vos interfaces. Quelques cartes Ethernet, par exemple, utilisent à la fois mémoire partagée et IRQ pour communiquer avec l'ordinateur. Si l'une ou l'autre entre en conflit avec la configuration d'un autre périphérique, la machine peut se comporter de manière très bizarre, ou ne pas fonctionner du tout. Vous pouvez modifier ces réglages sur les cartes d'extension à l'aide de cavaliers comme indiqué dans les manuels qui vous ont été fournis.

Le tableau 2.2 recense un certain nombre d'IRQ ou de canaux DMA utilisés par différents périphériques « standards » rencontrés sur la plupart des ordinateurs PC. Pratiquement tous les systèmes possèderont l'un ou l'autre de ces périphériques, vous devez donc éviter de régler l'IRQ ou le DMA d'une autre carte d'extension sur l'une de ces valeurs.

Périphérique	Adresse E/S	IRQ	DMA
`ttyS0` (`COM1`)	3f8	4	n/a
`ttyS1` (`COM2`)	2f8	3	n/a
`ttyS2` (`COM3`)	3e8	4	n/a
`ttyS3` (`COM4`)	2e8	3	n/a
`lp0` (`LPT1`)	378 - 37f	7	n/a
`lp1` (`LPT2`)	278 - 27f	5	n/a
`fd0`, `fd1` (disquettes 1 et 2)	3f0 - 3f7	6	2
`fd2`, `fd3` (disquettes 3 et 4)	370 - 377	10	3

TAB. 2.2 - *Périphériques courants*

2.5.2.2 Problèmes de reconnaissance disque dur ou contrôleur

Lors du démarrage de Linux, vous devriez voir toute une série de messages comme :

```
Console: colour EGA+ 80x25, 8 virtual consoles
Serial driver version 4.00 with no serial options enabled
tty00 at 0x03f8 (irq = 4) is a 16450
tty03 at 0x02e8 (irq = 3) is a 16550A
lpinit: lp1 exists, using polling driver
snd2 [SoundBlaster 1.5] at 0x220 irq 7 drq 1
Calibrating delay loop.. ok - 4.34 BogoMips
Configuring Adaptec at IO:330, IRQ 11, DMA priority 5
...
```

Ici, le noyau détecte ou initialise les différents périphériques présents dans votre machine. Au bout d'un moment, vous devriez voir la ligne

```
Partition check:
```

suivie d'une liste de partitions reconnues, par exemple:

```
Partition check:
sda: sda1 sda2
sdb: sdb1 sdb2 sdb3
hda: hda1 hda2 hda3 hda4
hdb: hdb1
```

Si, pour une raison quelconque, vos disques durs ou vos partitions ne sont pas reconnus, vous n'aurez aucun moyen d'y accéder.

Ceci peut avoir plusieurs causes:

- **Disque dur ou contrôleur non supporté.** Si vous utilisez un contrôleur (IDE, SCSI ou autre) qui n'est pas supporté par Linux, le noyau ne pourra pas reconnaître vos partitions.
- **Disque ou contrôleur mal configuré.** Même si le contrôleur est supporté par Linux, il peut très bien ne pas être correctement configuré. (C'est un problème fréquent pour les contrôleurs SCSI, la plupart des IDE fonctionnent sans aucune modification).

  Reportez vous à la documentation qui vous a été fournie avec votre disque dur et/ou votre carte contrôleur pour trouver les informations nécessaires pour résoudre un problème de ce type. En particulier, certains disques durs ont besoin du déplacement d'un cavalier s'ils sont utilisés comme disque « esclave » (par exemple, le second disque dur IDE). Le test à effectuer est de lancer MS-DOS ou tout autre système d'exploitation. Si vous avez accès au disque depuis un autre système, alors ce n'est pas une histoire de configuration matérielle.

Voyez la section 2.5.2.1 un peu plus haut, pour résoudre d'éventuels conflits matériels, et la section 2.5.2.3 un peu plus loin, pour la configuration des périphériques SCSI.

- **Contrôleur correctement configuré, mais non détecté.** Quelques contrôleurs dépourvus de BIOS nécessitent que l'utilisateur spécifie certaines caractèristiques lors de l'amorçage. La section 2.5.2.3, plus bas, décrit comment forcer la détection de ce type de contrôleurs.

- **Géométrie du disque dur non reconnue.** Quelques systèmes, comme l'IBM PS/ValuePoint, ne stockent pas les informations sur la géométrie du disque en mémoire CMOS, où Linux pense la trouver. De même, certaines cartes contrôleurs SCSI nécessitent qu'on leur précise où trouver cette géométrie du disque de manière à ce que Linux sache comment gérer votre disque dur.

  La plupart des distributions proposent une option pour spécifier la géométrie du disque lors de l'amorçage. En général, en amorçant le système sur le support d'installation, vous pouvez préciser cette géométrie à l'invite de LILO avec une commande comme :

  ```
  boot: linux hd=⟨cylindres⟩,⟨têtes⟩,⟨secteurs⟩
  ```

  où ⟨*cylindres*⟩, ⟨*têtes*⟩, et ⟨*secteurs*⟩ correspondent au nombre de cylindres, de têtes et de secteurs par piste de votre disque dur.

  Après avoir installé le système, vous pourrez alors installer LILO, qui vous permettra d'amorcer Linux directement depuis le disque dur. A ce moment, vous pourrez préciser la géométrie du disque à la procédure d'installation de LILO, afin d'éviter d'avoir à la préciser manuellement à chaque lancement du système. Voyez le chapitre 4 pour plus d'informations sur LILO.

2.5.2.3 Problèmes avec les périphériques et contrôleurs SCSI

Voici quelques uns des problèmes les plus fréquemment rencontrés avec les cartes contrôleurs SCSI et des périphériques comme les disques durs, CD-ROMS, lecteurs de bandes magnétiques. Si Linux rechigne à reconnaître votre contrôleur ou votre disque dur, lisez ce qui suit.

Le « Linux SCSI HOWTO » (voir annexe A) contient bien plus d'informations utiles que ce qui est cité ici. SCSI est une interface très universelle, mais dont la configuration est parfois critique.

- **Un périphérique SCSI est détecté sur toutes les « ID » possibles.** C'est dû au fait que vous avez configuré le périphérique à la même adresse que le contrôleur. Vous devez changer la position du cavalier sur le périphérique pour qu'il utilise une adresse différente de celle du contrôleur.

- **Linux reporte des erreurs de « sense » , même si les périphériques sont connus pour ne comporter aucune erreur.** Ce peut être dû à des câbles de liaison défectueux, ou à un mauvais terminateur. Si votre bus SCSI n'est pas terminé à chaque extrémité, vous pouvez avoir des erreurs d'accès aux périphériques. En cas de doute, testez vos câbles.

- **Des erreurs de « timeout » SCSI sont détectées.** Il s'agit généralement d'un conflit d'IRQ, de DMA ou d'adresses. Vérifiez également que les interruptions sont correctement validées sur le contrôleur.

- **Les contrôleurs SCSI qui utilisent un BIOS ne sont pas détectés.** La détection des contrôleurs utilisant un BIOS échouera si celui-ci n'est pas mis en service, ou si la « signature » du contrôleur n'est pas reconnue par le noyau. Voyez le « Linux SCSI HOWTO » pour plus d'informations à ce sujet.

- **Les contrôleurs utilisant une zone de mémoire partagée ne fonctionnent pas.** C'est le cas lorsque cette zone n'est pas correctement gérée par le cache. Soit vous devez marquer cette zone comme non cachable dans la configuration XCMOS, soit supprimer tout.

- **Lors du partitionnement, vous avez un avertissement à propos de « cylindres > 1024 » , ou vous n'arrivez pas à lancer le système depuis une partition utilisant plus de 1024 cylindres.** Le BIOS limite le nombre de cylindres à 1024, et toute partition utilisant plus de cylindres ne sera pas accessible depuis le BIOS. En ce qui concerne Linux, cela n'affecte que l'amorçage du système; une fois que Linux tourne vous pouvez accéder sans limites à cette partition. Vous pouvez choisir soit d'amorcer Linux depuis une disquette, soit depuis une partition comportant moins de 1024 cylindres. Voyez la section 2.3.7 pour créer une disquette d'amorçage ou l'installation de LILO.

- **Le lecteur de CD-ROM ou d'autres périphériques de media extractibles ne sont pas reconnus lors de l'amorçage.** Essayez avec un CD-ROM, ou disque, ou bande magnétique, dans le lecteur lors de la mise en route de l'ordinateur. Certains périphériques en ont besoin.

Si votre contrôleur SCSI n'est pas reconnu, vous pouvez avoir besoin de forcer sa détection. C'est particulièrement important pour les contrôleurs dépourvus de BIOS. La plupart des distributions vous permettent de préciser l'adresse et l'IRQ depuis le support d'installation. Par exemple, si vous utilisez un contrôleur TMC-8xx, vous devriez pouvoir spécifier

```
boot: linux tmx8xx=⟨interruption⟩,⟨adresse-mémoire⟩
```

à l'invite de LILO, où ⟨*interruption*⟩ est l'IRQ du contrôleur et ⟨*adresse-mémoire*⟩ la zone mémoire utilisée. Cette possibilité dépend de la distribution de Linux que vous utilisez; consultez sa documentation.

2.5.3 Problèmes lors de l'installation du système

En principe l'installation des fichiers composant le système devrait se faire sans encombre, si vous avez un peu de chance. La seule chose qui pourrait vous arriver serait de tomber sur un support défectueux au cours de la copie des fichiers ou d'être face à un manque de place sur vos partitions. Voici une liste de ces problèmes courants.

- **Vous voyez des messages comme « `Read error` » , « `file not found` » , ou d'autres erreurs lors de la copie du système.** Vous avez un problème avec votre support d'installation. S'il s'agit de disquettes, vous devez savoir que ce type de support est très sujet à ce genre d'erreurs. Assurez-vous toujours d'utiliser des disquettes de qualité, neuves, fraîchement formatées. Si vous avez une partition MS-DOS sur votre disque dur, beaucoup de distributions permettent d'installer Linux à partir de là. Ce sera beaucoup plus fiable et beaucoup plus rapide que de jouer au grille-pain avec le lecteur de disquettes.

  Si vous partez d'un CD-ROM, vérifiez qu'il n'est ni rayé, ni poussiéreux, et que c'est bien Linux et pas de la musique...

  La cause du problème peut également être l'utilisation d'un media incorrect. Par exemple, lors d'installation par disquettes, beaucoup de distributions de Linux nécessitent que celles-ci soient haute densité et formatées selon le format MS-DOS (la disquette d'amorçage étant une exception). Si vraiment rien ne va plus, procurez-vous un nouveau jeu de disquettes soigneusement réalisées.

- **Le système affiche des erreurs commes « `tar: read error` » ou « `gzip: not in gzip format` » .** Ce problème est généralement dû à des fichiers corrompus sur le media d'installation. En d'autre termes, votre disquette peut être en parfait état, mais les données qui sont dessus abîmées pour une raison inconnue. Si vous avez téléchargé ces fichiers par FTP en oubliant d'utiliser le mode binaire, vous avez obtenu des données corrompues qui provoqueront à coup sûr ce type d'erreur.

- **Le système affiche « `device full` » lors de l'installation.** Vous n'avez plus assez d'espace disque disponible pour continuer l'installation. Peu de distributions de Linux prévoient ce cas, vous êtes censés prévoir la place disque nécessaire avant. N'espérez pas arrêter le processus d'installation en trouvant un système utilisable à ce moment là.

La solution est souvent de recréer vos systèmes de fichiers (avec la commande **mke2fs**), ce qui efface d'un coup le système partiellement installé, et refait place nette. Vous pouvez alors tenter l'installation à nouveau, en sélectionnant cette fois moins de paquetages que précédemment. Dans d'autres cas, vous pouvez avoir besoin de revoir votre stragégie de partitionnement du disque dur et par conséquent tout recommencer.

- **Le système affiche des erreurs comme « `read_intr: 0x10` » lors de l'accès au disque dur** C'est généralement l'indication que votre disque dur possède des secteurs défectueux. Toutefois si ces messages surviennent durant l'utilisation de **mkswap** ou **mke2fs**, il se peut que le système n'arrive pas à accéder au disque dur. Ce peut être dû à une erreur matérielle (voir section 2.5.2), ou une mauvaise spécification de la géométrie du disque. Si vous avez utilisé l'option

  hd=⟨*cylindres*⟩,⟨*têtes*⟩,⟨*secteurs*⟩

  lors de l'amorçage pour forcer la détection de la géométrie du disque, et que les paramètres sont incorrects, vous obtiendrez ce type d'erreur. Ceci peut aussi se produire si la géométrie du disque est incorrectement paramétrée dans la mémoire CMOS de la machine.

- **Le système trouve des « `file not found` » ou « `permission denied` ».** Ce problème peut arriver s'il manque des fichiers sur le support d'installation (voir paragraphe suivant) ou si les fichiers à installer n'ont pas les bonnes permissions d'accès. Il est arrivé que certaines distributions de Linux soient mal réalisées ou que la procédure d'installation soit boguée. Si c'est le cas, contactez le responsable de la distribution pour signaler le problème afin qu'il soit corrigé le plus rapidement possible.

Si vous rencontrez d'autres erreurs étranges lors de l'installation de Linux (particulièrement si vous avez téléchargé vous même les fichiers), vérifiez bien que tous les fichiers nécessaires ont été récupérés lors de l'opération. Par exemple, certains utilisent la commande FTP

```
mget *.*
```

pour télécharger Linux par FTP. Ceci ne ramènera que les fichiers qui contiennent un « . » dans leur nom; s'il existe des fichiers sans « . » vous les raterez tous. La commande correcte pour tout télécharger est

```
mget *
```

Le meilleur conseil que l'on puisse vous donner, est de vérifier une à une les étapes lorsque quelque chose se passe mal. Vous pouvez penser avoir tout fait correctement, mais en réalité avoir oublié un tout petit détail crucial au cours de l'installation. Dans beaucoup de cas, recommencer les choses calmement en téléchargeant ou en recommencant l'installation à tête reposée résoudra le problème.

Si Linux « plante » de façon inattendue pendant l'installation, c'est peut être le signe d'un quelconque problème matériel. Voyez la section 2.5.2 dans ce cas.

2.5.4 Problèmes après l'installation

Vous venez de passer une après-midi entière à installer Linux. Pour lui faire de la place, vous avez sacrifié vos partitions MS/DOS et OS/2, et tristement effacé vos copies de Doom, SimCity et Wing Commander. Vous réamorcez le système, et rien ne se passe. Ou pire, *quelque chose d'anormal* se passe. Que devez-vous faire?

Dans la section 2.5.1, nous avons décrit les problèmes les plus courants qui peuvent arriver lors du lancement de Linux depuis le support d'installation. Beaucoup de ce qui y est dit s'applique également ici. De plus, vous pouvez être victime de l'une des maladies qui vont suivre.

2.5.4.1 Problèmes d'amorçage depuis une disquette

Si vous utilisez une disquette pour amorcer Linux, vous pouvez avoir besoin de spécifier l'emplacement de votre partition racine. Ceci est particulièrement vrai si vous utilisez la disquette d'installation elle même, et non une disquette d'amorçage faite sur mesure pendant l'installation.

Dans ce cas, pendant l'amorçage de la disquette, gardez appuyée la touche `shift` ou `ctrl`. Vous devriez voir apparaître un menu, pressez `tab` pour voir la liste des options disponibles. Par exemple, beaucoup de distributions vous permettent de taper

```
boot: linux hd=⟨partition⟩
```

où ⟨*partition*⟩ est le nom de la partition racine de Linux, comme par exemple `/dev/hda2`. Consultez la documentation de votre distribution pour plus de détails.

2.5.4.2 Problèmes d'amorçage depuis le disque dur

Si vous avez choisi d'installer LILO au lieu de créer une disquette d'amorce, vous devriez pouvoir lancer Linux depuis le disque dur. Toutefois la procédure d'installation de LILO utilisée par beaucoup de distributions n'est pas toujours parfaite. Elle peut se tromper sur votre shéma de partitionnement, auquel cas vous devrez réinstaller LILO proprement pour rétablir la situation. L'installation de LILO est décrite dans le chapitre 4.

- **La machine affiche** « `Drive not bootable---Please insert system disk.` »
  Vous aurez ce message d'erreur si le secteur maître d'amorçage du disque dur (master boot record) est corrompu. Dans la plupart des cas, ce n'est pas grave

du tout, et tout le reste du disque est intact. Il y a plusieurs solutions à ce problème:

1. Lors du partitionnement du disque avec **fdisk**, vous pouvez avoir supprimé la partition qui était marquée comme « active » . MS-DOS et les autres systèmes d'exploitation tentent d'amorcer sur la partition « active » (Linux peut ne pas s'en préoccuper). Vous devriez pouvoir lancer la machine avec une disquette MS-DOS, utiliser **FDISK** et re-activer la partition voulue, et tout ira bien.

   Une autre commande à essayer (avec MS-DOS 5.0 et plus) est

   ```
   FDISK /MBR
   ```

   Cette commande reconstruit le secteur maître d'amorçage du disque dur, supprimant LILO. Si vous n'avez plus MS-DOS sur votre disque, vous devrez lancer Linux depuis une disquette et tenter de réinstaller LILO plus tard.

2. Si vous avez créé une partition MS-DOS avec la version Linux de **fdisk**, ou vice versa, vous pouvez obtenir cette erreur. Utilisez toujours par précaution les commandes **fdisk** ou équivalent de chaque système d'exploitation pour travailler ses propres partitions. La meilleure solution dans ce cas est de repartir à zéro et de repartitionner le disque correctement.

3. La procédure d'installation de LILO peut avoir échoué. Dans ce cas, vous devez amorcer Linux depuis votre disquette d'amorçage (si vous en avez une), ou depuis le support d'installation original. Chacune de ces méthodes devrait vous permettre de préciser la partition racine lors de l'amorçage. Gardez `shift` ou `ctrl` appuyées pendant le lancement et pressez `tab` dans le menu qui s'affichera pour avoir la liste des options.

– **Lors de l'amorçage depuis le disque dur, c'est MS-DOS (ou un autre système) qui se charge au lieu de Linux.** Tout d'abord, assurez-vous de bien avoir installé LILO. Sinon, la machine continuera à charger le système d'exploitation que vous aviez avant. Vous devez installer LILO pour pouvoir lancer Linux directement depuis le disque dur.

  D'un autre côté, si vous avez *vraiment* installé LILO, c'est que celui-ci est configuré pour charger un autre système d'exploitation par défaut. Pendant que le système s'amorce, appuyez sur `shift` ou `ctrl`, et pressez `tab` ensuite. Cela vous présentera une liste des systèmes d'exploitation possibles, sélectionnez l'option appropriée (généralement, « **linux** ») pour lancer Linux.

  Si vous voulez que Linux soit le système chargé par défaut, vous devrez ré-installer LILO. Voir le chapitre 4 pour cette opération.

  Il est aussi possible que vous ayez tenté d'installer LILO mais que l'installation ait échoué quelque part. Voir l'item précédent.

2.5.4.3 Problèmes pour se connecter

Après avoir lancé Linux, votre écran devrait afficher quelque chose comme ceci:

```
linux login:
```

À ce point, soit la documentation de votre distribution, soit le système lui-même vous indiquera quoi faire. Dans le cas général, vous devez vous connecter simplement en tant que `root`, qui n'a pas encore de mot de passe. Si cela vous est refusé (peu probable), essayez pour voir `guest` ou `test` (voire `install` ou `setup`).

La plupart des systèmes Linux fraîchement installés ne devraient pas demander de mots de passe pour les sessions initiales. Toutefois, si un mot de passe vous est demandé, c'est un vrai problème. D'abord, tentez d'utiliser un mot de passe équivalent au nom de l'utilisateur, si vous vous connectez comme `root`, essayez `root` comme mot de passe.

Si vraiment vous ne pouvez pas accéder au système, ne paniquez pas. Lisez d'abord la documentation de votre distribution, l'utilisateur et le mot de passe à utiliser y sont peut être indiqués. Regardez bien l'écran : si ça se trouve, ces indications vous crèvent les yeux.

Une cause possible de ce problème peut se trouver dans les fichiers d'initialisation de Linux ou le programme login. Si c'est le cas, vous devrez sans doute réinstaller (au moins en partie) Linux, ou lancer le support d'installation pour tenter de régler ce problème manuellement. Voir dans le chapitre 4 comment pirater votre propre machine.

Si vous êtes certain de donner le bon mot de passe et qu'il est sans cesse refusé, vérifiez votre clavier : s'il est français, mais que Linux est resté configuré pour le modèle américain, certaines touches ne correspondront pas à ce que vous pensez taper...

2.5.4.4 Problèmes d'utilisation

Si vous avez pu vous loguer, vous devriez être sous shell avec une invite comme « `#` » (si vous êtes root) ou « `$` » (simple utilisateur), et vous pouvez commencer à faire un petit tour du propriétaire. Toutefois il peut y avoir quelques problèmes initiaux qui peuvent devenir énervants si on n'y remédie pas.

Une mauvaise configuration des permissions d'accès à des fichiers ou des répertoires peuvent provoquer le message d'erreur

```
Shell-init: permission denied
```

(en fait, chaque fois que vous voyez le message « `permission denied` » vous pouvez être certain qu'il s'agit d'un problème de droits d'accès, qui peut d'ailleurs être volontaire.)

Dans beaucoup de cas, il suffit d'utiliser la commande `chmod` pour corriger les permissions des fichiers ou répertoires en question. Par exemple, quelques mauvaises distributions de Linux utilisèrent une fois le mode (incorrect) 0644 pour le répertoire racine (/). La correction consistait simplement à taper la commande

```
# chmod 755 /
```

en tant qu'utilisateur `root`. Cependant, pour que cela marche, il fallait amorcer Linux depuis le support d'installation et monter le système de fichiers racine manuellement ; de quoi s'arracher les cheveux pour la majorité des nouveaux utilisateurs.

Au fur et à mesure de l'utilisation du système, vous pourrez trouver ici ou là des fichiers ou des répertoires ayant des droits d'accès incorrects, ou des programmes qui ne fonctionnent pas avec leur configuration initiale. Bienvenue dans le monde de Linux ! Bien que la plupart des distributions soient excellentes, aucune n'est vraiment parfaite. Il est bien entendu impossible de décrire dans ce livre tous les problèmes que vous pourriez rencontrer. Au lieu de cela, tout au long de cet ouvrage nous vous aiderons à résoudre la plupart de ces problèmes de configuration en vous apprenant comment les trouver et les corriger vous-même. Dans le chapitre 1, nous avons expliqué cette manière d'aborder Linux, et dans le chapitre 4 nous expliquons comment corriger beaucoup de ces erreurs grossières.

Chapitre 3

Didacticiel Linux

3.1 Introduction

Les nouveaux venus à UNIX et Linux peuvent être un peu impressionnés par la taille et la complexité apparente du système auquel ils sont confrontés. Il y a beaucoup d'excellents livres sur l'utilisation d'UNIX, pour tous les niveaux de connaissances, depuis le novice jusqu'à l'expert. Toutefois, aucun de ces ouvrages n'est dédié à Linux en particulier. Comme 95% de l'utilisation de Linux ne présente aucune différence avec d'autres systèmes UNIX, la meilleure façon de prendre contact avec votre nouveau système est de lire un guide d'apprentissage adapté spécialement à Linux. Voici donc un tel guide.

Ce chapitre ne s'embarrasse pas de détails et ne décrit pas les notions les plus complexes. Au lieu de cela, il est conçu pour apprendre pas à pas l'utilisation de Linux, de telle façon que l'utilisateur puisse ensuite lire sans être perdu un ouvrage plus général concernant UNIX, et être capable de repérer facilement et comprendre les quelques différences entre le système décrit et Linux.

Très peu de connaissances sont demandées ici, sauf peut-être quelque familiarité avec les micro-ordinateurs, et MS-DOS. Malgré tout, même si vous n'êtes pas un utilisateur de MS-DOS, vous devriez être capable de comprendre pratiquement tout ce qui est décrit. Au premier coup d'œil, UNIX ressemble beaucoup à MS-DOS (après tout, MS-DOS fut inspiré du système d'exploitation CP/M, qui lui même était inspiré d'UNIX). Toutefois cette ressemblance n'est que superficielle. Même si le monde des PC est totalement nouveau pour vous, ce guide d'initiation devrait pouvoir vous aider.

Et, avant de commencer : *n'ayez pas peur d'expérimenter.* Le système ne vous fera pas de mal. Vous ne pouvez rien abîmer en utilisant le système. UNIX, par conception, possède des notions de sécurité, afin d'éviter aux utilisateurs « normaux » (le rôle que vous allez jouer à partir de maintenant) de corrompre des données qui sont vitales pour le fonctionnement du système. Et même sans cela, le pire qui pourrait vous arriver serait d'effacer la totalité de vos fichiers; vous auriez alors à réinstaller totalement Linux. Donc pour l'instant, vous n'avez rien à perdre.

3.2 UNIX, Concepts de base

UNIX est un système d'exploitation multi-tâches et multi-utilisateurs. Cela veut dire qu'il peut y avoir plusieurs personnes utilisant la machine en même temps, travaillant sous différentes applications. (À la différence de MS-DOS, où seulement une personne peut utiliser l'ordinateur et qui ne peut exécuter qu'un seul programme à la fois). Sous UNIX, pour que les utilisateurs soient reconnus par le système, ils doivent d'abord s'y « connecter » , ce qui se fait en deux étapes : entrer son **nom d'utilisateur** (le **login**), puis donner son **mot de passe**, qui est une clé secrète associée à chaque utilisateur. Personne d'autre que vous ne peut usurper votre identité sur le système puisque vous êtes le seul à connaître votre mot de passe.

Sur les systèmes UNIX traditionnels, l'administrateur système vous affectera un nom d'utilisateur et un mot de passe initial lorsqu'il vous ouvrira un compte sur la machine. Cependant, comme vous êtes l'administrateur de votre propre système, vous devez réaliser ces opérations vous même. Lisez la section 3.2.1 pour savoir comment faire. Pour la démonstration, nous utiliserons un utilisateur imaginaire que nous appellerons « `dupont` » .

De plus, chaque système UNIX se voit assigner un nom, le **hostname**. C'est ce nom qui personnalise votre machine, et lui donne charme et caractère. Ce nom est utilisé pour identifier chaque machine sur un réseau, mais même si votre ordinateur est isolé, vous devez lui donner un nom car il est nécessaire à beaucoup de programmes. La section 4.10.2 explique comment procéder. Pour nos exemples, nous appellerons le système « `bistro` » .[1]

3.2.1 Création d'un compte utilisateur

Avant de pouvoir utiliser le système, vous devez créer un compte utilisateur pour vous-même. C'est impératif car il n'est généralement pas une bonne idée d'utiliser le compte `root` pour une utilisation courante. Le compte `root` doit être réservé à l'usage de commandes privilégiées pour la maintenance du système, comme expliqué dans la section 4.1. Ceci est très important ! Vous avez tous les droits si vous êtes `root`, y compris de tout casser. L'utilisation journalière d'un système UNIX ne doit se faire que comme utilisateur ordinaire, mettez-vous bien ça en tête.

1. L'orthographe vous autorise également « bistrot » . Un nom de machine doit être simple, court et mémorisable. « Mastroquet » ,« débit-de-boisson » , « Chez-Marcel » ou « café-de-la-paix » étaient résolument trop longs.

Afin de vous créer un compte, vous devez entrer en tant que `root` (oui, dans ce cas c'est indispensable) et utiliser soit la commande `useradd`, soit la commande `adduser`, ou encore `mkuser`, selon le système que vous utilisez. Voyez la section 4.4 pour plus d'informations sur cette procédure.

3.2.2 Se connecter sur le système

Au moment de vous connecter, vous verrez un message ressemblant à ceci sur votre écran:

```
bistro login:
```

Entrez votre nom d'utilisateur, et pressez la touche `Entrée`. Notre héros, `dupont`, taperait ceci:

```
bistro login: dupont
Password:
```

Maintenant entrez votre mot de passe. Il ne s'affichera pas à l'écran, (pour éviter que l'on regarde par dessus votre épaule), aussi entrez-le soigneusement. Si vous vous trompez, vous verrez le message

```
Login incorrect
```

et vous pourrez recommencer.

Une fois que vous avez entré correctement votre nom d'utilisateur et votre mot de passe, vous êtes officiellement entré dans le système, et libre de vous promener un peu dedans.

3.2.3 Consoles virtuelles

La **console** système correspond à l'écran et au clavier directement connectés à l'ordinateur. (UNIX étant multi-utilisateurs, vous pouvez avoir d'autres terminaux connectés sur la machine, chacun avec un écran et un clavier, mais ce ne serait pas la console). Linux, comme d'autres versions d'UNIX, offre des **consoles virtuelles** (ou VC), qui vous permettent d'avoir plus d'une session à la fois sur votre console.

Pour démontrer cette possibilité, connectez-vous sur votre système comme expliqué ci-dessus. Maintenant, pressez `alt-F2`. Vous devriez voir à nouveau l'invite `login:`. Vous voyez en fait la seconde console virtuelle, vous êtes déjà connecté sur la première. Pour revenir à la précédente, pressez `alt-F1`. *Et voilà !* Vous êtes de retour dans votre précédente session.

Un système Linux fraîchement installé ne vous donnera probablement accès qu'aux quatre premières consoles virtuelles, depuis `alt-F1` jusquà `alt-F4`. Vous pouvez avoir jusqu'à 12 consoles virtuelles, une par touche de fonction sur votre clavier, c'est une simple question de configuration. Comme vous le voyez, l'utilisation de consoles virtuelles peut être extrêmement utile, vous pouvez travailler sur différentes consoles en même temps.

Bien que l'utilisation de ces consoles virtuelles ait quelques limites (vous ne pouvez en voir qu'une seule à la fois), cela devrait vous donner une (toute petite) idée des capacités multi-utilisateurs d'UNIX.

3.2.4 Shells et commandes

Pour l'essentiel de vos explorations du monde UNIX, vous discuterez avec le système par le biais d'un **shell**, un interpréteur de commandes. Un shell n'est rien d'autre qu'un programme qui prend ce que l'utilisateur tape et traduit ces commandes en instructions destinées au système. Vous pouvez le comparer au programme `COMMAND.COM` sous MS-DOS, qui a essentiellement la même fonction. Le shell n'est juste qu'une interface. Les interfaces avec le système sont nombreuses et variées, depuis les nombreux shells disponibles jusqu'au système X Window, qui vous permet d'utiliser des commandes en utilisant le clavier ou la souris, avec affichage sur un écran graphique du plus bel effet.

Dès que vous vous connectez, le système lance le shell, et vous pouvez taper des commandes. Voici un petit exemple. Ici, Dupont entre sur le système, et se retrouve devant l'**invite** du shell, qui s'appelle le **prompt** en anglais.

```
bistro login: dupont
Password: mot de passe de dupont
Bienvenue au Bistrot !

/home/dupont$
```

« `/home/dupont$` » est l'invite du shell, indiquant qu'il est prêt à prendre des commandes. (plus de détails sur ce qu'indique cette invite plus tard). Essayons de demander au système de faire quelque chose d'intéressant:

```
/home/dupont$ servir a boire
servir: command not found
/home/dupont$
```

Comme c'est dommage! Visiblement la machine ne veut pas prendre notre commande.

Une question vous brûle aussitôt les lèvres : qu'est-ce qu'une commande ? qu'est ce qui se passe lorsque l'on tape « `servir a boire` » ? Le premier mot de la commande, « `servir` » , est le nom de la commande à exécuter. Tout le reste de la ligne est pris comme arguments de cette commande. Exemples:

```
/home/dupont$ cp toto titi
```

Ici, la commande s'appelle « `cp` » , et les arguments sont « `toto` » et « `titi` » .

Lorsque vous tapez une commande, le shell fait plusieurs choses. Tout d'abord, il regarde le nom de la commande et teste s'il s'agit d'une commande qu'il connaît en interne. (C'est à dire une commande qu'il sait exécuter tout seul. Il en existe un certain nombre que nous verrons plus tard). Ce shell regarde aussi si cette commande est un alias, un synonyme, d'une autre commande. Si aucune de ces conditions n'est vérifiée, le shell recherche alors un programme sur le disque, portant le nom de la commande. S'il trouve un tel programme, il l'exécute, en lui passant les arguments spécifiés sur la ligne de commande.

Dans notre exemple, le shell recherche un programme appelé « `servir` » , et ne le trouve pas, nous signale ce fait, et se remet en attente d'autres commandes. Cette situation est très fréquente, lorsque l'on fait une faute de frappe par exemple.

Souvent, ce genre de démonstration se fait en langue anglaise, (les ordinateurs parlent souvent anglais), à l'aide du programme appelé « `make` » , qui est destiné en principe à réaliser la compilation de gros programmes. « `Make` » , selon les arguments qu'on lui passe, peut afficher des messages d'erreurs amusant beaucoup l'anglophone malicieux. Essayez « `make love` » , si vous comprenez l'anglais, et imaginez-en d'autres.

3.2.5 Se déconnecter

Avant d'aller plus loin, nous devons vous expliquer comment vous déconnecter proprement du système. Ce n'est pas indispensable mais ça se fait. Notamment, si vous laissez votre terminal logé sur votre compte en votre absence, n'importe qui peut usurper votre identité. A l'invite du shell, tapez

```
/home/dupont$ exit
```

pour terminer votre session. Il existe d'autre méthodes, mais celle-ci est assurée de marcher partout.

3.2.6 Changer votre mot de passe

Vous devez aussi savoir comment changer votre mot de passe. La commande `passwd` vous demandera par sécurité votre ancien mot de passe, puis un nouveau. Elle vous demandera de rentrer une seconde fois le nouveau, pour bien vérifier que vous n'avez pas fait de faute de frappe, et l'enregistrera. Faites très attention, n'oubliez jamais votre mot de passe ! Les mots de passe sont cryptés et personne ne pourra vous le rappeler. Si vous le perdez, vous devrez demander à l'administrateur système de vous en remettre un autre. (Si l'administrateur système, c'est vous, voyez la section 4.4.)

3.2.7 Fichiers et répertoires

Sous la plupart des systèmes d'exploitation (y compris UNIX), il existe le concept de **fichier**, qui est un ensemble d'informations auquel on donne un nom significatif (appelé le **nom de fichier**). Des exemples de fichiers pourraient être un courrier électronique, le texte d'une recette de cuisine, ou un programme informatique qui peut être exécuté. Pratiquement, tout ce qui est écrit sur le disque dur est sauvé sous forme de fichier individuel.

Les fichiers sont identifiés par leur nom de fichier. Par exemple, le fichier contenant la recette de cuisine pourrait être enregistré sous le nom de `cassoulet`. Ces noms sont en général choisis de manière à être compréhensibles pour un humain. Il n'y a pas de format standard pour les noms de fichiers, comme il en existe sous MS-DOS ou ailleurs; en général, les noms de fichiers peuvent contenir n'importe quel caractère (sauf `/`, voir la discussion sur les chemins d'accès plus loin), et peuvent faire jusqu'à 256 caractères de longs, voire plus.

Au concept de fichier est associé la notion de répertoire. Un **répertoire** est simplement une collection de fichiers. On donne aussi bien sûr un nom à chaque répertoire, pour pouvoir l'identifier aussi facilement que les fichiers. De plus, les répertoires sont organisées de manière arborescente; c'est à dire que les répertoires peuvent contenir d'autres répertoires, indéfiniment.

Un fichier peut être référencé par son **chemin d'accès**, qui est constitué du nom de fichier, précédé par le nom du répertoire qui le contient. Par exemple, disons que Dupont possède un répertoire nommé `recettes`, qui contient trois fichiers: `cassoulet`, `confit`, et `soupe` (chacun de ses fichiers contenant le texte de la recette par exemple). Pour se référer au fichier `soupe`, Dupont peut spécifier le chemin d'accès au fichier:

```
recettes/soupe
```

Comme vous pouvez le voir, le répertoire et le nom de fichier sont séparés par un simple caractère, la barre oblique, (`/`) appelée « slash » . C'est pour cette raison que ce caractère est lui-même interdit dans les noms de fichiers. Les utilisateurs de MS-DOS trouveront cette convention familière, bien que sous ce système le séparateur soit l'anti-slash (`\`) pour compliquer les choses.

Comme nous l'avons dit, les répertoires peuvent être imbriqués les uns dans les autres. Par exemple, considérons que Dupont a un autre répertoire, à l'intérieur de `recettes`, qui se nomme `vomitifs`. Ce répertoire contiendrait les fichiers `macdo` et `pacific`. Le chemin d'accès du fichier `macdo` serait alors

```
recettes/vomitifs/macdo
```

Par conséquent, le chemin d'accès est vraiment un « chemin » qui vous mène directement au fichier spécifié. Le répertoire précédent un sous-répertoire est appelé le **répertoire père**. Ici, le répertoire **recettes** est le père de **vomitifs**.

3.2.8 Arborescence de répertoires

La plupart des systèmes UNIX ont une manière assez standard d'organiser les fichiers, de manière à pouvoir facilement localiser les ressources du système sans y passer des heures. Cet agencement forme une arborescence de répertoires, qui commence au répertoire « `/` » , connu comme le « répertoire racine » . Directement sous « `/` » sont placés quelques sous-répertoires importants: `/bin`, `/etc`, `/dev`, et `/usr`, entre autres. Ces répertoires contiennent à leur tour d'autres répertoires qui contiennent des programmes, des fichiers de configuration, et ainsi de suite.

En particulier, chaque utilisateur possède son **répertoire personnel**, ou **home directory**, qui est le répertoire réservé à cet utilisateur pour y stocker ses fichiers et y travailler. Dans l'exemple précédent, tous les fichiers de Dupont (comme `pacific` ou `cassoulet`) étaient situés dans son répertoire personnel. En général, ces répertoires personnels sont placés dans `/home`, et prennent le nom de l'utilisateur qui possède ce répertoire (mais ce n'est pas obligatoire). Par conséquent, le répertoire personnel de Dupont est `/home/dupont`.

Sur la figure 3.2.8, un exemple d'arborescence de répertoires est représenté. Il devrait vous donner une idée de la façon dont l'arborescence de fichiers de votre système est organisée.

3.2.9 Le répertoire courant

À tout moment, les commandes que vous tapez sous shell sont référencées par rapport à votre **répertoire courant**. Vous pouvez vous représenter ce répertoire comme celui dans lequel vous êtes présentement « positionné » . Lorsque vous vous connectez, votre répertoire courant est par défaut votre répertoire personnel, `/home/dupont` dans notre cas. Chaque fois que vous référencez un fichier, vous pouvez le faire en relation avec le répertoire courant, au lieu de donner son chemin d'accès complet.

Voici un exemple. Dupont possède le répertoire `recettes`, et `recettes` contient le fichier `soupe`. Si Dupont veut voir le contenu de ce fichier, il peut utiliser la commande

```
/home/dupont$ more /home/dupont/recettes/soupe
```

La commande `more` affiche simplement un fichier texte, un écran à la fois. Toutefois, puisque le répertoire courant de Dupont est actuellement `/home/dupont`, il peut au lieu de cela appeler le fichier de manière *relative* à ce répertoire courant. La commande serait alors

```
/home/dupont$ more recettes/soupe
```

Par conséquent, si vous commencez un nom de fichier (comme `recettes/soupe`) avec un caractère autre que « / » , le système considère que vous vous référez au fichier de manière relative au répertoire courant. Ceci est connu sous le nom de **chemin d'accès relatif**.

D'un autre côté, si vous commencez un nom de fichier par « / » , le système l'interprétera comme un chemin complet, c'est à dire commençant depuis le répertoire racine, /. Ceci est connu sous le nom de **chemin d'accès absolu**.

3.2.10 Se référer au répertoire personnel

Sous les deux shells `tcsh` et `bash`,[2] votre répertoire personnel peut être notifié par convention, par le caractère tilde (« ~ »). Par exemple, la commande

```
/home/dupont# more ~/recettes/cassoulet
```

est équivalente à

```
/home/dupont$ more /home/dupont/recettes/cassoulet
```

Le caractère « ~ » est simplement remplacé par le nom de votre répertoire personnel par le shell.

De plus, vous pouvez de cette manière référencer très facilement les répertoires personnels des autres utilisateurs. Le chemin « `~rene/boissons` » sera traduit en « `/home/rene/boissons` » par le shell (si `/home/rene` est bien le répertoire personnel de René). L'utilisation du tilde est un simple raccourci; il n'existe pas de répertoire « ~ » , c'est juste une convention syntaxique offerte par le shell.

3.3 Premiers pas sous UNIX

Avant d'entrer dans le vif du sujet, vous devez savoir que tous les noms de fichiers et les commandes sur un système UNIX, sont ce que l'on appelle « case-sensitive »

2. Les programmes `tcsh` et `bash` sont deux des *shells* disponibles sous Linux. Le shell est l'interpréteur de commandes qui lit ce que l'utilisateur tape au clavier et exécute l'action correspondante.

, c'est à dire que le système fait une différence entre les lettres majuscules et minuscules (contrairement à certains systèmes comme MS-DOS). Par exemple, la commande `date` est très différente de `Date` ou `DATE`, même chose pour les noms de fichiers.

3.3.1 Se déplacer

Maintenant que nous pouvons nous connecter, et référencer un chemin d'accès à un fichier, comment pouvons-nous changer de répertoire courant, pour se simplifier la vie ?

La commande permettant de se déplacer dans l'arborescence de fichiers est `cd`, ce qui peut se mémoriser par « change directory » . Vous noterez que les commandes les plus utilisées sous Unix ne font que deux ou trois lettres. On utilise `cd` de la façon suivante :

cd ⟨*répertoire*⟩

où ⟨*répertoire*⟩ est le nom du répertoire où vous voulez aller.

Comme nous l'avons vu, au départ vous êtes dans votre répertoire personnel. Si Dupont voulait se déplacer dans le sous-répertoire `recettes`, il utiliserait donc la commande:

```
/home/dupont$ cd recettes
/home/dupont/recettes$
```

Comme vous pouvez le voir, l'invite change pour refléter le répertoire courant (de manière à se repérer facilement). Cette possibilité dépend du shell utilisé. Maintenant qu'il est dans le répertoire `recettes`, notre ami peut lire sa recette de cassoulet directement par la commande

```
/home/dupont/recettes$ more cassoulet
```

Maintenant, Dupont est dans son sous-répertoire `recettes`. Pour revenir un cran au-dessus, dans le répertoire parent, il peut utiliser la commande

```
/home/dupont/recettes$ cd ..
/home/dupont$
```

(Notez bien l'espace entre le « `cd` » et les deux points « `..` » .) Chaque répertoire possède une entrée spéciale nommée « `..` » , qui se réfère au répertoire père. De façon similaire, chaque répertoire possède une entrée nommée « `.` » , qui se réfère à lui-même. Par conséquent la commande

```
/home/dupont/recettes$ cd .
```

nous déplace là où nous sommes, c'est à dire ne change rien.

Vous pouvez aussi indiquer des chemins d'accès absolus à la commande `cd` bien sûr. Pour aller dans le répertoire personnel de René, nous pouvons utiliser la commande

```
/home/dupont/recettes$ cd /home/rene
/home/rene$
```

L'utilisation de `cd` sans aucun argument nous ramène dans notre répertoire personnel, c'est un raccourci très pratique:

```
/home/rene$ cd
/home/dupont$
```

3.3.2 Afficher le contenu des répertoires

Maintenant que nous pouvons nous promener un peu partout, vous allez sans doute nous dire « Bon, et alors ? » . Il est vrai que ça ne nous avance pas à grand chose pour l'instant, aussi voyons une nouvelle commande, `ls`. `ls` affiche une liste de fichiers et de répertoires, par défaut depuis votre répertoire courant. Par exemple :

```
/home/dupont$ ls
Mail
adresses
recettes
/home/dupont$
```

Nous voyons ici que Dupont a trois entrées dans son répertoire courant: `Mail`, `recettes`, et `adresses`. C'est assez imprécis; s'agit-il de fichiers ou de répertoires ? Nous pouvons utiliser l'option `-F` de la commande `ls` pour en savoir un peu plus.

```
/home/dupont$ ls --F
Mail/
adresses/
recettes/
/home/dupont$
```

Le caractère `/` rajouté à chaque nom nous montre que ce sont en fait trois sous-répertoires.

L'utilisation de `ls -F` peut aussi rajouter « `*` » à la fin d'un nom de fichier. Cela indiquera alors que ce fichier est un **exécutable**, ou un programme qui peut être exécuté. Si rien n'est ajouté à la fin du nom, il s'agit alors d'un « bon vieux fichier » , c'est à dire ni un répertoire, ni un exécutable.

En général, chaque commande UNIX peut prendre un certain nombre d'options en plus de ces arguments habituels. Ces options commencent traditionnellement par un « - » , comme nous venons de le voir plus haut avec `ls -F`. L'option `-F` demande à `ls` de donner plus d'informations sur le type de fichiers rencontrés, dans ce cas, en affichant un / après chaque nom de répertoire.

Si vous donnez un nom de répertoire à `ls`, il affichera le contenu de ce répertoire.

```
/home/dupont$ ls --F recettes
cassoulet
confit
soupe
vomitifs/
/home/dupont$
```

Pour voir une liste bien plus intéressante, regardons ce qui se trouve dans le répertoire `/etc` du système.

```
/home/dupont$ ls /etc

Images       ftpusers     lpc          rc.new       shells
adm          getty        magic        rc0.d        startcons
bcheckrc     gettydefs    motd         rc1.d        swapoff
brc          group        mount        rc2.d        swapon
brc~         inet         mtab         rc3.d        syslog.conf
csh.cshrc    init         mtools       rc4.d        syslog.pid
csh.login    init.d       pac          rc5.d        syslogd.reload
default      initrunlvl   passwd       rmt          termcap
disktab      inittab      printcap     rpc          umount
fdprm        inittab.old  profile      rpcinfo      update
fstab        issue        psdatabase   securetty    utmp
ftpaccess    lilo         rc           services     wtmp
/home/dupont$
```

(Pour les utilisateurs de MS-DOS qui seraient parmi nous, remarquez que le nom de certains fichiers, montrant qu'il peut être bien plus long que 8 caractères, et peut contenir autant de points que nécessaires à n'importe quel endroit.)

Maintenant, déplacons nous tout en haut de l'aborescence, en utilisant la commande « `cd ..` » , puis redescendons dans un autre sous répertoire: `/usr/bin`.

```
/home/dupont$ cd ..
/home$ cd ..
/$ cd usr
/usr$ cd bin
/usr/bin$
```

Vous pouvez aussi vous déplacer d'un coup, comme dans `cd /usr/bin`.

Essayez de vous promener dans différents répertoires, en utilisant `cd` puis `ls`. Dans certains cas, vous pourrez vous voir afficher le message frustrant « `Permission denied` » , signifiant qu'il vous est interdit d'aller par là. C'est simplement la notion de sécurité que possède UNIX : pour exécuter `ls` ou `cd` dans un répertoire, vous devez en avoir l'autorisation. Nous parlerons de cela en détail dans la section 3.9.

3.3.3 Création de nouveaux répertoires

Il est maintenant temps d'apprendre à créer des répertoires. Cela se fait à l'aide de la commande `mkdir`. Essayez par exemple:

```
/home/dupont$ mkdir essais
/home/dupont$ ls -F
Mail/
adresses/
essais/
recettes/
/home/dupont$ cd essais
/home/dupont/essais$ ls
/home/dupont/essais$
```

Félicitations ! Vous venez de créer un nouveau répertoire et vous déplacer dedans. Comme il n'y a aucun fichier dans ce répertoire, apprenons à copier des fichiers d'un endroit à un autre.

3.3.4 Copier des fichiers

La copie de fichiers se réalise avec la commande `cp` :

```
/home/dupont/essais$ cp /etc/termcap  .
/home/dupont/essais$ cp /etc/shells  .
/home/dupont/essais$ ls --F
shells    termcap
/home/dupont/essais$ cp shells bells
/home/dupont/essais$ ls --F
bells    shells    termcap
/home/dupont/essais$
```

La commande `cp` copie les fichiers précisés sur la ligne de commande vers le fichier ou le répertoire donné comme dernier argument. Notez comment nous pouvons utiliser le répertoire « `.` » pour référencer facilement le répertoire courant.

3.3.5 Déplacer des fichiers

Une nouvelle commande appelée `mv` (abbréviation de move) permet de déplacer des fichiers, plutôt que d'en effectuer une copie. La syntaxe est évidente.

```
/home/dupont/essais$ mv termcap sells
/home/dupont/essais$ ls -F
bells    sells    shells
/home/dupont/essais$
```

Notez que `termcap` n'existe plus, mais qu'à sa place il y a le fichier `sells`. Ceci peut servir à renommer des fichiers, comme nous venons justement de le faire, mais aussi pour déplacer un fichier vers un nouveau répertoire.

◇ **Note :** `mv` et `cp` écraseront le contenu du fichier destination (si il existe déjà) sans vous prévenir. Faites très attention lorsque vous déplacez un fichier vers un autre répertoire : il peut déjà contenir un fichier portant le même nom, que vous écraserez alors irrémédiablement !

3.3.6 Effacer des fichiers et des répertoires

Nous sommes maintenant rompus à l'utilisation de `ls`, et cette commande nous montre qu'il serait temps de faire un peu le ménage ici ou là après une longue session de travail. Pour effacer un fichier, vous utiliserez la commande `rm`. (« `rm` » symbolise le terme anglais « remove »).

```
/home/dupont/essais$ rm bells sells
/home/dupont/essais$ ls -F
shells
/home/dupont/essais$
```

Il ne nous reste plus que shells. Notez que par défaut `rm` ne vous demande rien avant de supprimer les fichiers, aussi faites très attention.

La commande `rm` ne supprime pas les répertoires, pour cela vous devrez utiliser `rmdir`. La commande `rmdir` supprime une entrée de répertoire, mais uniquement si ce répertoire est vide. S'il contient un fichier ou des sous-répertoires, `rmdir` vous signalera qu'il ne peut pas agir.

3.3.7 Voir le contenu des fichiers

Les commandes `more` et `cat` sont utilisées pour voir le contenu de certains fichiers. `more` affiche le fichier écran par écran, alors que `cat` le fait en une seule fois. Si le fichier à lire contient plus de lignes que votre écran, mieux vaut utiliser `more`.

Pour voir le contenu du fichier `shells`, nous pouvons utiliser la commande

```
/home/dupont/essais$ more shells
```

Au cas où vous seriez intéressé par le contenu de `shells`, il s'agit d'une liste des programmes shells valides sur votre système. Cela inclut souvent `/bin/sh`, `/bin/bash`, et `/bin/csh` au minimum. Nous verrons la différence entre ces différents shells un peu plus tard.

Lors de l'utilisation de `more`, tapez `Espace` pour afficher la page de texte suivante, et `b` pour revenir à la précédente. Il y a beaucoup d'autres commandes disponibles, celles-ci sont juste une base à connaître. La touche `q` vous permettra de sortir de `more`.

Quittez `more` et essayez `cat /etc/termcap`. Le texte va probablement défiler beaucoup trop vite pour avoir le temps de lire quoi que ce soit. Le nom « `cat` » signifie quelque chose comme « concaténation » , qui est le véritable but de ce programme. La commande `cat` peut être utilisée pour concaténer le contenu de plusieurs fichiers et sauver le résultat dans un autre. Nous en reparlerons plus tard.

3.3.8 L'aide en ligne

Pratiquement tout système UNIX complet, y compris Linux, offre une facilité appelée « pages de manuel » . Ces pages de manuel contiennent la documentation de toutes les différentes commandes système, ressources, fichiers de configuration, et ainsi de suite.

La commande utilisée pour accéder à ces pages de manuel se nomme `man`. Par exemple, si vous voulez connaître les autres options de la commande `ls`, vous pouvez taper

```
/home/dupont$ man ls
```

et la page de manuel de `ls` s'affichera sur l'écran.

Malheureusement, la plupart de ces pages sont écrites pour ceux qui ont déjà une notion de ce que font les commandes. Pour cette raison, les pages de manuel contiennent seulement les renseignements techniques pratiques, ne comptez pas trouver tout un cours d'initiation dans le manuel UNIX. Toutefois, le manuel peut être très pratique lorsque vous avez un trou de mémoire, pour vous rappeler la syntaxe d'une commande. Vous apprendrez par ce manuel beaucoup de choses à propos de commandes qui ne sont pas décrites dans ce livre.

Nous vous suggérons d'essayer `man` pour toutes les commandes que nous avons déjà vu, et chaque fois que nous en introduisons une nouvelle. Vous noterez que parfois, certaines commandes n'ont pas de page de manuel; ce peut être pour plusieurs raisons. La première et la plus courante, est que la page n'a pas encore été écrite (Le Projet de Documentation Linux s'occupe aussi de rédiger les pages de manuel, et nous accumulons petit à petit toute la documentation, mais nous ne sommes que des êtres humains...) La seconde raison, est que la commande peut être interne au shell, ou bien un alias (comme expliqué dans la section 3.2.4), auquel cas elle n'aura pas sa propre page de manuel. Un bon exemple est la commande `cd`, qui est une commande interprétée directement par le shell. Il n'y a pas de programme séparé qui réalise cette fonction, donc il n'y a pas de page de manuel pour `cd`.

3.4 Résumé des commandes de base

Cette section présente quelques unes des commandes les plus utiles sur un système UNIX, y compris celles décrites dans la section prédédente.

Notez bien que les options commencent généralement par un « `-` » , et que dans la plupart des cas il est possible de combiner de multiples options d'une lettre dans un seul « `-` » . Par exemple, au lieu d'utiliser la commande `ls -l -F`, il est plus pratique de taper `ls -lF`.

Au lieu de donner la liste de toutes les options disponibles pour chacune de ces commandes, nous ne parlerons que de celles qui sont utiles ou importantes, pour l'instant. En réalité, la plupart de ces commandes comprennent une très grande quantité d'options que vous n'utiliserez jamais. Vous pouvez utiliser `man` si vous voulez en savoir plus.

Notez également que beaucoup de ces commandes prennent une liste de fichiers ou de répertoires comme arguments, dénotés par « ⟨*fichier1*⟩ ...⟨*fichierN*⟩ » . Par exemple, la commande `cp` prend une liste de fichiers à copier, suivie de la destination, fichier ou répertoire. Lorsque l'on copie plus d'un fichier à la fois, la destination doit être un répertoire.

`cd` Change le répertoire de travail courant.
Syntaxe: `cd` ⟨*répertoire*⟩
⟨*répertoire*⟩ est le répertoire de destination. (« `.` » se réfère au répertoire courant, « `..` » au parent.)
Exemple : `cd ../toto` positionne le répertoire courant sur `../toto`.

`ls` Affiche des informations sur les fichiers ou répertoires.
Syntaxe: `ls` ⟨*fichier1*⟩ ⟨*fichier2*⟩ ...⟨*fichierN*⟩
Où ⟨*fichier1*⟩ à ⟨*fichierN*⟩ sont les noms de fichiers ou répertoires.
Options: Il y a vraiment beaucoup d'options. Les plus couramment utilisées sont `-F` (pour afficher des informations sur le type de fichier), et `-l` (donnant un format « long » comprenant la taille du fichier, le propriétaire, les droits d'accès, et ainsi de suite. Ces détails seront expliqués plus tard.)

Exemple : `ls -lF /usr/bin` affichera le contenu du répertoire `/usr/bin`.

`cp` Copie un (des) fichier(s) dans un autre fichier ou répertoire.
Syntaxe: `cp` ⟨*fichier1*⟩ ⟨*fichier2*⟩ …⟨*fichierN*⟩ ⟨*destination*⟩
Où ⟨*fichier1*⟩ à ⟨*fichierN*⟩ sont les fichiers à copier, et ⟨*destination*⟩ est le fichier de destination ou le répertoire.
Exemple : `cp ../frog joe` copie le fichier `../frog` dans le fichier ou le répertoire `joe`.

`mv` Déplace un (des) fichier(s) vers un autre fichier ou répertoire. Cette commande réalise l'équivalent d'une copie suivie de la suppression de l'original. Elle peut être utilisée pour renommer des fichiers, comme dans la comande MS-DOS `RENAME`.
Syntaxe: `mv` ⟨*fichier1*⟩ ⟨*fichier2*⟩ …⟨*fichierN*⟩ ⟨*destination*⟩
Où ⟨*fichier1*⟩ à ⟨*fichierN*⟩ sont les fichiers à déplacer, et ⟨*destination*⟩ est le fichier ou le répertoire de destination.
Exemple : `mv ../frog joe` déplace le fichier `../frog` dans le fichier ou le répertoire `joe`.

`rm` Supprime des fichiers. Notez bien que lorsque des fichiers sont supprimés sous UNIX, c'est irrémédiable, il est impossible de revenir en arrière.
Syntaxe: `rm` ⟨*fichier1*⟩ ⟨*fichier2*⟩ …⟨*fichierN*⟩
Où ⟨*fichier1*⟩ à ⟨*fichierN*⟩ sont les fichiers à supprimer.
Options: `-i` demandera confirmation avant chaque effacement de fichier.
Exemple : `rm -i recettes/cassoulet vomitifs/macdo` effacera les fichiers cassoulet et macdo si l'opération est bien confirmée par l'opérateur.

`mkdir` Crée de nouveaux répertoires.
Syntaxe: `mkdir` ⟨*dir1*⟩ ⟨*dir2*⟩ …⟨*dirN*⟩
Où ⟨*dir1*⟩ à ⟨*dirN*⟩ sont les répertoires à créer.
Exemple : `mkdir /tmp/test` créera le répertoire `test` dans le répertoire `/tmp`.

`rmdir` Cette commande supprime les répertoires vides. Lors de l'utilisation de `rmdir`, votre répertoire courant ne doit pas faire partie du répertoire à supprimer.
Syntaxe: `rmdir` ⟨*dir1*⟩ ⟨*dir2*⟩ …⟨*dirN*⟩
Où ⟨*dir1*⟩ à ⟨*dirN*⟩ sont les répertoires à supprimer.
Exemple : `rmdir /tmp/test` supprime le répertoire `test` dans `/tmp`, si il est vide (et s'il existe).

`man` Affiche la page de manuel pour la commande ou la ressource don-

née. (c'est à dire, n'importe quel utilitaire système qui n'est pas une commande, comme une fonction de bibliothèque). Syntaxe:
`man` ⟨*commande*⟩
Où ⟨*commande*⟩ est le nom de la commande ou ressource sur laquelle on désire de l'aide.
Exemple : `man ls` donne la description de la commande `ls`.

`more` Affiche le contenu des fichiers, un écran à la fois.
Syntaxe: `more` ⟨*fichier1*⟩ ⟨*fichier2*⟩ …⟨*fichierN*⟩
Où ⟨*fichier1*⟩ à ⟨*fichierN*⟩ sont les fichiers à examiner.
Exemple : `more /etc/termcap` affiche le fichier `/etc/termcap`.

`cat` Normalement destinée à concaténer des fichiers, la commande `cat` est aussi utilisée pour afficher tout le contenu d'un fichier d'un coup.
Syntaxe: `cat` ⟨*fichier1*⟩ ⟨*ficher2*⟩ …⟨*fichierN*⟩
Où ⟨*fichier1*⟩ à ⟨*fichierN*⟩ sont les fichiers concernés.
Exemple : `cat /etc/passwd` affiche le contenu de `/etc/passwd`.

`echo` Affiche simplement les arguments qu'on lui passe.
Syntaxe: `echo` ⟨*arg1*⟩ ⟨*arg2*⟩ …⟨*argN*⟩
Où ⟨*arg1*⟩ à ⟨*argN*⟩ sont les arguments à afficher.
Example: `echo "Bonjour tout le monde"` affichera la chaîne de caractères « `Bonjour tout le monde` » .

`grep` Affiche toutes les lignes dans le (les) fichier(s) correspondant à l'expression donnée.
Syntaxe: `grep` ⟨*expression*⟩ ⟨*fichier1*⟩ ⟨*fichier2*⟩ …⟨*fichierN*⟩
Où ⟨*expression*⟩ est une expression rationnelle, et ⟨*fichier1*⟩ à ⟨*fichierN*⟩ les fichiers dans lesquels la rechercher.
Exemple : `grep local /etc/hosts` affichera toutes les lignes du fichier `/etc/hosts` qui contiennent l'expression « `local` » .

3.5 Exploration du système de fichiers

Le **système de fichiers** est l'ensemble des fichiers et la hiérarchie de répertoires de votre système. Nous vous avons promis plus haut de vous accompagner dans une petite visite de cette arborescence, il est temps de se mettre en route.

Vous avez maintenant les connaissances de base pour comprendre ce dont nous allons parler, et vous avez un plan du domaine. (Cf figure 3.2.8 à la page 152).

Tout d'abord, allez dans le répertoire racine (`cd /`), et faites un `ls -F`. Vous verrez probablement les répertoires suivants[3] :

`bin`, `dev`, `etc`, `home`, `install`, `lib`, `mnt`, `proc`, `root`, `tmp`, `user`, `usr`, et `var`.

Jetons un œil à chacun d'eux.

`/bin` — `/bin` signifie « binaires » , ou exécutables. C'est l'endroit où se trouvent beaucoup de programmes système essentiels. Utilisez la commande « `ls -F /bin` » pour obtenir une liste de ces fichiers. Vous pourrez reconnaître certaines commandes que nous avons décrite plus haut: Lorsque vous utilisez la commande `cp`, vous exécutez le programme `/bin/cp`.

Avec `ls -F`, vous verrez que pratiquement tous ces fichiers possèdent une astérisque (« `*` »), rajoutée par `ls` à la fin de leur nom. C'est l'indication qu'ils sont exécutables, comme nous l'avons vu section 3.3.2.

`/dev` — Notre prochain arrêt est `/dev`. Observez, toujours avec `ls -F`.

Les « fichiers » contenus dans `/dev` sont connus comme **pilotes de périphériques** (device-drivers); ils sont utilisés pour accéder aux périphériques et ressources du système, comme les disques durs, modems, mémoire, souris, et ainsi de suite. Sous UNIX, vous pouvez lire de la même façon les données contenues dans un fichier, que par exemple celles fournies par la souris en accédant à `/dev/mouse`.

Les fichiers commençant par `fd` sont les pilotes des lecteurs de disquettes. `fd0` est le premier lecteur, `fd1` le second. Vous constatez qu'il y a bien plus de pilotes que de lecteurs de disquettes dans la machine: chacun d'eux représente un type particulier de disquette. Par exemple, `fd1H1440` permettra d'accéder aux disquettes haute-densité de 3.5" sur le second lecteur.

Voici une liste des pilotes les plus utilisés. Vous pouvez ne pas tous les posséder sur votre système, mais c'est peu probable.

- `/dev/console` correspond à la console système; l'écran qui est connecté directement à la machine.
- les différents `/dev/ttyS` et `/dev/cua` sont utilisés pour accéder aux ports série de l'ordinateur. Par exemple, `/dev/ttyS0` correpond en gros à « `COM1` » sous MS-DOS. Les pilotes `/dev/cua` correspondent aux mêmes ports séries, mais pour une utilisation « sortante » , comme un appel à l'aide d'un modem.

3. Vous en verrez surement d'autres, ou moins. Ne vous affolez pas. Chaque version de Linux peut être légèrement différente.

- Les pilotes dont le nom commence par `hd` permettent d'accéder aux disques durs. `/dev/hda` correspond à *tout* le premier disque dur, alors que `hda1` représente la première *partition* du disque `/dev/hda`.
- Les pilotes dont le nom commence par `sd` correspondent aux disques SCSI. Si vous avez un disque dur SCSI, vous ne pourrez pas y accéder par `/dev/hda`, mais uniquement par `/dev/sda`. Les bandes magnétiques SCSI sont accessibles par `st`, et les CD-ROM SCSI par les pilotes `sr`.
- Les pilotes dont le nom commence par `lp` correspondent aux ports imprimante parallèle. `/dev/lp0` représente en fait le « `LPT1` » du monde MS-DOS.
- `/dev/null` est utilisé comme un « trou noir » ; c'est un puits sans fond. Toute donnée envoyée dans ce pilote est absorbée sans autre forme de procès. A quoi cela sert-il ? Et bien, si vous voulez supprimer la sortie d'une commande qui affiche trop de choses à votre goût, vous pourrez envoyer cette sortie sur `/dev/null`. Nous reparlerons de ça un peu plus tard.
- Les pilotes dont le nom commence par `/dev/tty` correspondent aux « consoles virtuelles » de votre système. (accessibles par les touches `alt-F1` à `alt-F12`). `/dev/tty1` est la première `/dev/tty2` la seconde, et ainsi de suite.
- Les pilotes dont le nom commence par `/dev/pty` sont des « pseudo-terminaux » . Ils sont utilisés pour offrir un « terminal » aux sessions distantes réalisées par le réseau. Par exemple, si votre machine est connectée sur un réseau, les sessions ouvertes par `telnet` utiliseront chacune l'un des pilotes `/dev/pty`.

`/etc` — `/etc` contient presque tout ce qui pourrait être appelé « et cætera » ! Beaucoup de fichiers de configuration système, programmes et utilitaires. La plupart des programmes rencontrés dans `/etc` sont à l'usage exclusif de l'administrateur système. La tendance actuelle est à supprimer les fichiers exécutables de `/etc` pour les déplacer dans `/sbin`, et de ne conserver dans la mesure du possible que des fichiers de configuration dans `/etc`; nous verrons tout cela dans le chapitre 4.

`/sbin` — `sbin` est utilisé pour les binaires essentiels au système, nécessaires à l'administrateur.

`/home` — `/home` contient traditionnellement les répertoires personnels des utilisateurs. Par exemple, `/home/rene` est le répertoire personnel de René, dont le nom utilisateur est `rene`. Sur un système venant juste d'être installé, il peut ne pas y avoir grand monde dans ce répertoire.

`/lib` — `/lib` contient les **bibliothèques partagées**. Ces bibliothèques partagées contiennent le code que beaucoup de programmes partagent ensemble. Au lieu d'avoir dans chaque programme, une copie individuelle de ces routines, le code est stocké une seule fois au même endroit pour tout le monde, dans `/lib`. Les exécutables résultants sont beaucoup plus petits et la place disque économisée est assez conséquente.

`/proc` — `/proc` est un système de fichiers « virtuel » , les fichiers sont simulés en mémoire, pas sur le disque. [4] Ils correspondent aux différents **processus** présents sur le système, et permettent d'obtenir des informations de manière souple et élégante. Nous entrerons dans le détail de ce système de fichiers section 3.11.1.

`/tmp` — Énormément de programmes ont besoin de générer de temps à autres des fichiers temporaires. Ces fichiers sont toujours créés dans `/tmp`.

`/usr` — `/usr` est un répertoire très important. Il contient un certain nombre de sous-répertoires qui à leur tour, contiennent les programmes ou les fichiers de configuration les plus utiles du système.

Ces différents répertoires sont essentiels pour le fonctionnement du système, mais beaucoup de choses situées dans `/usr` sont optionnelles. Néanmoins, ce sont ces options là qui rendent le système utile et intéressant. Sans `/usr`, vous auriez un ordinateur capable juste d'effectuer des opérations ennuyeuses comme `cp` ou `ls`. `/usr` contient les applications les plus importantes, et leurs fichiers de configuration.

`/usr/X386` — `/usr/X386` contient X Window, si vous l'avez installé. X Window est un important et puissant environnement graphique qui offre une infinité d'utilitaires et programmes graphiques, affichés dans des « fenêtres » sur votre écran. Si vous êtes familier avec des environnements comme Microsoft Windows ou Macintosh, X Window vous semblera familier. (Tout en étant beaucoup plus puissant). Le répertoire `/usr/X386` contient tous les fichiers relatifs à cet environnement, qui sera décrit dans la section 5.1 page 187

`/usr/bin` — `/usr/bin` est vraiment le magasin à programmes sur tout système UNIX. Il contient la plupart des exécutables qui ne se trouvent

4. N'ayez pas peur de l'énorme fichier `/proc/kcore` que vous allez y trouver. Il symbolise toute la mémoire de votre machine, fera donc 32 Mo si vous avez 32Mo de RAM, mais ne prend aucune place nulle part.

pas ailleurs, comme `/bin`.

`/usr/etc` Tout comme `/etc` contient différents programmes et fichiers de configuration, `/usr/etc` contient encore plus de fichiers de ce type. En général, les fichiers trouvés dans `/usr/etc` ne sont pas essentiels au système, comme le sont ceux de `/etc`. La tendance actuelle est de déplacer les exécutables de ce répertoire pour les mettre dans `/usr/sbin`, ne conservant que ce qui est configuration pure dedans.

`/usr/include` `/usr/include` contient les « **fichiers include** » pour le compilateur C. Ces fichiers (dont la plupart se terminent par `.h`, pour « header ») déclarent les structures, noms, fonctions, constantes utilisées lors de l'écriture de programmes en langage C. Si vous êtes un familier de la programmation C, ce répertoire ne devrait pas vous surprendre.

`/usr/g++-include`
`/usr/g++-include` contient les fichiers de définition du compilateur C++, comme `/usr/include`).

`/usr/lib` `/usr/lib` contient les bibliothèques nécessaires à l'édition de liens des programmes que l'on réalise. Il y a généralement pour chaque, deux versions: l'une statique et l'autre partagée, qui est le pendant de ce que l'on trouve dans `/lib` pour l'exécution de ces programmes. De plus, divers autres programmes vont chercher leurs fichiers de configuration dans `/usr/lib`.

`/usr/local` `/usr/local` ressemble beaucoup au répertoire `/usr`: Il contient des programmes et fichiers divers qui ne sont pas indispensables, mais qui rendent le système encore meilleur. En général, ce que l'on met dans `/usr/local` est spéficique à chaque machine et dépend de l'administrateur et des besoins des utilisateurs; c'est pourquoi `/usr/local` est toujours différent d'une machine à une autre.

Vous pouvez y trouver par exemple de grosses applications comme TeX (avec lequel ce livre est réalisé), ou Emacs (un puissant éditeur de texte), aussi bien que de petits utilitaires en tous genres faits sur mesure.

`/usr/man` Ce répertoire contient les pages de manuel. Il y a deux sous répertoires pour chaque « section » de manuel (utilisez la commande `man man` pour avoir des détails sur le manuel). Par exemple, le répertoire `/usr/man/man1` contient les sources originales (les pages de manuel non formatées) de la section 1, et `/usr/man/cat1` les mêmes pages, une fois formatées.

`/usr/src` — `/usr/src` contient le code source (les programmes non compilés) de divers programmes composant votre système. Le plus important sous Linux est `/usr/src/linux`, qui contient tout le code source du noyau du système.

Vous ne trouverez sur aucun autre système UNIX, de telles sources, à moins d'être développeur dans la compagnie qui le programme, ou de payer quelques millions de francs [5].

`/var` — `/var` contient des répertoires qui sont sujets à des changements de taille imprévisibles, ou tentent à s'agrandir. Beaucoup d'entre eux se trouvaient à l'origine dans `/usr`, mais comme il est préférable que celui-ci soit assez fixe, les répertoires qui changent souvent ont été déplacés dans `/var`. Notez qu'ils sont toujours accessibles depuis `/usr` pour des raisons de compatibilité, grâce à des liens symboliques. En voici quelques-uns :

`/var/adm` — Il contient différents fichiers utiles à l'administrateur système, particulièrement les traces du système, qui enregistrent toute erreur ou problème pouvant se produire. D'autres fichiers tracent les accès à la machine, les tentatives de fraude, etc. Nous verrons cela dans le chapitre 4.

`/var/spool` — `/var/spool` contient des fichiers qui sont « spoulés » [6] pour d'autres programmes Par exemple, si votre machine est connectée à un réseau, le courrier arrivé sera mis en réserve dans `/usr/spool/mail`, jusqu'à ce qu'il soit lu. Les articles des *News Usenet* sont stockés pareillement dans `/usr/spool/news`, et ainsi de suite.

3.6 Les différents shells

Comme nous l'avons mentionné beaucoup trop de fois, UNIX est un système d'exploitation multi-tâches et multi-utilisateurs. Le multi-tâches est très utile, et une fois que vous en avez pris l'habitude, vous ne pourrez plus revenir en arrière sur un système qui ne sait utiliser qu'un programme unique à la fois. Dans peu de temps, vous serez capable de lancer des applications en « tâche de fond » , de suivre l'évolution de multiples programmes, d'enchaîner vos commandes en tous sens, de ne laisser aucun répit à votre microprocesseur...

Beaucoup des possibilités que nous décririons dans cette section sont fournies par le shell directement. Faites attention à ne pas confondre UNIX (le système d'exploitation) avec le shell; le shell est juste une interface avec le système qui est dessous.

5. Mais il existe bien entendu sur les rares autres UNIX gratuits disponibles, comme FreeBSD.
6. Je ne rigole pas. Il parait que c'est le terme à employer en français. C'était pourtant une belle langue...

Le shell fournit un grand nombre de fonctionnalités, offertes en réalité par UNIX.

Le shell n'est pas qu'un simple interpréteur de commandes [7]. C'est également un puissant langage de programmation, qui vous permet d'écrire des **shell scripts**, ou fichiers de commandes. Les utilisateurs de MS-DOS reconnaîtront une similarité avec leurs « fichiers batch » , en beaucoup plus puissant toutefois. L'utilisation de shell scripts forme un outil très puissant, qui vous permettra d'automatiser et étendre votre système UNIX de manière souple et facile. Voyez la section 3.13.1 à ce sujet.

Il y a différents types de shells dans le monde UNIX. Les deux types les plus importants sont le « Bourne shell » et le « C shell » . Le shell Bourne utilise une syntaxe similaire à celle du shell original des tout premiers systèmes UNIX, comme System III. Le nom du shell Bourne sur la plupart des systèmes UNIX est `/bin/sh` (où `sh` signifie « shell »). Le C shell utilise une syntaxe très différente, ressemblant beaucoup au langage C, et s'appelle en général `/bin/csh` sous UNIX.

Sous Linux, (et d'autres UNIX modernes), nous avons en plus plusieurs variantes de ces shells à notre disposition. Les deux les plus couramment utilisés sont le « Bourne Again Shell » , ou « Bash » (`/bin/bash`), et Tcsh (`/bin/tcsh`). Bash est un Bourne Shell incluant beaucoup de caractéristiques supplémentaires que l'on trouve dans le C shell. Comme Bash supporte un sur-ensemble de la syntaxe du Bourne shell, tous les scripts écrits pour le shell Bourne standard fontionneront avec. Pour ceux qui préfèrent utiliser la syntaxe du C shell, Linux propose Tcsh, qui est une version très améliorée du C shell original.

Le type de shell que vous déciderez d'utiliser est un choix vraiment personnel. Certains préfèrent le shell Bourne, d'autres le C shell, peu importe. Tant qu'il s'agit de lancer des commandes ordinaires, les deux fonctionneront de la même manière, sans distinction. La différence n'apparaît que lorsque vous commencez à écrire un shell-script ou vouloir utiliser les possibilités complexes qu'ils proposent.

Nous signalerons éventuellement chaque fois que ce sera utile les différences entre ces deux shells dans les explications qui vont suivre. Toutefois, pour ce que nous utiliserons dans ce guide, les différences sont minimes. (Si vous voulez vraiment entrer dans le détail, commencez par lire les pages de manuel système de `bash` et de `tcsh`).

3.7 Caractères génériques

Une fonctionnalité très puissante dans la plupart des shells UNIX est la possibilité de référencer plus d'un nom de fichier à l'aide de caractères spéciaux. Ces **caractères génériques** vous permettent de désigner tous les fichiers dont, par exemple,

7. et voila pourquoi, Monsieur le Ministre de Notre Culture, je ne peux vraiment pas utiliser « interpréteur de commandes » à la place du mot « shell » . Et puis en français on dit « interprète » il me semble.

le nom contient la lettre « **n** » .

Le caractère « ***** » remplace n'importe quel caractère ou chaîne de caractères dans un nom de fichier. Par exemple, lorsque vous utilisez « ***** » dans un nom de fichier, le shell le remplace par toutes les substitutions possibles à partir des fichiers du répertoire que vous désignez.

Voici un petit exemple. Supposons que Marcel possède les fichiers `cognac`, `rhum` et `litron` dans son répertoire personnel.

```
/home/marcel$ ls
cognac     litron     rhum
/home/marcel$
```

Pour accéder à tous les fichiers dont le nom contient la lettre « o » , nous pouvons utiliser la commande

```
/home/marcel$ ls *o*
cognac     litron
/home/marcel$
```

Comme vous le voyez, le caractère « ***** » a été remplacé par toutes les substitutions possibles à partir des fichiers contenus dans le répertoire.

L'usage de « ***** » tout seul correspond simplement à tous les fichiers, puisque tous les caractères lui correspondent.

```
/home/marcel$ ls *
cognac     litron     rhum
/home/marcel$
```

Voici encore quelques exemples.

```
/home/marcel$ ls r*
rhum
/home/marcel$ ls *on
litron
/home/marcel$ ls *r*
litron     rhum
/home/marcel$ ls c*c
cognac
/home/marcel$
```

L'action de changer un caractère générique en noms de fichier est appelée **expansion**, et est réalisée par le shell. Ceci est important : les commances individuelles, comme `ls`, ne voient *jamais* le caractère « ***** » dans leur liste de paramètres. Le shell expanse l'expression pour inclure tous les noms de fichiers possibles avant d'appeler le programme. En fait, la commande

```
/home/marcel$ ls *o*
```

este expansée par le shell et devient en réalité

```
/home/marcel$ ls cognac litron
```

Une note importante à propos du caractère « `*` » . L'utilisation de ce caractère générique ne prendra pas en compte les noms de fichiers qui commencent par un simple point (« `.` »). C'est une convention ; ces fichiers sont traités comme fichiers "cachés"; ils ne sont pas vraiment cachés, mais par défaut ils ne se voient pas, et ne sont pas concernés par le caractère « `*` » .

Voici un exemple. Nous avons déja expliqué que chaque répertoire possède deux entrées spéciales: « `.` » qui correspond au répertoire courant, et « `..` » au parent. Lorsque vous utilisez `ls`, ces deux entrées ne sont pas affichées.

```
/home/marcel$ ls
cognac    litron    rhum
/home/marcel$
```

Si vous utilisez l'option `-a` de la commande `ls`, alors vous pouvez voir tous les fichiers qui commencent par un point. Observez :

```
/home/marcel$ ls -a
.    ..    .bash_profile    .bashrc    cognac    litron    rhum
/home/marcel$
```

Là, nous voyons les deux entrées spéciales, « `.` » et « `..` » , ainsi que deux autres fichiers « cachés » : `.bash_profile` et `.bashrc`. Ces deux fichiers sont des fichiers de démarrage utilisés par `bash`. Ils sont expliqués dans la section 3.13.3.

Notez que lorsque nous utilisons le caractère « `*` » , aucun de ces fichiers commençant par « `.` » n'est affiché.

```
/home/marcel$ ls *
cognac    litron    rhum
/home/marcel$
```

C'est en partie une question de sécurité: Si « `*` » prenait en compte les fichiers començant par « `.` » , il prendrait aussi en compte les deux entrées spéciales « `.` » et « `..` » , ce qui peut être très dangereux lors de l'utilisation de certaines commandes.

Un autre caractère générique : « `?` » . Le caractère « `?` » ne représente qu'un seul caractère. Par conséquent, « `ls?` » n'affichera que les noms de fichiers d'une seule lettre, et « `ls termca?` » pourra afficher « `termcap` » mais jamais « `termcap.backup` » . Voici un autre exemple:

```
/home/marcel$ ls rhu?
rhum
/home/marcel$ ls li??on
litron
/home/marcel$ ls ?????c
cognac
/home/marcel$
```

Comme vous voyez, les caractères génériques permettent de spécifier de multiples fichiers en une seule fois. Dans le petit résumé des commandes de la page 109, nous avons dit que les commandes `cp` et `mv` pouvaient copier ou déplacer plusieurs fichiers en une seule fois. Par exemple,

```
/home/marcel$ cp /etc/s* /home/marcel
```

copiera tous les fichiers de `/etc` dont le nom commence par la lettre « `s` » dans le répertoire `/home/marcel`. Par conséquent, le format de la commande `cp` est en réalité

cp ⟨*fichier1*⟩ ⟨*fichier2*⟩ ⟨*fichier3*⟩ ... ⟨*fichierN*⟩ ⟨*destination*⟩

où ⟨*fichier1*⟩ à ⟨*fichierN*⟩ est une liste de fichiers à copier, et ⟨*destination*⟩ est le fichier ou répertoire de destination. La commande `mv` possède une syntaxe identique.

Notez bien que si vous copiez ou déplacez plus d'un fichier, le paramètre ⟨*destination*⟩ doit être un répertoire. Vous ne pouvez copier ou déplacer dans un fichier qu'un seul autre fichier.

3.8 Entrées et sorties sous UNIX

3.8.1 Entrée standard et sortie standard

Beaucoup de commandes d'UNIX prennent leurs entrées sur ce qui est connu comme l'**entrée standard** et envoient leur affichage sur la **sortie standard**, très souvent abrégées par « stdin » (pour l'entrée) et « stdout » pour la sortie. Votre shell paramètre les choses de telle manière que l'entrée standard soit votre clavier, et la sortie standard votre écran.

Voici un exemple utilisant la commande `cat`. Normalement, `cat` lit les données de tous les fichiers spécifiés sur sa ligne de commande et envoie ce contenu directement sur la sortie standard. Par conséquent, la commande

```
/home/rene/notes$ cat memo courses
```

affichera le contenu du fichier `memo` suivi de celui du fichier `courses`.

Toutefois, si aucun fichier n'est précisé à `cat`, il lira les données sur l'entrée standard (stdin), et bien sûr les renverra sur la sortie standard (stdout). Voici un exemple:

```
/home/rene/notes$ cat
Bonjour tout le monde.
Bonjour tout le monde.
Allez, salut.
Allez, salut.
ctrl-D
/home/rene/notes$
```

Comme vous pouvez le voir, chaque ligne que l'utilisateur tape (imprimée en italiques) est immédiatement affichée par la commande `cat` sur l'écran. Lors de la lecture depuis l'entrée standard, les commandes savent qu'il n'y a plus de données lorsqu'elles recoivent un signal EOT (end-of-text, fin du texte). En général, ceci est généré en pressant `ctrl-D`.

Voici un autre exemple. La commande `sort` lit des lignes de texte (toujours, depuis stdin, sauf si des fichiers sont précisés en ligne de commande), et envoie le résultat trié sur la sortie standard. Essayez ce qui suit.

```
/home/rene/notes$ sort
bordeaux
cahors
bergerac
ctrl-D
bergerac
bordeaux
cahors
/home/rene/notes$
```

Nous pouvons trier la liste des courses par ordre alphabétique... UNIX est fantastique !

3.8.2 Rediriger les entrées et les sorties

Maintenant, admettons que nous voulions envoyer la sortie de `sort` dans un fichier, pour sauver notre liste de courses quelque part. Le shell nous permet de **rediriger** la sortie standard dans un fichier, à l'aide du symbole « `>` » . Voici comment ça marche.

```
/home/rene/notes$ sort > liste.courses
gauloises
bordeaux
cahors
ctrl-D
/home/rene/notes$
```

Vous voyez que le résultat de la commande `sort` ne s'est pas affiché, au lieu de cela il est enregistré dans le fichier `liste.courses`. Jetons un œil à ce fichier.

```
/home/rene/notes$ cat liste.courses
bordeaux
cahors
gauloises
/home/rene/notes$
```

Non seulement nous pouvons trier la liste des courses, mais en plus, la conserver ! Mais supposons que nous notions notre liste originale, dans le désordre, dans le fichier `bricoles` par exemple. Une méthode pour trier cette liste serait de donner à `sort` le nom du fichier à lire, au lieu de l'entrée standard, et rediriger la sortie standard comme nous l'avons fait ci-dessus. Comme ceci:

```
/home/rene/notes$ sort bricoles > liste.courses
/home/rene/notes$ cat liste.courses
bordeaux
cahors
gauloises
/home/rene/notes$
```

Toutefois il y a une autre méthode pour réaliser cela. Car non seulement nous pouvons rediriger la sortie standard, mais c'est aussi possible pour l'*entrée* standard, en utilisant le symbole « `<` » .

```
/home/rene/notes$ sort < bricoles
bordeaux
cahors
gauloises
/home/rene/notes$
```

Techniquement, `sort <bricoles` est équivalent à `sort bricoles`, mais la première forme nous permet de démontrer ceci : `sort <bricoles` se comporte comme

si les données contenues dans le fichier `bricoles` étaient tapées sur l'entrée standard. Le shell prend en charge la redirection. La commande `sort` n'a jamais vu ni connu le nom du fichier `bricoles` à lire, pour elle, elle était encore en train de lire l'entrée standard comme si c'est vous qui tapiez les données sur votre clavier.

Ceci nous permet d'introduire le concept de **filtre**. Un filtre est un programme qui lit des données sur l'entrée standard, fait quelque chose avec, et envoie le résultat sur la sortie standard. En utilisant les redirections, les entrées/sorties standard peuvent être référencées depuis des fichiers. `sort` est un filtre simple: il trie les données qui arrivent et envoie le résultat sur la sortie standard. `cat` est encore plus simple: il ne fait rien du tout des données qu'il reçoit, il ressort tel quel tout ce qui lui est fourni.

3.8.3 Faites des pipes !

Nous avons vu comment utiliser `sort` en tant que filtre. Cependant ces exemples considéraient que vous aviez des données situées dans un fichier ou que vous les tapiez vous même sur l'entrée standard. Comment faire si les données que vous voulez trier proviennent de la sortie d'une autre commande, comme `ls`? Par exemple, vous voudriez pouvoir trier par ordre alphabétique inverse, grâce à l'option `-r` de la commande `sort`, la liste des fichiers présents dans votre répertoire courant. Une solution pour cela serait:

```
/home/rene/notes$ ls
bricoles
liste.courses
problemes
telephones
/home/rene/notes$ ls > liste.fichiers
/home/rene/notes$ sort -r liste.fichiers
telephones
problemes
liste.courses
bricoles
/home/rene/notes$
```

Ici, nous avons sauvé la sortie de `ls` dans un fichier, puis utilisé `sort -r` sur ce fichier. Mais ce n'est pas très pratique et nous oblige à utiliser un fichier temporaire pour récupérer la sortie de `ls`.

La solution est d'utiliser un **pipe**.[8] Les pipes sont une autre caractéristique offerte par le shell, qui permettent de connecter la sortie d'une commande à un « tuyau »

8. Certains traduisent vraiment **pipe** par **tuyau**, ou **tube**. Il est vrai que c'en est la signification exacte et qu'il est employé dans ce sens; néanmoins le mot **pipe** étant universellement utilisé, nous conserverons l'usage de ce mot dans ce guide, afin qu'il reste compréhensible par le commun des mortels. Et puis ça permet des jeux de mots encore plus drôles...

, et l'entrée d'une autre à l'autre bout de ce tuyau. Le résultat est que l'entrée standard de la seconde commande est directement connectée à la sortie de la première, et ainsi de suite. Ici, nous voulons connecter la sortie standard de `ls` à l'entrée standard de `sort`. On utilise le symbole « `|` » pour créer un pipe :

```
/home/rene/notes$ ls | sort -r
telephones
problemes
liste.courses
bricoles
/home/rene/notes$
```

Cette commande est bien plus courte, et beaucoup plus facile à taper.

Un autre exemple très utile, une commande comme

```
/home/rene/notes$ ls /usr/bin
```

va afficher une très longue liste, qui défilera bien trop vite sur l'écran pour avoir le temps de lire quoi que ce soit. A la place, utilisons donc `more` pour afficher écran par écran les fichiers présents dans `/usr/bin`.

```
/home/rene/notes$ ls /usr/bin | more
```

Nous pouvons maintenant lire cette liste tranquillement.

Mais on peut faire encore mieux ! Nous pouvons enchaîner plus de deux commandes avec des pipes. La commande `head` est un filtre qui affiche les premières lignes d'un flux (ici, l'entrée standard depuis un pipe). Si nous voulions afficher le dernier fichier par ordre alphabétique dans le répertoire courant, nous utiliserions :

```
/home/rene/notes$ ls | sort -r | head -1
telephones
/home/rene/notes$
```

où `head -1` affiche simplement la première ligne qu'il reçoit en entrée (dans ce cas, la sortie triée par ordre alphabétique inverse de la commande `ls`).

3.8.4 Redirection non destructive

L'utilisation de « `>` » pour rediriger la sortie vers un fichier est destructive: la commande

```
/home/rene/notes$ ls > liste.fichiers
```

écrase à chaque fois le contenu de **liste.fichiers**. Si vous utilisez le symbole « >> » pour effectuer la redirection, la sortie sera rajoutée au contenu déja existant du fichier.

```
/home/rene/notes$ ls >> liste.fichiers
```

rajoutera la sortie de la commande **ls** à la fin du fichier **liste.fichiers**.

N'oubliez jamais que les pipes et redirections sont des possibilités offertes par le shell. C'est le shell qui gère cette syntaxe à base de « > » ou « >> » et « | » . Cela n'a rien à voir avec les commandes elles-mêmes, c'est le shell qui s'occupe de tout.

La lecture d'un ouvrage plus général s'impose si vous désirez approfondir ce sujet. Il serait trop long d'introduire ici l'opérateur « << » et surtout la notion de « sortie standard d'erreurs » , ainsi que les différences entre les différents shells qui commencent alors à entrer dans le jeu.

3.9 Droits d'accès aux fichiers

3.9.1 Notions de permissions d'accès

Du fait qu'UNIX puisse supporter de nombeux utilisateurs, le système contient un mécanisme destiné à protéger les fichiers de la curiosité de certains. Vous n'aimeriez sans doute pas que n'importe qui puisse lire votre courrier, n'est-ce pas ? Ce mécanisme, appelé **permissions d'accès**, permet aux utilisateurs d'être « propriétaires » de certains fichiers ou répertoires. Par exemple, Marcel est propriétaire de tous les fichiers qu'il a créé dans son répertoire personnel, et peut en faire ce qu'il veut.

UNIX permet également de partager des fichiers entre utilisateurs et groupes d'utilisateurs. Si Marcel le désirait, il pourrait interdire tout accès à ses fichiers, de telle manière que personne d'autre ne puisse y accéder. Néanmoins sur la plupart des systèmes, par défaut les autres utilisateurs peuvent lire vos fichiers, mais ne peuvent pas les modifier ou les effacer. C'est à vous de les protéger davantage si vous en ressentez le besoin.

Comme nous l'avons dit, chaque fichier appartient à un utilisateur particulier. Mais les fichiers appartiennent également à un **groupe**, qui est un groupe d'utilisateurs défini dans la configuration du système. Chaque utilisateur fait partie d'au moins un groupe lorsque son compte est créé. L'administrateur système peut également placer cet utilisateur dans plusieurs groupes à la fois.

Les groupes sont généralement définis par le type d'utilisateurs qui ont accès à la machine. Par exemple, sur un système UNIX d'une université, les utilisateurs pourraient être placés dans des groupes comme **etudiant**, **professeur**, **faculte** ou **invite**. Il existe aussi quelques groupes définis par le système pour contrôler l'accès aux ressources (comme `bin` et `admin`), mais les utilisateurs appartiennent très rarement à ces groupes.

Les droits d'accès se divisent en trois catégories principales: lecture, écriture et exécution. Ces permisions peuvent être accordées à trois classes d'utilisateurs: le propriétaire du fichier, le groupe d'utilisateurs auquel ce fichier appartient, et tous les autres utilisateurs, quel que soit le groupe.

L'autorisation de lecture permet de lire le contenu du fichier, ou bien dans le cas de répertoires, de lire le contenu de ce répertoire (avec `ls`). Pour les répertoires, le droit d'écriture permet à l'utilisateur de créer de nouveaux fichiers ou d'en supprimer, à l'intérieur de ce répertoire. Finalement, la permission d'exécution autorise l'utilisateur à exécuter le fichier en tant que programme ou shell-script. (Si le fichier est un programme ou un shell-script bien sûr). Dans le cas des répertoires, posséder le droit d'exécution autorise par convention l'utilisateur à traverser le répertoire en question (commande `cd`).

3.9.2 Interpréter les permissions d'accès

Voyons une démonstration pratique. La commande `ls`, utilisée avec son option `-l` affiche un résultat sous un format « long » , montrant les droits d'accès.

```
/home/rene/notes$ ls -l liste.courses

-rw-r--r--   1 rene    users         505 Mar 13 19:05 liste.courses

/home/rene/notes$
```

Le premier champ affiché représente les différentes permissions d'accès au fichier. Le troisième indique le propriétaire (`rene`), et le quatrième indique le groupe auquel ce fichier appartient (`users`). Le dernier champ est bien évidemment le nom du fichier (`liste.courses`), et nous verrons les autres plus tard.

Ce fichier appartient à `rene`, ainsi qu'au groupe `users`. Regardons les permissions d'accès. La chaîne `-rw-r--r--` donne dans l'ordre, les droits qu'ont respectivement le propriétaire, le groupe, et tous les autres sur ce fichier.

Le premier caractère (« `-` ») représente le type du fichier. Un « `-` » signifie simplement qu'il s'agit d'un fichier ordinaire (et non pas un répertoire ou un pilote de périphérique par exemple). Les trois lettres suivantes (« `rw-` ») représentent les droits que possède le propriétaire de ce fichier (`rene`), concernant sa lecture, son écriture ou son exécution. Le « `r` » symbolise « read » (lecture), et le « `w` » veut dire « write » (écriture). Par conséquent nous voyons que `rene` possède les permissions de lecture et d'écriture sur le fichier `liste.courses`.

Comme nous l'avons mentionné, en dehors des permissions de lecture et d'écriture, il y a également l'autorisation d'« exécution » , représentée par la lettre « `x` » , donc apparamment René n'a pas la permission d'exécuter ce fichier. C'est tout à fait logique, le fichier `liste.courses` n'est pas un programme exécutable. Bien sûr puisqu'il en est propriétaire, René pourrait positionner la permission d'exécution sur ce fichier s'il le désirait. Dans ce cas précis, ça n'aurait pas grand intérêt. Mais nous verrons bientôt que ce peut être très utile.

Les trois caractères suivants, `r--`, représentent les permissions pour le groupe auquel appartient ce fichier, qui est ici le groupe `users`. Comme on ne trouve que « `r` » , tout utilisateur appartenant à ce groupe a le droit de lire ce fichier, et c'est tout ce qu'il peut faire.

Les trois derniers caractères, qui sont ici aussi `r--`, représentent les permissions accordées à tous les autres utilisateurs du système (autres que le propriétaire et ceux du groupe `users`). Encore une fois, comme il n'y a juste qu'un « `r` » , les autres utilisateurs ont uniquement le droit de lire le fichier.

Voici quelques autres exemples de droits d'accès.

`-rwxr-xr-x` Le propriétaire du fichier peut le lire, l'écire et l'exécuter. Les utilisateurs qui sont dans le groupe du fichier, et tous les autres aussi, peuvent lire et exécuter le fichier. (mais pas le modifier).

`-rw-------` Le propriétaire du fichier peut le lire et l'écrire. Personne d'autre ne peut accéder à ce fichier.

`-rwxrwxrwx` Tout le monde peut écrire, lire, et exécuter le fichier.

3.9.3 Dépendances

Il est important de noter que les permissions d'accès à un fichier dépendent aussi de celles du répertoire dans lequel il se trouve. Par exemple, même si un fichier a les permissions `-rwxrwxrwx`, les autres utilisateurs ne peuvent accéder à ce fichier que si ils ont les droits de lecture et d'exécution sur le répertoire dans lequel le fichier se trouve. Par exemple, si René voulait restreindre l'accès à tous ses fichiers, il pourrait simplement positionner les permissions de son répertoire personnel `/home/rene` sur `-rwx------`. De cette manière, aucun autre utilisateur ne pourrait accéder à son répertoire, et par conséquent aux fichiers qu'il contient. René n'aurait alors plus besoin de se soucier des permissions particulières de chaque fichier.

En d'autres termes, pour avoir accès à un fichier, vous devez avoir la permission d'exécution sur tous les répertoires rencontrés sur son chemin d'accès, et au moins le droit de lecture (ou d'exécution) de ce fichier.

D'habitude, les utilisateurs d'un système UNIX ne font pas mystère de leurs fichiers. Les permissions par défaut des fichiers sont souvent `-rw-r--r--`, ce qui autorise tout le monde à prendre connaissance du fichier, sans pouvoir le modifier par inadvertance. Les répertoires possèdent en général les droits `-rwxr-xr-x`, ce qui autorise tout le monde à les traverser, sans pouvoir toutefois créer ou effacer de fichiers dedans.

Cependant, certains utilisateurs désirent garder secrètes certaines données personnelles, ce qui est bien normal. Mettre les droits `-rw-------` sur un fichier le rendra inaccessible au reste du monde. De la même façon, un répertoire ayant les permissions `-rwx------` empêchera tout les utilisateurs autres que le propriétaire d'aller y faire un tour.

3.9.4 Changer les permissions

La commande `chmod` est utilisée pour positionner les permissions d'un fichier ou répertoire. Seul le propriétaire d'un fichier a le droit d'en modifier les droits d'accès. La syntaxe de `chmod` est:

`chmod {a,u,g,o}{+,-}{r,w,x}` ⟨*fichiers*⟩

En bref, vous spécifiez un ou plus de **a**ll (tous), **u**ser (propriétaire), **g**roup (groupe) , ou **o**ther (les autres). Puis vous précisez si vous ajoutez des droits (`+`) ou en supprimez (`-`). Finalement, vous précisez l'un ou plus de **r**ead (lecture) , **w**rite (écriture) , et e**x**ecute (exécution). Quelques exemples seront plus clairs :

`chmod a+r fichier`
: Donne l'accès en lecture à tous les utilisateurs.

`chmod +r fichier`
: Comme ci-dessus, si aucun des `a`, `u`, `g`, or `o` n'est spécifié, `a` est sous-entendu.

`chmod og-x fichier`
: Supprime la permission d'exécution aux utilisateurs qui ne sont pas propriétaires du fichier.

`chmod u+rwx fichier`
: Autorise le propriétaire du fichier à le lire, l'écrire ou l'exécuter.

`chmod o-rwx fichier`
: Retire les permissions de lecture, écriture et exécution aux utilisateurs autres que le propriétaire et les utilisateurs étant dans le même groupe que le fichier.

3.10 Les liens sur les fichiers

Sous UNIX, un même fichier peut avoir plusieurs noms, grâce à la magie des liens. Les fichiers sont physiquement identifiés par le système par leur **numéro d'i-nœud**, qui est l'unique identificateur de ce fichier [9]. Un répertoire n'est en fait qu'une liste d'i-nœuds avec leur nom de fichiers correspondants. Chaque nom de fichier dans un répertoire, est un **lien** vers un i-nœud particulier, et il est tout à fait concevable que plusieurs noms différents pointent sur le même i-nœud.

3.10.1 Liens physiques

Il s'agit comme nous venons de le dire, d'associer le même i-nœud à plusieurs noms de fichiers différents. La commande `ln` est destinée à cette opération; prenons donc exemple. Supposons que vous avez un fichier nommé `toto` dans un répertoire. La commande `ls -i` nous montrerait son numéro d'i-nœud :

```
$ ls -i toto
22192 toto
$
```

Ici, le fichier `toto` correspond à l'i-nœud 22192 du système de fichiers. Nous pouvons utiliser `ln` pour créer un autre lien, que nous appellerons par exemple `titi` :

```
$ ln toto titi
```

La commande `ls -i` nous montre alors que nous avons maintenant deux fichiers, qui correspondent au même i-nœud :

```
$ ls -i toto titi
22192 toto   22192 titi
$
```

Maintenant, accéder au fichier `toto` ou au fichier `titi` donnera le même résultat, ce sont les mêmes données, le même fichier physique sur le disque. Si vous faites des modifications sur le fichier `titi`, elles seront immédiatement répercutées sur le fichier `toto`, puisque c'est le même.

Ces liens sont connus comme *liens physiques*, ou *hard links*, parce qu'ils relient directement les noms de fichiers aux i-nœuds. Notez que vous ne pouvez réaliser cette opération que sur un même système de fichiers; vous ne pouvez pas créer de tels liens entre deux partitions d'un disque dur par exemple. Les liens symboliques (que nous allons voir plus loin) n'ont pas cette limitation.

9. La commande `ls -i` vous montrera les numéros d'i-nœuds de chaque fichier.

Lorsque vous effacez un fichier par `rm`, vous ne supprimez que le lien correspondant au nom de ce fichier. Si vous faites :

```
$ rm toto
```

seul le lien nommé `toto` est supprimé, le fichier `titi` existe toujours. Un fichier n'est totalement supprimé du disque que lorsqu'il n'a plus aucun lien. En général, les fichiers ne comportent qu'un seul nom, par conséquent la commande `rm` efface physiquement le fichier. Mais si il comporte plusieurs liens, `rm` n'en supprimera qu'un; pour supprimer totalement le fichier du disque vous devrez utiliser `rm` sur chaque nom que peut avoir ce fichier.

La commande `ls -l` affiche le nombre de liens que possède un fichier (entre autres):

```
$ ls -l titi toto
-rw-r--r--   2 root     root          12 Aug  5 16:51 titi
-rw-r--r--   2 root     root          12 Aug  5 16:50 toto
$
```

La seconde colonne, qui contient « `2` » , nous indique le nombre de liens sur ce fichier.

Un répertoire n'est jamais qu'un fichier contenant des informations relatives aux noms de fichiers et leur i-nœud correspondant. Du coup, chaque répertoire possède au moins deux liens physiques: « `.` » (un lien pointant sur lui-même), et « `..` » (un lien pointant sur son répertoire parent). Le répertoire racine (`/`) est un cas particulier, il n'y a rien au dessus de lui et son lien « `..` » pointe aussi sur (`/`).

3.10.2 Les liens symboliques

C'est une autre sorte de lien, qui fonctionne d'une toute autre manière que ce que nous venons de voir. Un lien symbolique permet bien de donner un nom supplémentaire à un fichier, mais n'utilise pas l'i-nœud physique du fichier. Il est simplement interprété par le système, qui va suivre l'indication contenue pour aller chercher un autre fichier, c'est un peu comme un panneau indiquant une direction à suivre.

La commande `ln -s` créera un lien symbolique vers un fichier. Par exemple, si nous utilisons la commande

```
$ ln -s toto titi
```

nous créerons le lien symbolique `titi` qui pointera sur le fichier `toto`. Si nous tapons `ls -i`, nous verrons que les deux fichiers ont des i-nœuds différents:

```
$ ls -i toto titi
```

```
22195 toto   22192 titi
#
```

Mais la commande `ls -l` nous indiquera clairement que le fichier `titi` est un lien symbolique pointant vers `toto` :

```
$ ls -l toto titi
lrwxrwxrwx   1 root     root          3 Aug  5 16:51 titi -> toto
-rw-r--r--   1 root     root         12 Aug  5 16:50 toto
$
```

Les permissions sur un lien symbolique ne sont pas utilisées (elles apparaissent toujours comme `rwxrwxrwx`). Ce sont celles du fichier sur lequel il pointe qui déterminent les droits d'accès (notre fichier toto, dans cet exemple).

Les liens physiques et les liens symboliques sont très similaires dans leur fonctionnement, mais il y a quelques différences. En premier lieu, il est possible de créer des liens symboliques sur des fichiers qui n'existent pas; ce qui est irréalisable avec les liens physiques. Les liens symboliques sont interprétés différemment par le noyau, ce n'est qu'une différence technique mais qui quelquefois peut avoir son importance. Les liens symboliques sont très utiles car ils permettent de traverser la barrière des différents systèmes de fichiers; et enfin ils identifient clairement le fichier sur lequel ils pointent, qu'il est facile de retrouver; alors qu'identifier quels fichiers sont liés au même i-nœud est bien moins évident.

Les systèmes UNIX font un abondant usage des liens, Linux n'échappe pas à la règle. Les liens symboliques sont particulièrement importants pour les bibliothèques partagées dans le répertoire `/lib`. Voyez la section 4.7.2 à ce propos.

3.11 Contrôle de processus

3.11.1 Les processus

Le contrôle de processus est une possibilité offerte par beaucoup de shells (Bash et Tcsh inclus) qui permet de contrôler plusieurs exécutions de commandes depuis un même terminal. Avant que nous ne développions plus avant, il nous faut parler des **processus**.

Chaque fois que vous lancez un programme, vous démarrez ce que l'on appelle un **processus**; c'est juste un nom pratique pour désigner un programme un cours d'exécution. La commande `ps` affiche une liste des processus en cours sur la machine. Voici un exemple :

```
/home/marcel$ ps
```

```
  PID TT STAT  TIME COMMAND
   24  3 S     0:03 (bash)
  161  3 R     0:00 ps

/home/marcel$
```

Le `PID` donné dans la première colonne est l' **identificateur du processus** (process ID), un nombre unique donné à chaque processus tournant sur le système. La dernière colonne, `COMMAND`, est le nom de la commande correspondante. Ici, seuls sont représentés les processus appartenant à Marcel[10]. Ce sont `bash` (son shell) et la commande `ps` elle-même. Comme vous le voyez, `bash` fonctionne en même temps que `ps`. Le shell `bash` a exécuté le programme `ps` lorsque Marcel a tapé la commande. Après que `ps` se soit terminé (lorsque la table des processus a été affichée), le contrôle est repassé à `bash`, qui affiche l'invite, prêt à recevoir une nouvelle commande.

Vous entendrez souvent les termes *job* et *contrôle de jobs* à ce propos. Les mots *processus* et *job* sont interchangeables, mis à part que le second n'est pas français. Toutefois, un processus est généralement appelé *job* dans le contexte du *contrôle de job*, même lorsque l'on appelle ça le *contrôle de processus*. À vous de voir, l'essentiel est de se faire comprendre.

Très souvent les utilisateurs n'exécutent qu'un job à la fois : la commande qu'ils ont entrée en dernier. Mais en utilisant le contrôle de processus, vous pouvez exécuter plusieurs jobs à la fois, en les commutant comme vous le désirez. À quoi cela peut-il servir ? Et bien, supposons que vous soyez en train d'éditer un texte et que vous ayez soudainement besoin d'interrompre ce travail et de faire autre chose. Avec le contrôle de processus, vous pouvez suspendre temporairement l'exécution de l'éditeur, et une fois retourné à l'invite du shell, commencer à travailler sur autre chose. Une fois ce travail terminé, vous pouvez relancer l'éditeur et tout retrouver dans l'état où vous l'avez laissé, comme si vous n'aviez jamais quitté le programme. Ce n'est qu'un exemple (pas très bon puisque les consoles virtuelles de Linux vous auraient permis quelque chose de presque équivalent), le contrôle de processus peut être très utile dans beaucoup de circonstances.

3.11.2 Avant-plan et arrière-plan

Les processus peuvent être soit en **avant-plan**, soit en **arrière-plan** ; appelé aussi parfois **tâche de fond**. Le processus en avant-plan est celui avec lequel vous dialoguez, il reçoit des données de votre clavier et envoie des messages sur votre écran. (Sauf bien sûr si vous avez redirigé ces entrées/sorties comme expliqué dans la section 3.8). À l'opposé, un processus en arrière-plan ne reçoit rien de votre terminal; en général il tourne tranquillement sans jamais rien demander à personne.

10. Il y a beaucoup d'autres processus actifs sur la machine, la commande « `ps -aux` » les montre tous.

Certains programmes ont des temps de traitement très longs, et ne font rien d'intéressant pendant qu'ils travaillent. La compilation d'un gros programme, ou le compactage d'un gros fichier sont des processus de ce type. Il n'y a aucune raison de rester cloué devant votre écran en attendant des heures que ce genre de travail se termine; vous pouvez les lancer en tâche de fond et faire autre chose de plus intéressant pendant ce temps là.

Les processus peuvent aussi être **suspendus**. Un processus suspendu est un job qui n'est pas actif, mais arrêté temporairement, comme figé. Après avoir suspendu un processus, vous pouvez lui indiquer de continuer, en avant-plan ou en arrière-plan, au moment ou vous le voulez.

Notez bien que la suspension d'un processus est différente de son *interruption*. Lorsque vous interrompez un programme (avec la touche d'interruption, qui est généralement `ctrl-C`) [11], le programme se termine, définitivement. Une fois qu'un processus est « tué » ainsi, vous ne pourrez plus le rappeler, il vous faudra relancer la commande. Notez également que certains programmes interceptent le signal d'interruption, et que dans ce cas `ctrl-C` ne tuera pas forcément le programme immédiatement. C'est souvent le cas de programmes devant effectuer certains nettoyages avant de se terminer. Vous rencontrerez même des programmes qui sont prévus pour ne pas pouvoir être interrompus du tout.

3.11.3 Mise en tâche de fond et interruption

Commençons par un exemple simple. La commande `yes` est un programme pouvant paraître inutile qui se borne à afficher indéfiniment des séries de `y`' sur la sortie standard (C'est utilisé pour répondre automatiquement « yes » à des programmes qui demandent de multiples confirmations). Essayez la.

```
/home/rene$ yes
y
y
y
y
y
```

Les `y` continueront à défiler *à l'infini*. Vous pouvez tuer le processus en pressant votre touche d'interruption, qui est probablement `ctrl-C`. Afin de ne pas être gênés par ce flux de `y`, redirigeons la sortie du programme vers `/dev/null`. Vous vous souvenez sans doute que `/dev/null` absorbe toutes les données, tout ce qu'on lui passe disparaît à jamais. C'est une méthode très efficace pour faire taire un programme.

```
/home/rene$ yes > /dev/null
```

11. La touche d'interruption peut être modifiée à l'aide de la commande `stty`. La valeur par défaut sur la plupart des systèmes est `ctrl-C`, mais peut être différente sur votre système.

Ah, c'est beaucoup mieux comme ça ! Rien ne s'affiche, mais l'invite du shell ne revient pas. C'est parce que **yes** est toujours en train de tourner, et envoie frénétiquement des milliers de **y** dans **/dev/null**. Là encore, pour arrêter le programme, utilisez la touche d'interruption.

Supposons maintenant que nous voulions que **yes** continue à fonctionner, mais tout en récupérant l'interactivité de notre shell pour travailler sur autre chose. Nous pouvons passer **yes** en arrière-plan, ce qui lui permettra de tourner, sans avoir besoin de notre clavier.

La première méthode traditionnelle pour lancer un programme en tâche de fond, est de rajouter le caractère « **&** » à la fin de la ligne de commande.

```
/home/rene$ yes > /dev/null &
[1] 164
/home/rene$
```

Hop, vous voyez, l'invite de notre shell est de retour. Mais que signifient les chiffres « `[1] 164` » ? Et est-ce que **yes** est réellement en train de tourner ?

Le numéro « `[1]` » représente le **numéro de job** du processus **yes**. Le shell assigne un tel numéro pour chaque job. Comme **yes** est le seul pour l'instant, il porte le numéro `1`. Le nombre « `164` » est l'identificateur de processus sur le système, le PID. L'un ou l'autre de ces nombres peut être utilisé pour se référer au job, comme nous le verrons plus loin.

Vous avez donc maintenant le processus **yes** en tâche de fond, envoyant continuellement des **y** sur **/dev/null**. Pour vérifier l'état de ce processus, utilisez la commande interne du shell : `jobs`.

```
/home/rene$ jobs
[1]+  Running                 yes >/dev/null  &
/home/rene$
```

D'accord, il est bien là. Vous pourriez aussi utiliser la commande **ps** comme nous l'avons vu au début, pour vérifier si le programme tourne bien.

Pour terminer le processus, il faut utiliser la commande **kill**. Cette commande prend en argument, soit un identificateur de processus, soit un numéro de job. C'était le job numéro 1, aussi la commande

```
/home/rene$ kill %1
```

« tuera » le processus. Lorsque vous utilisez le numéro de job pour spécifier le processus, vous devez préfixer le nombre par le caractère « **%** » (pourcent).

Maintenant que nous avons tué le processus, nous pouvons essayer la commande `jobs` encore une fois pour voir son état :

```
/home/rene$ jobs

[1]+  Terminated              yes >/dev/null

/home/rene$
```

En fait le job est terminé, et si nous utilisons `jobs` encore une fois, plus rien ne s'affichera à l'écran.

Vous pouvez également tuer le job en utilisant le PID (identificateur de processus), qui s'est affiché lorsque vous l'avez lancé. Dans notre exemple, le PID est 164, aussi la commande

```
/home/rene$ kill 164
```

est équivalente à

```
/home/rene$ kill %1
```

Vous ne devez pas rajouter le « `%` » lorsque vous indiquez le PID.

3.11.4 Stopper et relancer des processus

Il y a encore une autre méthode pour passer un processus en tâche de fond. Vous pouvez lancer le programme tout à fait normalement (en avant-plan), puis le **stopper**, et ensuite le relancer en arrière-plan. Nous allons voir comment.

D'abord, lancez la commande `yes` en avant-plan, de manière tout à fait ordinaire:

```
/home/rene$ yes > /dev/null
```

Vous n'avez plus la main sur le shell, puisque le programme est en avant-plan.

Maintenant, au lieu de tuer le programme avec `ctrl-C`, nous allons *suspendre* le processus. Suspendre un processus ne le tue pas: il l'arrête temporairement jusqu'à ce que vous le relanciez. Pour réaliser cela, vous pressez la touche de suspension, qui est généralement `ctrl-Z`.

```
/home/rene$ yes > /dev/null
ctrl-Z
[1]+  Stopped                 yes >/dev/null
/home/rene$
```

Tant que le job est suspendu, il ne tourne simplement pas. Aucun temps machine n'est utilisé pour le processus. Mais vous pouvez le relancer, et il se remettra à tourner comme si rien ne s'était passé, il repartira là où il en était.

Pour relancer le processus en avant-plan, utilisez la commande `fg` (qui signifie « foreground » , avant-plan).

```
/home/rene$ fg
yes >/dev/null
```

Le shell affiche le nom de la commande pour nous rafraîchir la mémoire, et nous indiquer quel est le job que nous venons de rappeler. Stoppez-le encore une fois, avec `ctrl-Z`. Cette fois, utilisez la commande `bg` (qui signifie background, en arrière plan) qui le relancera, mais en tâche de fond. Le résultat sera alors exactement le même que si vous aviez dès le départ lancé la commande avec un « `&` » au bout.

```
/home/rene$ bg
[1]+ yes >/dev/null &
/home/rene$
```

Et nous retrouvons notre invite de shell. La commande `jobs` devrait indiquer que `yes` est en train de tourner, et nous pouvons tuer le processus avec `kill` comme nous l'avons fait tout à l'heure.

Comment pouvons nous stopper à nouveau ce processus ? La touche `ctrl-Z` ne marchera pas, puisqu'il est en tâche de fond. La solution est de repasser le job en avant-plan, avec `fg`, et ensuite de le stopper. Vous pouvez utiliser `fg` aussi bien sur les processus suspendus que ceux qui tournent en tâche de fond.

Il y a une grande différence entre un processus en arrière-plan et un processus suspendu. Un job suspendu ne fonctionne pas, il ne fait rien, il est figé. Il occupe juste de la mémoire. Un processus en arrière-plan fonctionne, utilise des ressources système, bien sûr. Parfois, un processus tournant en tâche de fond peut vouloir écrire du texte sur votre terminal, ce qui peut devenir gênant si vous travaillez sur autre chose pendant ce temps là. Par exemple si vous aviez utilisé la commande

```
/home/rene$ yes &
```

sans rediriger la sortie sur `/dev/null`, un flot de `y` aurait envahi votre écran, ce qui n'est pas très pratique pour saisir son courrier pendant ce temps là... Et ce sans possibilité d'interruption, car `ctrl-C` ne fonctionne pas sur un programme en tâche de fond. La seule solution aurait été d'utiliser la commande `fg`, pour ramener le job en avant-plan afin de pouvoir l'interrompre.

Une autre remarque. Les commandes `fg` et `bg` passent en avant ou arrière plan le dernier processus qui a été suspendu (indiqué par « `+` » à coté du numéro de job sur ce qu'affiche la commande `jobs`). Si vous faites tourner plusieurs processus à la fois, vous pouvez passer le numéro de job en argument à **`fg`** ou **`bg`** pour préciser celui sur lequel vous voulez agir, comme ceci :

```
/home/rene$ fg %2
```

(pour passer en avant-plan le job numéro 2), ou

```
/home/rene$ bg %3
```

(pour passer en arrière-plan le job numéro 3). Vous ne pouvez pas utiliser les PID avec **`fg`** ou **`bg`**.

De plus, utiliser % et le numéro de job seuls, comme dans

```
/home/rene$ %2
```

est équivalent à

```
/home/rene$ fg %2
```

N'oubliez jamais que le contrôle de processus est une possibilité du shell. Les commandes **`fg`**, **`bg`** et **`jobs`** sont internes au shell. Si pour une raison quelconque vous utilisez un shell qui ne supporte pas le contrôle de processus, n'espérez pas y trouver ces commandes.

De plus, certains aspects du contrôle de jobs diffèrent entre Bash et Tcsh. En fait, certains shells ne supportent pas du tout le contrôle de processus, mais la plupart de ceux disponibles sous Linux le gèrent sans problèmes, soyez rassuré, vous avez un système moderne !

3.12 Utilisation de l'éditeur `vi`

Un **éditeur de texte** est un programme destiné à créer ou modifier aisément des fichiers qui contiennent du texte, comme un courrier, un programme en langage C, ou un fichier de configuration. Bien qu'il y ait beaucoup d'éditeurs de cette sorte disponibles sous Linux, le seul que vous soyez assuré de trouver sur tous les systèmes UNIX du monde est **`vi`**, le « visual editor » . **`vi`** est loin d'être le plus facile à utiliser, ni le plus convivial. Toutefois il est si répandu dans le monde UNIX, et il peut tellement vous être indispensable par moments, que nous devons en parler un peu ici.

Le choix d'un éditeur est surtout une question de préférences personnelles et de style. Beaucoup d'utilisateurs préfèrent utiliser l'énorme **Emacs** (un éditeur qui sait faire bien plus que de gérer du texte; il comporte un langage de programmation basé sur LISP) mais étant donné sa taille, vous risquez de ne pas le trouver sur tous les systèmes. Le programme `vi` est quand à lui tout petit et très puissant, mais plus compliqué à utiliser. Malgré tout, une fois que vous aurez pris le temps (en vous énervant quelquefois...) d'apprendre à vous en servir, vous finirez par trouver `vi` facile et souple. Comme beaucoup de choses, c'est souvent l'apprentissage qui est pénible.

Cette section est une introduction pratique à `vi` ; nous n'exposerons pas toutes ses possibilités, mais justes celles que vous avez besoin de connaître pour débuter. Vous pourrez plus tard vous reporter à sa page de manuel si vous désirez en apprendre un peu plus. Ou bien encore, vous pouvez lire le livre *Learning the vi Editor* aux éditions O'Reilly and Associates. Pour plus d'informations, voir l'annexe A.

3.12.1 Concepts

Sous `vi`, vous êtes à tout moment dans l'un des trois modes d'opération possibles. ces modes sont *le mode commande*, *le mode édition*, et *le mode dernière ligne*.

Lorsque vous lancez `vi`, vous êtes en *mode commande*. Ce mode vous autorise un certain nombre de commandes permettant d'éditer des fichiers, ou de basculer dans un autre mode. Par exemple, taper « `x` » en mode commande efface le caractère qui se trouve sous le curseur. Les touches fléchées déplacent le curseur dans le texte que vous éditez. Généralement, les commandes utilisées dans ce mode font un ou deux caractères de long.

Pour insérer ou écrire du texte vous devez utiliser le *mode édition*. C'est probablement celui sous lequel vous passerez le plus de temps lorsque vous utiliserez `vi`. Pour entrer en mode édition, vous devez taper (depuis le mode commande) « `i` » , (pour « insertion »), et là vous pouvez taper du texte qui s'insérera à la position courante du curseur. Pour sortir du mode édition et revenir en mode commande, pressez simplement la touche `esc`.

Le *mode ex*, ou *mode dernière ligne* est un mode spécial utilisé pour passer certaines commandes étendues à `vi`. Lorsque l'on tape ces commandes, elles apparaissent sur la dernière ligne de l'écran (d'où le nom). Par exemple, lorsque vous tapez « `:` » depuis le mode commande, vous passez dans ce mode dernière ligne, et vous pouvez alors taper des commandes comme « `wq` » (qui écrit le fichier et termine le programme) ou « `q!` » (pour quitter `vi` sans sauver les modifications). Le mode dernière ligne est en général utilisé pour des commandes plus longues qu'un seul caractère. Dans ce mode, on entre une commande d'une ligne que l'on valide par la touche `enter`.

3.12.2 Appel de `vi`

Le mieux pour que vous puissiez assimiler ces concepts est d'appeler `vi` et d'éditer un fichier tout en lisant ces lignes. Dans les exemples qui vont suivre, nous n'allons montrer que quelques lignes de texte, comme si l'écran ne faisait que six lignes de haut (au lieu de vingt-quatre lignes probablement chez vous, ou bien plus si vous utilisez X Window et une très grande fenêtre).

La syntaxe de `vi` est

```
vi ⟨fichier⟩
```

où ⟨*fichier*⟩ est le nom du fichier que vous voulez éditer.

Appelez `vi` en tapant

```
/home/marcel$ vi test
```

pour éditer le fichier `test`. Vous devriez voir quelque chose qui ressemble à :

```
_
~
~
~
~
~
"test" [New file]
```

La colonne de caractères « ~ » indique que vous êtes à la fin du fichier.

3.12.3 Insertion de texte

Vous êtes donc en mode commande ; pour insérer du texte dans le fichier, tapez `i` (ce qui vous placera en mode édition), et commencez à taper votre texte.

```
C'est maintenant au tour du patron de payer sa tournée_
~
~
~
~
~
```

Vous pouvez entrer autant de lignes que vous voulez lors de l'insertion de texte, (en pressant la touche `Entrée` à la fin de chaque ligne, bien entendu), et vous pouvez corriger vos erreurs immédiatement par la touche « backspace » (retour arrière, celle qui est au dessus de `Entrée` en principe).

Pour sortir du mode édition et revenir au mode commande, tapez `esc`.

Une fois en mode commande, vous pouvez utiliser les touches fléchées pour vous déplacer dans le texte. Ici, comme nous n'avons encore qu'une ligne, les touches haut et bas seront inopérantes et vous serez alerté par un signal sonore.

Il y a plusieurs façons d'insérer du texte, autres que la commande `i`. Par exemple la commande `a` permet l'insertion de texte juste *après* la position du curseur, et non pas juste dessus. Essayez par exemple d'utiliser la touche fléchée gauche afin de déplacer le curseur entre les mots « du » et « patron » .

```
C'est maintenant au tour du_patron de payer sa tournée.
~
~
~
~
~
```

Tapez `a`, pour passer en mode édition, puis entrez « **`généreux`** » , et enfin pressez `esc` pour revenir en mode commande.

```
C'est maintenant au tour du généreux_patron de payer sa tournée.
~
~
~
~
~
```

Pour insérer du texte une ligne au dessous, utilisez la commande `o`. Par exemple, pressez `o` et rajoutez une ligne ou deux :

```
C'est maintenant au tour du généreux patron de payer sa tournée.
Après, nous irons nous taper un cassoulet chez la mère Tapedur_
~
~
~
~
```

Gardez à l'esprit simplement, qu'à tout moment vous êtes soit en mode commande (où les actions comme `i`, `a`, ou `o` sont valides), ou en mode insertion (où le texte saisi s'affiche, suivi de `esc` pour revenir en mode commande), ou enfin en mode dernière ligne.

3.12.4 Suppression de texte

Depuis le mode commande, la touche `x` efface le caractère situé sous le curseur. Si vous tapez `x` 16 fois, vous obtiendrez :

```
C'est maintenant au tour du généreux patron de payer sa tournée.
Après, nous irons nous taper un cassoulet chez_
~
~
~
~
```

Maintenant, pressez la touche a, insérez un peu de texte, suivi par esc:

```
C'est maintenant au tour du généreux patron de payer sa tournée.
Après, nous irons nous taper un cassoulet chez le père Naud.
~
~
~
~
```

Vous pouvez supprimer des lignes entières par la commande `dd` (c'est à dire taper deux fois d sur une ligne). Si votre curseur est sur la seconde ligne, et que vous tapez `dd`,

```
C'est maintenant au tour du généreux patron de payer sa tournée.
~
~
~
~
~
```

Pour supprimer tout le mot sous lequel se trouve le curseur, utilisez la commande `dw`. Placez le curseur sur le mot « généreux » , et tapez avec assurance `dw`, car les doses servies étaient petites :

```
C'est maintenant au tour du patron de payer sa tournée.
~
~
~
~
~
```

Si vous aviez tapé les caractères accentués dans le texte, il se peut que la commande ne fonctionne pas sur les mots accentués, selon l'origine de votre version de `vi`. Bienvenue dans le monde où ceux qui ne parlent pas anglais sont très souvent ignorés.

3.12.5 Remplacement de texte

Vous pouvez remplacer des parties de texte à l'aide de la commande `R`. Placez le curseur sur la première lettre de « payer » , tapez R, puis entrez le mot « remettre » .

```
C'est maintenant au tour du patron de remettre_tournée.
~
~
~
~
~
```

La commande `R` est très semblable aux commandes `i` et `a`, mais elle remplace le texte existant au lieu d'effectuer une insertion.

Corrigeons notre phrase avant d'aller plus loin, vous devriez d'ores et déjà être capable d'insérer ce qu'il faut tout seul :

```
C'est maintenant au tour du patron de remettre sa_tournée.
~
~
~
~
~
```

Voilà qui est plus raisonnable !

La commande `r` remplace le caractère qui se trouve sous le curseur. Par exemple, déplacez le curseur sur le début de « sa » , que nous venons de rajouter, et tapez `r` suivi de `l`, et vous obtenez :

```
C'est maintenant au tour du patron de remettre la tournée.
~
~
~
~
~
```

La commande « `~` » bascule la lettre située sous le curseur de minuscule en majuscule, et vice versa; si vous placez par exemple le curseur sur le « e » de « C'est » , et pressez de manière répétée la touche `~`, vous finirez par obtenir :

```
C'EST MAINTENANT AU TOUR DU PATRON DE REMETTRE LA TOURNéE.
~
~
~
~
~
```

Vous constaterez probablement encore une fois que les caractères accentués ne sont pas pris en compte.

3.12.6 Commandes de déplacement

Vous savez déjà comment utiliser les touches fléchées pour vous déplacer dans le document. Il est aussi possible de se déplacer de la même façon par les commandes `h`, `j`, `k`, et `l`, qui déplacent le curseur respectivement à gauche, en bas, en haut, et à droite. Il est très utile de connaître ces commandes lorsque les touches fléchées ne fonctionnent pas pour une raison quelconque. (Par exemple, si le clavier n'en est pas équipé). Cela permet de se sortir des pires situations.

La commande `w` déplace le curseur au début du mot suivant ; la commande `b` le place au début du mot précédent.

La commande `0` (c'est un zéro) déplace le curseur au début de la ligne courante, et `$` le place à la fin de la ligne.

Lors de l'édition de gros fichiers, vous aurez besoin de vous déplacer par écrans entiers pour aller plus vite. Pressez `ctrl-F` pour avancer d'un écran, et `ctrl-B` pour remonter dans le fichier.

Pour aller directement à la fin du fichier, tapez `G`. Vous pouvez également vous positionner sur une ligne arbitraire; par exemple si vous tapez la commande `10G` vous vous retrouverez sur la ligne 10 du fichier. Pour vous positionner au début du fichier, utilisez la commande `1G`.

Vous pouvez coupler les commandes de déplacement avec d'autres, comme la suppression. Par exemple, `d$` supprimera tout le texte à partir du curseur jusqu'à la fin de la ligne, `dG` supprimera tout ce qui se trouve à partir du curseur jusquà la fin du fichier, et ainsi de suite.

3.12.7 Sauver le fichier et quitter vi

Pour quitter `vi` sans conserver les modifications dans le fichier édité, utilisez la commande `:q!`. Lorsque vous tapez « : » , le curseur se déplacera sur la dernière ligne de l'écran et vous serez en mode dernière ligne.

```
C'EST MAINTENANT AU TOUR DU PATRON DE REMETTRE LA TOURNéE.
~
~
~
~
~
:_
```

En mode dernière ligne, certaines commandes étendues sont disponibles. L'une d'elle est `q!`, qui sort de l'éditeur sans sauver le fichier. La commande `:wq` enregistre le fichier puis termine `vi`. La commande `ZZ` (depuis le mode commande, donc sans le « : ») est équivalente à `:wq`. Rappelez-vous que vous devez taper `enter` après toute commande entrée en mode dernière ligne.

Pour enregistrer le fichier, mais sans quitter l'éditeur, tapez simplement `:w`.

3.12.8 Éditer un autre fichier

Pour éditer un autre fichier, utilisez la commande `:e`. Par exemple, pour cesser d'éditer `test`, et éditer le fichier `bidule` à la place, utilisez la commande

```
C'EST MAINTENANT AU TOUR DU PATRON DE REMETTRE LA TOURNéE.
~
~
~
~
~
:e bidule_
```

Si vous utilisez `:e` sans sauver préalablement le premier fichier, vous aurez le message d'erreur

```
No write since last change (":edit!" overrides)
```

qui signifie simplement que `vi` ne veut pas changer de fichier tant que vous n'avez pas sauvé le premier, par sécurité. À ce moment vous pouvez soit utiliser `:w` pour enregistrer le fichier en question, puis utiliser `:e`, soit taper la commande

```
C'EST MAINTENANT AU TOUR DU PATRON DE REMETTRE LA TOURNéE.
~
~
~
~
~
:e! bidule_
```

Le « `!` » indique à `vi` que vous insistez vraiment pour éditer le premier fichier et que vous êtes bien au courant que vous allez perdre le premier.

3.12.9 Inclure un autre fichier

À l'aide de la commande `:r`, vous pouvez inclure le contenu d'un autre fichier dans celui que vous êtes en train d'éditer. Par exemple la commande

```
:r signature.txt
```

insérera le contenu du fichier `signature.txt` à la position courante du curseur.

3.12.10 Lancer une commande shell

Vous pouvez également lancer des commandes depuis l'éditeur `vi`. La commande `:r!` fonctionne comme `:r`, mais au lieu de lire un fichier, elle insère la sortie de la commande indiquée, à partir de la position courante du curseur. Par exemple si vous utilisez la commande

```
:r! uname -a
```

vous aboutirez à

```
C'EST MAINTENANT AU TOUR DU PATRON DE REMETTRE LA TOURNéE.
Linux plux 1.1.20 #11 Fri Jun 17 17:19:46 MET DST 1994 i486
~
~
```

Vous pouvez aussi « obtenir un shell » à partir de `vi`, en d'autres termes, lancer une commande sans quitter l'éditeur, et retourner à votre édition de fichier lorsque vous avez terminé. Par exemple en faisant

```
:! ls -F
```

la commande `ls -F` sera exécutée, et le résultat s'affichera à l'écran, mais ne sera pas inséré dans le fichier que vous éditez. Si vous utilisez la commande

```
:shell
```

`vi` lancera une copie du shell, vous permettant temporairement de mettre l'éditeur « en attente » pendant que vous faites autre chose. Lorsque vous sortirez de ce shell (en utilisant sa commande `exit`), vous retrouverez `vi`.

3.12.11 Obtenir de l'aide

`Vi` n'offre pas beaucoup d'aide interactive (peu de programmes UNIX le font), mais vous pouvez toujours lire la page de manuel qui lui est consacrée (`man vi`). En fait, cet éditeur est une interface plein écran de l'éditeur `ex`; c'est en réalité `ex` qui gère l'essentiel des commandes passées en mode dernière ligne. Aussi, en plus de la lecture de la page de manuel de `vi`, jetez un œil également à celle de `ex`.

3.13 Configurer votre environnement

Le shell offre de nombreux mécanismes pour configurer votre environnement de travail. Nous l'avons souvent répété, le shell est bien plus qu'un bête interpréteur de commandes, c'est également un langage de programmation puissant. Bien que l'écriture de shell-scripts soit un sujet extrêmement vaste, nous allons essayer de vous présenter quelques trucs qui vous permettront de vous faciliter la vie sur un système UNIX, en utilisant quelques unes des possibilités du shell.

Nous vous avons déjà expliqué que les différents shells ont des syntaxes différentes. Par exemple, Tcsh utilise un langage ressemblant au langage de programmation C, alors que les shells Bourne ont une syntaxe compatible avec les shells des premiers systèmes UNIX. Dans cette section, nous ne ferons que des choses simples, mais nous considérerons que le shell-scripts sont exécutés par un shell de type Bourne.

3.13.1 Shell scripts

Admettons que vous utilisiez souvent une série de commandes, et que vous vouliez les grouper toutes sous un même nom pour ne pas perdre de temps à les taper une par une à chaque fois. Par exemple, les commandes

```
/home/rene$ cat chapitre1 chapitre2 chapitre3 > bouquin
/home/rene$ wc -l bouquin
/home/rene$ lpr bouquin
```

concaténerait les trois fichiers dans `bouquin`, puis afficherait le nombre total de lignes contenues, et enfin imprimerait le fichier à l'aide de la commande `lpr`. Au lieu de taper toutes ces commandes, vous pourriez les grouper dans un **shell script**. Nous avons abordé rapidement ces fichiers de commandes dans la section 3.13.1. Ce script ressemblerait donc à

```
#!/bin/sh
# Un shell script pour creer et imprimer le livre

cat chapitre1 chapitre2 chapitre3 > bouquin
wc -l bouquin
lpr bouquin
```

Ce script étant sauvé dans le fichier `makebook` (par exemple), il suffit de taper la commande

```
/home/rene$ makebook
```

pour qu'aussitôt les commandes qu'il contient soient exécutées. Les shell-scripts sont de purs fichiers texte; vous pouvez les réaliser avec un éditeur comme `emacs` ou `vi` [12].

Regardons cet exemple de shell-script. La première ligne, « `#!/bin/sh` » , indique que le fichier est un script, et indique au shell de quelle façon l'exécuter. Dans ce cas, il devra le passer à `/bin/sh`, qui est le shell lui-même dans ce cas. Pourquoi est-ce important ? Sur beaucoup de systèmes UNIX, `/bin/sh` est un shell Bourne, comme Bash. En forçant le shell à utiliser `/bin/sh`, on s'assure que le script sera bien exécuté par le bon shell, quelque soit celui qui est utilisé par défaut. Même si vous êtes sous Tcsh, ce script fonctionnera correctement.

12. L'utilisation de `vi` est décrite dans la section 3.12, page 137.

La deuxième ligne est un *commentaire*. Les commentaires commencent par le caractère « **#** » et s'étendent jusqu'à la fin de la ligne. Ils sont ignorés par le shell, ils permettent de mettre des explications ou des notes pour que le programmeur s'y retrouve.

Les autres lignes sont juste des commandes, exactement comme vous les taperiez au clavier directement. Le shell lit chaque ligne du script et l'exécute comme si vous veniez de la taper.

Les permissions d'accès des shell-scripts sont importantes: lorsque vous créez un tel fichier, vous devez vous assurer d'avoir la permission d'exécution pour pouvoir le lancer[13]. La commande

```
/home/rene$ chmod u+x makebook
```

peut être utilisée à cet effet.

3.13.2 Les variables et l'environnement

Le shell permet de définir des **variables**, comme la plupart des langages de programmation. Une variable est une donnée à qui l'on attribue un nom.

◇ Notez que Tcsh, tout comme d'autres C-shells, utilise un mécanisme différent de ce qui est décrit ici pour assigner des valeurs aux variables. Nous considérons ici un shell Bourne, comme Bash (que vous utilisez probablement), ou le **/bin/sh** de la plupart des systèmes UNIX. Voyez la page de manuel de Tcsh si vous voulez savoir comment ce shell fonctionne.

Lorsque vous assignez une valeur à une variable (avec l'opérateur « **=** »), vous pouvez alors accéder à cette variable en rajoutant un « **$** » devant son nom, voici un exemple :

```
/home/rene$ toto=« Salut la dedans »
```

Nous avons donné la valeur « **Salut la dedans** » à la variable que nous avons appelée **toto**. Nous pouvons maintenant nous référer à cette variable par son nom, préfixé par le caractère « **$** » . La commande

```
/home/rene$ echo $toto
Salut la dedans
/home/rene$
```

13. Lorsque vous créez des fichiers texte, les droits d'accès par défaut ne comprennent généralement pas la permission d'exécution.

produit maintenant le même résultat que

```
/home/rene$ echo « Salut la dedans »
Salut la dedans
/home/rene$
```

Ces variables sont internes au shell. Cela veut dire que seul le shell peut y avoir accès. C'est très utile dans les shell-scripts; si vous voulez utiliser un nom de fichier, vous pouvez le stocker dans une variable, comme ci-dessus. La commande `set` affiche une liste de toutes les variables définies à un moment donné.

Mais le shell vous permet d'**exporter** des variables dans l'**environnement**. L'environnement est un ensemble de variables auquel toutes les commandes que vous exécutez, scripts ou programmes binaires, ont accès. Une fois que vous avez défini une variable dans le shell, le fait de l'exporter la place dans cet environnement et la rend connue de tous vos programmes. La commande `export` est utilisée pour cette opération.

◇ Là encore, la méthode est différente entre Bash et Tcsh. Sous les C-shells, les variables d'environnement sont positionnées selon une autre syntaxe (la commande `setenv`), reportez-vous à la page de manuel.

L'environnement est une chose très importante dans les systèmes UNIX. Il vous permet de configurer certains programmes, simplement en positionnant des variables ad hoc.

Voici un petit exemple. La variable d'environnement `PAGER` est utilisée par la commande `man`; elle désigne le programme à utiliser pour afficher les pages. Si vous positionnez `PAGER` sur un nom de commande, il utilisera cette commande pour afficher les pages de manuel, au lieu de `more` qui est le programme utilisé par défaut.

Assignez « `cat` » à la variable `PAGER`. Cela aura pour effet de faire afficher par `man`, les pages en une seule fois, sans s'arrêter à chaque écran.

```
/home/rene$ PAGER=« cat »
```

Maintenant, exportez `PAGER` dans l'environnement.

```
/home/rene$ export PAGER
```

Essayez la commande `man ls`. La page de manuel va défiler sur votre écran, car elle est affichée en un seul bloc.

Maintenant, si nous positionnons `PAGER` sur « `more` » , c'est cette commande qui sera utilisée pour l'affichage.

```
/home/rene$ PAGER=« more »
```

Notez que nous n'avons pas besoin d'utiliser la commande **export** une seconde fois, la variable est déjà exportée dans l'environnement ; une fois suffit. Tout changement de sa valeur sera pris en compte.

Les pages de manuel des différentes commandes que vous utilisez vous renseigneront sur le nom des éventuelles variables d'environnement qu'elles peuvent utiliser; par exemple la page de manuel de **man** explique l'usage de **PAGER**. Certaines commandes partagent des variables; par exemple de nombreux programmes utilisent la variable **EDITOR** pour savoir quel est l'éditeur de texte préféré de l'utilisateur qu'il faudra utiliser par défaut.

L'environnement est aussi utilisé pour stocker certaines informations importantes de votre session. Un bon exemple, la variable **HOME**, qui contient le nom de votre répertoire personnel.

```
/home/rene/notes$ echo $HOME
/home/rene
```

Une autre variable d'environnement qui vous intéressera sûrement: la variable `PS1` définit l'invite principale de votre shell. Par exemple,

```
/home/rene$ PS1=« Votre commande: »
Votre commande:
```

Pour remettre l'invite que vous aviez avant (qui contenait le répertoire courant suivi du symbole « **$** »),

```
Votre commande: PS1=« \w£  »
/home/rene$
```

La page de manuel de **bash** décrit la syntaxe utilisée pour l'invite, lisez-la, on peut y mettre des choses très pratiques !

3.13.2.1 La variable d'environnement `PATH`

Lorsque vous utilisez la commande **ls**, comment le shell sait-il où trouver le programme **ls** dans tout ce fatras de fichiers qui est sur votre disque dur ? En fait, **ls** se trouve dans **/bin/ls** sur la plupart des systèmes. Le shell utilise la variable d'environnement **PATH** pour rechercher les fichiers exécutables des commandes que vous tapez.

Par exemple, votre variable **PATH** a sans doute une valeur qui ressemble à :

```
/bin:/usr/bin:/usr/local/bin:.
```

C'est une liste de répertoires dans lesquels le shell ira faire sa recherche, séparés par le caractère « : » . Quand vous utilisez la commande `ls`, le shell commence par rechercher `/bin/ls`, s'il ne trouve pas il va essayer `/usr/bin/ls`, et ainsi de suite.

Notez bien que le `PATH` n'a rien à voir avec la recherche de fichiers ordinaires, il ne concerne que les commandes. Par exemple si vous tapez

```
/home/rene$ cp toto titi
```

Le shell n'utilisera pas `PATH` pour rechercher les fichiers `toto` et `titi`, seul l'exécutable de la commande `cp` sera recherché en fonction du `PATH`.

Cette variable fait gagner un temps précieux: vous n'avez pas besoin de vous souvenir du chemin d'accès complet aux exécutables. Les programmes sont souvent dispersés en différents endroits, comme `/usr/bin`,`/bin`, `/usr/local/bin`, ou `/usr/bin/X11`. Au lieu d'être obligé de donner le chemin complet (comme `/usr/bin/cp`), vous pouvez simplement assigner à `PATH` la liste de tous les répertoires où vos commandes peuvent se trouver, et le shell fera tout le travail pour vous.

Notez que le `PATH` contient « . » , qui correspond au répertoire courant. Vous pouvez ainsi créer un script ou un programme et le lancer, depuis l'endroit où vous vous trouvez, sans être obligé de le copier ailleurs ou de spécifier exactement ce répertoire (comme dans `./makebook`). Si un répertoire n'est pas dans votre `PATH`, le shell n'ira pas y chercher les commandes pour vous, et ceci est valable pour le répertoire courant.[14]

3.13.3 Scripts d'initialisation

En plus des shell-scripts que vous créez vous-même, il y a un certain nombre de scripts que le shell utilise pour certains usages. Les plus importants sont vos **scripts d'initialisation**, qui sont automatiquement exécutés par le shell lorsque vous vous connectez sur le système.

Ce sont de simple shell-scripts, semblables à ceux décrits plus haut. Toutefois ils sont très utiles pour configurer votre environnement, en exécutant automatiquement des commandes lorsque vous arrivez sur le système. Par exemple, si vous utilisez toujours la commande `mail` pour vérifier votre courrier à ce moment là, vous pouvez placer cette commande dans votre script d'initialisation de manière à ce qu'elle soit automatiquement exécutée.

14. Les utilisateurs de MS-DOS doivent bien noter ce fait. MS-DOS utilise une toute autre stratégie, il recherche par défaut d'abord dans le répertoire courant, et seulement ensuite utilise le PATH. UNIX ne fait pas de cas particulier pour le répertoire courant, s'il n'est pas dans le PATH la recherche ne s'y fera pas. C'est la meilleure solution pour des raisons de sécurité et d'administration. En général, soit on ne met pas le répertoire courant dans le PATH, soit on le met en dernière position de recherche, surtout pas en première; ces notions de sécurité dépassent largement le cadre de cet ouvrage.

Bash et Tcsh distinguent tous deux ce que l'on appelle un **login shell**, et les autres invocations du shell. Un « login shell » est celui invoqué lorsque vous vous connectez sur la machine; c'est généralement le seul que vous utilisiez, avec lequel vous dialoguez. Cependant, si vous appelez un shell depuis un autre programme, comme depuis `vi`, vous lancez une seconde instance du shell, qui n'est pas votre login shell. De plus, chaque fois que vous exécutez un script, vous lancez automatiquement un autre shell qui interprète ce fichier.

Les fichiers d'initialisation utilisés par Bash son: `/etc/profile` (configuré par l'administrateur du système, et exécuté par tous les shells Bourne lorsqu'un utilisateur se connecte) ; `$HOME/.bash_profile` (exécuté par tous les login shells Bash) ; et `$HOME/.bashrc` (exécuté par tous les Bash qui ne sont pas login shell). Si `.bash_profile` n'existe pas, `.profile` est utilisé à la place.

Tcsh utilise les fichiers d'initialisation suivants: `/etc/csh.login` (exécuté par tous les C-shell lorsqu'un utilisateur se connecte), `$HOME/.tcshrc` (exécuté par tous les login shells et toutes les autres instances de Tcsh), et `$HOME/.login` (exécuté au login, après `.tcshrc`). Si `.tcshrc` n'existe pas, `.cshrc` est utilisé à la place.

Pour bien maîtriser les fonctions et l'usage de ces fichiers, vous devez approfondir davantage vos connaissances du shell. La programmation shell est un sujet compliqué, dépassant largement du cadre de ce livre. Lisez les pages de manuel de `bash` et/ou `tcsh` pour en savoir un peu plus, et si les shell-scripts vous passionnent offrez-vous un ouvrage sur le sujet.

3.14 Vous vous lancez dans l'aventure?

Nous vous avons fourni suffisamment d'informations pour que vous puissiez commencer à utiliser le système. N'oubliez pas que la plupart des aspects les plus intéressants et les plus importants de Linux ne vous ont pas été présentés, vous n'avez vu que les notions de base. Avec ces connaissances, dans très peu de temps vous pourrez utiliser de complexes applications et étendre les possibilités de votre système à l'infini. Si tout cela ne vous semble pas extraordinaire pour l'instant, ne désespérez pas : vous avez encore beaucoup à apprendre.

Les pages de manuels du système sont un outil d'apprentissage indispensable. Bien que la plupart d'entre elles puissent apparaître confuses au premier abord, vous y trouverez en approfondissant des tonnes d'informations passionnantes.

Nous vous suggérons également de lire un ouvrage complet traitant de l'utilisation d'UNIX. Vous en trouverez une liste dans l'annexe A.

```
/ ─┬─ bin
   ├─ dev
   ├─ etc
   ├─ home ─┬─ dupont
   │        └─ rene
   ├─ lib
   ├─ proc
   ├─ tmp
   └─ usr ─┬─ X386
           ├─ bin
           ├─ emacs
           ├─ etc
           ├─ g++-include
           ├─ include
           ├─ lib
           ├─ local ─┬─ bin
           │         ├─ emacs
           │         ├─ etc
           │         └─ lib
           ├─ man
           ├─ spool
           ├─ src ─── linux
           └─ tmp
```

FIG. 3.1 - *Une arborescence de répertoires typique sous UNIX (abrégée).*

Chapitre 4

Administration système

Ce chapitre est un aperçu de l'administration système sous Linux, et comprend un certain nombre d'explications plus poussées qui ne sont pas uniquement utiles aux administrateurs de la machine. À chaque système est attaché un administrateur, et sa gestion est une tâche très importante, demandant parfois beaucoup de temps, même si vous êtes l'unique utilisateur de votre ordinateur.

Nous avons essayé de décrire ici les choses les plus importantes que vous devez absolument connaître à propos de l'administration système pour pouvoir utiliser Linux, avec suffisamment de détails pour faciliter vos débuts. Afin que cette partie reste courte et supportable, nous n'avons traité que des choses de base, et avons omis beaucoup de détails importants. Vous devez lire le *Linux System Administrator's Guide* si vous avez de sérieux problèmes. Il vous aidera à mieux comprendre comment tout cela est organisé et fonctionne. Au minimum, feuilletez-le, de manière à savoir ce qu'il contient et quel genre de renseignements vous pouvez y trouver.

4.1 Les dangers du pouvoir

Vous le savez, UNIX différencie les divers utilisateurs, de manière à sécuriser leurs actions sur le système (éviter que quelqu'un ne lise votre carnet d'adresses par exemple). Chaque utilisateur possède son propre **compte**, avec son répertoire personnel, son nom d'utilisateur, etc. Parallèlement à ces utilisateurs ordinaires, il existe des comptes spéciaux définis dans le système qui possèdent certains privilèges. Le plus important est le compte `root`.

4.1.1 L'utilisateur `root`

Les utilisateurs ordinaires ont généralement des privilèges tels qu'ils ne peuvent rien faire qui puisse gêner les autres utilisateurs ou la bonne marche du système. Les droits d'accès aux fichiers sont positionnés de telle façon que l'utilisateur normal ne puisse ni effacer, ni modifier des fichiers dans les répertoires partagés par tous (comme `/bin` et `/usr/bin`). Beaucoup protègent également leurs propres fichiers en leur donnant des permissions telles que les autres personnes utilisant le système ne puissent y accéder.

Il n'y a pas de telles restrictions pour `root`. L'utilisateur `root` peut lire, modifier, supprimer n'importe quel fichier du système, changer les permissions et les propriétaires, exécuter certains programmes, comme ceux destinés à partitionner les disques durs ou créer des systèmes de fichiers. Le principe est simple: la personne (ou les personnes) responsables de la gestion de l'ordinateur utilise le compte `root` chaque fois qu'il est nécessaire d'effectuer une opération normalement interdite aux utilisateurs normaux. Elle est la seule à en posséder le mot de passe, et utilise ce privilège parcimonieusement: `root` ayant tous les droits, il lui est très facile de faire des erreurs pouvant avoir des conséquences catastrophiques.

Par exemple, en tant qu'utilisateur normal, si par inadvertance vous tentiez d'effacer tous les fichiers présents dans `/etc`, le système vous l'interdirait. Si c'est l'utilisateur `root` qui fait la même manipulation, il en aura le droit et les fichiers seront irrémédiablement perdus, rendant la machine quasiment inutilisable. Il est très facile de détériorer le système sous le compte `root`. La meilleure façon d'éviter les accidents, est de suivre ces conseils:

- Vérifiez bien les commandes que vous tapez avant de presser `Entrée`. Par exemple, si vous êtes en train de nettoyer un répertoire, avant de taper frénétiquement sur `Entrée`, relisez bien ce qui est à l'écran et assurez-vous que vous n'allez pas effacer des données cruciales. Dans le doute utilisez `rm -i` plutôt que `rm` sans paramètres; mieux vaut perdre du temps à confirmer chaque opération que perdre un an de travail en une fraction de seconde.

- Ne prenez pas l'habitude d'utiliser le compte `root` tout le temps. Sinon vous finiriez par ne plus réaliser que vous êtes privilégié et confondre `root` avec votre compte ordinaire. Les conséquences sur certaines erreurs de commande sont fort différentes !

- Utilisez une invite différente pour l'utilisateur `root`. La tradition veut que le dernier caractère soit « `#` » si l'on est `root` et « `$` » pour le reste des utilisateurs. Les fichiers d'initialisation des shells servent aussi à ça. [1]

- Ne soyez `root` que lorsque c'est absolument nécessaire, et aussitôt que vous n'en aurez plus besoin, quittez ce compte. Moins vous utiliserez `root`, moins

1. Si vous utilisez le shell `bash`, vous rendre cela automatique, en mettant « \$ » à la fin de la définition de `PS1`. Bash passera tout seul de « `$` » à « `#` » selon les privilèges dont vous disposez.

vous prendrez le risque de faire de grosses bêtises.

Bien sûr, vous rencontrerez des experts UNIX irréductibles qui utilisent `root` pour pratiquement tout faire. Mais interrogez-les... Chacun d'eux a fait au moins une fois une erreur stupide et détruit son système. La règle générale est que, tant que vous ne maîtrisez pas parfaitement l'absence de restrictions sur `root`, et tant que l'utilisation en tant qu'utilisateur normal vous est facile, utilisez `root` le moins possible et avec beaucoup de précautions.

Bien entendu, tout le monde fait des erreurs. Linus Torvalds lui-même a effacé par erreur l'intégralité de son système de fichiers. Des jours de travails perdus à jamais. Heureusement pour lui, en raison de sa connaissance du code gérant les systèmes de fichiers dans le noyau, il fut capable de relancer son système et de reconstruire à la main toute l'arborescence perdue. N'espérez pas savoir en faire autant de sitôt...

Vu d'une autre façon, si vous vous imaginez le compte `root` comme une armure magique qui vous donne la puissance suprême de pouvoir d'un simple claquement des doigts, diriger ou détruire votre système, régner en maître sur vos utilisateurs éventuels, une bonne idée serait de faire attention où vous mettez les doigts. Un faux mouvement est si vite arrivé... Evitez donc plutôt d'endosser cette armure trop souvent. Le pouvoir a déjà tourné la tête à trop de gens en ce bas monde.

4.1.2 Abus de pouvoir

Le sentiment de puissance amène souvent le désir de nuire. C'est l'un des grands pièges de l'administration de systèmes UNIX, et tout le monde tombe plus ou moins dedans un jour où l'autre. La majorité des utilisateurs d'UNIX ne pourra jamais connaître ces impressions : sur les systèmes des grandes compagnies ou des universités, seuls quelques spécialistes hautement qualifiés et responsables, très cher payés, ont la possibilité d'utiliser le compte `root` afin d'administrer le système. En fait, dans la plupart des sites UNIX, le mot de passe de `root` est un secret extrêmement bien gardé : Il est traité comme le trésor le plus précieux qui soit. Les rumeurs courent, des légendes s'installent, pouvoir se connecter en tant que `root` est dépeint comme un acte magique donnant la puissance suprême, réservée à quelques élus dominant de leur supériorité le reste du monde informatique.

De telles attitudes envers le compte `root` sont justement le genre de choses qui provoquent la convoitise et l'envie de nuire. Ce compte `root` est tellement porté aux nues, que dès qu'un utilisateur a l'opportunité pour la première fois d'être `root` (que ce soit sur un système Linux ou ailleurs), il aura tendance à abuser des privilèges qui lui sont octroyés. On a vu tant de prétendus « administrateurs système » lire le courrier des autres utilisateurs, effacer leurs fichiers sans prévenir, et d'une manière générale se comporter comme des gamins dès lors qu'on leur offre un « jouet » aussi puissant.

Parce que être `root` offre de tels privilèges sur le système, il faut beaucoup de calme et surtout de maturité pour utiliser ce compte dans son véritable but : faire marcher la machine. Il y a un accord tacite, un code d'honneur entre les utilisateurs et leur administrateur. Que penseriez-vous si votre administrateur système lisait votre courrier personnel ou espionnait tout ce que vous faites dans vos fichiers ? Il n'y a pas encore de précédent juridique en ce domaine. Sur les systèmes UNIX, l'utilisateur `root` a la possibilité d'outrepasser toutes les sécurités et mécanismes de protection. Il est indispensable que lui et les utilisateurs travaillent dans un climat de confiance et de respect mutuel. Nous ne le répéterons jamais assez.

4.1.3 Relations avec les utilisateurs

La sécurité sous UNIX est plutôt laxiste, par conception. La notion de sécurité fut introduite après-coup: le système fut à l'origine développé dans un milieu où il n'était pas pensable que certains utilisateurs pensent à mal. Pour cette raison, même en prenant des mesures de sécurité, certains utilisateurs normaux peuvent encore trouver le moyen de nuire, souvent involontairement.

Les administrateurs système peuvent agir de deux manières envers les utilisateurs qui abusent : Ils peuvent être paranoïaques ou confiants. L'administrateur parano fait en général plus de mal que de bien. La majorité des utilisateurs n'ont ni la possibilité ni les connaissances pour réellement endommager le système. Dans 90% des cas, lorsqu'un utilisateur gêne le fonctionnement de la machine (par exemple en remplissant les partitions utilisateurs par d'énormes fichiers, ou exécutant de nombreuses instances d'une très grosse application), il n'a absolument pas conscience que ce qu'il fait pose un gros problème. Il ne cherche pas à nuire, il ne comprend simplement pas ce qu'il fait.

Lorsque vous vous trouvez face à un utilisateur qui provoque un problème, ne l'accusez pas immédiatement: tant que rien ne prouve qu'il est coupable, il est innocent, cette règle s'applique aussi ici. Le mieux à faire est de discuter avec lui, et lui parler calmement de ce qui s'est passé, au lieu d'être agressif. Evitez à tout prix de vous mettre l'utilisateur à dos, ce qui aurait pour effet de vous rendre d'un seul coup très suspect aux yeux de beaucoup, vous pourriez vous voir accusé de tous les maux. Au contraire, expliquez-lui ce qui s'est passé, guidez-le et il vous en sera reconnaissant, et vous pourrez être assuré qu'il ne fera plus jamais la même fausse manipulation par ignorance.

Si vous avez vraiment la preuve que quelqu'un a tenté de « pirater » le système sciemment, ou a essayé volontairement de troubler son fonctionnement, là encore ne soyez pas agressif. Avertissez-le qu'il est découvert, mais soyez souple. Dans beaucoup de cas vous pourrez le prendre « la main dans le sac » , à ce moment envoyez-lui un message, lui disant de ne pas recommencer. Mais assurez-vous bien d'avoir la preuve que c'est un acte délibéré et pas une erreur de sa part.

4.1.4 Établir les règles

On n'administre pas un système UNIX à coups de cravache. Le monde UNIX n'est pas fait pour la discipline bête et méchante, laissons cela aux militaires. Le mieux est d'établir un certain nombre de règles souples, un guide de conduite des utilisateurs; mais dites-vous bien que moins il y aura de règles, moins vous aurez de chances d'avoir des infractions. Même si elles sont parfaitement claires et raisonnables, les utilisateurs finiront toujours par les enfreindre involontairement de temps à autres. C'est particulièrement vrai pour les nouveaux venus à UNIX, qui apprennent tant bien que mal les méandres du système, ses possibilités et ses limites. Par exemple, Il n'est pas évident au premier abord pour un complet novice, de comprendre qu'il ne faut pas télécharger 1 gigaoctet de programmes et les envoyer à tout le monde par courrier. Les utilisateurs ont besoin d'aide, et doivent comprendre la raison des règles qu'on leur impose.

Si vous rédigez un guide de conduite pour votre système, assurez-vous que tout soit bien clair et compréhensible, et que chaque limitation imposée puisse être comprise par tout le monde. Si vous ne le faites pas, les utilisateurs trouveront toutes sortes d'astuces plus créatives les unes que les autres pour arriver à leurs fins, sans se rendre compte qu'ils abusent.

4.1.5 Mais encore?

Nous ne pouvons pas vous donner de recette idéale pour administrer votre système, tout dépend de la manière dont vous l'utilisez. Les choses sont fort différentes selon le nombre d'utilisateurs, l'isolement ou la connexion réseau de la machine, le type d'accès, etc. Malgré tout, quelle que soit votre situation, même si vous êtes le seul utilisateur de votre ordinateur, comprendre les tâches qui attendent l'administrateur du système est toujours une bonne idée.

Être l'administrateur système ne fait pas de vous un expert UNIX. Il y a de nombreux administrateurs qui ont très peu de connaissances d'UNIX. Tout comme il y a beaucoup d'utilisateurs « normaux » qui connaissent beaucoup mieux UNIX que leur administrateur système. Et encore une fois, vos privilèges ne vous autorisent pas à léser vos utilisateurs. Ce n'est pas parce que vous avez la possibilité de détruire tous leurs fichiers que vous avez le droit de le faire.

Enfin, être l'administrateur système n'a rien d'extraordinaire. Que votre machine soit un petit 386 ou un Cray n'a aucune importance, la situation est la même. Connaître le mot de passe de `root` ne vous apportera jamais argent et célébrité. Cela vous permettra simplement de faire tourner le système correctement. C'est tout.

4.2 Amorçage du système

On peut amorcer le système de plusieurs façons, soit depuis une disquette, soit depuis le disque dur.

4.2.1 Utilisation d'une disquette d'amorce

Beaucoup d'utilisateurs lancent le système depuis une disquette d'amorce (boot floppy) qui contient une copie du noyau de Linux. Ce noyau contient à un endroit bien déterminé, deux octets « magiques » qui lui permettent de savoir sur quel périphérique aller chercher le système de fichiers racine. (La commande **rdev** peut être utilisée pour ajuster cette valeur directement dans cette l'image du noyau; voir plus loin). C'est le type de disquettes créé par certaines procédures d'installation de Linux, par exemple.

Pour créer votre propre disquette d'amorce, cherchez tout d'abord l'image de votre noyau sur le disque dur. Elle devrait normalement se trouver sous le nom de /usr/src/linux/zImage. Toutefois si les sources du système ne sont pas installées, vous la trouverez selon la distribution dans /zImage, ou /vmlinux, ou /vmlinux.z.

Le suffixe « z » indique que le noyau de Linux est compressé. Un noyau compressé se décompacte en mémoire au moment du chargement, et prend beaucoup moins de place disque. C'est aussi une technique permettant de le charger dans la limite fatidique des 640 Ko de mémoire disponible tant que l'ordinateur n'est pas encore passé en mode protégé.

Une fois que vous avez trouvé ce fichier, indiquez le périphérique racine de votre système dans cette image du noyau avec la commande **rdev**. Le format de la commande est:

rdev ⟨*fichier-noyau*⟩ ⟨*périphérique-racine*⟩

où ⟨*fichier-noyau*⟩ est votre fichier contenant l'image du noyau et ⟨*périphérique-racine*⟩ le nom de la partition supportant le système de fichiers racine. Par exemple, s'il est sur **/dev/hda2** et que votre noyau s'appelle zImage, utilisez la commande:

```
# rdev zImage /dev/hda2
```

rdev peut configurer d'autres options également, comme le mode vidéo par défaut. Lancez « **rdev -h** » pour obtenir un message d'aide.

Après cette opération, vous pouvez tout simplement copier ce fichier image sur votre disquette. Avant de copier des données sur une disquette, il peut être pratique de la formater à la norme MS-DOS avant. Cela positionne les informations secteur et pistes, ce qui permet de détecter ensuite si elle est haute ou basse densité.

Par exemple, pour copier le noyau zImage sur la disquette qui est dans `/dev/fd0` (le premier lecteur de disquettes), utilisez la commande:

```
# cp zImage /dev/fd0
```

Cette disquette devrait alors pouvoir amorcer Linux.

4.2.2 Utilisation de LILO

LILO offre une autre méthode pour amorcer Linux, directement depuis le disque dur. C'est un programme qui réside dans le secteur d'amorce de votre disque dur. Il est alors exécuté chaque fois que vous allumez l'ordinateur, et peut automatiquement lancer Linux depuis un noyau stocké sur le disque dur lui-même.

LILO peut aussi être utilisé comme chargeur pour amorcer plusieurs autres systèmes d'exploitation, vous permettant de choisir au démarrage de l'ordinateur celui que vous désirez utiliser (comme Linux ou MS-DOS). Lorsque vous utilisez LILO, le système d'exploitation configuré par défaut est amorcé, sauf si vous pressez `ctrl`, `alt`, ou `shift` pendant la séquence d'amorçage. Si vous appuyez sur l'une de ces touches, vous verrez apparaître une invite, où vous pourrez taper le nom du système désiré (comme « `linux` » ou « `msdos` »). Si vous tapez `tab`, vous verrez s'afficher la liste des différents systèmes d'exploitation disponibles.

LILO est situé dans le répertoire `/sbin`, ou peut être sur des distributions plus anciennes dans `/etc/lilo`, bref vous le trouverez bien quelque part si vous le possédez. La façon la plus simple de l'installer est d'éditer son fichier de configuration, `/etc/lilo.conf`, et de lancer la commande

```
# /sbin/lilo
```

(Dans le cas ou il se trouve dans `/sbin` bien sûr).

Le fichier de configuration de LILO contient une « strophe » pour chaque système d'exploitation qu'il doit prendre en charge. La meilleure explication consiste à donner un exemple d'un tel fichier. Celui qui suit est paramétré pour un système dont la racine est sur `/dev/hda1`, et possède un système MS-DOS sur `/dev/hda2`.

```
# Indique a LILO de modifier le programme d'amorcage sur /dev/hda
# (le premier disque dur non SCSI). Si vous amorcez depuis un disque
# different, changez la ligne suivante.
boot = /dev/hda

# Nom du programme d'amorcage. Ne modifiez pas  cette ligne sauf
# si vous etes developpeur et travaillez a l'amelioration de LILO.
install = /boot/boot.b

# Laisse LILO faire quelques optimisations
compact
```

```
# Strophe pour Linux, partition racine sur /dev/hda1.
image = /etc/Image   # Chemin d'acces au noyau
   label = linux     # Nom du systeme (pour le menu d'amorcage)
   root = /dev/hda1  # Nom de la partition racine
   vga = ask         # Indique au noyau de demander le mode video au demarrage

# Strophe pour MS-DOS sur la partition /dev/hda2.
other = /dev/hda2    # Nom de la partition supportant le systeme
   table = /dev/hda  # Disque supportant la table de partitions pour /dev/hda2
   label = msdos     # Nom du systeme (pour le menu d'amorcage)
```

La première strophe du fichier correspond au système qui doit être lancé par défaut. Vous pourrez choisir les autres comme nous l'avons expliqué plus haut.

Attention: Chaque fois que vous mettez à jour le noyau de Linux sur le disque, vous devez lancer la commande `/sbin/lilo` afin que la modification soit prise en compte. Ceci même si le fichier porte le même nom et est situé au même endroit ! LILO conserve non pas un chemin d'accès, mais une position physique sur le disque et un nombre de blocs à charger. Si ces informations sont erronées, le système ne démarrera pas.

Notez aussi que si vous utilisez la ligne « `root =` » , vous n'avez pas besoin d'utiliser `rdev` pour configurer le noyau, LILO se chargera de passer le bon paramètre.

Vous trouverez beaucoup plus de détails sur LILO dans la « FAQ » Linux (voir annexe A), y compris comment utiliser LILO avec le « boot manager » d'OS/2.

4.3 Arrêter le système

L'arrêt d'un système Linux doit se faire avec soin. Vous ne devez jamais éteindre l'ordinateur ou appuyer sur le bouton de réinitialisation pendant le fonctionnement. Le noyau conserve en mémoire les entrées/sorties disque dur : si vous arrêtez la machine sans qu'il ait eu le temps d'écrire ces données, vous corromprez à coup sûr vos systèmes de fichiers.

La procédure d'arrêt du système comprend également d'autres précautions : un signal est envoyé à tous les processus, qui peuvent alors se terminer proprement, et les systèmes de fichiers sont correctement démontés. Si vous le désirez, il est également possible d'alerter tous les utilisateurs que la machine va être arrêtée afin de leur laisser une chance de se déconnecter proprement et de sauver leurs travaux en cours.

La méthode la plus simple pour effectuer tout cela est d'utiliser la commande `shutdown`. Elle s'utilise ainsi :

`shutdown` ⟨*temps*⟩ ⟨*message-utilisateurs*⟩

L'argument ⟨*temps*⟩ est l'heure d'arrêt désirée (au format *hh:mm:ss*), et ⟨*message-utilisateurs*⟩ est l'avertissement envoyé sur le terminal de chaque utilisateur. Vous pouvez aussi indiquer « `now` » comme paramètre ⟨*temps*⟩, dans ce cas l'opération aura lieu immédiatement. La commande peut prendre l'option `-r` qui lui indique de relancer le système une fois l'arrêt effectué.

Par exemple, pour arrêter puis redémarrer Linux à vingt heures, tapez :

```
# shutdown -r 20:00
```

Vous pouvez aussi utiliser la commande `halt` pour forcer un arrêt immédiat, sans prévenir personne. Cette commande est très utile si vous êtes le seul utilisateur.

◇ N'arrêtez pas l'ordinateur tant que vous n'avez pas vu s'afficher le message:

```
The system is halted
```

Nous répétons qu'il est très important de stopper le système proprement à l'aide de `shutdown` ou `halt`. Sur certaines configuration, la combinaison `ctrl-alt-del` sera reconnue et appellera `shutdown` ; mais sur d'autres elle relancera immédiatement l'ordinateur et provoquera un désastre dans vos fichiers.

4.4 Gérer les utilisateurs

Que vous ayez ou non plusieurs utilisateurs sur votre système, il est important de comprendre certains aspects de la gestion des utilisateurs sous Linux. Même si vous êtes la seule personne à accéder à votre machine, vous avez probablement un compte séparé pour vous-même pour votre utilisation quotidienne, en plus du compte `root` pour les travaux d'administration.

Toute personne utilisant le système doit posséder son propre compte. Partager un même compte entre plusieurs utilisateurs est rarement une bonne idée. Cela pose non seulement un problème de sécurité, mais les comptes utilisateurs sont là pour identifier chacun séparément sur la machine. Vous devez pouvoir être capable de savoir qui fait quoi.

4.4.1 Concepts de base

Le système conserve certaines informations sur chaque utilisateur. Elles sont indiquées ci-dessous.

nom d'utilisateur

C'est l'identificateur unique donné à chaque utilisateur du système. Il peut comprendre des lettres et des chiffres, et les caractères « _ » (souligné) et « . » (point). Il vaut mieux limiter leur taille à 8 caractères, pour des raisons techniques. Pour les mêmes raisons, utilisez toujours des lettres minuscules et ne mettez **jamais** de caractères accentués dedans. Exemples: « `marcel` » , « `rene` » , « `dugenou` » sont des noms d'utilisateurs valides.

identificateur d'utilisateur
Appelé UID, (user ID), c'est un nombre unique donné à chaque utilisateur du système. En effet, c'est par ce numéro que chaque utilisateur est identifié et non pas par son nom, qui est un équivalent pratique pour les humains.

identificateur de groupe
Appelé GID (group ID), c'est un numéro qui représente le groupe dans lequel l'utilisateur se trouve par défaut. Nous avons présenté cette notion de groupe dans la section 3.9 ; chacun appartient à un ou plusieurs groupes définis par l'administrateur système. Nous donnerons plus de détails un peu plus bas.

mot de passe
Le système conserve le mot de passe de chaque utilisateur sous forme cryptée. On peut utiliser la commande `passwd` pour changer ce mot de passe.

nom réel Le « vrai » nom de chacun est également enregistré. Par exemple, l'utilisateur `marcel` peut s'appeler « Marcel Dugenou » dans la vie courante ; le système permet de conserver cette information. Connu également sous le nom de « champ GCOS » , en hommage à un système de chez Bull.

répertoire personnel
C'est le répertoire où l'utilisateur est automatiquement placé lorsqu'il arrive sur la machine. Chaque utilisateur doit posséder son propre répertoire personnel, généralement dans `/home`.

C'est le « login shell » de l'utilisateur, celui qui est lancé lorsque l'utilisateur se connecte sur le système. Ce peut être par exemple `/bin/bash`, `/bin/tcsh`, `/bin/csh`, `/bin/ksh`, ou bien d'autres.

Toutes ces informations sont contenues dans le fichier `/etc/passwd`. Chaque ligne de ce fichier contient les informations d'un utilisateur; le format de chaque ligne est le suivant:

```
nom-utilisateur:mot-de-passe-crypté:UID:GID:nom-réel:répertoire-perso:login-shell
```

Ce qui donne par exemple:

```
marcel:Xv8Q981g71oKK:102:100:Marcel DUGENOU:/home/marcel:/bin/bash
```

Comme vous pouvez le voir, le premier champ, « `marcel` » , est le nom d'utilisateur.

Le champ suivant, « `Xv8Q981g71oKK` » , correspond à son mot de passe crypté. Les mots de passe ne sont pas enregistrés dans un format lisible, mais encryptés avec une clé secrète. En d'autres termes, vous devez connaître ce mot de passe pour le décrypter, il est impossible de le reconstituer à partir de la seule information codée. Cette méthode assure déjà une excellente sécurité.

Quelques systèmes utilisent la technique des « shadow passwords » , où les mots de passe sont déplacés dans un fichier nommé `/etc/shadow`. En effet, `/etc/passwd` est lisible par tout le monde, et bien qu'il soit impossible de décoder les mots de passe, certains pirates munis d'ordinateurs très puissants et d'un dictionnaire électronique peuvent arriver à trouver ceux qui sont trop simples. Le fichier `/etc/shadow` n'est lisible que par `root` et les deux ou trois programmes ayant besoin d'accéder aux mots de passe. La sécurité du système est alors accrue dans de grandes proportions. Les « shadow passwords » offrent aussi d'autres possibilités, comme limiter la durée de validité des comptes, nous n'entrerons pas dans de tels détails dans ce guide ; consultez un ouvrage traitant de la sécurité sous UNIX pour en savoir plus.

Le troisième champ, « `102` » est l'identificateur utilisateur, l'UID. Il doit être unique pour chacune des personnes ayant accès au système. Le quatrième champ, « `100` » , correspond au GID. Cet utilisateur appartient au groupe numéro 100. Les informations sur les groupes, tout comme celles sur les utilisateurs associés, sont enregistrées dans le fichier `/etc/group`. Voyez la section 4.4.5 page 164 pour plus d'informations.

Le cinquième champ contient le nom réel de l'utilisateur, ici « `Marcel DUGENOU` » . Enfin, les deux derniers champs sont respectivement le répertoire personnel (`/home/marcel`) et le shell utilisé (`/bin/bash`). Il n'est pas nécessaire que le répertoire personnel ait le même nom que l'utilisateur ; mais il est plus facile de s'y retrouver ainsi.

4.4.2 Ajouter des utilisateurs

L'ajout manuel d'utilisateurs se fait en plusieurs étapes. Tout d'abord, il faut attribuer une entrée dans `/etc/passwd`, avec un unique numéro d'identification (l'UID). Les GID, noms réels, et les autres informations doivent également être précisés dans cette ligne. Il faut ensuite créer le répertoire personnel de cet utilisateur, et positionner ses permissions d'accès de telle manière que cet utilisateur en soit propriétaire. Il faut installer dans ce répertoire les fichiers d'initialisation du shell, et prendre en considération d'autres choses dépendant du système (par exemple, créer éventuellement une boîte aux lettres).

Bien qu'il ne soit pas difficile d'ajouter des utilisateurs à la main (beaucoup le font), lorsque vous administrez un système comportant beaucoup de monde et des configurations complexes, il est facile d'oublier quelque chose. La meilleure méthode est alors d'utiliser un programme interactif qui vous demande les informations nécessaires et paramètre le système automatiquement. Ce programme peut s'appeler `useradd` , `adduser`, voire `mkuser` ou `newuser` selon la version de votre système. Les pages de manuel de celui dont vous disposez devraient vous guider dans son utilisation, qui en général est évidente.

4.4.3 Supprimer des utilisateurs

Cela peut se faire de la même façon, avec des programmes comme `userdel` ou `deluser` selon votre version de système.

Si vous désirez interdire provisoirement l'accès à la machine à un utilisateur particulier, sans lui supprimer son compte, vous pouvez simplement rajouter une astérisque (« `*` ») devant son mot de passe crypté dans le fichier `/etc/passwd` (ou `/etc/shadow` selon le cas). Par exemple, modifier l'entrée de `marcel` ainsi :

```
marcel:*Xv8Q981g71oKK:102:100:Marcel DUGENOU:/home/marcel:/bin/bash
```

l'empêchera de se connecter, son mot de passe n'étant plus correct.

4.4.4 Modifications des paramètres d'un utilisateur

Après avoir créé un utilisateur, vous pouvez avoir besoin de changer certains de ses attributs, comme son répertoire personnel, ou son mot de passe. Le plus simple est de changer les valeurs directement dans le fichier `/etc/passwd`. En ce qui concerne son mot de passe, utilisez la commande `passwd`. Par exemple,

```
# passwd marcel
```

vous permettra de changer le mot de passe de `marcel`. Seul `root` est habilité à modifier les mots de passe des autres utilisateurs de cette façon, bien entendu. Les autres n'ont accès qu'au leur.

Sur certains systèmes, les commandes `chfn` et `chsh` pourront exister et permettre aux utilisateurs de modifier eux-même leur nom réel et le shell qu'ils désirent utiliser. Si ces commandes n'existent pas, ils devront demander ces modifications à l'administrateur du système.

4.4.5 Les groupes

Comme nous l'avons dit, chaque utilisateur appartient à un ou plusieurs groupes. Ceci n'a d'importance que pour les permissions d'accès aux fichiers ; comme vous l'avez lu dans la section 3.9, chaque fichier appartient à un groupe, et ses droits d'accès de groupe définissent dans quelle mesure les utilisateurs en faisant partie peuvent utiliser ce fichier.

Il existe plusieurs groupes définis pour le système comme `bin`, `mail` et `sys` par exemple. Aucun utilisateur ne doit faire partie de ces groupes là : ils doivent appartenir à des groupes généraux comme `users`. Si vous voulez détailler, vous pouvez aussi créer différentes catégories de groupes comme `etudiant`, `expert`, `amis`, et d'autres encore.

Le fichier `/etc/group` contient les informations relatives aux groupes. Le format de chaque ligne est le suivant :

```
nom-du-groupe:mot-de-passe:GID:autres membres de ce groupe
```

Quelques exemples:

```
root:*:0:
users:*:100:rene,marcel,dugenou,mdw
guest:*:200:
other:*:250:marcel,dugenou
```

Le premier groupe, `root`, est un groupe spécial au système réservé au compte `root`. Le second groupe, `users`, est destiné aux utilisateurs ordinaires. Il a un GID de 100. Les utilisateurs `rene`, `marcel`, `dugenou` et `mdw` ont accès à ce groupe. Souvenez vous que dans `/etc/passwd` chaque utilisateur possède un groupe par défaut. Cependant, tout utilisateur peut faire partie de plusieurs groupes, en ajoutant simplement son nom sur la ligne correspondante dans `/etc/group`. La commande `groups` vous permet d'afficher tous les groupes auxquels vous appartenez.

Le troisième groupe, `guest`, est destiné aux invités et le groupe `other` est pour les « autres » utilisateurs. Marcel et Dugenou ont également accès à ce groupe.

Les mots de passe de groupe sont très rarement utilisés. Cela peut servir à demander un mot de passe pour accéder à un groupe particulier, ce qui peut être quelque fois nécessaire. Pour interdire totalement l'accès aux groupes privilégiés (avec la commande `newgroup`), mettez le caractère « `*` » en guise de mot de passe.

Les commandes `addgroup` ou `groupadd` peuvent être utilisées pour rajouter des groupes à votre système. En général il est beaucoup plus simple d'ajouter ces entrées manuellement dans `/etc/group`, puisque c'est la seule manipulation à faire pour rajouter un groupe. La suppression s'effectue tout simplement en supprimant la ligne adéquate dans ce fichier.

4.5 Archivage et compression de fichiers

Avant que nous puissions aborder les sauvegardes de données, nous devons présenter les outils destinés à réaliser des archives sur les systèmes UNIX.

4.5.1 Utilisation de `tar`

La commande `tar` est la plus utilisée, tous systèmes confondus.

La syntaxe de cette commande est :

`tar` ⟨*options*⟩ ⟨*fichier1*⟩ ⟨*fichier2*⟩ . . . ⟨*fichierN*⟩

où ⟨*options*⟩ est une liste de commandes et d'options pour `tar`, et ⟨*fichier1*⟩ à ⟨*fichierN*⟩ est la liste de fichiers à ajouter ou à extraire.

Par exemple, la commande

```
# tar cvf backup.tar /etc
```

rassemblera tous les fichiers présents dans `/etc` dans l'archive `backup.tar`. Le premier argument de `tar` (ici, « `cvf` ») est la « commande » d'archivage. La lettre « `c` » indique à `tar` de créer une nouvelle archive, le « `v` » passe le programme en mode verbeux (il affiche toute ses actions à l'écran), et la lettre « `f` » indique à `tar` que l'argument suivant (ici, `backup.tar`) est le nom de l'archive à créer. Le reste de la ligne de commande sont les fichiers et/ou répertoires à inclure dans l'archive.

La commande

```
# tar xvf backup.tar
```

extraira tous les fichiers contenus dans `backup.tar` à partir du répertoire courant. C'est quelquefois dangereux, car les anciens fichiers de même nom qui peuvent éventuellement être présents, seront écrasés.

Avant d'extraire le contenu d'un fichier tar, il vaut mieux en connaitre le contenu, pour se rendre compte de ce qui se passera au moment de l'extraction. Par exemples, supposons que vous ayez archivé les fichiers suivants: `/etc/hosts`, `/etc/group`, et `/etc/passwd`. Si vous utilisez la commande

```
# tar cvf backup.tar /etc/hosts /etc/group /etc/passwd
```

le nom complet du répertoire `/etc/` fera partie de l'archive pour chaque fichier. Pour extraire ces fichiers et les voir se retrouver à la bonne place, vous devrez utiliser les commandes :

```
# cd /
# tar xvf backup.tar
```

car les fichiers sont extraits avec leur chemin d'accès; mais par sécurité la plupart des versions de tar correctes suppriment le tout premier « / » , afin d'éviter de contenir des chemins absolus pour des raisons de sécurité évidentes.

Si vous avez réalisé l'archive avec les commandes

```
# cd /etc
# tar cvf backup.tar hosts group passwd
```

le nom du répertoire ne sera pas enregistré. Vous aurez alors besoin de vous déplacer dans « `cd /etc` » avant d'extraire les fichiers. Vous voyez que la manière dont est réalisée l'archive joue un grand rôle ; la commande

```
# tar tvf backup.tar
```

permet d'afficher le contenu de l'archive avant l'extraction. Ainsi vous pouvez tester son contenu et déterminer la meilleure façon (ou la plus sûre) de récupérer les fichiers qu'il contient.

4.5.2 `compress` et `gzip`

Contrairement aux différents utilitaires d'archivage rencontrés sous MS-DOS, la commande `tar` ne compresse pas les fichiers. Par conséquent, si vous sauvegardez 2 mégaoctets de fichiers, l'archive résultante fera 2 mégaoctets elle aussi. Les commandes `compress` ou `gzip` (au choix) peuvent être utilisées pour compacter le fichier (qui peut être quelconque, et pas forcément une archive tar). Par exemple,

```
# gzip backup.tar
```

compactera `backup.tar` et vous donnera le fichier `backup.tar.gz`, d'une taille beaucoup plus réduite. Seule la version compressée est conservée, l'originale est détruite. (Sauf en cas de problème).

Le décompactage se fait par la commande `gunzip`, ou de manière totalement équivalente par « `gzip -d` » .

Le programme `gzip` est un outil relativement nouveau dans la communauté UNIX. Depuis de nombreuses années, la commande standard pour compresser les fichiers était (et est toujours) `compress`. Toutefois, pour différentes raisons,[2] `compress` est en train de tomber en désuétude.[3]

Les fichiers compactés avec `compress` se terminent par l'extension `.Z`. Par exemple, `backup.tar.Z` est la version compactée par compress de `backup.tar`, alors que

2. Ces raisons sont en partie un problème légal sur l'algorithme utilisé dans `compress` et le fait que `gzip` est beaucoup plus efficace.

3. Attention tout de même. La commande standard présente sur tous les UNIX du monde, Linux y compris, est toujours `compress`. Le programme `gzip` est un programme GNU, libre et gratuit. Mais il existe hélas des gens qui payent une fortune des systèmes commerciaux et qui soit ne connaissent pas `gzip`, soit refusent d'utiliser ce qui est gratuit. Si vous ne savez pas où part l'archive que vous réalisez, utilisez toujours `compress` pour être certain d'être compatible avec les plus récalcitrants.

`backup.tar.gz` est la version compressée par `gzip`.[4] La commande `uncompress` sert à décompacter les fichiers traités par `compress`, et ne sait faire que ça, alors que la commande `gunzip` sait automatiquement reconnaître le format et décompacte indifféremment du `gzip` ou du `compress`.

4.5.3 Combiner tout ça

Par conséquent, pour archiver un groupe de fichiers et compresser le résultat, vous pouvez utiliser les commandes :

```
# tar cvf backup.tar /etc
# gzip backup.tar
```

Le résultat sera `backup.tar.gz`. Pour extraire cette archive, utilisez les commandes inverses :

```
# gunzip backup.tar.gz
# tar xvf backup.tar
```

Bien sûr, vérifiez toujours que vous êtes dans le bon répertoire avant d'extraire les fichiers d'une archive.

Vous pouvez utiliser les possibilités d'UNIX pour taper toutes ces commandes en une seule fois, à savoir :

```
# tar cvf - /etc | gzip -9 > backup.tar.gz
```

Ici, nous envoyons le fichier tar vers « `-` » , qui signifie pour `tar` la sortie standard. Un pipe envoie le tout à `gzip`, qui à son tour compresse ce qu'il reçoit et le résultat est sauvé dans `backup.tar.gz`.

L'option `-c` de `gzip` indique au programme d'envoyer sa sortie sur la sortie standard, qui est redirigée vers `backup.tar.gz` ; et l'option 9 demande la compression maximale.

Une commande unique pour extraire et décompresser cette archive serait :

```
# gunzip -c backup.tar.gz | tar xvf -
```

Là encore, `gunzip` décompacte le contenu de `backup.tar.gz` et envoie le résultat (le fichier tar) sur la sortie standard. Celle ci est envoyée par un pipe vers le programme `tar`, qui ici lit « `-` » , qui cette fois signifie son entrée standard.

4. Pour compliquer encore un peu plus les choses, l'extension `.z` (en minuscule) fut utilisée parfois au tout début de la carrière de `gzip`. L'extension officielle est maintenant `.gz` mais vous pourrez encore rencontrer d'anciens fichiers datant de cette époque.

Heureusement, sous Linux nous utilisons la version GNU de la commande `tar`. Comme tout ce qui est GNU, elle offre par rapport à la version standard des autres systèmes un certain nombre de facilités et améliorations. Cette version comprend l'option `-z` qui permet de compacter/décompacter automatiquement les archives en cours de traitement, en utilisant « `gzip` » de manière transparente. (De la même façon, l'option `-Z` en majuscule utiliserait `compress`).

Par exemple, la commande

```
# tar cvfz backup.tar.gz /etc
```

est équivalente à

```
# tar cvf backup.tar /etc
# gzip backup.tar
```

Tout comme la commande

```
# tar xvfz backup.tar.gz
```

peut être utilisée à la place de

```
# gunzip backup.tar.gz
# tar xvf backup.tar
```

Consultez les pages de manuel de `tar` et de `gzip` pour plus d'informations.

4.6 Disquettes et sauvegardes

Les disquettes sont souvent utilisées comme support de sauvegarde, surtout chez les particuliers. Si vous ne possédez pas de lecteur de bandes sur votre machine, vous pouvez réaliser vos sauvegardes sur disquettes (bien que ce soit beaucoup plus long et fastidieux, et bien moins fiable).

Vous pouvez également utiliser des disquettes pour stocker des systèmes de fichiers individuels, puis utiliser la commande **mount** pour monter la disquette et accéder à vos données.

4.6.1 Sauvegarde sur disquettes

La méthode la plus simple pour sauvegarder des données sur disquettes est d'utiliser la commande `tar` :

```
# tar cvfzM /dev/fd0 /
```

Cet exemple réalisera une sauvegarde totale de votre système sur le lecteur de disquettes `/dev/fd0`. L'option « `M` » de la commande `tar` permet de réaliser une archive multivolume; c'est à dire que lorsqu'une disquette sera pleine, `tar` vous en demandera une autre [5]. La commande

```
# tar xvfzM /dev/fd0
```

peut être utilisée pour restaurer une archive complète comme celle-ci. Cette méthode peut aussi être utilisée sur un lecteur de bandes (`/dev/rmt0`) connecté à la machine.

La sauvegarde complète d'un système peut être très longue et consommer beaucoup de ressources. La plupart des administrateurs système utilisent une méthode de sauvegarde incrémentale, dans laquelle ils réalisent une sauvegarde totale une fois par mois seulement, et chaque semaine ne sauvent que les fichiers qui ont été réellement modifiés ces derniers jours. De cette façon, si un gros accident se produit, il suffit de restaurer la totalité de la sauvegarde mensuelle et la dernière sauvegarde hebdomadaire (ou journalière).

La commande `find` peut être très utile pour localiser les fichiers qui ont changé depuis une certaine date. Vous trouverez au gré des serveurs que vous fréquentez de nombreux scripts réalisant des sauvegardes incrémentales.

4.6.2 Systèmes de fichiers sur disquettes

Vous pouvez créer un système de fichiers sur une disquette, exactement de la même façon que sur une partition d'un disque dur. Il n'y a pas de différences, si ce n'est la taille (et le temps d'accès). Par exemple,

```
# mke2fs /dev/fd0 1440
```

crée un système de fichiers (de type ext2) sur `/dev/fd0`. La taille doit correspondre à ce que supporte la disquette; les disquettes haute-densité de 3,5" font 1,44 Mo, soit 1440 blocs. Les disquettes de 5,25" haute densité font 1,2 Mo soit 1200 blocs.

5. Bien qu'alléchante, il faut vous méfier de cette méthode multivolume. Il suffit qu'une seule des disquettes soit mauvaise lorsque vous voulez restaurer l'archive pour que tout ce qui suit soit irrécupérable.Une disquette mauvaise ne veut pas seulement dire abîmée, mais parfois simplement illisible sur un lecteur particulier. Selon l'alignement de ces périphériques, une disquette écrite sur une machine peut ne pas être relue correctement sur une autre; c'est particulièrement vrai par exemple entre les machines Sun et IBM-PC. De plus, cette facilité multivolume n'est offerte que par la version GNU de la commande `tar`. Si vous êtes vraiment obligé de réaliser une telle archive sur disquettes, *vérifiez toujours* avec l'option `t` de `tar` si vous pourrez relire et par conséquent restaurer toutes les disquettes. Cela double le temps de sauvegarde, déjà très long sur des disquettes, mais vous verrez que c'est quasiment indispensable.

Ensuite, pour avoir accès à ce système de fichiers, vous devez le « monter » , en utilisant la commande **mount** :

```
# mount -t ext2 /dev/fd0 /mnt
```

montera le système de fichiers contenu sur la disquette qui est dans `/dev/fd0` dans le répertoire `/mnt`. Maintenant, tous les fichiers présents sur la disquette apparaîtront dans /mnt. L'option « `-t ext2` » spécifie un système de fichiers Linux de type ext2. Si vous avez créé un autre type de système de fichiers, précisez-le de la même manière.

Le répertoire dans lequel vous montez le système de fichiers (ici, `/mnt`) doit exister lorsque vous utilisez `mount`, et doit bien entendu être vide. S'il n'existe pas, créez-le avec la commande `mkdir`.

Voyez la section 4.8 pour plus d'informations sur les systèmes de fichiers et les commandes `mount` et `umount`.

◇ Faites bien attention, toutes les entrées/sorties disques sont traitées sous UNIX, de la même façon sur disquettes et sur disque dur. Lorsque vous copiez un fichier sur la disquette, vous ne verrez probablement pas le lecteur se mettre en route tout de suite, les données ne seront physiquement écrites que lorsque le système le décidera. Par conséquent il est très important de ne jamais retirer une disquette du lecteur sans l'avoir « démontée » à l'aide de la commande `umount` :

```
# umount /dev/fd0
```

Si vous changez les disquettes à tout moment comme vous le faites peut être sous MS-DOS, les plus grands désastres sont à prévoir. Utilisez toujours `umount` si vous avez « monté » un système de fichiers avant d'ôter la disquette du lecteur.

4.7 Mise à jour et installation de nouveaux programmes

L'administrateur système doit aussi parfois installer des mises à jour des programmes ou du système, et de nouvelles applications.

La communauté Linux est très dynamique. Il ne se passe pas quelques semaines sans qu'une nouvelle version du noyau n'apparaisse, et le reste des programmes est mis à jour presque aussi souvent. Les nouveaux utilisateurs de Linux se sentent souvent obligés de suivre le rythme pour avoir un système toujours le plus récent possible, dans tous ses détails. Non seulement ce n'est pas nécessaire, mais c'est une énorme perte de temps: si vous vouliez suivre vraiment tout ce qui sort chaque jour, vous passeriez votre temps à ce jeu et ne pourriez jamais utiliser votre système.

Alors quand? Certains pensent qu'il faut tout réinstaller dès qu'une nouvelle version de leur distribution préférée est disponible, alors que c'est aussi une perte de temps; le téléchargement de 20 ou 30 disquettes alors que probablement seulement 10% des programmes ont légèrement été modifiés est ridicule.

La meilleure façon de faire vos mises à jour est de le faire manuellement, petit à petit, et lorsque c'est réellement utile. Ne changez que ce qui doit être changé. Certains veulent trop souvent savoir quand ils doivent faire une mise à jour, quoi changer, et comment le faire, et ce qui risque de ne plus marcher une fois l'opération terminée. Avec Linux, ce n'est pas la bonne démarche. Vous devez, comme tous les autres, qu'ils soient développeurs ou utilisateurs, y mettre un peu du vôtre. En fait, une fois que vous avez un système installé, qui fonctionne bien et dont vous êtes content, vouloir réinstaller par dessus la toute nouvelle distribution à la mode détruira sûrement une grande partie de vos précieuses configurations, peaufinées au fil des mois. La bonne démarche est de suivre un peu la vie de Linux, de savoir quels sont les parties qui méritent d'être mises à jour, et d'avoir les connaissances de base nécessaires pour le faire vous même, et correctement: ce n'est pas bien difficile.

Vous vous rendrez vite compte que la plupart du temps, changer juste un élément de votre système n'aura pas d'influence sur les autres programmes, qui continueront à fonctionner comme avant. (Il existe bien entendu des exceptions à cette règle). Il n'y a aucune raison de changer un programme dont vous êtes content, juste pour en posséder la toute dernière version. Linux n'est pas MS-DOS ou Microsoft Windows. Si quelque chose de nouveau dont vous avez besoin arrive, installez-le, sinon ce n'est pas forcément la peine.

Les choses les plus importantes à tenir à jour sur votre système sont le noyau, les bibliothèques, et le compilateur `gcc`. Ce sont les trois parties essentielles, et dans certains cas elles dépendent toutes les unes des autres pour fonctionner correctement. La plupart des autres programmes n'ont pas besoin d'être changés très souvent.

4.7.1 Mise à jour du noyau

La mise à jour du noyau est simple, il suffit de récupérer les nouvelles sources et de les recompiler. Vous devez toujours compiler vous-même votre noyau, afin de le configurer correctement pour votre système personnel. La compilation d'un noyau Linux est enfantine.

Les sources des nouvelles versions du noyau sont diffusées bien entendu sur tous les sites proposant Linux, téléchargez les simplement depuis votre site ftp ou BBS préféré. Elles sont fournies sous forme d'archive tar compactée par gzip [6]. Par

6. Très souvent, les nouvelles versions sont simplement un « patch » vous permettant de modifier les sources précédentes pour les amener à la version courante. Si vous ne savez ni programmer ni utiliser cette méthode et le programme `patch`, attendez que la version complète

exemple, le fichier contenant les sources du noyau Linux 1.0 (déjà largement obsolète à l'heure où nous écrivons ces lignes) s'appelle `linux-1.0.tar.gz`.

Décompactez cette archive depuis le répertoire `/usr/src`; cela créera le répertoire `/usr/src/linux` qui contiendra toutes les sources du noyau. Vous devrez auparavant renommer ou effacer le répertoire `/usr/src/linux` contenant votre ancienne version.

Ensuite, vous devez vous assurer que deux liens symboliques très importants dans `/usr/include` sont corrects. Pour les créer, utilisez les commandes

```
# ln -sf /usr/src/linux/include/linux /usr/include/linux
# ln -sf /usr/src/linux/include/asm /usr/include/asm
```

Une fois qu'ils existent, vous n'avez plus besoin de les refaire à chaque fois que vous installerez une nouvelle version des sources (voyez la section 3.10 pour avoir des détails sur les liens symboliques).

Notez bien que pour pouvoir compiler le noyau, vous devez posséder le compilateur `gcc` correctement installé sur votre système. Vous pourrez éventuellement avoir besoin de la toute dernière version de ce compilateur, lisez la section 4.7.3, plus bas, pour en savoir plus.

Pour compiler le noyau, allez tout d'abord dans le répertoire `/usr/src/linux`. Lancez la commande `make config`. Cette commande vous posera des tas de questions concernant la configuration que vous désirez, vous répondrez par « y » ou « n » (oui ou non) selon vos préférences ou besoins. Vous pourrez choisir le type de système de fichiers supportés, les pilotes de périphériques voulus ou nécessaires, bref vous préparer un noyau sur mesures.

Ensuite, éditez le fichier `/usr/src/linux/Makefile`. Assurez-vous que la définition de `ROOT_DEV` est correcte, elle indique le périphérique à utiliser pour la racine lors de l'amorçage du système. La définition habituelle est :

```
ROOT_DEV = CURRENT
```

En principe si c'est bien celle-là, vous n'avez aucune raison de la modifier (sauf si vous savez ce que vous faites).

Puis, tapez la commande `make dep`, ce qui aura pour effet de réaliser une bonne fois les dépendances correctes pour la compilation des sources. C'est une étape très importante, si vous êtes programmeur vous comprendrez aisément pourquoi.

Tout est alors prêt pour la compilation. La commande `make zImage` compilera le noyau et vous trouverez au bout du compte le fichier `zImage`, votre tout nouveau

---

soit diffusée, pour éviter les crises de nerfs... Ou bien, apprenez.

noyau Linux [7]. Selon la version, il se trouvera soit dans `/usr/src/linux`, soit dans le sous-répertoire `arch/i386/boot`.

Vous pouvez alors installer ce nouveau système soit sur une disquette d'amorce, soit utiliser LILO comme expliqué dans la section 4.2.2. Nous vous conseillons de l'essayer tout de même un peu à partir d'une disquette avant de supprimer définitivement votre ancienne version.

Note: La compilation du noyau est une lourde tâche pour le système, consomme beaucoup de temps machine et surtout de mémoire. Il est rare de compiler d'aussi grosses applications. Selon la vitesse de votre machine, et la quantité de mémoire dont vous disposez, cela peut prendre entre 30 minutes et plusieurs heures. Libérez le maximum de mémoire avant de lancer la compilation, vous accélérerez le processus d'autant.

4.7.2 Mise à jour des bibliothèques

L'essentiel des programmes sous Linux utilisent les bibliothèques partagées, qui contiennent les routines les plus courantes afin d'économiser mémoire et place disque, nous l'avons déjà expliqué.

Si vous voyez un message ressemblant de près ou de loin à:

```
Incompatible library version
```

lorsque vous tentez de lancer un programme, c'est que vous devez mettre à jour la version des bibliothèques partagées, celles que vous avez sont trop anciennes. Ces bibliothèques assurent la compatibilité pour les anciens programmes; un programme compilé pour utiliser une vieille version tournera parfaitement avec les nouvelles, mais l'inverse n'est pas vrai (il peut fonctionner quand même mais ne comptez pas trop dessus).

Les versions à jour des bibliothèques partagées se trouvent sur tous les sites diffusant Linux, tout comme les sources du noyau et le reste du système. Généralement le répertoire où elles se trouvent s'appelle « GCC » (elles sont très liées au compilateur), cherchez « /pub/linux/packages/GCC/image.... » ou quelque chose d'approchant. En bref, vous devez télécharger les fichiers `image-`*`version`*`.tar.gz` et `inc-`*`version`*`.tar.gz`, où `version` est la version des bibliothèques à installer, comme par exemple `4.5.26`. Ce sont des archives tar compactées par gzip; le fichier `image` contient les images des bibliothèques à installer dans `/lib` et `/usr/lib`, tandis que le fichier `inc` contient les fichiers qui vont dans `/usr/include`.

Lisez absolument le fichier `release-`*`version`*`.tar.gz`, il contient toutes les explications nécessaire à l'installation et n'en ratez pas une ligne: certaines grosses

7. Ne vous affolez pas si vous voyez apparaître des messages de « Warning » ici ou là. Ce sont souvent des détails mineurs signalés par le compilateur qui est positionné sur l'option la plus verbeuse. Il est rare que soient diffusées des sources du noyau comportant de gros problèmes; seuls les messages « Error » qui arrêtent la compilation sont graves.

anomalies de fonctionnement constatées par un grand nombre d'utilisateurs récemment, étaient dues au fait qu'ils avaient oublié de faire une manipulation qui était pourtant bien expliquée dans cette notice. En général, les fichiers *.a et *.sa s'installent dans `/usr/lib`, ce sont les bibliothèques nécessaires pour réaliser les compilations des programmes. Les fichiers `libc.so.`*version* sont installés dans `/lib`. ce sont les images des bibliothèques partagées qui sont chargées au lancement des programmes qui les utilisent. Chaque bibliothèque possède un lien symbolique utilisant le numéro de version principal, c'est sous ce nom là que les programmes cherchent ces fichiers.

Par exemple, la version 4.5.26 de la `libc` a un numéro de version de `4`. Le fichier contenant la bibliothèque est `libc.so.4.5.26`. Un lien symbolique du nom de `libc.so.4` est créé dans `/lib`, pointant sur ce fichier. Vous devez éventuellement vérifier et changer manuellement ce lien si l'installation s'est mal passée.

En réalité, ce ne sont pas les programmes qui cherchent les bibliothèques partagées, mais un programme « éditeur de liens dynamiques » , qui s'appelle `/lib/ld.so`[8]. Cet éditeur de lien doit être configuré par la commande `ldconfig` chaque fois que vous changez de version de bibliothèques partagées, c'est lui qui va automatiquement créer les bons liens symboliques, pour toutes les bibliothèques qu'il trouvera sur le système. En général tout se passe très bien et il n'y a rien à retoucher à la main. Mais, dans le pire des cas, lisez ce qui suit.

◊ Si vous devez corriger à la main les liens symboliques sur les bibliothèques partagées, il est très important de le faire en une seule étape, comme nous vous le montrons ci-dessous. Si vous supprimez le lien symbolique `libc.so.4`, plus aucun programme qui en dépend ne fonctionnera (y compris les commandes de base comme `ls` et `cat`). Utilisez la commande suivante pour mettre à jour le lien symbolique `libc.so.4`, afin qu'il pointe par exemple sur `libc.so.4.5.26`:

```
# ln -sf /lib/libc.so.4.5.26 /lib/libc.so.4
```

Vous devrez aussi faire la même manipulation sur le lien `libm.so.`*version* (Pour en savoir plus sur les liens symboliques, lisez la section 3.10). Mais encore une fois, la commande `ldconfig` devrait faire tout cela pour vous sans se tromper. Elle se trouve probablement dans `/etc` ou dans `/sbin`, selon votre distribution de Linux.

8. Si votre système Linux ne comporte pas `/lib/ld.so`, c'est qu'il est vraiment très ancien, c'est probablement une vieille version Alpha test de Linux. Mettez vous à jour, ces versions n'ont plus lieu d'être depuis longtemps. Si votre système est très récent et que ce fichier n'est pas là, c'est peut être aussi qu'il a été remplacé par une nouvelle méthode révolutionnaire et que ce livre est dépassé depuis des années, sait on jamais!

4.7.3 Mise à jour de gcc

Le compilateur C et C++ `gcc` est le compilateur standard [9] sous Linux, qui vous servira à écrire tous les programmes qui vous passeront par la tête, et également compiler les applications à installer. Il est bien entendu utilisé pour compiler le noyau, et souvent les nouvelles versions du noyau dépendent étroitement de la qualité du compilateur, aussi faut-il souvent utiliser la dernière version de `gcc` pour compiler proprement le dernier noyau Linux en date. (Ce n'est pas une règle bien sûr). Les nouvelles versions de `gcc` pour Linux sont disponibles comme tout le reste de Linux sur les sites diffusant le système, souvent dans un répertoire nommé `/pub/Linux/GCC` ou quelque chose d'approchant. Vous pouvez bien sûr également télécharger les sources originales GNU et compiler vous même le programme, néanmoins les archives binaires des portages Linux sont plus rapides à installer et sont sans surprises. Lisez la notice d'installation qui est toujours très détaillée.

4.7.4 Mise à jour des autres programmes

La mise à jour du reste du système se borne en règle générale au téléchargement de la nouvelle version, et de l'installer directement s'il s'agit de binaires, ou de les compiler au préalable dans le cas des sources. La compilation est une affaire de routine, il suffit dans la plupart des cas de taper `make` et tout se passe bien. L'essentiel des programmes composant le système est disponible sur les sites Linux sous forme d'archives tar compressées avec gzip, rassemblant une section particulière. Si par hasard un de ces paquetages n'était pas disponible, cherchez-le dans les sources originales GNU, les programmes et utilitaires GNU se compilant parfaitement sous Linux, il arrive que personne ne prenne la peine de préparer une archive spéciale; dans ce cas vous vous trouvez dans le même cas que le reste du monde, il vous suffit d'installer l'archive GNU, comme les autres.

La lecture du forum Usenet `comp.os.linux.announce` est la meilleure façon de se tenir au courant des nouvelles versions ou des nouveaux programmes disponibles pour Linux. Chaque fois que vous recherchez un programme sur un site FTP, une bonne méthode pour localiser un programme est de télécharger l'index `ls-lR` de ce site et d'utiliser la commande `grep` pour trouver les fichiers en question. Si vous disposez d' `archie`, il peut vous être également très utile [10]. Voir l'annexe A pour plus de détails.

Les archives de certaines distributions binaires sont une source pratique de programmes prêts à installer. Souvent, les disquettes contiennent des fichiers `.tar.gz` ou `.tgz` (qui est la même chose en abrégé pour tenir sur des disquettes MS-DOS); il vous suffit de télécharger non pas les disquettes entières mais choisir parmi les

9. Ne vous posez plus la question : Sous UNIX le compilateur C standard s'appelle `cc`, et les développeurs préfèrent souvent installer et utiliser `gcc` pour de nombreuses raisons. Sous Linux, `gcc` étant le compilateur natif, les commandes `cc` et `gcc` sont strictement équivalentes.

10. Si vous n'avez pas `archie` , vous pouvez faire un telnet sur un serveur archie comme `archie.rutgers.edu`, connectez-vous sous « `archie` » et utilisez la commande « `help` » .

archives qui les composent, les applications dont vous avez besoin. Faites attention toutefois: selon la distribution, les programmes peuvent être compilés avec des options incompatible avec celles nécessaire pour votre propre système, ce cas est toutefois assez rare.

Encore une fois, répétons qu'il est inutile et dangereux de vouloir changer tout le système. La mise à jour doit se faire programme par programme, petit à petit, lorsque c'est nécessaire. Vous n'achetez sûrement pas un nouveau rasoir chaque fois qu'il sort un nouveau modèle? Et bien sous Linux, c'est pareil...

4.8 Gestion des systèmes de fichiers

L'administrateur système doit entretenir les systèmes de fichiers. L'essentiel de cette tâche consiste à les tester régulièrement pour détecter toute anomalie; beaucoup de configurations font cela automatiquement lors du démarrage du système.

4.8.1 Montage des systèmes de fichiers

D'abord, quelques notions élémentaires. Pour qu'un système de fichiers soit accessible, il doit être **monté** sur un répertoire. Par exemple, si vous avez installé un système de fichiers sur une disquette, vous devez le monter dans un répertoire vide quelconque, disons `/mnt`, afin de pouvoir accéder aux fichiers qui s'y trouvent (voir section 4.6.2). Ensuite, tous les fichiers ou répertoires présents apparaîtront dans le répertoire `/mnt`. Après avoir démonté la disquette, le répertoire `/mnt` sera à nouveau vide.

C'est aussi valable pour les disques durs. Le système monte automatiquement les systèmes de fichiers de votre disque dur lors de l'amorçage. Celui que l'on appelle « racine » est monté sur `/`, si vous avez une partition séparée pour `/usr` par exemple, elle est montée dans le répertoire `/usr`, et ainsi de suite.

On utilise la commande `mount` pour monter un système de fichiers. La commande

```
mount -av
```

est exécutée lors du démarrage du système par le fichier d'initialisation `/etc/rc` (voir section 4.10.1). Cette commande `mount -av` récupère les informations concernant les systèmes de fichiers et points de montage depuis le fichier le `/etc/fstab`. Voici un exemple d'un tel fichier :

```
# peripherique  repertoire   type   options   freq pass
/dev/hda2       /            ext2   defaults  1    1
/dev/hda3       /usr         ext2   defaults  1    2
/dev/hda4       none         swap   sw        0    0
/proc           /proc        proc   none      0    0
```

Le premier champ est le nom de la partition à monter. Le second est le point de montage, le répertoire dans lequel cette partition doit être montée. Le troisième champ indique le type du système de fichiers qui est sur cette partition, comme par exemple **`ext2`** (pour ext2fs), **`minix`** (pour Minix) voire **`msdos`** (dans le cas d'une partition MS-DOS). Le tableau 4.1 liste les différents types de systèmes de fichiers reconnus par Linux[11]. Ils ne seront sans doute pas tous disponibles sur votre système ; il faut que votre noyau ait été compilé en intégrant le support de chacun d'eux. Consultez la section 4.7 pour savoir comment construire votre propre noyau.

Système de fichiers	Identification	Commentaires
Étendu, version 2	`ext2`	Le plus courant sous Linux, très performant.
Étendu	`ext`	Obsolète, remplacé par ext2.
Minix	`minix`	Le tout premier, celui de minix; peu utilisé.
Xia	`xia`	Un peu mieux que *minix*, rarement utilisé
UMSDOS	`umsdos`	Permet d'avoir Linux sur une partition DOS.
MS-DOS	`msdos`	Celui de MS-DOS, pour accéder à ces fichiers.
HPFS	`hpfs`	Partitions HPFS, en lecture seulement.
`/proc`	`proc`	Syst. de fichiers virtuel, pour infos noyau.
ISO 9660	`iso9660`	La plupart des CD-ROM.
Xenix	`xenix`	Celui du système d'exploitation Xenix.
System V	`sysv`	Accès aux fichiers de différents System V.
Coherent	`coherent`	Celui du système d'exploitation Coherent.

TAB. 4.1 - *Les différents types de systèmes de fichiers supportés par Linux*

Le champ **`options`** est destiné à indiquer les options à passer à la commande **`mount`**, généralement il est positionné sur « **`defaults`** » .

Les deux derniers champs (**`freq`** et **`pass`**) sont facultatifs, et ne seront pour l'instant utilisés que par le programme **`e2fsck`**, destiné à la vérification et aux réparations éventuelles des systèmes de fichiers de type **`ext2`**. Consultez les pages de manuels et les documentations correspondantes en cas de besoin.

Vous pouvez voir que les partitions de swap sont aussi indiquées dans **`/etc/fstab`**. Le fichier de démarrage **`/etc/rc`** lance la commande **`swapon -a`**, qui met en service automatiquement toutes les partitions de swap indiquées dans **`/etc/fstab`**.

Le fichier **`fstab`** contient une entrée spéciale, pour le système de fichiers **`/proc`**. Comme nous l'avons vu section 3.11.1, **`/proc`** sert à stocker des informations sur les différents processus et autres paramètres du système. Si **`/proc`** n'est pas monté, certaines commandes comme **`ps`** ne fonctionneront pas.

◇ La commande **`mount`** ne peut être utilisée que par **`root`**, pour des raisons de sécurité ; il serait très dangereux que les utilisateurs normaux puissent monter ou démonter des partitions à tout moment. Il existe différents utilitaires en domaine

11. Il s'agit de ceux gérés à l'heure où nous écrivons ces lignes, où la version du noyau est 1.1.93

public qui peuvent permettre d'autoriser les utilisateurs à monter des systèmes de fichiers (des disquettes par exemple) en toute sécurité.

La commande `mount -av` monte automatiquement toutes les partitions autres que la racine (dans l'exemple ci-dessusn `/dev/hda2`) qui sont citées dans `/etc/fstab`. La racine est montée au moment de l'amorçage de Linux, directement par le noyau.

Au lieu d'utiliser `mount -av`, vous pouvez monter un système de fichiers à la main. La commande

```
# mount -t ext2 /dev/hda3 /usr
```

revient à monter le système de fichiers de `/dev/hda3` dans l'exemple de `fstab` ci-dessus.

En général, vous ne devriez pas avoir besoin de monter ou démonter les partitions manuellement (sauf dans le cas des disquettes). La commande `mount -av` lancée par `/etc/rc` s'occupe de tout lors du démarrage du système. Les partitions sont automatiquement démontées par les commandes `shutdown` ou `halt` avant l'arrêt de la machine.

4.8.2 Vérification des systèmes de fichiers

Il est conseillé de vérifier régulièrement l'intégrité de vos partitions, afin de vous assurer qu'aucun fichier n'est corrompu. Quelques systèmes le font automatiquement au démarrage (avec la commande appropriée dans `/etc/rc`).

La commande utilisée pour cette opération dépend du type du système de fichiers en question. Pour le type ext2fs (le plus courant), cette commande est `e2fsck`. Par exemple, la commande

```
# e2fsck -av /dev/hda2
```

testera le système de fichiers ext2fs sur la partition `/dev/hda2` et corrigera automatiquement toute erreur rencontrée.

Il faut impérativement démonter un système de fichiers avant de le tester. La commande

```
# umount /dev/hda2
```

démontera la partition `/dev/hda2`, après quoi vous pourrez la tester et la réparer en toute sécurité. La seule exception concerne la partition racine, qu'il est impossible de démonter. La solution employée par la plupart des distributions binaires, est d'indiquer au noyau lors du démarrage de la monter en lecture seule (read-only); une fois cette partition testée, et éventuellement réparée, le fichier `/etc/rc` appelle la commande `mount` avec une option permettant de la remonter, cette fois en lecture et écriture. Si vous voulez vérifier cette partition racine à la main, vous n'avez d'autre solution que d'utiliser une disquette de maintenance (voir 4.11.1).

Il est impossible de démonter une partition lorsque l'un de ses fichiers est « occupé » , c'est à dire utilisé par un quelconque processus. Par exemple, si le répertoire courant d'un utilisateur est sur ce système de fichiers, vous aurez le message d'erreur « `Device busy` » tant qu'il sera là.

Les autres types de systèmes de fichiers nécessitent leur **`fsck`** particulier, comme **`xfsck`** pour **`xiafs`**. Certaines installations sont faites de telle manière que vous pouvez vous contenter d'utiliser la commande **`fsck`**, qui déterminera automatiquement le type et exécutera le programme approprié.

◇ Il est important de relancer le système si fsck a détecté et corrigé des erreurs sur la racine, sinon les informations présentes en mémoire ne correspondraient plus à la réalité.

Bien sûr, le système de fichiers `/proc` n'a jamais besoin d'être vérifié. Il est purement virtuel, et géré directement en mémoire par le noyau.

4.9 Utilisation d'un fichier de swap

Au lieu de réserver une petite partition individuelle pour la zone de swap, vous pouvez utiliser un simple fichier. Toutefois vous devrez avoir installé et configuré tout le système *avant* de pouvoir créer ce fichier.

Si vous êtes dans ce cas, vous pouvez utiliser les commandes qui vont suivre pour créer un fichier de swap. Nous allons ci-dessous initialiser un fichier de swap de 8208 blocs (environ 8 Mégaoctets).

```
# dd if=/dev/zero of=/swap bs=1024 count=8208
```

Cette commande crée le fichier. Remplacer la valeur du paramètre « **`count=`** » par la taille (en blocs) dont vous avez besoin, et le nom associé à « **`of=`** » par celui du fichier désiré (avec son chemin d'accès).

```
# mkswap /swap 8208
```

Ici nous avons initialisé le fichier ; remplacez également le nom et la taille du fichier par les valeurs appropriées.

```
# sync
# swapon /swap
```

Nous venons d'activer cette zone de swap, après avoir utilisé la commande *sync* pour être assuré que le fichier a bien été écrit physiquement sur le disque.

L'inconvénient majeur de tels fichiers est que tous les accès à cette zone de swap sont faits par l'intermédiaire de la gestion des systèmes de fichiers, ce qui implique que les blocs constituant cette zone peuvent ne pas être contigus sur le disque. Par conséquent, les performances peuvent être inférieures à celles que l'on obtiendrait avec une partition dédiée, sur laquelle cette zone est continue et dont les accès sont réalisés directement sur le périphérique.

Ces fichiers de swap peuvent rendre de grands services lorsque vous avez momentanément besoin d'une large zone de swap. Par exemple, si vous compilez un trés gros programme et venez à manquer de mémoire virtuelle, vous pouvez créer et activer pour l'occasion un tel fichier et l'utiliser en plus de votre partition de swap habituelle.

Pour supprimer un fichier de swap, utilisez tout d'abord la commande `swapoff` pour le désactiver, comme par exemple :

```
# swapoff /swap
```

Puis vous pouvez effacer le fichier en toute sécurité :

```
# rm /swap
```

Notez bien que chaque zone de swap (fichier ou partition) ne peut dépasser 16 Mégaoctets, mais que vous pouvez en utiliser jusqu'à 8 simultanément, ce qui porte tout de même la mémoire virtuelle possible à 128 Mo.

4.10 Divers

Croyez-le ou non, il y a un certain nombre de tâches diverses qui attendent l'administrateur système, ne pouvant être classées dans aucune catégorie.

4.10.1 Fichiers d'initialisation système

Lorsque le système démarre, il exécute automatiquement un certain nombre de scripts avant de pouvoir accueillir des utilisateurs. Voici une description de ce qui se passe.

Après l'amorçage, le noyau exécute le programme `/etc/init`. Il s'agit d'un programme qui lit son fichier de configuration, `/etc/inittab`, et lance d'autres processus en fonction du contenu de ce fichier. L'un des plus importants, c' est `getty`, qui est lancé sur chaque console virtuelle. Le programme `getty` initialise cette console et gère la connexion des utilisateurs en leur proposant l'invite « login » . C'est ce qui vous permet de vous connecter sur chaque console virtuelle; si `/etc/inittab` ne prévoit pas un `getty` sur une certaine console, celle-ci vous sera inaccessible. Le processus `init` tourne en permanence, et aussitôt qu'un utilisateur se déconnectera, il relancera automatiquement `getty` sur le terminal correspondant.

Le script principal de démarrage se nomme `/etc/rc`; c'est un simple shell-script qui exécute toute commande nécessaire à l'initialisation du système, comme monter les partitions (voir 4.8) ou lancer le swap.

Votre système peut exécuter également d'autres scripts d'initialisation, comme `/etc/rc.local`, qui contient généralement des commandes spécifiques à votre propre machine; il peut être lancé depuis `/etc/rc` ou bien directement du fichier `/etc/inittab`.

Selon votre type de configuration, les commandes nécessaires au démarrage peuvent être scindées en de multiples shell-scripts, lancés depuis `/etc/rc`, et se trouvant dans un sous-répertoire. Cela permet de classifier les différentes actions. Ces scripts se trouvent souvent dans un répertoire nommé `/etc/rc.d`, vous pourrez par exemple trouver pour le lancement des programmes réseau des fichiers comme `/etc/rc.d/rc.inet1` et `/etc/rc.d/rc.inet2`.

4.10.2 Baptiser la machine

Dans un environnement réseau, afin de pouvoir identifier de manière unique les différentes machines, chacune se voit attribuer un nom: le *hostname*. C'est exactement comme un animal domestique: vous pouvez toujours appeler votre chien "le chien", mais il est beaucoup plus élégant de le baptiser d'un nom ridicule. Donner un nom à votre machine, lorsque vous n'êtes pas en réseau, n'est pas uniquement une affaire d'originalité, un certain nombre de programmes ont besoin de ce paramètre.

Initialiser cette variable *hostname* consiste simplement à utiliser la commande du même nom. Si vous êtes en réseau, vous devez soit donner le nom complet (avec le domaine) de la machine, soit utiliser conjointement la commande *domainname*. Comme cela dépend de la manière dont votre système et le réseau sont configurés (utilisation ou non de NIS), nous n'entrerons pas dans les détails. Nous considérerons simplement que votre machine est isolée, ou bien juste sur un réseau local, qui n'est pas relié au reste du monde.

Amettons que vous ayez décidé d'appeler votre système « ouzo » . Vous taperez la commande :

```
# hostname ouzo
```

Vérifiez dans votre page de manuel la syntaxe de la commande `hostname` dont vous disposez. Il en circule différentes versions.

Pour que votre machine soit réellement baptisée, vous devez indiquer ce nom dans le fichier `/etc/hosts`, et lui assigner une adresse IP.

Si par exemple, vous n'êtes pas connecté à un réseau TCP/IP, mettez simplement cette ligne dans `/etc/hosts` :

```
127.0.0.1       ouzo localhost
```

Ceci assigne à votre machine, l'adresse de *loopback* 127.0.0.1, le nom `localhost` étant un alias universellement utilisé pour cette adresse très particulière.

Si vous êtes sur un réseau TCP/IP, votre adresse IP doit apparaître dans le fichier `/etc/hosts`. Un exemple arbitraire :

```
127.0.0.1       localhost
128.253.154.32       ouzo
```

En général, la commande `hostname` est exécutée automatiquement depuis l'un des scripts d'initialisation du système, comme `/etc/rc`, `/etc/rc.local`, ou l'un de ceux présents dans le répertoire `/etc/rc.d`. Dans ce cas il vous suffit de trouver dans quel fichier elle est appelée et de remplacer le nom qui est mis par défaut par celui que vous avez attribué à votre machine, puis vous relancerez le système.

4.11 Que faire en cas d'urgence

Il arrive que l'administrateur système doive tenter de réparer des catastrophes, comme la perte du mot de passe root, ou des systèmes de fichiers détériorés. Le meilleur conseil que l'on puisse donner dans une telle situation : *restez calme.* Tout le monde fait des erreurs stupides, c'est le métier qui rentre.

Linux est un système très stable. En fait, l'auteur a eu beaucoup moins de plantages ou de « Kernel Panic » sous Linux qu'avec d'autres versions d'UNIX commerciales, sur différentes architectures. Linux a aussi l'avantage d'avoir un nombre non négligeable de spécialistes joignables sur Usenet, pouvant être d'une aide appréciable.

La première chose à faire lorsque vous êtes face à un problème, c'est de tenter de le résoudre tout seul. Essayez de comprendre ce qui se passe, aidez-vous des documentations, voire des sources. Trop souvent, des personnes postent des messages désespérés avant même d'avoir tenté de résoudre le problème. La plupart du temps vous vous rendrez compte que la solution est simple, et que la trouver vous même n'est pas compliqué. C'est également un bon moyen de progresser.

Il y a très peu de cas où la réinstallation complète du système soit nécessaire. Beaucoup de nouveaux utilisateurs, dès qu'ils ont effacé par erreur quelques fichiers essentiels, se ruent sur leurs disquettes d'installation et recommencent tout. Ce n'est pas une bonne idée. Avant d'en arriver aux solutions extrêmes, réfléchissez [12] au problème et demandez éventuellement un peu d'aide. Dans la plupart des cas, vous pourrez réparer votre erreur depuis une disquette de maintenance.

12. Surtout avec la tête.

4.11.1 Utilisation d'une disquette de maintenance

La disquette de maintenance est un outil indispensable pour tout administrateur Linux. Il s'agit de ce qu'on appelle la disquette « boot/root »[13], qui permet de faire tourner un système Linux réduit entièrement sur disquette, indépendant de tout disque dur. Créer une telle disquette est extrêmement simple : vous réalisez un système de fichiers tenant sur un tel support, sur lequel vous mettez tous les utilitaires dont vous pourriez avoir besoin, en essayant de caser tout ça dans 1.44 Mo, sans oublier d'y mettre un noyau Linux. Vous installez LILO dessus, et voilà. Il y a une autre technique, permettant d'avoir encore plus de place, qui consiste à utiliser deux disquettes : une d'amorçage, qui ne contient que le noyau, et une seconde contenant la racine, qui sera insérée à la place lorsque le noyau sera chargé. Dans les deux cas, le résultat est le même: vous faites tourner Linux entièrement sur disquette.

L'exemple typique d'un tel support est la disquette d'installation incluse dans la plupart des distributions de Linux. Vous pouvez l'utiliser comme base de départ et l'adapter à vos besoins.

L'utilisation d'un tel système est très simple: amorcez l'ordinateur depuis cette disquette, connectez-vous en tant que `root`. Pour accéder aux fichiers de votre disque dur, vous devrez le monter manuellement. Par exemple la commande

```
# mount -t ext2 /dev/hda2 /mnt
```

montera un système de fichiers de type ext2fs présent sur la partition `/dev/hda2` sous le répertoire `/mnt`. Par conséquent, votre fichier `/etc/passwd` habituel est maintenant `/mnt/etc/passwd`, et vous pouvez corriger tout ce que vous désirez dedans.

4.11.2 Corriger le mot de passe de root

Si vous avez oublié le mot de passe de root, aucun problème. Lancez le système sur la disquette de maintenance, montez comme nous venons de l'expliquer votre disque dur sur `/mnt`, et nettoyez le champ du mot de passe de `root` dans `/mnt/etc/password`, comme ceci:

```
root::0:0:root:/:/bin/sh
```

Maintenant, `root` n'a plus de mot de passe, lorsque vous relancerez le système depuis le disque dur, vous pourrez entrer sous le compte `root` et en remettre un nouveau à l'aide de la commande `passwd`.

13. Disquette bite et couteau, en langage populaire.

Vous voyez qu'apprendre à utiliser l'éditeur `vi` peut rendre d'énormes services ! Sur une disquette de maintenance, il est impossible de faire tenir de gros programmes, pratiquement seul `vi` sera utilisable.

4.11.3 Corriger un système de fichiers endommagé

Si pour une raison quelconque vous devez tenter de réparer un système de fichiers endommagé, vous pouvez depuis votre disquette de maintenance, lancer la commande `e2fsck` (s'il est du type ext2 bien sûr, voyez la section 4.8 pour les autres types), afin de réparer les erreurs dans la mesure du possible. Lors de cette opération, vous ne devez pas monter les partitions à réparer.

L'une des causes les plus courantes de corruption de systèmes de fichiers, est la corruption du superbloc. Le *superblock* est l'« en-tête » du système de fichiers, il contient toutes les informations sur sa taille, les blocs libres, et bien d'autres données générales. Si cette partie est en mauvais état, le système pourra ne pas reconnaître du tout le système de fichiers en question. Toute tentative de montage de cette partition échouera, et `e2fsck` ne saura pas corriger le problème.

Heureusement, le système de fichiers *ext2fs* enregistre des copies de ce superbloc sur le disque, (généralement, tous les 8 Ko). Vous pouvez demander à `e2fsck` d'utiliser une copie du superbloc avec une commande comme

```
# e2fsck -b 8193 ⟨partition⟩
```

où ⟨*partition*⟩ est la partition sur laquelle le système de fichiers endommagé réside. L'option `-b 8193` indique à `e2fsck` d'utiliser la copie du superbloc située au bloc 8 193 de ce système de fichiers.

4.11.4 Récupérer des fichiers

Si vous avez accidentellement effacé des fichiers importants, il n'y a aucun moyen de les récupérer. Néanmoins, s'il ne s'agit que de fichiers courants, vous pouvez copier ceux de la disquette. Si par exemple vous avez effacé `/bin/login` (ce qui empêche tout accès au système), amorcez le système sur la disquette de maintenance, montez votre disque dur sur `/mnt`, et copiez le `/bin/login` qu'elle contient :

```
# cp -a /bin/login /mnt/bin/login
```

L'option `-a` demande à `cp` de conserver les permissions originales du fichier, ce qui vous évite d'avoir à rechercher quelles sont les bonnes et d'avoir à utiliser `chmod` ensuite.

Bien sûr, si les fichiers perdus ne sont pas des choses essentielles qui se trouvent sur la disquette de maintenance, vous ne pouvez rien. Dans ce cas, il reste à espérer que vous ayez de bonnes sauvegardes.

4.11.5 Corriger une erreur sur les bibliothèques

Si vous avez accidentellement supprimé les liens symboliques sur les bibliothèques partagées dans `/lib`, votre système ne doit pratiquement plus être utilisable. La solution la plus simple est de procéder comme ci-dessus, et de corriger manuellement la situation (voir section 4.7.2), vos bibliothèques étant dans ce cas dans `/mnt/lib`.

Chapitre 5

Techniques avancées

Ce chapitre va vous présenter quelques unes des possibilités avancées de Linux. Cela sous-entend que vous avez un minimum d'expérience UNIX et que vous maîtrisez les notions que nous avons présentées dans les précédents chapitres.

L'un des aspects les plus importants de Linux, qui le distingue d'autres implémentations d'UNIX est son ouverture. Linux n'a pas été développé par une équipe réduite de programmeurs dirigés par des commerciaux n'ayant qu'un seul but en tête. Ce sont au contraire de nombreux développeurs, formant des groupes de spécialistes augmentant de jour en jour, qui ont contribué à l'écriture du système, chacun participant à l'implémentation de ce dont il avait besoin dans un système personnel. Les possibilités de Linux sont vastes et les types de programmes supportés sont très variés. Certains n'aiment pas ce manque de cohérence et d'uniformité, pourtant beaucoup considèrent que c'est l'une des plus grandes qualités de Linux.

5.1 Le Système X Window

Le système X Window est un environnement graphique très puissant (et quelque peu complexe) pour les systèmes UNIX. Le code original de X Window est développé au MIT; les différents constructeurs ont fait de X le standard de l'industrie sur les plate-formes UNIX. Pratiquement toute station de travail dans le monde utilise une version ou un dérivé de X Window.

Un portage de X Version 11, révision 6 (X11R6) pour les systèmes UNIX tournant sur architecture 80386/80486 a été développé par une équipe de programmeurs dirigés par David Wexelblat[1]. Cette version, connue sous le nom de XFree86[2], est disponible pour System V/386, 386BSD, et d'autres Unix pour i386, y compris

1. David peut être joint à l'adresse `dwex@mtgzfs3.att.com`.
2. XFree86 est une marque déposée de *The XFree86 Project, Inc.*

bien entendu Linux. Il contient tout ce qui est nécessaire : binaires, fichiers de configuration, bibliothèques, et outils.

La configuration et l'utilisation de X Window sort largement du cadre de ce livre. Nous vous encourageons à lire *The X Window System User's Guide*, par Valerie Quercia et Tim O'Reilly ; voyez l'annexe A pour les références de cet ouvrage. Ici, nous allons simplement entrevoir les grandes lignes de l'installation et la configuration de X Window pour Linux, mais ce ne sera qu'un survol. Les pages de manuel et les différentes documentations fournies dans la distribution Linux de XFree86 vous seront indispensables si vous voulez approfondir la chose.

Le fichier « Linux XFree86 HOWTO » contient tout le détail nécessaire pour mener à bien l'installation et la configuration de X Window. Tous les lecteurs intéressés devront se procurer ce document.

5.1.1 Configuration matérielle nécessaire

À partir de XFree86 version 3.1, sortie en septembre 1994, les processeurs vidéo suivants sont supportés. La documentation de votre carte d'affichage doit en principe contenir cette information. Si vous avez l'intention d'acheter une nouvelle carte vidéo, ou une machine neuve en étant déjà équipée, demandez-en à votre fournisseur les caractéristiques exactes (constructeur, modèle, processeur). Il devra peut-être entrer en contact avec le support technique du fabriquant, en général les vendeurs sérieux vous rendent ce service avec plaisir. Expliquez que votre système (mentionnez Linux et XFree86) ne supporte pas tous les adaptateurs vidéo du marché et que vous devez obtenir des informations très détaillées.

Vous pouvez aussi déterminer quel type de circuit vidéo équipe votre ordinateur en lançant le programme `SuperProbe` fourni avec la distribution XFree86. Nous verrons cela en détail plus loin.

Les circuits standard SVGA suivants sont supportés :

- Tseng ET3000, ET4000AX, ET4000/W32
- Western Digital/Paradise PVGA1
- Western Digital WD90C00, WD90C10, WD90C11, WD90C24, WD90C30, WD90C31, WD90C33
- Genoa GVGA
- Trident TVGA8800CS, TVGA8900B, TVGA8900C, TVGA8900CL, TVGA-9000, TVGA9000i, TVGA9100B, TVGA9200CX, TVGA9320, TVGA9400CX, TVGA9420
- ATI 18800, 18800-1, 28800-2, 28800-4, 28800-5, 28800-6, 68800-3, 68800-6, 68800AX, 68800LX, 88800

- NCR 77C22, 77C22E, 77C22E+
- Cirrus Logic CLGD5420, CLGD5422, CLGD5424, CLGD5426, CLGD5428, CLGD5429, CLGD5430, CLGD5434, CLGD6205, CLGD6215, CLGD6225, CLGD6235, CLGD6420
- Compaq AVGA
- OAK OTI067, OTI077
- Advance Logic AL2101
- MX MX68000, MX680010
- Video 7/Headland Technologies HT216-32

Les circuits SVGA avec fonctions accélératrices suivants sont supportés :

- 8514/A (et vrais clones)
- ATI Mach8, Mach32
- Cirrus CLGD5420, CLGD5422, CLGD5424, CLGD5426, CLGD5428, CLGD5429, CLGD5430, CLGD5434, CLGD6205, CLGD6215, CLGD6225, CLGD6235
- S3 86C911, 86C924, 86C801, 86C805, 86C805i, 86C928, 86C864, 86C964
- Western Digital WD90C31, WD90C33
- Weitek P9000
- IIT AGX-014, AGX-015, AGX-016
- Tseng ET4000/W32, ET4000/W32i, ET4000/W32p

Les cartes vidéo utilisant ces processeurs sont supportées sur tous types de bus, y compris VLB et PCI.

Tous peuvent être employés en mode 256 couleurs ou monochrome, à l'exception d'Advance Logic, MX et Video 7, qui ne sont supportés qu'en 256 couleurs exclusivement. Si votre carte vidéo possède assez de mémoire, beaucoup peuvent être utilisées dans les modes 16 ou 32 bits par pixel (en particulier, quelques Mach32, P9000, S3 et Cirrus). La configuration habituelle est de 8 bits par pixels (c'est-à-dire 256 couleurs).

Le serveur monochrome supporte aussi les cartes VGA ordinaires, ainsi que les cartes Hercules, Hyundai HGC1280, Sigma LaserView, et Apollo. Sur la Compaq AVGA, seuls 64K de mémoire vidéo sont supportés, et la GVGA n'a pas été testée avec plus de mémoire.

Cette liste n'est bien entendu pas à jour, et est régulièrement étendue. La documentation de la version de XFree86 que vous utilisez vous renseignera bien mieux.

Les développeurs de XFree86 ont parfois des problèmes pour obtenir les informations nécessaires à la programmation des cartes vidéo. Certains constructeurs refusent de fournir ces renseignements, ou bien exigent la signature d'une clause de non divulgation de ces documents. Ceci empêcherait bien évidemment la libre distribution de XFree86, ce que l'équipe de développement n'a pas l'intention de faire. Pendant longtemps, il était par exemple impossible d'utiliser certaines cartes vidéo de marque Diamond ; mais depuis la version 3.1 de XFree86, le constructeur a commencé à travailler avec l'équipe de développement afin d'être en mesure de fournir des pilotes gratuits pour ses cartes.

La configuration typique conseillée pour utiliser XFree86 sous Linux est une machine 486 comportant au moins 8 mégaoctets de RAM, et une carte vidéo parmi celles citées ci-dessus. Pour des performances optimales, nous vous suggérons d'acquérir une carte accélérée, à base de S3 par exemple. Vérifiez la documentation de XFree86 pour savoir si le produit que vous comptez acheter est bien supporté avant de sauter le pas. Des tests de comparaison entre divers adaptateurs vidéo disponibles sur le marché sont postés régulièrement dans le forum Usenet `comp.windows.x.i386unix`.

Pour la petite histoire, la machine personnelle de l'auteur est un 486DX2-66 avec 20 mégaoctets de RAM, et est équipée d'une carte VLB S3-864 comportant 2 mégaoctets de mémoire. Des tests ont été effectués sur cette configuration, et sur des stations de travail Sun Sparc IPX. Le système Linux est environ 7 fois plus rapide que les Sparc IPX (pour les curieux, XFree86-3.1 jauge 171 000 xstones sur cette machine Linux, la Sparc IPX n'en fait que 24 000). En règle générale, XFree86 sous Linux et une carte vidéo accélérée vous donnera de bien meilleures performances que beaucoup de stations de travail sous des systèmes UNIX commerciaux.

Vous aurez besoin d'un minimum de 4 Mégaoctets de RAM, et 16 Mo de mémoire virtuelle (par exemple, 8 Mo de RAM et 8 Mo de swap). N'oubliez pas que la mémoire physique est beaucoup plus rapide ! Un système équipé de seulement 4 Mo de RAM, sera *beaucoup* plus lent (jusqu'à 10 fois) que le même avec 8 Mo. De plus, le swap est trop lent pour compenser le manque de mémoire. Il est vraiment recommandé d'avoir au moins 8 Mo de mémoire physique. Disposer de 16 Mo de RAM offre un réel confort.

5.1.2 Installation de XFree86

La distribution binaire de XFree86 pour Linux peut être téléchargée depuis de nombreux sites diffusant Linux. Sur l'Internet, vous la trouverez par exemple en France sur **`ftp.ibp.fr`**. À l'heure où nous écrivons ces lignes, la version courante est 3.1 ; des mises à jour sont diffusées périodiquement.

Il est possible que vous ayez obtenu XFree86 avec une distribution de Linux, dans ce cas vous n'aurez bien sûr rien à télécharger. Dans le cas contraire, ce tableau donne la liste des fichiers que vous devriez posséder.

L'un des serveurs X suivant est indispensable :

Fichier	Description
`XF86-3.1-8514.tar.gz`	Serveur pour cartes à base de 8514.
`XF86-3.1-AGX.tar.gz`	Serveur pour cartes à base de AGX.
`XF86-3.1-Mach32.tar.gz`	Serveur pour cartes à base de Mach32.
`XF86-3.1-Mach8.tar.gz`	Serveur pour cartes à base de Mach8.
`XF86-3.1-Mono.tar.gz`	Serveur pour modes vidéo monochromes.
`XF86-3.1-P9000.tar.gz`	Serveur pour cartes à base de P9000.
`XF86-3.1-S3.tar.gz`	Serveur pour cartes à base de S3.
`XF86-3.1-SVGA.tar.gz`	Serveur pour cartes de type Super VGA.
`XF86-3.1-VGA16.tar.gz`	Serveur pour cartes de type VGA/EGA.
`XF86-3.1-W32.tar.gz`	Serveur pour cartes de type ET4000/W32.

Tous les fichiers suivants sont requis :

Fichier	Description
`XF86-3.1-bin.tar.gz`	Le reste des binaires X11R6.
`XF86-3.1-cfg.tar.gz`	Fichiers de configuration de `xdm`, `xinit` et `fs`.
`XF86-3.1-doc.tar.gz`	Documentation et pages de manuel.
`XF86-3.1-inc.tar.gz`	Fichiers *include*.
`XF86-3.1-lib.tar.gz`	Bibliothèques partagées X et fichiers de support.
`XF86-3.1-fnt.tar.gz`	Fontes de base.

Les fichiers suivants sont facultatifs :

Fichier	Description
`XF86-3.1-ctrb.tar.gz`	Quelques programmes (contributions).
`XF86-3.1-extra.tar.gz`	Serveurs XFree86 et binaires supplémentaires.
`XF86-3.1-lkit.tar.gz`	Kit de reconstruction des serveurs.
`XF86-3.1-fnt75.tar.gz`	Fontes 75-dpi.
`XF86-3.1-fnt100.tar.gz`	Fontes 100-dpi.
`XF86-3.1-fntbig.tar.gz`	Fontes Kanji et autres grandes fontes.
`XF86-3.1-fntscl.tar.gz`	Fontes Speedo, Type1, etc.
`XF86-3.1-man.tar.gz`	Pages de manuel.
`XF86-3.1-pex.tar.gz`	Binaires PEX binaries, includes et bibliothèques.
`XF86-3.1-slib.tar.gz`	Bibliothèques statiques X et fichiers de support.
`XF86-3.1-usrbin.tar.gz`	Démons situés dans /usr/bin.
`XF86-3.1-xdmshdw.tar.gz`	Version de xdm avec mots de passe *shadow*.

Le répertoire XFree86 devrait contenir des fichiers `README` (lisez moi) et des notes d'installation pour la version courante.

Tout ce que vous avez à faire pour installer XFree86 est de vous procurer les fichiers ci-dessus, créer le répertoire `/usr/X11R6` (en tant que `root`) et décompacter les archives depuis `/usr/X11R6` par une commande du genre :

```
# gzip -dc XF86-3.1-bin.tar.gz | tar xfBp -
```

Souvenez-vous que ces fichiers tar sont relatifs à `/usr/X11R6`, il est donc très important de les extraire en étant dans ce répertoire.

Ensuite, vous devez tout d'abord créer un lien `/usr/X11R6/bin/X` vers le serveur que vous allez utiliser. Par exemple, si votre carte vidéo nécessite le serveur couleur SVGA, `/usr/bin/X11/X` doit pointer sur `/usr/X11R6/bin/XF86_SVGA`. Si vous devez utiliser le serveur monochrome à la place, refaites le lien vers `XF86_MONO` par la commande :

```
# ln -sf /usr/X11R6/bin/XF86_MONO  /usr/X11R6/bin/X
```

Ceci s'applique à tout autre serveur bien entendu.

Si vous ne savez pas trop quel serveur utiliser, ou ne connaissez pas avec quel processeur votre carte vidéo utilise, vous pouvez lancer le programme `SuperProbe` qui se trouve dans `/usr/X11R6/bin` (inclus dans l'archive `XF86-3.1-bin` citée plus haut). Il effectue différents tests dans le but de tenter de deviner le processeur vidéo et quelques autres paramètres. Notez ce qu'il indiquera, vous en aurez besoin plus tard.

Vous devez vous assurer que `/usr/X11R6/bin` est dans votre path. Pour cela, vous pouvez éditer le fichier global `/etc/profile` ou `/etc/csh.login` (en fonction du shell que vous ou vos utilisateurs emploient) ; ou bien simplement rajouter ce répertoire à votre path personnel en modifiant `/etc/.bashrc` ou `/etc/.cshrc`, toujours selon le shell utilisé.

Vous devez également être certain que `/usr/X11R6/lib` pourra être trouvé par `ld.so`, l'éditeur de liens dynamique. Pour ce faire, ajoutez la ligne

```
/usr/X11R6/lib
```

au fichier `/etc/ld.so.conf`, et, en tant qu'utilisateur `root`, exécutez la commande `/sbin/ldconfig` pour que cette modification soit prise en compte.

5.1.3 Configuration de XFree86

Dans la majorité des cas, la configuration de XFree86 n'est pas difficile. Néanmoins, si vous devez utiliser du matériel pour lequel les pilotes sont en développement, ou désirez obtenir les meilleures performances ou résolutions de cartes vidéo de haut niveau, cela peut vous prendre beaucoup de temps.

Nous allons voir dans cette section comment créer le fichier `XF86Config`, qui constitue la configuration du serveur X. Souvent, il est préférable de commencer par quelque chose de simple, comme le mode vidéo 640x480, qui est supporté par toutes les cartes vidéo et moniteurs. Cette résolution est beaucoup trop basse pour être utilisée, mais ensuite vous pourrez commencer à modifier la configuration pour exploiter les possibilités de votre matériel. Le principe à suivre, est de vérifier que XFree86 fontionne sur votre équipement, avant de tenter des réglages complexes.

En plus des informations que vous allez trouver ici, vous devez absolument lire les documents suivants :

- La documentation de XFree86 qui se trouve dans `/usr/X11R6/lib/X11/doc` (contenu dans l'archive `XFree86-3.1-doc`). Vous devez particulièrement prendre connaissance du fichier `README.Config`, qui est un didacticiel sur la configuration.
- Plusieurs processeurs vidéo font l'objet de fichiers `README` dans le répertoire précédent (comme `README.Cirrus` and `README.S3`). Lisez ceux qui vous concernent.
- La page de manuel de `XFree86`.
- La page de manuel de `XF86Config`.
- La page de manuel du serveur particulier que vous utilisez (comme `XF86_SVGA` or `XF86_S3`).

Le fichier principal de configuration de XFree86 version 3 se trouve dans le répertoire `/usr/X11R6/lib/X11`, et s'appelle `XF86Config`. Il contient des informations sur la souris, la carte vidéo le moniteur, etc. Copiez l'exemple `XF86Config.eg` fourni sous le nom de `XF86Config` et éditez ce dernier, c'est un bon point de départ.

La page de manuel de `XF86Config` explique le format de ce fichier ; si vous n'en avez pas encore pris connaissance, lisez-là dès maintenant.

Nous allons vous présenter un exemple de fichier `XF86Config`, morceau par morceau. Notre exemple pourra légèrement différer de celui contenu dans la distribution, mais sa structure sera la même.

◇ Notez que le format du fichier `XF86Config` peut changer à chaque nouvelle version de la distribution XFree86. Les informations données ici ne sont valables que pour XFree86 version 3.1.

◇ Vous ne devez pas tenter de faire fonctionner le serveur X avec l'exemple donné ici, sans l'avoir adapté à votre propre système. L'utilisation d'un fichier de configuration non adapté à votre équipement peut amener le moniteur ou la carte vidéo à des valeurs hors limites ; et certains écrans de mauvaise facture peuvent être irrémédiablement détériorés. (C'est déjà arrivé, particulièrement sur des moniteurs à fréquence fixe). Répétons-le : soyez absolument certain que le fichier `XF86Config` correspond à votre équipement matériel avant de tenter de l'utiliser.

Chaque partie du fichier `XF86Config` est délimitée par un couple de lignes « `Section` ⟨*nom-de-la-section*⟩ ...`EndSection` ». La première s'appelle `Files` (fichiers), et ressemble à ceci :

```
Section "Files"
    RgbPath     "/usr/X11R6/lib/X11/rgb"
    FontPath    "/usr/X11R6/lib/X11/fonts/misc/"
    FontPath    "/usr/X11R6/lib/X11/fonts/75dpi/"
EndSection
```

La ligne `RgbPath` indique le chemin d'accès à la base de données des couleurs RGB de X11R6, et chaque ligne `FontPath` indique un chemin d'accès de fontes. En général, vous n'aurez pas besoin de modifier ces lignes ; assurez-vous simplement qu'il existe une ligne `FontPath` pour chaque police de caractères que vous aurez installée (c'est-à-dire, pour chaque répertoire dans `/usr/X11R6/lib/X11/fonts`).

La partie suivante, `ServerFlags`, spécifie la validation de certaines options du serveur. En général, elle est vide.

```
Section "ServerFlags"
# Les commentaires originaux sont en anglais. Nous les avons traduits
# pour vous faciliter la tâche.
#
# Validez cette ligne pour provoquer un core-dump au point exact de
# la réception d'un signal. La console peut alors rester dans un état
# indeterminé, inutilisable.
#    NoTrapSignals

# Validez cette ligne pour invalider l'arrêt du serveur sur la
# combinaison de touches Control-Alt-Backspace.
#    DontZap
EndSection
```

Ici, toutes les lignes sont commentées, elles ne seront donc pas prises en compte.

La section suivante, `Keyboard`, concerne le clavier. Elle se passe facilement de commentaires.

```
Section "Keyboard"
    Protocol    "Standard"
    AutoRepeat  500 5
    ServerNumLock
    LEFTALT meta
    RIGHTALT modeshift
    SCROLLLOCK modelock
EndSection
```

D'autres options sont disponibles, consultez l'exemple de la distribution si vous voulez modifier la configuration clavier. Les valeurs ci-dessus devraient fontionner sur la plupart des machines (y compris sur les claviers français).

La section suivante s'appelle `Pointer` et spécifie les paramètres nécessaires pour que la souris fonctionne.

```
Section "Pointer"

    Protocol    "MouseSystems"
    Device      "/dev/mouse"

# Baudrate and SampleRate ne sont requis que pour quelques souris Logitech
#    BaudRate   9600
#    SampleRate 150

# Emulate3Buttons n'est nécessaire que pour les souris à 2 boutons
#    Emulate3Buttons

# ChordMiddle est une option de quelques souris 3 boutons Logitech
#    ChordMiddle

EndSection
```

Les seules options qui vous concernent pour l'instant sont `Protocol` et `Device`. `Protocol` désigne le *protocole* utilisé par la souris (et non pas sa marque). Les types valides (sous Linux ; car il en existe pour d'autres systèmes d'exploitation) sont :

- `BusMouse`
- `Logitech`
- `Microsoft`
- `MMSeries`
- `Mouseman`
- `MouseSystems`
- `PS/2`
- `MMHitTab`

`BusMouse` doit être utilisé pour les souris bus Logitech. Notez que les anciennes souris emploient `Logitech`, mais que les nouveaux modèles de ce constructeur nécessitent soit `Microsoft` ou `MouseSystems`. C'est un des cas où le protocole utilisé n'a rien à voir avec la marque de la souris.

`Device` désigne le périphérique utilisé. Sur beaucoup d'installations Linux, il s'agit de `/dev/mouse`; qui est en réalité un lien vers le port série approprié (comme

`/dev/cua0`), ou vers le bon pilote de souris bus pour ce type d'accessoire. Assurez-vous au moins que le périphérique indiqué dans `Device` existe.

La partie suivante, `Monitor`, indique les caractéristiques de votre moniteur vidéo. Comme pour d'autres sections de `XF86Config`, il peut y avoir plusieurs sections `Monitor`. Cela permet de pouvoir utiliser différents écrans sur le même système, ou utiliser le même fichier de configuration pour plusieurs machines différentes. La plupart du temps vous n'aurez toutefois besoin que d'une entrée.

```
Section "Monitor"

    Identifier  "CTX 5468 NI"

    # Ces valeurs sont exclusivement pour un CTX 5468NI !
    # Ne tentez pas de les utiliser avec votre moniteur !

    Bandwidth    60
    HorizSync    30-38,47-50
    VertRefresh  50-90

    # Modes: Name      dotclock  horiz               vert

    ModeLine "640x480"  25       640 664 760 800     480 491 493 525
    ModeLine "800x600"  36       800 824 896 1024    600 601 603 625
    ModeLine "1024x768" 65       1024 1088 1200 1328 768 783 789 818

EndSection
```

La ligne `Identifier` permet de donner un nom arbitraire à l'entrée `Monitor`. Ce peut être n'importe quelle chaîne de caractères ; elle servira à désigner le moniteur à utiliser, un peu plus loin dans le fichier.

`Bandwidth` indique la bande passante vidéo maximum de l'appareil, en MHz. En résumé, il s'agit de la vitesse maximum à laquelle la carte vidéo peut envoyer des pixels au moniteur. Le manuel de votre écran doit indiquer cette valeur dans la partie spécifications techniques. Si elle est absente, vous devez consultez le constructeur ou votre vendeur pour l'obtenir. Il y a néanmoins d'autres sources d'information, citées plus bas.

`HorizSync` spécifie les différentes fréquences de synchronisation horizontale supportées par votre moniteur, en KHz. S'il s'agit d'un appareil « multisync », ce sera une ou plusieurs gammes de valeurs (comme dans l'exemple ci-dessus) ; s'il est à fréquence fixe, ce sera une liste de valeurs séparées par des virgules, comme ceci :

```
    HorizSync    31.5, 35.2, 37.9, 35.5, 48.95
```

Là encore, la notice de votre appareil doit contenir ces renseignements.

`VertRefresh` indique les fréquences de synchronisation verticale valides pour votre moniteur, en Hz. La syntaxe est la même que pour la synchro horizontale ; et comme toujours la documentation fournie avec l'écran doit les indiquer.

En fait, `Bandwidth`, `HorizSync`, et `VertRefresh` ne sont utilisées que pour vérifier que les résolutions que vous demandez sont compatibles avec votre moniteur, pour vous éviter de détruire l'appareil par un mode de fontionnement pour lequel il n'a pas été conçu.

La directive `ModeLine` sert à spécifier une résolution d'affichage. Son format est le suivant :

```
ModeLine ⟨nom⟩ ⟨horloge⟩ ⟨horizontal⟩ ⟨vertical⟩
```

Le champ ⟨*nom*⟩ est une chaîne arbitraire, qui vous servira à référencer ce mode plus tard. Le champ ⟨*horloge*⟩, est la fréquence exprimée en MHz, à laquelle la carte vidéo devra travailler, c'est la vitesse à laquelle elle enverra les pixels au moniteur pour cette résolution donnée. Les champs ⟨*horizontal*⟩ et ⟨*vertical*⟩ contiennent deux valeurs chacuns, déterminant à quel moment le canon du tube cathodique doit envoyer des électrons (« s'allumer »), et quand les impulsions verticales et horizontales doivent apparaître.

Comment déterminer ces valeurs en fonction des caractéristiques de votre moniteur ? Le fichier `VideoModes.doc`, inclus dans la distribution de XFree86, décrit en détail comment les calculer pour chaque résolution désirée. En tout premier lieu, ⟨*horloge*⟩ doit correspondre à l'une des valeurs que votre carte vidéo peut produire. Car plus loin, vous aurez à spécifier ces fréquences d'horloge, et il n'est possible d'utiliser un mode vidéo donné que si il est compatible avec ce que sait faire votre carte d'affichage.

Il existe pour vous aider deux fichiers dans la distribution de XFree86 : `modeDB.txt` and `Monitors`, ils se trouvent dans `/usr/X11R6/lib/X11/doc`. Vous pourrez peut-être y trouver des lignes `ModeLine` prévues pour votre modèle de moniteur vidéo.

Vous devrez commencer en prenant des valeurs `ModeLine` pour les modes standard VESA, supportés par la majorité des moniteurs. Le fichier `modeDB.txt` comprend les résolutions standard VESA, sous la forme suivante :

```
# 640x480@60Hz Non-Interlaced mode
# Horizontal Sync = 31.5kHz
# Timing: H=(0.95us, 3.81us, 1.59us), V=(0.35ms, 0.064ms, 1.02ms)
#
# name        clock   horizontal timing     vertical timing      flags
 "640x480"     25.175  640  664  760  800    480  491  493  525
```

Il s'agit ici de valeurs pour la résolution standard VESA 640x480. Il utilise une fréquence d'horloge de 25.175, que votre carte vidéo doit supporter pour pouvoir l'utiliser (nous reviendrons sur ce sujet). Pour inclure cette entrée dans votre fichier `XF86Config`, vous utiliserez donc la ligne :

```
ModeLine "640x480" 25.175  640 664 760 800  480 491 493 525
```

Notez que l'argument ⟨*nom*⟩ de `ModeLine` (ici,`"640x480"`) est une chaîne arbitraire ; par convention on y met la résolution, mais techniquement ce pourrait être n'importe quoi, à vous de décider.

Pour chaque `ModeLine`, le serveur vérifiera que les spécifications sont compatibles avec celles indiquées par `Bandwidth`, `HorizSync` et `VertRefresh`. Si ce n'est pas le cas, il affichera un message d'erreur lorsque vous le lancerez (nous y reviendrons). En tout état de cause, la fréquence d'horloge utilisée dans un mode ne doit pas être supérieure à la valeur indiquée par `Bandwidth` (mais souvent, une valeur très légèrement supérieure est sans danger pour le moniteur).

Si les valeurs standard VESA ne fonctionnent pas pour vous, les fichiers `modeDB.txt` et `Monitors` contiennent des valeurs adaptées à de nombreux types de moniteurs. Vous pourrez à partir de là créer des entrées `ModeLine`. N'utilisez que ce qui correspond au moniteur que vous possédez ; notez que beaucoup de petits écrans 14 et 15 pouces ne peuvent supporter les plus hautes résolutions, et souvent ne fonctionnent en 1024x768 qu'à des fréquences assez basses. Cela signifie que si vous ne pouvez trouver de modes haute résolution pour votre moniteur dans ces fichiers, c'est qu'il ne supporte probablement pas mieux que ce qui est indiqué.

Si vous êtes complètement perdu et ne trouvez vraiment aucune valeur adaptée pour votre cas, vous pouvez suivre les instructions contenues dans le fichier `VideoModes.doc` inclus dans la distribution de XFree86, afin de générer vous même la ligne `ModeLine` pour votre moniteur. Bien que la réalisation manuelle de cette configuration ne soit pas forcément très précise, vous pourrez à coup sûr vous en sortir. Le fichier `VideoModes.doc` décrit également le format de la directive `ModeLine` et quelques autres aspects du serveur XFree86 avec moult détails.

Enfin, si vous obtenez des valeurs qui sont presque bonnes, mais pas parfaites, il est possible de les modifier très légèrement pour aboutir au réglage désiré. Par exemple, si l'image est un peu décalée sur l'écran, ou semble défiler dans un sens ou un autre, suivez les instructions contenues dans le fichier `VideoModes.doc` pour corriger ces problèmes. Mais auparavant, vérifiez qu'il ne s'agit pas d'un mauvais réglage du moniteur, qui possède en principe tout ce qu'il faut pour ajuster la qualité de l'affichage ! Dans de nombreux cas, il est au moins nécessaire d'ajuster les tailles horizontales et verticales de l'image, ainsi que le cadrage.

◊ Vous ne devez pas utiliser des valeurs adaptées à d'autres moniteurs que le vôtre dans la directive `ModeLine`, car vous pourriez amener votre appareil à fonctionner hors des limites prévues par le constructeur et le détruire irrémédiablement.

La partie suivante du fichier `XF86Config` s'appelle `Device`, et indique les paramètres de votre carte vidéo. En voici un exemple.

```
Section "Device"
        Identifier "#9 GXE 64"

        # Rien pour l'instant. Nous remplirons ceci plus tard.

EndSection
```

Cette section définit les propriétés d'une carte vidéo donnée. `Identifier` est une chaîne de caractères arbitraire qui permettra de désigner la carte plus tard.

Au départ, vous n'avez pas besoin de remplir cette section `Device`, car nous utiliserons le serveur X pour détecter ces caractéristiques, que nous pourrons ensuite indiquer afin d'éviter ce test à chaque démarrage de X Window. Le serveur XFree86 est capable de détecter processeur vidéo, fréquences d'horloges utilisables, RAMDAC, et mémoire vidéo installée sur la carte.

Auparavant, nous devons toutefois terminer la réalisation de notre fichier de configuration `XF86Config`. La section suivante, `Screen`, spécifie la combinaison moniteur/carte vidéo à utiliser pour un serveur particulier.

```
Section "Screen"
    Driver     "Accel"
    Device     "#9 GXE 64"
    Monitor    "CTX 5468 NI"
    Subsection "Display"
        Depth      16
        Modes      "1024x768" "800x600" "640x480"
        ViewPort   0 0
        Virtual    1024 768
    EndSubsection
EndSection
```

La ligne `Driver` spécifie quel type de serveur X vous utiliserez. Les valeurs possibles sont :

- `Accel` : pour les serveurs `XF86_S3`, `XF86_Mach32`, `XF86_Mach8`, `XF86_8514`, `XF86_P9000`, `XF86_AGX`, and `XF86_W32` ;
- `SVGA` : pour le serveur `XF86_SVGA` ;
- `VGA16` : pour le serveur `XF86_VGA16` ;
- `VGA2` : pour le serveur `XF86_Mono`
- `Mono` : pour les modes monochromes non VGA des serveurs `XF86_Mono` et `XF86_VGA16`.

Vous devez vous assurer que `/usr/X11R6/bin/X` est bien un lien symbolique vers le serveur particulier que vous utilisez (la section « 3.10.2 » du chapitre 3 vous apprendra de quoi il s'agit).

La ligne `Device` spécifie l'identificateur (`Identifier`) de la section `Device` correspondant à la carte vidéo que ce serveur doit utiliser. Plus haut, nous avons créé une section `Device` contenant la ligne

```
Identifier "#9 GXE 64"
```

Par conséquent, nous mettrons ici `"#9 GXE 64"` sur cette ligne `Device`.

De même, la ligne `Monitor` indique le nom de la section `Monitor` à utiliser avec ce serveur. Ici, `"CTX 5468 NI"` est l'identificateur (`Identifier`) utilisé dans la section `Monitor` décrite auparavant.

`Subsection "Display"` définit plusieurs propriétés du serveur XFree86 correspondant à votre combinaison écran/carte vidéo. Le fichier `XF86Config` fourni en exemple décrit toutes ces options en détails, beaucoup ne sont que du perfectionnisme et inutiles pour faire fonctionner le système.

Les options que vous devez connaître sont :

- `Depth`. Définit le nombre de plans de couleurs (le nombre de bits par pixel). Généralement, `Depth` vaut 8. Avec le serveur `VGA16`, ce sera 4 et pour le serveur monochrome, cette valeur vaudra 1. Si vous utilisez une carte vidéo accélérée avec suffisamment de mémoire permettant de supporter plus de couleurs, vous pourrez initialiser `Depth` à 16, 24 ou 32. Si vous avez des problèmes, mettez 8 ; vous essaierez de déboguer plus tard.

- `Modes`. Il s'agit de la liste de noms des modes vidéo que vous avez défini par la directive `ModeLine` dans la section `Monitor`. Plus haut, nous avons utilisé des `ModeLines` appelées `"1024x768"`, `"800x600"`, et `"640x48"0`. Par conséquent, notre ligne `Modes` sera :

  ```
  Modes    "1024x768" "800x600" "640x480"
  ```

  Le premier mode indiqué sera celui par défaut, dans lequel XFree86 démarrera. Vous pourrez alors commuter ces différentes résolutions par les combinaisons de touches `control`-`alt`-`+` et `control`-`alt`-`-`.

  Il est peut-être préférable, lors des essais de configuration, de mettre les résolutions les plus basses en premier. Une fois que tout fonctionnera correctement, vous pourrez modifier `XF86Config` pour supporter des résolutions supérieures.

- `Virtual`. Initialise la taille du bureau virtuel. XFree86 a la capacité d'utiliser toute mémoire additionnelle de votre carte vidéo pour étendre la taille de votre bureau. Lorsque vous déplacez le curseur de la souris vers le bord de l'écran, vous verrez la partie auparavant cachée du bureau : votre écran se comporte comme une fenêtre ouverte sur une surface beaucoup plus grande. Par conséquent, même si vous utilisez une faible résolution comme 800x600, vous pouvez initialiser `Virtual` à la valeur maximum que votre carte vidéo peut supporter (une carte équipée de 1Mo de mémoire peut faire 1024x768 avec 8 bits par pixel, une carte de 2Mo fera 1280x1024, ou encore 1024x768 avec 16 bits par pixel).

  Ce bureau virtuel est très pratique pour utiliser toute la mémoire de votre carte vidéo, mais il est assez limité. Si vous désirez employer un vrai bureau virtuel, nous vous suggérons d'utiliser le gestionnaire de fenêtres `fvwm` (ou un autre du même genre). Vous pourrez utiliser de multiples bureaux bien plus grands, avec toutes sortes de facilités qui amélioreront votre confort ; consultez la page de manuel de `fvwm` pour en savoir plus. La plupart des distributions binaires de Linux contenant X Window utilisent `fvwm` par défaut.

– `ViewPort`. Si vous utilisez le bureau virtuel du serveur décrit ci-dessus, `ViewPort` indique les coordonnées du coin supérieur gauche de la portion à afficher au démarrage de XFree86. On utilise souvent `Virtual 0 0` ; si rien n'est indiqué, la fenêtre sera centrée (ce qui peut s'avérer peu pratique).

Cette section permet encore de nombreuses autres options, consultez la page de manuel de `XF86Config` pour en avoir une description complète. En pratique elles ne sont pas nécessaires pour l'installation initiale du serveur X Window.

5.1.4 Indiquer les caractéristiques de la carte vidéo

Votre fichier `XF86Config` est maintenant prêt à l'emploi, mais il ne contient aucune information sur la carte vidéo. Nous allons donc laisser le serveur X tester cette partie, puis récupérer les informations qu'il trouvera pour les intégrer au fichier de configuration.

Il est possible d'éviter cette opération, si vous trouvez la description de votre carte dans les fichiers `modeDB.txt`, `AccelCards`, et `Devices`. Ils se trouvent dans `/usr/X11R6/lib/X11/doc`. Vous trouverez également dans ce répertoire des fichiers `README` pour certains circuits particuliers. Si vous trouvez les informations relatives à votre équipement, vous pouvez les inclure dans `XF86Config` ; si l'une des valeurs requises est absente, il vous faudra procéder de la manière qui va suivre.

Dans les exemples suivants, nous allons utiliser une carte vidéo #9 GXE 64, qui utilise le processeur S3 et par conséquent le serveur `XF86_S3`. Il se trouve que l'auteur utilise ce matériel ; mais tout ce qui va être dit s'appliquera bien entendu à n'importe quelle autre type de configuration.

La première chose à faire est de déterminer le circuit vidéo utilisé sur la carte. Le programme `SuperProbe` (qui se trouve dans `/usr/X11R6/bin`) vous donnera cette information, mais vous devrez connaître le nom associé à ce processeur, tel qu'il est connu par le serveur X.

Pour cela, tapez la commande :

```
X -showconfig
```

Elle vous indiquera les noms connus par le serveur X (la page de manuel de chaque serveur les indique également). Par exemple, avec le serveur `XF86_S3` pour cartes accélérées, nous obtenons :

```
XFree86 Version 3.1 / X Window System
(protocol Version 11, revision 0, vendor release 6000)
Operating System: Linux
Configured drivers:
  S3: accelerated server for S3 graphics adaptors (Patchlevel 0)
      mmio_928, s3_generic
```

Les noms valides seront donc ici `mmio_928` et `s3_generic`. La page de manuel de `XF86_S3` décrit ces circuits et indique quelles cartes vidéo les utilisent. Dans le cas de la #9 GXE 64, c'est `s3_generic` qui s'applique.

Si vous ne savez vraiment pas quel nom de circuit utiliser, le serveur X peut le tester pour vous. Tapez la commande :

```
X -probeonly \superieura /tmp/x.out 2>&1
```

Si vous n'utilisez pas `bash`, mais `csh`, essayez : if you use `bash` as your shell. If you use `csh`, try:

```
X -probeonly &\superieura /tmp/x.out
```

Vous ne devez effectuer ce test que lorsque le système n'est pas chargé, c'est-à-dire lorsqu'il n'y a pas d'autre activité sur la machine, car cette commande testera également les temporisations à utiliser, et un système chargé faussera les mesures.

La sortie de la commande ci-dessus (redirigée dans le fichier `/tmp/x.out` devrait contenir des lignes ressemblant à ceci :

```
XFree86 Version 3.1 / X Window System
(protocol Version 11, revision 0, vendor release 6000)
Operating System: Linux
Configured drivers:
  S3: accelerated server for S3 graphics adaptors (Patchlevel 0)
      mmio_928, s3_generic
Plusieurs lignes supprimées...
(--) S3: card type: 386/486 localbus
(--) S3: chipset:   864 rev. 0
(--) S3: chipset driver: s3_generic
```

Ici, nous voyons que les deux noms de processeur valides pour ce serveur (`XF86_S3` dans notre cas) sont `mmio_928` et `s3_generic`. Le serveur a détecté une carte utilisant le processeeur `s3_generic`.

Dans la section `Device` du fichier `XF86Config`, ajoutons une ligne `Chipset`, contenant ce nom. Donc par exemple :

```
Section "Device"
        # We already had Identifier here...
```

```
        Identifier "#9 GXE 64"
        # Add this line:
        Chipset "s3_generic"
EndSection
```

Nous devons maintenant déterminer les fréquences de travail utilisées par cette carte vidéo. Ces fréquences d'horloge correspondent aux vitesses auxquelles la carte peut envoyer des pixels au moniteur. Comme nous l'avons vu, chaque résolution se voit assigner une fréquence d'horloge particulière. Nous devons donc déterminer quelles sont celles offertes par notre équipement.

Tout d'abord, jetez un œil aux fichiers (`modeDB.txt`, etc) mentionnés plus haut, pour voir si ces valeurs y sont indiquées. Ce sont généralement des listes de 8 ou 16 valeurs, toutes exprimées en MHz. Par exemple, dans `modeDB.txt` nous trouvons une entrée pour la carte vidéo Cardinal ET4000, qui ressemble à ceci :

```
# chip    ram   virtual   clocks                           default-mode  flags
 ET4000   1024  1024 768   25  28  38  36  40  45  32   0  "1024x768"
```

Nous voyons que cette carte supporte des fréquences d'horloge de 25, 28, 38, 36, 40, 45, 32, et 0 MHz.

Vous devez donc ajouter dans la section `Devices` du fichier `XF86Config` une ligne `Clocks` contenant la liste de ces valeurs pour votre carte vidéo. Par exemple, pour les valeur ci-dessus, nous ajouterions la ligne

```
        Clocks 25 28 38 36 40 45 32 0
```

à la section `Devices`, après `Chipset`. Notez que l'ordre de ces valeurs est très important ! Ne les triez surtout pas et n'éliminez pas les doublons !

Si vous ne trouvez pas cette liste toute faite, le serveur X peut tester ces valeurs pour vous. Avec la commande `X -probeonly` décrite plus haut, vous devriez obtenir des lignes comme ceci :

```
(--) S3: clocks:  25.18  28.32  38.02  36.15  40.33  45.32  32.00  00.00
```

Nous pouvons alors rajouter une ligne `Clocks` contenant toutes ces valeurs, telles qu'affichées. Vous pouvez utiliser plusieurs lignes `Clocks` dans le fichier `XF86Config`, si il y a trop de valeurs pour que tout tienne sur une ligne. Assurez-vous simplement de conserver l'ordre correct.

Vérifiez, avant de lancer `X -probeonly`, qu'il n'y ait pas déjà de ligne `Clocks` dans la section `Devices` ; sinon le serveur n'effectuera pas le test, mais utilisera ces valeurs.

Notez que quelques cartes vidéo accélérées utilisent un générateur d'horloge programmable (voyez la page de manuel de `XF86_Accel` pour plus de détails, c'est généralement le cas des cartes S3, AGX et XGA-2). Cela permet essentiellement au serveur X d'indiquer à la carte quelle fréquence utiliser. Si c'est le cas, il se peut que vous ne trouviez pas de liste de valeurs d'horloge pour la carte en question dans les fichiers cités précédemment, ou que la liste générée par la commande `X -probeonly` ne contienne qu'une ou deux valeurs, le reste n'étant qu'une suite de zéros.

Pour ce type de carte, vous devrez utiliser une ligne `ClockChip`, en lieu et place de la ligne `Clocks`. Elle donne le nom du circuit ; les pages de manuel de chaque serveur XFree86 décrivent ce qu'il en est. Par exemple, dans le fichier `README.S3`, nous pouvons lire que plusieurs cartes S3-864 utilisent un générateur « ICD2061A », et que nous devons utiliser la ligne

```
ClockChip "icd2061a"
```

dans le fichier `XF86Config`, au lieu de `Clocks`. Tout comme cette dernière, elle se place dans la section `Devices`, après `Chipset`.

De même, certaines cartes accélérées nécessitent que l'on spécifie le type de circuit RAMDAC dans le fichier `XF86Config`, par une ligne `Ramdac`. La page de manuel de `XF86_Accel` décrit cette option. En général, le serveur X détermine cette valeur correctement.

Certains types de cartes vidéo demandent que l'on spécifie plusieurs options dans la section `Devices` du fichier `XF86Config`. Ces options sont décrites dans la page de manuel du serveur que vous utilisez, ou dans les documentations supplémentaires fournies (comme `README.cirrus` ou `README.S3`). Elles sont validées par une ligne `Option` ; par exemple la carte #9 GXE 64 en nécessite deux :

```
Option "number_nine"
Option "dac_8_bit"
```

En règle générale, le serveur X fonctionnera sans ces options, mais elles sont nécessaires pour obtenir des performances optimales. Il en existe tellement qu'il est impossible de les citer ici, et elles dépendent de l'équipement que vous employez. Si vous devez utiliser de telles options, les pages de manuel et les diverses documentations vous indiqueront celles qui s'appliquent dans votre cas.

Donc, au bout du compte, vous devriez avoir obtenu une section `Devices` ressemblant à celle-ci :

```
Section "Device"
        # Device section for the #9 GXE 64 only!
        Identifier "#9 GXE 64"
        Chipset "s3_generic"
        ClockChip "icd2061a"
        Option "number_nine"
        Option "dac_8_bit"
EndSection
```

La plupart des cartes vidéo demanderont une ligne `Clocks`, au lieu de `ClockChip`, comme nous l'avons expliqué. L'entrée ci-dessus n'est valide que pour une carte particulière, la #9 GXE 64. Elle n'est ici qu'à titre d'exemple.

5.1.5 Démarrage de XFree86

Avec votre beau fichier `XF86Config` tout neuf, vous devriez maintenant être prêt à lancer X Window. Vérifiez bien que `/usr/X11R6/bin` est dans votre path.

La commande à utiliser est alors :

```
startx
```

Il s'agit d'un frontal au programme `xinit` (au cas où vous seriez habitué à utiliser `xinit` sur d'autres systèmes UNIX).

Cette commande va lancer le serveur X et exécuter les instructions trouvées dans le fichier `.xinitrc` de votre répertoire personnel. Ce fichier n'est qu'un shell-script contenant les clients X que vous désirez lancer au départ. S'il n'existe pas, c'est le fichier de configuration global `/usr/X11R6/lib/X11/xinit/xinitrc` qui sera pris en compte.

Un fichier `.xinitrc` de base ressemble à ceci :

```
#!/bin/sh

xterm -fn 7x13bold -geometry 80x32+10+50 &
xterm -fn 9x15bold -geometry 80x34+30-10 &
oclock -geometry 70x70-7+7 &
xsetroot -solid midnightblue &

exec twm
```

Ce script va lancer deux clients `xterm`, une pendule `oclock`, et peindre en bleu foncé la fenêtre principale (le fond d'écran). Il lancera ensuite `twm`, le gestionnaire de fenêtres. Notez que `twm` est exécuté par l'instruction `exec` du shell, ce qui aura pour effet de remplacer le processus `xinit` par `twm`. Par conséquent, en sortant de ce gestionnaire de fenêtres, le serveur X s'arrêtera. Vous pouvez sortir de `twm` en utilisant son menu principal : pressez le premier bouton de la souris lorsque le curseur n'est pas dans une fenêtre, un menu s'affichera qui vous proposera différentes options, dont celle de terminer la session (`Exit Twm`).

Assurez-vous bien que la dernière commande dans `.xinitrc` est lancée par `exec`, et qu'elle ne l'est pas en tâche de fond (pas de signe « et commercial » en fin de ligne). Sinon le serveur X s'arrêterait aussitôt les clients lancés.

Vous pouvez aussi sortir de X en pressant la combinaison de touches `control`-`alt`-`backspace`. Tout s'arrêtera d'un coup.

La configuration ci-dessus est vraiment minimale. Vous pourrez obtenir bien mieux avec un peu de travail, et organiser votre environnement comme vous l'entendez. Par exemple, le gestionnaire de fenêtres `fvwm` vous offrira un bureau virtuel efficace, vous pourrez choisir les couleurs, fontes, tailles et positions de fenêtres, et bien d'autres choses selon vos goûts personnels. Bien que le système X Window puisse paraître minimaliste au premier abord, il est en réalité l'un des environnements graphiques les plus puissants et universels.

Si vous êtes un nouveau venu à X Window, nous vous suggérons fortement de vous procurer un livre comme *The X Window System: A User's Guide*.

5.1.6 En cas de problème

Lors du tout premier lancement de X Window, il est très fréquent que quelque chose ne se passe pas bien. C'est presque toujours dû à un problème dans votre fichier de configuration `XF86Config`. Généralement, les caractéristiques du moniteur sont commentées ou absentes, ou les temporisations de la carte vidéo ne sont pas indiquées correctement. Si votre affichage semble défiler, ou si les bords sont déformés ou décalés, ce peut être une indication de mauvaises valeurs pour la carte vidéo. Vérifiez bien que vous avez spécifié le processeur vidéo correct, ainsi que d'autres options de la section `Device`. Assurez-vous d'utiliser le bon serveur X et que `/usr/X11R6/bin/X` est bien un lien symbolique vers ce programme.

Si vraiment tout ceci ne suffit pas, essayez de lancer le serveur X « tout nu » ; c'est à dire, tapez une commande comme :

```
X > /tmp/x.out 2>&1
```

Vous pourrez ensuite l'arrêter (s'il a tout de même démarré) par les touches `control`-`alt`-`backspace`, et examiner le contenu du fichier `/tmp/x.out`. Le serveur aura indiqué toutes les erreurs rencontrées.

Le fichier `VideoModes.doc` inclus dans la distribution XFree86 contient beaucoup de trucs et astuces pour ajuster les valeurs contenues dans votre `XF86Config`.

Souvenez-vous que vous pouvez utiliser `control`-`alt`-`+` et `control`-`alt`-`-` pour changer de mode vidéo (il s'agit des touches `+` et `-` du pavé numérique), et choisir ainsi entre les résolutions indiquées dans les lignes `Modes` de la section `Screen` du fichier `XF86Config`. Si les résolutions les plus hautes ne fonctionnent pas bien, essayez les plus faibles. Vous pourrez au moins déterminer ainsi quelles parties de votre configuration fontionnent correctement, et lesquelles seront à revoir.

Également, vérifiez les réglages de votre moniteur. Dans la plupart des cas il est nécessaire d'ajuster la taille de l'image, son cadrage, et parfois la stabilité verticale ou horizontale. Tout moniteur vidéo correct permet ces réglages, que ce soit par des boutons rotatifs, un réglage par microprocesseur ou de simples potentiomètres ajustables à l'aide d'un tournevis.

Enfin, le forum Usenet `comp.windows.x.i386unix` est dédié aux discussions sur XFree86. Il peut être très utile de parcourir les messages, votre problème pourrait y être décrit ; sinon vous pourrez y poser les questions appropriées.

5.2 Accéder aux fichiers MS-DOS

Si, pour quelque raison tordue, vous avez besoin d'accéder à des fichiers MS-DOS, c'est une opération très simple à réaliser sous Linux.

La méthode ordinaire permettant d'avoir accès à ces fichiers est de monter une partition ou une disquette MS-DOS sous Linux, ce qui vous autorise à accéder aux données directement comme sur tout autre système de fichiers. Si par exemple vous avez une disquette MS-DOS dans le lecteur `/dev/fd0`, la commande

```
# mount -t msdos /dev/fd0 /mnt
```

la montera sur `/mnt`. Voyez la section 4.6.2 pour plus d'information sur la manière de monter des disquettes.

Vous pouvez bien sûr monter une partition MS-DOS de votre disque dur pour y accéder sous Linux. Si cette partition est `/dev/hda1`, la commande

```
# mount -t msdos /dev/hda1 /mnt
```

la montera le plus simplement du monde. Assurez-vous de bien démonter la partition (avec `umount`) lorsque vous aurez fini de l'utiliser. Vous pouvez faire que vos partitions MS-DOS soient automatiquement montées lors du démarrage du système si vous prévoyez des entrées adéquates dans `/etc/fstab`; plus de détails section 4.8. Par exemple, la ligne suivante dans `/etc/fstab` montera une partition MS-DOS `/dev/hda1` dans le répertoire `/dos` :

```
/dev/hda1      /dos      msdos       defaults
```

L'ensemble de programmes `Mtools` peut également être utilisé pour accéder aux fichiers MS-DOS. Il n'est pas spécifique à Linux et peut être installé sur toute machine UNIX pourvue d'un lecteur de disquettes compatible avec celui de l'IBM-PC. Par exemple, les commandes `mcd`, `mdir`, et `mcopy` se comportent comme leurs équivalents MS-DOS. Si vous avez installé les `Mtools`, vous devriez avoir les pages de manuel correspondantes : comme toujours, lisez-les.

Il est bien évident que les différentes méthodes de partitions MS-DOS compressées ne peuvent pas êtres supportées, et que les noms de fichiers et permissions d'accès sont limités à ce que sait faire ce système de fichiers, c'est à dire pas grand chose...

Lire ou écrire des fichiers MS-DOS est une chose; exécuter des programmes DOS en est une autre. Il existe un émulateur MS-DOS pour Linux, qui est disponible un peu partout ; il est toutefois toujours en développement. Il est déjà suffisamment correct pour exécuter sous Linux un certain nombre de programmes, y compris WordPerfect. Cela dit, Linux et MS-DOS sont des systèmes extrêmement différents ; la puissance de tout émulateur DOS sous UNIX est toujours assez limitée.

Il existe également un projet d'émulateur Microsoft Windows, dont les versions tests permettent déjà de faire tourner certains programmes MS-Windows (comme le Solitaire par exemple) sous X Window. Tenez-vous au courant de l'avancement de ce projet en lisant les forums Usenet appropriés.

5.3 Réseau TCP/IP

Linux supporte une implémentation complète des protocoles réseau TCP/IP (Transport Control Protocol/Internet Protocol). Le succès de TCP/IP dans le monde entier n'est plus à démontrer ; avec Linux et une carte Ethernet, vous pouvez connecter votre machine à un réseau local, ou (avec la connectivité adéquate), à l'Internet, le plus grand réseau mondial.

Installer un petit réseau local de machines UNIX est très facile. Il suffit d'une carte Ethernet dans chaque ordinateur et un câblage correct. Si votre compagnie ou votre université offre un accès à l'Internet, vous pouvez très facilement ajouter votre système Linux à cet énorme réseau.

Linux supporte TCP/IP non seulement sur Ethernet, mais aussi sur ligne série grâce à SLIP[3] et PPP[4]. Si vous pouvez avoir un accès SLIP ou PPP (travail, université, fournisseur de services IP) vous ouvrant les portes d'Internet, vous pourrez en connectant un modem à votre système Linux, accéder à l'intégralité du réseau d'un simple coup de téléphone. Alternativement, si votre machine Linux est déjà connectée au réseau par Ethernet, vous pouvez en faire un serveur SLIP ou PPP.

Pour une description complète de l'installation et de la configuration de TCP/IP sous Linux, nous vous encourageons à lire le « Linux NET-2 HOWTO, » (voir annexe A), qui est un guide détaillé traitant aussi de SLIP et PPP. Vous pouvez également prendre connaissance du livre *Linux Network Administrator's Guide*, du groupe de documentation Linux, qui est disponible depuis quelque temps[5] comme le présent manuel, par téléchargement sous forme de fichiers LaTeX, dvi, postscript ou ASCII.

La lecture de *TCP/IP Network Administration*, par Craig Hunt, est également intéressante; il décrit la configuration et l'utilisation de TCP/IP sur les systèmes UNIX, tout ce qui y est dit s'applique donc parfaitement à Linux.

3. SLIP : Serial Line Internet Protocol.
4. PPP: Point to Point Protocol.
5. Uniquement en anglais et allemand, pour l'instant.

5.3.1 Configuration matérielle

Pour connecter votre machine sur un réseau Ethernet, vous devrez posséder l'une de ces cartes (la liste s'agrandit de mois en mois) :

3com 3c503, 3c503/16, 3c505, 3c507,3c509/3c579; Novell NE1000, NE2000; Western Digital WD8003, WD8013; Hewlett Packard HP27245, HP27247, HP27250; SMC Ultra support, AT1500, DEPCA, EtherExpress, AT1700, Apricot Xen-II, D-Link DE600, D-Link DE620, AT-LAN-TEC/RealTek,...

Les clones de ces cartes fonctionnent, par exemple : clones WD-80x3 : LANNET LEC-45 ; clones NE2000 : Alta Combo, Artisoft LANtastic AE-2, Asante Etherpak 2001/2003, D-Link Ethernet II, LTC E-NET/16 P/N 8300-200-002, Network Solutions HE-203, SVEC 4 Dimension Ethernet, 4-Dimension FD0490 EtherBoard 16, et D-Link DE-600, SMC Elite 16.

Voyez le « Linux Ethernet HOWTO » et les sources du noyau que vous utilisez pour plus de précisions sur les cartes Ethernet supportées et leur configuration.

5.3.2 Configuration de TCP/IP sur votre système

Nous allons voir comment configurer une connexion TCP/IP Ethernet sur votre machine Linux. Notez que cette méthode devrait fonctionner sur de nombreux systèmes, mais certainement pas sur tous. Vous devriez trouver ici suffisamment d'explications pour vous permettre de comprendre comment configurer le réseau sur votre machine, mais il est impossible de donner une procédure exacte commune à tout le monde, tant il existe de cas particuliers et de pièges possibles. [6]

Tout d'abord, nous considérerons que vous disposez d'un système Linux équipé pour TCP/IP. Il comprend donc les clients de base comme `telnet` et `ftp`, les commandes d'administration comme `ifconfig` et `route` (que vous trouverez sans doute dans `/etc` ou `/sbin`), et les fichiers de configuration réseau (comme par exemple `/etc/hosts`). Si ce n'est pas le cas, consultez les documents cités précédemment afin de vous procurer et installer les programmes nécessaires.

Nous considérerons également que votre noyau a été configuré et compilé en validant le support de TCP/IP. Voyez la section 4.7 pour trouver des informations sur la manière de recompiler le noyau ; vous devrez répondre « yes » (oui) aux questions relatives au réseau lors de l'opération `make config`, avant la compilation.

Vous devrez commencer par modifier un certain nombre de fichiers de configuration utilisés pour le réseau. C'est une opération très simple pour la plupart d'entre eux; malheureusement l'emplacement de ces fichiers dans les différentes distributions de Linux varie énormément. Vous pourrez les trouver la plupart du temps soit dans `/etc`, `/usr/etc`, voire `/usr/etc/inet`; ou bien carrément à d'autres endroits

6. Certaines parties contenues ici sont une adaptation du NET-2-HOWTO de Terry Dawson et Matt Welsh.

beaucoup plus bizarres. Dans le pire des cas, il vous faudra utiliser la commande `find` pour trouver dans quel recoin de votre système ils peuvent bien se cacher. Notez que les programmes exécutables et leurs fichiers de configuration ne sont pas nécessairement dans le même répertoire.

Les informations qui vont suivre s'appliquent principalement aux connexions Ethernet. Si vous avez l'intention d'utiliser SLIP ou PPP, lisez cette section pour en assimiler les notions, puis reportez-vous ensuite aux instruction spécifiques à SLIP.

5.3.2.1 Votre configuration réseau

Avant de pouvoir configurer TCP/IP, vous devez posséder certaines informations sur votre site. Dans la plupart des cas, l'administrateur de votre réseau local pourra vous fournir les renseignements nécessaires, qui sont :

- Votre adresse IP. Il s'agit d'une adresse unique attribuée à la machine, sous forme de quatre nombres décimaux séparés par des points, comme par exemple 128.253.153.54. Votre administrateur réseau pourra vous la fournir.

  Si vous ne configurez que votre système local, c'est à dire une machine isolée du monde (sans Ethernet, SLIP ou PPP), ne faisant que des connexions TCP/IP sur elle-même (« loopback »), alors votre adresse IP est par convention 127.0.0.1.

- Votre masque réseau (« netmask »). C'est un nombre similaire à votre adresse IP, qui détermine quelle portion de l'adresse spécifie le sous-réseau, et quelle portion spécifie la machine (host) sur ce sous-réseau. (Si ce jargon TCP/IP vous affole, nous vous suggérons de lire un ouvrage quelconque d'initiation à l'administration réseau). Le masque réseau est un ensemble de bits, qui en fonction d'une adresse de votre réseau, permet de déterminer à quel sous-réseau appartient cette adresse. Ceci est très important pour le routage ; si par exemple vous vous apercevez que vous pouvez vous connecter à l'extérieur de votre réseau, mais pas à certaines machines de votre réseau local, il y a de grandes chances que votre masque réseau soit faux.

  Vos administrateurs réseau ont choisi le masque lors de l'installation du réseau, par conséquent ils doivent être capable de vous fournir la valeur correcte à utiliser. La plupart des réseaux sont des sous-réseaux de classe C, qui utilisent un masque de 255.255.255.0. D'autres réseaux de classe B utiliseront 255.255.0.0; Linux sélectionnera automatiquement un masque par défaut ne considérant aucun sous-réseau si vous ne lui précisez rien.

  Tout ceci s'applique également au port « loopback » : comme son adresse IP est 127.0.0.1, le masque réseau est toujours 255.0.0.0. Vous pouvez soit le préciser explicitement, soit faire confiance au masque par défaut attribué par Linux.

- Votre adresse réseau. Il s'agit de votre adresse IP masquée bit à bit par un ET logique avec le masque réseau. Par exemple, si votre masque est 255.255.255.0, et votre adresse IP 128.253.154.32, alors votre adresse réseau vaut 128.253.154.0. Si le masque était 255.255.0.0, elle vaudrait 128.253.0.0.

  Si vous êtes isolé, sur 127.0.0.1, vous n'avez pas d'adresse réseau.

- Votre adresse broadcast. Elle est utilisée pour diffuser des paquets sur toutes les machines de votre sous-réseau. Par conséquent, si l'adresse de votre machine est déterminée par le dernier octet de l'adresse IP (masque réseau 255.255.255.0), votre adresse broadcast sera le résultat d'un OU entre votre adresse réseau et 0.0.0.255.

  Par exemple, si votre adresse IP est 128.253.154.32, et votre masque réseau vaut 255.255.255.0, votre adresse de broadcast est alors 128.253.154.255.

  Notez que d'anciennes configurations réseau utilisent l'adresse réseau comme adresse de broadcast. Si vous avez un doute, demandez à votre administrateur réseau (dans la plupart des cas il suffit de copier la configuration réseau d'une autre machine du sous-réseau et de remplacer l'adresse IP bien sûr).

  Si vous êtes isolé, sur 127.0.0.1, vous n'avez pas d'adresse broadcast.

- L'adresse de la passerelle. Il s'agit de l'adresse de la machine qui est votre « passerelle » vers le monde extérieur (par exemple les machines qui ne sont pas sur votre sous-réseau). Très souvent la machine passerelle a une adresse IP identique à la vôtre mais se terminant par .1; par exemple si votre adresse est 128.253.154.32, la passerelle pourrait être 128.253.154.1. Vos administrateurs réseaux vous fourniront cette adresse.

  En fait, vous pouvez avoir plusieurs passerelles. Une *passerelle* (gateway) est simplement une machine connectée à plusieurs réseaux (elle a des adresses IP sur différents sous-réseaux), et qui route les paquets entre eux. Beaucoup de réseaux ont une seule passerelle vers « le monde extérieur » (le réseau à côté du vôtre), mais dans certains cas il peut être nécessaire d'accéder à plusieurs autres réseaux.

  Si vous ne sortez pas de votre réseau local, vous n'avez pas de passerelle, vous n'en avez pas non plus sur 127.0.0.1 bien sûr.

- L'adresse du serveur de noms. La plupart des machines d'un réseau important utilisent un serveur de noms, qui tranforme les noms des machines en adresses IP. Votre administrateur réseau vous indiquera l'adresse de la machine serveur de noms ; vous pouvez aussi avoir un tel serveur localement sur votre système en utilisant le programme `named`, dans ce cas il aura pour adresse 127.0.0.1. À moins que vous ne *deviez absolument* utiliser votre propre serveur de noms, nous vous suggérons d'utiliser celui fourni par votre réseau (s'il existe, ce n'est pas obligatoire). La configuration de `named` est très complexe et sort du cadre de ce guide, dont le but est uniquement de vous permettre de vous connecter et d'utiliser le réseau. Vous pourrez toujours plus tard revenir sur la résolution des adresses si vous en ressentez le besoin.

Utilisateurs de SLIP ou PPP: Vous pouvez ou non avoir besoin de certaines des informations précédentes, à l'exception de l'adresse du serveur de noms. Lors de l'utilisation de SLIP ou PPP, votre adresse IP peut être déterminée de deux façons: Soit vous avez une adresse « statique » , qui est la même chaque fois que vous vous connectez au réseau, soit cette adresse est « dynamique » , c'est à dire qu'elle vous est attribuée à partir d'un groupe d'adresses disponibles lorsque vous vous connectez au serveur. Ces détails seront expliqués dans la section concernant la configuration de SLIP, un peu plus loin.

L'implémentation du réseau sous Linux est très complète. Nous avons décrit ci-dessus la configuration TCP/IP de base la plus courante ; la vôtre peut être assez différente. En cas de doute, consultez votre administrateur réseau et lisez les pages de manuel de `route` et `ifconfig`. La configuration complète d'un réseau TCP/IP sort vraiment du cadre de ce livre, nous nous bornons ici à vous permettre de débuter correctement.

5.3.2.2 Les scripts de démarrage réseau

Ce sont des fichiers shell-scripts de configuration générale lancés lors du démarrage du système par `init`, qui lancent les programmes système de base comme `sendmail`,`cron`, et configurent des paramètres comme les adresses réseau, le nom de la machine, et bien d'autres choses. Comme nous l'avons déjà dit, ils peuvent se trouver à différents endroits selon la distribution de Linux en votre possession, et avoir plusieurs formes. Souvent on les trouve dans le répertoire `/etc/rc.d`, mais ils peuvent simplement se trouver dans `/etc`, et se réduire à un seul script. Nous considérerons ici le cas le plus fréquent rencontré sous Linux, le répertoire `/etc/rc.d`.

Nous allons donc décrire ici les fichiers `rc` (scripts de démarrage) utilisés pour configurer TCP/IP. Vous en trouverez très souvent deux : `rc.inet1` et `rc.inet2`. `rc.inet1` s'occupe de configurer les paramètres réseau de base (comme les adresses IP et le routage) et `rc.inet2` lance les programmes réseau comme `telnetd`, `ftpd`, et ainsi de suite.

Beaucoup de systèmes combinent ces deux fichiers en un seul, parfois appelé `rc.inet` ou `rc.net`. Les noms donnés à ces fichiers `rc` n'ont aucune importance, du moment qu'ils effectuent les bonnes opérations et qu'ils sont exécutés par `init` lors du démarrage du système. Pour vous en assurer, vous pourrez avoir besoin de modifier `/etc/inittab` ou `/etc/rc`, et dans le pire des cas vous aurez à créer vous même `rc.inet1` et `rc.inet2`.

Comme nous l'avons dit, `rc.inet1` configure les interfaces réseau. Ceci comprend votre adresse IP et réseau, et les informations de routage nécessaires. Les tables de routage sont utilisées pour diriger les paquets réseau entrants et sortants vers les autres machines. Sur les configurations les plus simples, vous avez trois routes: L'une pour envoyer des paquets vers votre propre machine, une autre à destination des autres machines de votre réseau, et une dernière pour diriger les paquets vers le monde extérieur (par la passerelle). Deux programmes sont nécessaires pour configurer ces paramètres : `ifconfig` et `route`.

La commande `ifconfig` permet de configurer l'interface réseau avec les valeurs nécessaires (adresse IP, masque réseau, etc), et `route` permet de créer et modifier des entrées dans la table de routage du système.

Dans la plupart des cas, un script `rc.inet1` comme celui-ci devrait fontionner. Vous devrez bien sûr éditer de fichier en fonction de votre propre système. N'utilisez *surtout pas* les exemples d'adresses IP et réseau indiquées ici pour votre propre machine : Elles appartiennent à une machine existant réellement sur l'Internet.

```
#!/bin/sh
# Fichier  /etc/rc.d/rc.inet1 -- Configuration des interfaces reseau

# D'abord, configuration du loopback

HOSTNAME=`hostname`

/etc/ifconfig lo 127.0.0.1      # masque reseau par defaut 255.0.0.0
/etc/route add 127.0.0.1        # une route sur nous meme

# Ensuite, configuration de l'interface Ethernet. Si vous n'en n'avez
 pas, supprimez ou commentez toutes les lignes restantes.

# Editez en fonction de votre configuration.
IPADDR="128.253.154.32"         # REMPLACEZ par VOTRE adresse IP
NETMASK="255.255.255.0"         # REMPLACEZ par VOTRE masque reseau
NETWORK="128.253.154.0"         # REMPLACEA par VOTRE adresse reseau
BROADCAST="128.253.154.255"     # REMPLACEZ par VOTRE adresse de broadcast, si
                                # vous en avez une. Sinon, laissez en blanc.
GATEWAY="128.253.154.1"         # REMPLACEZ par VOTRE passerelle.

/etc/ifconfig eth0 ${IPADDR} netmask ${NETMASK} broadcast ${BROADCAST}

# Si vous n'avez pas d'adresse de broadcast, remplacez la ligne ci-dessus par:
# /etc/ifconfig eth0 ${IPADDR} netmask ${NETMASK}

/etc/route add ${NETWORK}

# La ligne suivante n'est necessaire que si vous avez une passerelle, c'est
# a dire que votre reseau est connecte au monde exterieur.
/etc/route add default gw ${GATEWAY} metric 1

# Fin de la configuration Ethernet
```

Vous pourrez avoir besoin de travailler sur ce fichier pour qu'il fonctionne correctement. Tel quel, il devrait être suffisant pour la majorité des configurations réseau simples, mais sûrement pas pour toutes.

`rc.inet2` lance différents serveurs utilisés par l'ensemble de programmes TCP/IP. Le plus important est `inetd`. Le programme `inetd` fonctionne en arrière-plan et écoute différents ports du réseau. Lorsqu'une machine essaie de se connecter à un certain port (par exemple, le port d'entrée `telnet`), `inetd` lance une copie du programme approprié (dans le cas du port `telnet`, `inetd` lance `in.telnetd`). C'est une méthode bien plus simple que de laisser tourner en permanence des copies individuelles de ces programmes : `inetd` lance uniquement ceux qui sont nécessaires, au moment où ils sont demandés.

`syslogd` est le démon [7] qui enregistre toutes les activités du système et de différentes applications dans des fichiers de trace, selon sa configuration précisée dans `/etc/syslogd.conf`. `routed` est un serveur utilisé pour maintenir un routage dynamique. Lorsque votre système tente d'envoyer des paquets vers un autre réseau, il peut avoir besoin d'un routage additionnel pour y parvenir. Le démon `routed` s'occupe alors de manipuler lui-même les tables de routage sans intervention manuelle.

L'exemple de fichier `rc.inet2` que nous vous donnons ci-après, ne lance que le strict minimum de choses. Il y a beaucoup d'autres serveurs disponibles, un certain nombre ont trait à la configuration de NFS. Lorsque vous commencez la configuration de TCP/IP sur un système tout neuf, il est beaucoup plus prudent de réduire le plus possible les programmes utilisés ; une fois que tout fonctionne, vous avez tout loisir de vous attaquer aux parties les plus ardues (comme NFS).

Notez que dans le fichier ci-dessous, nous considérons que tous les démons réseaux se trouvent dans `/usr/etc`. Comme d'habitude, adaptez ces informations à votre propre configuration.

```
#! /bin/sh
# Exemple de fichier /etc/rc.d/rc.inet2 minimal.

# Lancement de syslogd
if [ -f /usr/etc/syslogd ]
then
      /usr/etc/syslogd
fi

# Lancement du super-serveur inetd
if [ -f /usr/etc/inetd ]
then
      /usr/etc/inetd
fi
```

7. Démon est une francisation familière du vocable informatique anglais `daemon`, qui signifie *Disk And Extension MONitor*, c'est-à-dire qui n'est pas invoqué manuellement mais attend en tâche de fond que quelque chose se passe, que quelque condition se produise. Ce terme fut introduit au départ sous CTSS (Compatible Time Sharing System), un ancêtre du système MULTICS, lui même parent d'UNIX (Unix is not Multics), mais ceci nous entraînerait beaucoup trop loin, pour une note de bas de page qui ne fait pas partie de l'ouvrage original, mais qui n'est là que parce que le traducteur est extrêmement consciencieux et tient à ce que vous compreniez bien ce que l'auteur explique. Le dit traducteur remercie Jon Collins et Steve Pate pour l'étymologie de l'acronyme `daemon`.Thank you guys.

```
# Fin !
```

Parmi tous les serveurs additionnels disponibles, vous pouvez lancer `named` par ce fichier. Le démon `named` est le serveur de noms, son rôle est de transformer les adresses IP en noms, et vice versa. Si vous n'avez aucun serveur de noms disponible sur votre réseau, ou si vous voulez fournir des noms locaux aux autres machines dans votre domaine, il peut être nécessaire d'utiliser `named`. Dans la plupart des cas simples, `named` n'est pas utile et vous pourrez vous en passer. Sa configuration est longue et complexe, les lecteurs intéressés doivent se procurer un ouvrage sérieux traitant de l'administration de réseaux TCP/IP.

5.3.2.3 /etc/hosts

Le fichier `/etc/hosts` contient une liste d'adresses IP et les noms des machines (« hostnames ») correspondants. En général, `/etc/hosts` ne contient que les entrées de votre propre machine, et celles de quelques autres systèmes « importants » (comme votre serveur de noms ou votre passerelle). Votre serveur de noms local s'occupera du reste. Si vous n'en avez pas, vous pouvez bien entendu dupliquer sur chaque ordinateur de votre réseau local tous les couples adresses/noms, mais au delà d'une dizaine de lignes, la maintenance d'autant de fichiers `/etc/hosts` devient vite incontrôlable.

Par exemple, si votre machine est `pastis.troquet.com` et a pour adresse IP 128.253.154.32, votre fichier `/etc/hosts` pourra ressembler à ceci :

```
127.0.0.1             localhost
128.253.154.32        pastis.troquet.com  pastis
```

Si vous n'êtes pas en réseau, `/etc/hosts` ne contiendra que la ligne 127.0.0.1, avec à la fois `localhost` et le nom de la machine juste derrière.

5.3.2.4 /etc/networks

Le fichier `/etc/networks` contient les noms et adresses de votre propre réseau, ainsi que de certains autres. Il est utilisé par la commande `route` et vous permet de désigner un réseau par son nom, plutôt que son adresse peu parlante.

Chaque réseau vers lequel vous voulez ajouter une route à l'aide de la commande `route` *doit* avoir une entrée dans `/etc/networks`.

Par exemple,

```
default         0.0.0.0         # route par defaut - obligatoire
loopnet         127.0.0.0       # loopback         - obligatoire
monrezo         128.253.154.0   # Mettez votre propre adresse reseau
```

5.3.2.5 /etc/host.conf

Ce fichier permet de spécifier au système la manière de résoudre les adresses IP en fonction des noms. Si vous utilisez les services d'un serveur de noms, il doit contenir ces deux lignes:

```
order hosts,bind
multi on
```

Elles indiquent aux fonctions bibliothèques de commencer par rechercher les noms dans `/etc/hosts`, et si elles ne trouvent pas ce qu'il leur faut, de demander ensuite ces renseignements au serveur de noms (bind). Le mot-clé `multi` vous permet d'avoir plusieurs adresses IP pour un même nom de machine dans `/etc/hosts`.

Si vous n'utilisez pas de serveur de noms, il est préférable que `/etc/host.conf` soit comme ceci :

```
order hosts
multi on
```

Mais ce n'est que préférable, pas obligatoire.

5.3.2.6 /etc/resolv.conf

Ce fichier doit contenir l'adresse de votre serveur de noms (si vous en utilisez un) et votre domaine. Si par exemple vous êtres une machine parfaitement qualifiée sur l'Internet par le nom `pastis.troquet.com`, votre domaine est juste `troquet.com`. Un exemple pratique ; votre machine s'appelle janot.bistro.org et utilise un serveur de noms qui a pour adresse 128.253.154.5; alors votre fichier `/etc/resolv.conf` sera :

```
domain      bistro.org
nameserver  127.253.154.5
```

Vous pouvez spécifier plusieurs serveurs de noms, chacun doit avoir sa propre ligne `nameserver` dans `resolv.conf`.

Vous devez positioner votre nom de machine à l'aide de la commande `hostname`. Ceci se fait généralement depuis `/etc/rc` ou `/etc/rc.local`; recherchez simplement dans tous vos scripts de démarrage l'endroit où cette commande est invoquée. Par exemple, si votre nom complet est `pastis.bistro.com`, la commande à passer sera :

```
/bin/hostname pastis.bistro.com
```

Notez que le binaire `hostname` peut très bien ne pas être dans `/bin` sur votre système.

5.3.2.7 Premiers essais

Une fois tous ces fichiers édités et correctement en place, vous devriez pouvoir réamorcer votre système et tenter d'utiliser le réseau. Cela peut très bien fonctionner du premier coup, mais il y a tellement d'endroits où des problèmes peuvent se poser qu'il vaut mieux tester chaque aspect de la configuration individuellement. (C'est à dire que c'est une très mauvaise idée de commencer par vouloir lancer Mosaic depuis une station connectée à l'Internet en guise de test).

Vous pouvez utiliser la commande `netstat` pour afficher les tables de routage, qui sont la source de la plupart des ennuis rencontrés. La page de manuel de `netstat` décrit la syntaxe exacte de cette commande. Afin de tester la connexion au réseau, nous vous suggérons d'employer un client comme `telnet` pour tenter de vous connecter à des machines d'abord sur votre réseau local, puis en dehors de chez vous si vous êtes relié à d'autres réseaux. Cela vous permettra de cerner les éventuels problèmes (par exemple si vous pouvez vous connecter au monde extérieur et pas aux machines locales, vous pourrez suspecter la valeur de votre masque réseau). Vous pouvez bien sûr utiliser manuellement la commande `route` pour modifier et tester vos tables de routage (vous devrez être `root` pour avoir le droit de faire des modifications).

Vous pouvez aussi vérifier la connectivité en spécifiant directement des adresses IP au lieu des noms de machines. Par exemple, si vous avez des problèmes avec la commande:

```
$ telnet casanis.troquet.com
```

c'est peut être un problème de configuration du serveur de noms, essayez dans ce cas de passer a `telnet` l'adresse IP de la machine en question. Si vous vous connectez de cette façon, vous saurez alors que votre configuration réseau est pratiquement correcte, mais que le problème vient de la spécification de l'adresse de votre serveur de noms. (Ou que cette machine ne figure pas dans `/etc/hosts` si vous n'utilisez pas les services d'un serveur de noms bien sûr).

Le déboguage d'une configuration réseau peut être une tâche très difficile, et nous ne pouvons traiter ce sujet dans ce livre. Si vous ne pouvez pas vous faire aider d'un spécialiste, nous vous recommandons la lecture du *Linux Network Administrators' Guide*.

5.3.3 Configuration de SLIP

SLIP (Serial Line Internet Protocol) vous permet d'utiliser TCP/IP sur un port série, que ce soit sur une ligne de téléphone avec un modem, ou toute liaison asynchrone sur ligne louée. Bien sûr, pour utiliser SLIP vous devrez avoir un accès à un serveur SLIP accessible par téléphone dans votre région. Beaucoup d'universités et d'entreprises proposent un tel accès pour une somme modique[8].

Les deux principaux programmes relatifs à SLIP sont `dip` et `slattach`. Tous deux servent à initialiser une connexion SLIP sur un un port série. Il est *indispensable* d'utiliser l'un ou l'autre de ces programmes pour établir une connexion SLIP; il ne suffit pas de téléphoner au serveur SLIP à l'aide d'un programme de communications quelconque comme `kermit` et d'utiliser les commandes `ifconfig` et `route`. En effet, `dip` et `slattach` exécutent un appel système *ioctl()* spécial pour passer la discipline du périphérique série utilisé en mode SLIP.

Le programme `dip` peut être utilisé pour initialiser un modem, téléphoner au serveur et dialoguer avec lui pour lancer la session SLIP. À l'opposé, `slattach` ne fait rien de tout cela, il se contente de passer la ligne série en mode SLIP. Il est très utile si vous avez une liaison permanente avec votre serveur SLIP et que par conséquent aucun dialogue de numérotation ou d'établissement de session n'est à faire. La plupart des utilisateurs auront donc besoin de `dip`, pour piloter leur modem et téléphoner afin établir la liaison avec leur fournisseur de services.

Vous pouvez également utiliser `dip` pour configurer votre système Linux en tant que serveur SLIP, que d'autres machines pourront appeler et pourront ainsi se connecter à votre réseau local ou au monde extérieur via une seconde carte Ethernet présente dans votre machine. Lisez la documentation de `dip` pour plus d'informations sur le sujet. (Et regardez le programme `sliplogin` qui est beaucoup plus adapté à cet usage).

SLIP est différent d'Ethernet, en ce sens qu'il n'existe seulement que deux machines sur le « réseau »: le serveur SLIP et votre machine. Pour cette raison, on qualifie souvent une liaison SLIP de connexion « point à point » . Une généralisation de cette idée, sous la forme du protocole PPP (point-to-point protocol), est également disponible sous Linux. PPP est en passe de remplacer SLIP sur la plupart des sites utilisant TCP/IP sur ligne série; il offre beaucoup plus de possibilités.

Lorsque vous vous connectez à un serveur SLIP, celui-ci vous attribuera une adresse IP selon (généralement) l'une ou l'autre des deux méthodes suivantes. Soit cette adresse sera « statique » , auquel cas elle sera la même à chaque connexion, soit elle sera « dynamique » , et différente à chaque connexion, ce qui est le cas le plus courant. En principe, le serveur SLIP affiche les valeurs de votre adresse SLIP et de la passerelle lors de la connexion. C'est au programme utilisé pour se connecter de savoir récupérer ces données, `dip` est capable de faire cela et de paramétrer l'interface SLIP en fonction.

La configuration d'une connexion SLIP est essentiellement similaire à tout autre connexion TCP/IP, les principales différences vont être expliquées ci-dessous; reportez vous à la section précédente pour la configuration TCP/IP de base.

8. Rappelons que l'auteur ne vit pas en France...

5.3.3.1 Connexions SLIP avec adresse IP statique à l'aide de `dip`

Si votre serveur vous alloue toujours la même adresse IP, vous pouvez mettre les entrées correspondantes dans votre fichier `/etc/hosts`, et configurer également à demeure `rc.inet2`, `host.conf`, et `resolv.conf`.

Configurez également `rc.inet1` comme expliqué plus haut, mais ne lui faites pas exécuter `ifconfig` et `route` pour cette liaison SLIP; `dip` exécutera lui-même les commandes appropriées pour vous. (Si vous utilisez juste `slattach`, vous devrez par contre prendre en compte la liaison SLIP dans `rc.inet1`; nous allons voir ça).

`dip` *devrait* configurer vos tables de routage correctement lors de la connexion. Dans certains cas hélas, son comportement n'est pas adapté et vous devrez corriger manuellement la chose, ce qui peut se faire par un script lancé automatiquement après `dip`, qui exécutera les commandes `ifconfig` et/ou `route` avec les bons paramètres. Dans la plupart des cas, votre passerelle est l'adresse de votre serveur SLIP. Vous pouvez la connaître à l'avance, ou bien la faire lire par DIP lorsque le serveur l'affiche lors de la connexion. Le script de dialogue de `dip` (décrit ci-dessous) peut récupérer ces informations.

`ifconfig` peut nécessiter l'utilisation de l'argument `pointopoint`, si `dip` n'a pas configuré proprement l'interface. Par exemple, si votre serveur SLIP a pour adresse 128.253.154.2, et que votre adresse est 128.253.154.32, vous pouriez avoir besoin de taper la commande

```
ifconfig sl0 128.253.154.32 pointopoint 128.253.154.2
```

en tant que `root`, après la connexion avec `dip`. La page de manuel de `ifconfig` vous sera très utile.

Notez que les périphériques utilisés pour SLIP s'appellent `sl0`, `sl1` et ainsi de suite, alors que dans le cas de l'Ethernet ce sont `eth0`, `eth1`, etc.

Dans la section 5.3.4 ci-dessous, nous expliquerons comment configurer `dip` pour vous connecter à un serveur SLIP.

5.3.3.2 Connexions SLIP avec adresse IP statique à l'aide de `slattach`

Si vous êtes directement relié à votre serveur SLIP par un câble série ou une ligne fixe, vous n'avez pas besoin de `dip` pour initialiser la connexion. La commande `slattach` peut être avantageusement utilisée pour cela.

Dans ce cas, votre fichier `/etc/rc.inet1` devra ressembler à quelque chose du genre:

```
#!/bin/sh
IPADDR="128.253.154.32"        # Remplacez par votre adresse IP
REMADDR="128.253.154.2"        # Remplacez par l'adresse de votre serveur SLIP
```

```
SERIALDEV="/dev/ttyS0"         # Remplacez par le port serie que vous utlisez

slattach -p cslip -s 19200 $SERIALDEV
/etc/ifconfig sl0 $IPADDR pointopoint $REMADDR up
/etc/route add default gw $REMADDR
```

`slattach` alloue la première interface SLIP libre (`sl0`, `sl1`, ...), à la ligne série spécifiée.

Le premier paramètre passé à `slattach` est le protocole SLIP à utiliser. Les seules valeurs possibles sont `slip` et `cslip`. `slip` est le SLIP ordinaire, vous vous en doutiez, et `cslip` correspond a SLIP avec compression des en-têtes; c'est celui à utiliser par défaut car beaucoup plus performant en réponse interactive, sauf si vous vous connectez à un serveur hors d'âge ne supportant pas ce protocole.

Si vous avez plus d'une interface SLIP, vous devrez alors penser aux différents routages désirés; ces décisions dépendant de ce que vous voulez faire et de la configuration actuelle de vos connexions réseau. Un bon livre traitant de la configuration d'un réseau TCP/IP, ainsi que les pages de manuel de `route`, vous seront d'un grand secours.

5.3.3.3 Connexions SLIP avec adresse IP dynamique à l'aide de `dip`

Si votre serveur SLIP vous alloue une adresse dynamiquement à la connexion, vous ne pouvez donc pas la connaître à l'avance, et par conséquent il vous est impossible de la mettre dans `/etc/hosts`. (Mais vous devez bien entendu avoir une entrée pour 127.0.0.1 dans ce fichier).

Beaucoup de serveurs SLIP affichent lors de la connexion l'adresse IP qu'ils vous ont attribué, ainsi que leur propre adresse, par exemple comme ceci:

```
Your IP address is 128.253.154.44.
Server address is 128.253.154.2.
```

`dip` peut détecter ces nombres automatiquement et les utiliser pour configurer l'interface SLIP.

Voyez la section 5.3.3.1 plus haut, afin de savoir comment configurer les différents fichiers relatifs à TCP/IP pour l'utilisation sous SLIP. Ci-dessous, nous allons voir comment utiliser `dip` pour se connecter au serveur SLIP.

5.3.4 Utilisation de `dip`

Le programme `dip` permet de simplifier la procédure de connexion au serveur, le dialogue avec celui-ci et la configuration de l'interface SLIP. Si vous utlisez un modem pour appeler votre fournisseur de services IP, vous devrez utiliser `dip`.

Vous devrez écrire un « script de dialogue » , qui est une liste de commandes utilisées pour communiquer avec le serveur SLIP pour établir la connexion. Ces commandes peuvent envoyer automatiquement votre nom d'utilisateur et votre mot de passe au serveur, et récupérer l'adresse IP qu'il vous assignera.

Voici un exemple d'un tel fichier, pour utiliser un serveur assignant des adresses IP dynamiques; si les adresses sont statiques, il vous suffira d'assigner leurs valeurs aux variables `$local` et `$remote`, au début du sript. (local c'est vous, « remote » c'est le serveur distant). Consultez la page de manuel de `dip` pour plus de détails.

```
main:
  # Ajustage du MTU. (Maximum Transfer Unit). C'est la taille maximum
  # des paquets transmis sur l'interface SLIP. En cas de doute, consultez
  # votre administrateur reseau.
  get $mtu 1500

  # Positionnement de la route par defaut sur la liaison SLIP
  default

  # Initialisation du port serie desire et de la vitesse
  port cua03
  speed 38400

  # Remise a zero du modem et de la ligne. Si ceci vous pose des
  # problemes, supprimez simplement cette ligne.
  reset

  # Preparation a la numerotation, remplacez tout ceci
  # par la chaine d'initialisation necessaire pour votre modem.
  send ATT&C1&D2\\N3&Q5%M3%C1N1W1L1S48=7\r
  wait OK 2
  if $errlvl != 0 goto error
  # Appel du serveur SLIP
  dial 25460000
  if $errlvl != 0 goto error
  wait CONNECT 60
  if $errlvl != 0 goto error

  # Nous sommes connectes. Entree dans le systeme
login:
  sleep 3
  send \r\n\r\n
  # Attente de l'invite login
  wait login: 10
  if $errlvl != 0 goto error

  # Envoi du nom d'utilisateur
  send USERNAME\n

  # Attente de la demande du mot de passe
  wait ord: 5
  if $errlvl != 0 goto error

  # Envoi du mot de passe
  send PASSWORD\n
```

```
  # Attente de l'invite du serveur SLIP (ici serveur de terminaux Annex)
  wait annex: 30
  if $errlvl != 0 goto error

  # Envoi de la commande necessaire au serveur
  send slip\n
  wait Annex 30

  # Recuperation de l'adresse IP affichee par le serveur.
  # Les commandes `get...remote' lisent un texte de la forme
  # xxx.xxx.xxx.xxx, et assignent ces valeurs aux variables donnees
  # en second argument (ici, $remote).
  get $remote remote
  if $errlvl != 0 goto error
  wait Your 30

  # Recuperation de l'adresse IP  qui nous est assignee par le serveur
  get $local remote
  if $errlvl != 0 goto error

  # Etablissement de la connexion SLIP
done:
  print CONNECTE a  $remote avec adresse  $local
  mode SLIP
  goto exit
error:
  print Connexion SLIP ratee.

exit:
```

`dip` exécutera automatiquement les commandes `ifconfig` et `route` avec les valeurs des variables `$local` and `$remote`. Ici, ces variables sont initialisées par les commandes `get...remote`, qui récupèrent le texte affiché par le serveur SLIP.

Si ces commandes que `dip` exécute pour vous ne fontionnent pas correctement, vous pouvez soit les faire exécuter par un script après l'exécution de `dip`, soit modifier le source de `dip` pour lui faire faire ce que vous désirez (si vous savez programmer un tant soit peu bien sûr). Vous pouvez lancer `dip` avec l'argument `-v` pour qu'il affiche pendant la tentative de connexion toutes sortes d'informations de déboguage qui vous aideront à cerner les éventuels problèmes.

Maintenant, en considérant que le script de dialogue ci-dessus a été sauvé dans un fichier nommé `mondialogue`, vous devrez pouvoir établir la connexion SLIP en tapant une commande comme :

```
dip -v mondialogue 2>&1
```

Tout ce que nous venons de voir devrait être suffisant pour vous permettre de commencer à utiliser le réseau sous Linux, que ce soit par l'Ethernet ou par SLIP. Nous insistons encore une fois sur la nécessité de posséder un livre spécialisé sur l'administration réseau, particulièrement si votre configuration demande des routages particuliers que nous ne pouvons décrire ici.

5.4 Transferts de données par UUCP

UUCP (UNIX-to-UNIX Copy) est une très ancienne méthode de communication destinée à transférer des données entre systèmes UNIX. Avec UUCP, les machines se téléphonent entre elles par l'intermédiaire d'un modem, et transfèrent automatiquement des fichiers, du courrier électronique, des messages Usenet. Si vous ne pouvez accéder à un réseau TCP/IP ouvert sur le monde, vous pouvez utiliser UUCP pour communiquer, et recevoir votre courrier de cette manière.

Le *Linux Network Administrator's Guide* décrit en détail la configuration et l'utilisation de UUCP sous Linux. Le document « Linux UUCP HOWTO » est également très utile. Le livre de référence, si vous désirez connaître tous les détails, s'appelle *Managing UUCP and USENET*, par Tim O'Reilly et Grace Todino ; voyez l'annexe A pour plus d'informations.

5.5 Courrier électronique

Comme la plupart des systèmes UNIX, Linux possède toute une panoplie d'applications destinées au courrier électronique. Ce peut être simplement du courrier local (c'est à dire limité à votre machine entre ses différents utilisateurs) ou en réseau (votre courrier est transporté par TCP/IP ou UUCP, vers d'autres machines). Le courrier électronique consiste généralement en deux couches: le « mailer » et le transport. Le « mailer » est le programme utilisateur permettant de composer, lire et gérer les messages, parmi les plus populaires on peut citer `elm` et `mailx`. Le transport s'occupe, par le biais de différents programmes de diriger et distribuer les messages vers les machines destinataires, selon divers protocoles. L'utilisateur ne voit jamais la couche transport ; mais en tant qu'administrateur système vous devez en comprendre le fonctionnement afin de savoir configurer corrrectement le courrier électronique sur votre machine.

Sous Linux, beaucoup choisissent d'utiliser `Smail` pour le transport : ce programme est très simple à configurer, et est capable de gérer à la fois du courrier local, ou distant par TCP/IP et/ou UUCP. L'agent de transport le plus puissant, `sendmail`, qui est universellement utilisé sous UNIX fonctionne bien entendu parfaitement sous Linux. Néanmoins, en raison de sa lourdeur de configuration, il a beaucoup moins de succès dans la communauté Linux.

Le « Linux Mail HOWTO » , document régulièrement mis à jour et diffusé sur les sites Linux, fait le point sur les différents programmes de courrier électronique, et leur configuration. Ce fichier vous guidera dans la configuration de votre système Linux pour qu'il puisse échanger du courrier aussi bien local, que par UUCP ou TCP/IP. (cf annexe A).

5.6 Les News Usenet

Linux supporte tous les programmes UNIX destinés à gérer les News Usenet. Si votre machine est isolée, vous pouvez néanmoins choisir d'installer un serveur local, qui permettra à vos utilisateurs de poster des « articles » dans différents « forums » , c'est une méthode de discussion très efficace. Mais si vous avez accès au monde extérieur, que ce soit par TCP/IP ou UUCP, vous pouvez participer à Usenet, le service de forums mondiaux s'appuyant largement sur le réseau Internet.

Les programmes de gestion des News se décomposent en deux parties : le serveur, et le client. Le serveur est l'application qui contrôle les différents forums et s'occupe de gérer les différents articles, c'est à dire les distribuer aux autres systèmes, les offrir aux programmes lecteurs, et effectuer un certain nombre de tâches administratives automatiquement. Le client, ou *lecteur*, est l'application utilisateur qui se connecte à ce serveur et autorise les utilisateurs à lire et poster des articles confortablement.

Il existe principalement deux serveurs de News sous UNIX, et par conséquent sous Linux. Ils sont tous deux conformes aux mêmes protocoles et totalement compatibles, bien entendu; il s'agit de « C News » et « INN » . Il existe de très nombreux lecteurs, comme par exemple `rn` et `tin`. Le choix du lecteur est une affaire de goût, ils fonctionnent tous sous Linux quel que soit le serveur utilisé.

Si vous ne désirez qu'une configuration locale (c'est à dire, ne pas faire partie de Usenet), vous devrez absolument installer un serveur sur votre machine, ainsi qu'un lecteur pour vos utilisateurs. Le serveur gèrera les articles dans un répertoire comme `/usr/spool/news`, et le lecteur sera configuré pour aller lire ces fichiers dans ce répertoire.

Sinon, pour avoir les News de manière normale, vous avez plusieurs possibilités. L'utilisation par réseau TCP/IP utilise un protocole nommé NNTP (Network News Transmission Protocol), qui permet aux programmes lecteurs de se connecter au serveur par l'intermédiaire du réseau, sur n'importe quelle machine distante. NNTP permet également au serveur d'envoyer les articles aux autres sites, c'est là dessus que Usenet est construit. La plupart des grandes compagnies et universités possèdent un ou plusieurs serveurs de News, paramétré pour gérer la totalité de Usenet pour ce site[9]. Toutes les autres machines de ce site utilisent un simple lecteur se connectant en NNTP au serveur, ce qui signifie que seul le système supportant le serveur NNTP stocke les articles sur son disque dur.

Voici quelques scénarios possibles pour la configuration des News.

- Vous êtes isolé du monde et voulez un système totalement local. Dans ce cas, vous devez installer un serveur C-News ou INN sur votre machine, et un lecteur pour lire les articles localement.

9. L'auteur ne peut bien sûr pas connaître la situation particulière en France et dans certains pays européens, en matière de réseaux informatiques et de tarification des télécommunications.

- Vous avez accès à un réseau TCP/IP et un serveur NNTP. Vous pouvez alors lire et écrire des articles depuis votre machine Linux, simplement en installant un lecteur fonctionnant en NNTP. Dans ce cas, vous n'avez pas besoin d'installer de serveur et d'espace disque pour conserver les articles, le lecteur se connecte par réseau au système distant qui s'occupe de tout cela pour vous. Bien sûr, cela implique d'avoir TCP/IP et l'accès au réseau correctement configurés (voir section 5.3).

- Vous avez accès à un réseau TCP/IP mais pas de serveur NNTP. Vous pouvez dans ce cas installer un serveur de News NNTP sur votre machine Linux, qui fournira le service à toutes les machines du réseau, (y compris la vôtre) qui devront simplement avoir un lecteur de News configuré pour communiquer par NNTP. Le serveur pourra également communiquer avec d'autres serveurs NNTP si votre réseau est connecté à l'Internet pour transférer les News Usenet.

- Vous n'avez pas de possibilité réseau, mais pouvez communiquer avec le reste du monde par UUCP. Dans ce cas vous devez installer un serveur de News local sur votre machine, et vous pourrez le configurer pour qu'il émette et reçoive les articles Usenet par UUCP, en connexion avec un site distant vous offrant ce service. Le transfert se fait en échangeant à intervalles réguliers des archives compressées contenant les articles Usenet regroupés par paquets. Vous devrez installer un lecteur local pour accéder aux articles.

Pour un utilisateur final, l'utilisation de la plupart des lecteurs est simple et conviviale, à la portée de n'importe qui.

Par contre, l'installation et la configuration des programmes est un travail complexe d'administration système. Les applications doivent être compilées en fonction de l'environnement dans lequel elles seront utilisées et il faut parfaitement maîtriser leur fonctionnement, ainsi que posséder de bonnes bases de programmation en langage C pour mener à bien cette opération. Il est dans l'état actuel des choses, impossible de fournir un serveur de News « clé en main » sous forme binaire, prêt à fonctionner. Une fois installé, la maintenance est très réduite et le serveur rendra service de nombreuses années.

Pour plus d'informations, consultez le « Linux News HOWTO » , ainsi que le « *Linux Network Administrator's Guide* » du groupe de documentation Linux. Du côté commercial, le livre *Managing UUCP and Usenet*, de Tim O'Reilly et Grace Todino, est un excellent guide pour l'installation et la configuration de UUCP et des programmes de News. Pour comprendre de quoi il s'agit si vous débutez, vous pouvez lire « How to become a USENET site » , qui est un texte publié tous les mois dans différents forums Usenet, que vous trouvez à coup sûr en lisant le groupe `news.answers`.

Annexe A

Sources d'informations sur Linux

Cette annexe recense diverses sources d'informations sur Linux, comme les textes disponibles en téléchargement et les différentes publications. Beaucoup de ces documents existent à la fois sous forme imprimée, et sous forme électronique sur l'Internet et certains BBS. La plupart des distributions de Linux contiennent l'essentiel de cette documentation, aussi si vous avez installé Linux il est probable que vous trouviez ces fichiers dans un répertoire de votre disque dur.

La quasi totalité des documents décrits ici sont rédigés en anglais. Pour l'instant, seul l'ouvrage que vous avez entre les mains existe en version française, un projet d'internationalisation de Linux est en cours mais la France est très peu représentée, par rapport aux autres pays.

A.1 Documents à télécharger

Ces documents sont disponibles sur tous les sites FTP qui archivent Linux (liste en annexe C). Si vous n'avez pas d'accès Internet, vous les trouverez sur les BBS diffusant Linux, et si vous pouvez faire du courrier électronique, il vous est possible de les recevoir par `ftpmail`. Voyez l'annexe C pour plus d'informations.

En particulier, vous trouverez en France tous ces documents sur `ftp.ibp.fr` dans le répertoire `/pub/linux/docs`. Ce site propose également certains textes concernant Linux, rédigés en français, dans le répertoire `/pub/linux/french`.

Certains sites proposent l'accès à ces documents par `gopher` ou `WWW`. Lisez régulièrement les forums Usenet dédiés à Linux pour trouver des informations à ce sujet.

La plupart de ces documents sont disponibles sous plusieurs formats, selon vos préférences : Texte ASCII pur, sources TeX, HTML, fichier dvi, postscript, grâce à un système de réalisation de documents basé sur un marquage SGML, qui a été constitué spécialement pour les documentations Linux par Matt Welsh.

The Linux Frequently Asked Questions List
: Il s'agit de la liste des questions les plus fréquemment posées à propos de Linux, avec les réponses bien entendu. Ce document est destiné à fournir un guide de secours pour tout problème courant rencontré sous Linux, chaque nouvel utilisateur du système *doit absolument* lire ou au moins posséder ce document à portée de main. Il est maintenu par Ian Jackson, `ijackson@nyx.cs.du.edu`.

Linux META-FAQ
: C'est une collection de « meta-questions » à propos de Linux, c'est à dire des pointeurs sur les sources d'informations ou d'autres sujets très généraux. C'est un document utile pour les utilisateurs d'Internet cherchant comment trouver des renseignements sur le système. Il est maintenu par Michael K. Johnson, joignable à l'adresse `johnsonm@sunsite.unc.edu`.

Linux INFO-SHEET
: Il s'agit d'un texte de présentation de Linux. Il donne un aperçu de ce qu'est et peut faire le système, c'est un genre de prospectus très utile pour répondre à la traditionnelle question : « Linux, c'est quoi ? » . Maintenu par Michael K. Johnson.

Linux Software Map
: Connu sous le nom de "LSM", c'est une liste qui recense les applications disponibles pour Linux, où les trouver, qui en est responsable, et bien d'autres renseignements. Elle est réalisée en ASCII selon un format pouvant être traité par des méthodes informatisées. Cette liste est très loin d'être complète, il serait impossible de réaliser une liste contenant tout ce qui peut tourner sous Linux. Néanmoins, si vous cherchez une application, c'est un bon document de départ. Il est maintenu par Lars Wirzenius, `lars.wirzenius@helsinki.fi`.

Linux HOWTO Index
: Les « HOWTO » Linux sont un ensemble de documents, décrivant « comment faire » dans tel ou tel domaine. Chacun décrit un aspect particulier du système avec beaucoup de détails. Ils sont maintenus par Matt Welsh, `mdw@sunsite.unc.edu`. Ils sont fréquemment remis à jour pour refléter les derniers perfectionnements du système, et sont postés plusieurs fois par mois sous forme ASCII dans les forums Usenet dédiés à Linux. Cet « HOWTO In-

dex » est une liste de tous les documents disponibles dans cette série ; certains (mais pas tous) sont cités ci-dessous.

Linux Installation HOWTO
Il décrit comment se procurer et installer une distribution de Linux, et est assez semblable (avec plus de détails) à ce que nous vous avons présenté dans le chapitre 2.

Linux Distribution HOWTO
Ce document est une liste des différentes distributions binaires de Linux disponibles par FTP anonyme ou proposées en vente par correspondance par différents distributeurs. L'annexe B contient une liste non exhaustive de certains d'entre eux.

Linux XFree86 HOWTO
Vous trouverez dans ce guide les détails nécessaires à l'installation de X Window sous Linux, nous avons survolé X11 dans la section « 5.1 » .

Linux Mail, News, and UUCP HOWTOs
Ces trois documents détaillent l'installation et la configuration du courrier électronique, News Usenet, et UUCP sous Linux. Comme ces trois sujets sont intimement liés, nous vous conseillons de lire l'ensemble de ces documentations.

Linux Hardware HOWTO
Contient une liste sans cesse grandissante du matériel supporté par Linux. Bien que ce soit loin d'être complet, vous pourrez vous faire une idée assez précise des périphériques que Linux peut ou pourra gérer.

Linux SCSI HOWTO
Il s'agit d'un guide complet décrivant la configuration et l'utilisation des périphériques SCSI sous Linux, comme les disques durs, lecteurs de bandes et CD-ROM.

Linux NET-2-HOWTO
Ce document décrit l'installation et la configuration de TCP/IP sous Linux, que ce soit par Ethernet ou ligne série. Si vous devez connecter votre système à un réseau, vous *devez* lire cette documentation. La racine « NET-2 » provient de la version du code réseau implémenté dans Linux. Etant donné qu'il s'appelle maintenant NET-3, le nom de ce document changera sans doute un jour ou l'autre pour devenir plus général.

Linux Ethernet HOWTO
Très lié au précédent, ce texte décrit les différents périphériques

Ethernet supportés par Linux, et explique comment configurer chacun d'eux pour les utiliser correctement sous Linux.

Linux Printing HOWTO
Vous guidera dans la configuration des programmes d'impression sous Linux, comme `lpr`. L'impression et l'usage des imprimantes peuvent paraître complexes sous UNIX, ce document devrait vous éclairer sur le sujet.

Autres documents disponibles
Si vous avez jeté un œil au répertoire `docs` de n'importe quel site diffusant Linux, vous aurez constaté qu'il contient bien plus de fichiers qu'il n'en est décrit ici. Ce sont des documents difficiles à classer qui traitent de sujets divers. Si vous ne trouvez pas les informations que vous recherchez dans ceux que nous venons de citer, regardez tous les autres, il y aura sûrement de quoi vous dépanner.

Une grande partie des documents ci-dessus sont disponibles en versions françaises sur le serveur `ftp.ibp.fr` et la plupart des sites diffusant Linux en France.

A.2 Manuels du groupe de documentation Linux

Le groupe de documentation Linux travaille à la réalisation d'un ensemble de livres et documentations diverses sur Linux, y compris les pages de manuel du système. Ces ouvrages sont plus ou moins avancés, et toute aide à leur sujet sera grandement appréciée. Pour toute question à propos du « Linux Documentation Project » , contactez Matt Welsh (`mdw@sunsite.unc.edu`).

Ces livres sont disponibles par ftp anonyme sur la plupart des sites archive Linux, y compris `ftp.ibp.fr`, dans le répertoire `/pub/Linux/docs/linux-doc-project`. Quelques éditeurs vendent des copies imprimées de ces ouvrages; dans l'avenir vous pourrez probablement les trouver chez votre libraire habituel.

Linux Installation and Getting Started, par Matt Welsh
Un guide destiné au nouvel utilisateur de Linux, expliquant tout ce que ce nouveau venu a besoin de savoir pour se lancer dans l'aventure. C'est celui que vous êtes en train de lire !

The Linux System Administrators' Guide, par Lars Wirzenius
Il s'agit d'un guide très complet sur la configuration et l'utilisation d'un système Linux. L'administration système sous Linux n'est pas différente des autres UNIX, en ce sens qu'il existe malgré tout quelques spécificités qui ne peuvent être abordées dans un

ouvrage général. Ce livre est là pour vous aider à gérer un système Linux, avec tous les petits détails que vous devez connaître.

The Linux Network Administrators' Guide, par Olaf Kirch
Un guide détaillé du réseau sous Linux, comprenant TCP/IP, UUCP, SLIP et bien d'autres choses. Il contient une foule d'informations sur de nombreux sujets, et explique les points obscurs ou confus de la configuration d'un réseau.

The Linux Kernel Hackers' Guide, by Michael Johnson
Tous les secrets du noyau et du développement bas niveau sous Linux. Linux est unique, car les sources de tout le système sont disponibles gratuitement; cet ouvrage guidera le développeur désirant modifier le noyau ou lui ajouter des fonctionnalités, ou fera découvrir au programmeur le cœur d'un système d'exploitation.

A.3 Livres et autres publications

Le *Linux Journal* est un mensuel entièrement dédié à Linux. Il est distribué dans le monde entier, et c'est le meilleur moyen de se tenir au courant de la vie de la communauté Linux, particulièrement si vous n'avez pas d'accès à Usenet.

Á l'heure où nous écrivons ces lignes, l'abonnement d'un an coûte 19 $ aux USA, 24 $ au Canada, et 29 $ partout ailleurs. Tous renseignements en écrivant à *Linux Journal, PO Box 85867, Seattle, WA, 98145-1867, USA*. Leur adresse électronique est `linux@ssc.com`.

Comme nous l'avons dit, il existe très peu de livres traitant spécifiquement de Linux. Toutefois, si vous êtes nouveau dans le monde UNIX, ou si vous voulez plus d'informations qu'il ne vous en est présenté ici, nous vous suggérons la lecture des quelques ouvrages qui suivent.

A.3.1 Ouvrages dédiés à Linux

Titre :	*Le système Linux*
Auteur :	Matt Welsh & Lar Kaufman
Éditeur :	O'Reilly and Associates, 1995
ISBN :	??,

Traduction française de *Running Linux*, plus de 600 pages qui vous initieront à Linux, ainsi qu'aux applications les plus courantes. Une bonne partie est une amélioration de l'ouvrage gratuit que vous tenez entre les mains ; dans le reste vous apprendrez à utiliser le compilateur

C, le débogueur, Perl, Tcl/TK, à installer votre propre serveur WWW, le courrier électronique, les News Usenet, et bien d'autres choses pratiques. C'est un ouvrage de référence à posséder ; il sera utile même aux utilisateurs d'autres systèmes Unix.

Titre : *Administration réseau sous Linux*
Auteur : Olaf Kirch
Éditeur : O'Reilly and Associates, 1995
ISBN : ??,

Il s'agit de la traduction française de l'ouvrage Network Administration Guide, du groupe de documentation Linux, décrit précédemment. Passionnant.

A.3.2 Utilisation d'UNIX

Titre : *Learning the UNIX Operating System*
Auteur : Grace Todino & John Strang
Éditeur : O'Reilly and Associates, 1987
ISBN : 0-937175-16-1, $9.00

Une excellente introduction aux systèmes UNIX. L'essentiel des informations contenues devrait s'appliquer sans problème à Linux. Nous vous suggérons de lire cet ouvrage si vous êtes un nouveau venu à UNIX et désirez vraiment tirer parti rapidement de votre système.

Titre : *Learning the `vi` Editor*
Auteur : Linda Lamb
Éditeur : O'Reilly and Associates, 1990
ISBN : 0-937175-67-6, $21.95

Un livre entièrement dédié à l'éditeur `vi`, puissant éditeur de texte présent sur chaque système UNIX du monde. Il est souvent indispensable de savoir utiliser `vi`, car il n'est pas toujours possible de disposer d'un autre éditeur.

A.3.3 Administration système

Titre : *Essential System Administration*

Auteur : Æleen Frisch
Éditeur : O'Reilly and Associates, 1991
ISBN : 0-937175-80-3, $29.95

Un livre qui démystifie l'administration système sous UNIX en offrant une introduction claire et pratique aux tâches que toute personne responsable d'un système UNIX doit assumer.

Titre : *TCP/IP Network Administration*
Auteur : Craig Hunt
Éditeur : O'Reilly and Associates, 1990
ISBN : 0-937175-82-X, $24.95

Un guide détaillé sur l'installation et l'utilisation d'un réseau TCP/IP. Bien que ce livre ne soit pas dédié à Linux, 90% de son contenu s'y applique. Associé au « Linux NET-2-HOWTO » et au *Linux Network Administrator's Guide*, il est idéal pour assimiler TCP/IP.

Titre : *Managing UUCP and Usenet*
Auteur : Tim O'Reilly and Grace Todino
Éditeur : O'Reilly and Associates, 1991
ISBN : 0-937175-93-5, $24.95

Ce livre explique l'installation et la configuration d'UUCP et des programmes associés, jusqu'à la configuration des News Usenet. Si ces sujets vous intéressent, c'est une lecture indispensable.

A.3.4 X Window

Titre : *The X Window System: A User's Guide*
Auteur : Niall Mansfield
Éditeur : Addison-Wesley
ISBN : 0-201-51341-2, ??

À la fois didacticiel plus que complet et guide de référence pour l'utilisation du Système X Window. Si vous avez installé X sur votre machine Linux, et désirez en tirer le meilleur parti, vous devez lire ce livre. Contrairement à d'autres interfaces graphiques, la puissance offerte par X n'apparaît pas au premier abord.

A.3.5 Programmation

Titre : *The C Programming Language*
Auteur : Brian Kernighan and Dennis Ritchie
Éditeur : Prentice-Hall, 1988
ISBN : 0-13-110362-8, $25.00

C'est l'ouvrage indispensable à toute personne désirant programmer en langage C. Bien qu'il ne soit pas particulièrement orienté UNIX, c'est l'ouvrage de référence à posséder.

Titre : *The Unix Programming Environment*
Auteur : Brian Kernighan and Bob Pike
Éditeur : Prentice-Hall, 1984
ISBN : 0-13-937681-X, ??

Un aperçu de la programmation sous UNIX. Couvre toute la gamme des outils utilisés ; sa lecture permet de se familiariser avec cette discipline parfois déroutante.

Titre : *Advanced Programming in the UNIX Environment*
Auteur : W. Richard Stevens
Éditeur : Addison-Wesley
ISBN : 0-201-56317-7, $50.00

Ce livre contient tout ce que vous devez savoir pour programmer sous UNIX au niveau système : entrées/sorties, communication interprocessus, signaux, etc. Il est orienté sur différents standards UNIX, y compris POSIX.1, auquel Linux se conforme.

A.3.6 Kernel Hacking

Titre : *The Design of the UNIX Operating System*
Auteur : Maurice J. Bach
Éditeur : Prentice-Hall, 1986

ISBN : 0-13-201799-7, ??

Il décrit les algorithmes internes du noyau UNIX. Il n'est pas spécifique à une version particulière d'UNIX, toutefois il tend vers System V. C'est la meilleure introduction que vous puissiez trouver pour comprendre le coeur de Linux.

Titre : *The Magic Garden Explained*
Auteur : Berny Goodheart and James Cox
Éditeur : Prentice-Hall, 1994
ISBN : 0-13-098138-9, ??

Cet ouvrage décrit en détail le noyau Sytem V R4. À l'opposé du livre de Bach, qui se concentre sur les algorithmes internes, celui-ci présente l'implémentation SVR4 de manière plus technique. Bien que Linux et SVR4 ne soient que de lointains cousins, vous pourrez vous faire une idée de ce que représente la réalisation d'un noyau UNIX ; c'est également un livre très récent, publié en 1994.

Annexe B

Distributeurs de Linux

Voici la liste d'un certain nombre de distributeurs vendant Linux sur disquettes, bande magnétique, et CD-ROM ; ou proposant certaines informations relatives à ce système. Beaucoup d'entre eux fournissent également les documentations Linux, du support ou d'autres services. Cette liste est bien entendu loin d'être complète ; si vous avez acheté cet ouvrage sous forme imprimée, il est fort possible que vous puissiez trouver une distribution de Linux chez le même fournisseur.

L'auteur ne garantit pas l'exactitude des informations contenues dans cette annexe. Cette liste n'est fournie que pour rendre service au lecteur, et non pas en tant que publicité.

DP-TOOL club
BP 745
59657 Villeneuve d'Ascq
France

Fintronic Linux Systems
1360 Willow Rd., Suite 205
Menlo Park, CA 94025 USA
Tel: +1 415 325-4474
Fax: +1 415 325-4908
`linux@fintronic.com`

InfoMagic, Inc.
PO Box 30370
Flagstaff, AZ 86003-0370 USA
Tel: +1 800 800-6613, +1 602 526-9565
Fax: +1 602 526-9573
`Orders@InfoMagic.com`

Lasermoon Ltd
2a Beaconsfield Road, Fareham,
Hants, England. PO16 0QB.
Tel: +44 (0) 329 826444.
Fax: +44 (0) 329 825936.
`info@lasermoon.co.uk`

Linux Journal
P.O. Box 85867
Seattle, WA 98145-1867 USA
Tel: +1 206 527-3385
Fax: +1 206 527-2806
`linux@ssc.com`

Linux Systems Labs
18300 Tara Drive
Clinton Twp, MI 48036 USA
Tel: +1 313 954-2829, +1 800 432-0556
Fax: +1 313 954-2806
`info@lsl.com`

Morse Telecommunication, Inc.
26 East Park Avenue, Suite 240
Long Beach, NY 11561 USA
Tel: +1 800 60-MORSE
Fax: +1 516 889-8665
`Linux@morse.net`

Nascent Technology
Linux from Nascent CDROM
P.O. Box 60669
Sunnyvale CA 94088-0669 USA
Tel: +1 408 737-9500
Fax: +1 408 241-9390
`nascent@netcom.com`

Red Hat Software
P.O. Box 4325
Chapel Hill, NC 27515 USA
Tel: +1 919 309-9560
`redhat@redhat.com`

SW Technology
251 West Renner Suite 229

Richardson, TX 75080 USA
Tel: +1 214 907-0871
`swt@netcom.com`

Takelap Systems Ltd.
The Reddings, Court Robin Lane,
Llangwm, Usk, Gwent, United Kingdom NP5 1ET.
Tel: +44 (0)291 650357
Fax: +44 (0)291 650500
`info@ddrive.demon.co.uk`

Trans-Ameritech Enterprises, Inc.
2342A Walsh Ave
Santa Clara, CA 95051 USA
Tel: +1 408 727-3883
`roman@trans-ameritech.com`

Unifix Software GmbH
Postfach 4918
D-38039 Braunschweig
Germany
Tel: +49 (0)531 515161
Fax: +49 (0)531 515162

Yggdrasil Computing, Incorporated
4880 Stevens Creek Blvd., Suite 205
San Jose, CA 95129-1034 USA
Tel: +1 800 261-6630, +1 408 261-6630
Fax: +1 408 261-6631
`info@yggdrasil.com`

Annexe C

Didacticiel FTP et liste des sites

FTP (« File Transfer Protocol ») est un protocole implémenté par différents programmes, utilisé pour transférer des fichiers entre différentes machines sur l'Internet. La plupart des systèmes d'exploitation (comme UNIX, VMS, MS-DOS) possèdent un programme appelé `ftp` qui vous permet le transfert de fichiers, et c'est la meilleure solution pour se procurer Linux si vous avez un accès au réseau. Cette annexe explique les notions de base permettant l'utilisation de ftp; bien entendu, il y a bien plus de fonctions et de possibilités que ce que nous décrirons.

À la fin de cette annexe, nous donnons une liste de sites FTP où vous pourrez trouver à coup sûr Linux. Si vous n'avez pas une connexion directe à l'Internet mais pouvez recevoir du courrier électronique (en quantité), nous présenterons aussi les services `ftpmail`.

Si vous utilisez MS-DOS, UNIX, ou VMS pour télécharger des fichiers sur l'Internet, alors `ftp` est pour vous un programme interactif. Cependant, il existe d'autres implémentations de `ftp` ici et là, comme la version Macintosh (appelée `Fetch`), utilisant une interface à base de menus, très simple d'emploi. Même si vous n'utilisez pas la version standard de `ftp`, les informations contenues ici devraient vous être très utiles.

`ftp` peut être utilisé à la fois pour envoyer et recevoir des fichiers entre différents ordinateurs. Dans la plupart des cas, vous allez surtout avoir besoin de recevoir des programmes. L'Internet fourmille de nombreux **sites archive FTP** publics, qui autorisent le téléchargement à tout le monde. Par exemple, en France, un site comme `ftp.ibp.fr` contient des Mégaoctets d'achives UNIX, DOS, MS-Windows, FreeBSD et est un site **miroir** de Linux et FreeBSD, c'est à dire qu'il rapatrie journellement les nouveaux fichiers des sites de diffusion de référence, les répertoires devenant alors des copies conformes et constamment à jour de ces sites.

C.1 Lancer `ftp`

Notez que les copies d'écran reproduites ci-dessous ne montrent que les informations les plus importantes, et que ce que vous pourrez voir chez vous peut être légèrement différent. Les commandes imprimées en *italiques* représentent ce que vous tapez sur votre clavier.

Pour lancer `ftp` et vous connecter à un site, utilisez simplement la commande

`ftp` ⟨*hostname*⟩

où ⟨*hostname*⟩ est le nom du site auquel vous désirez vous connecter. Par exemple, pour vous connecter à `bidule.truc.com`, nous pouvons utiliser la commande:

```
ftp bidule.truc.com
```

C.2 S'identifier

Lorsque la connexion s'établit vous devriez voir quelque chose ressemblant à:

```
Connected to bidule.truc.com
220 Bidule.truc.com FTPD ready at 15 Dec 1992 08:20:42 EDT
Name (bidule.truc.com:rene):
```

Ici, `ftp` nous demande de nous identifier. Par défaut, il propose `rene`, qui est le nom d'utilisateur sur le système qui appelle. Comme il est fort probable que nous n'ayons pas de compte personnel sur cette machine, il est impossible d'utiliser ce nom. Heureusement, les serveurs d'accès public prévoient un compte universel, accessible à tous: `anonymous`, et la tradition veut que le mot de passe à utiliser soit l'adresse électronique de la personne qui va utiliser ce service. Aussi, nous entrons:

```
Name (bidule.truc.com:rene): anonymous
331-Guest login ok, send e-mail address as password.
Password: rene@renux.frmug.fr.net
230- Welcome to bidule.truc.com.
ftp>
```

Bien sûr vous pouvez mettre autre chose que votre adresse électronique, c'est juste une convention, une manière d'être poli et d'annoncer sa visite. Elle ne s'affichera pas comme ici, puisque c'est une saisie de mot de passe; nous l'avons fait figurer dans un souci de clarté.

C.3 Promenons nous un peu

Bien, nous sommes connectés. Nous avons l'invite `ftp>`, et le programme attend nos commandes. Vous devez connaître quelques commandes élémentaires, tout d'abord

`ls` ⟨*fichier*⟩

et

`dir` ⟨*fichier*⟩

affichent toutes deux les listes de fichiers (où ⟨*fichier*⟩ est un argument optionnel spécifiant un fichier ou répertoire particulier). La différence est que `ls` procure généralement un affichage court alors que `dir` donne plus d'informations sur les fichiers; comme les tailles, dates, etc.

La commande

`cd` ⟨*repertoire*⟩

vous déplacera dans le répertoire en question (exactement comme la commande `cd` de tout système UNIX ou MS-DOS). Vous pouvez utiliser la commande

`cdup`

pour remonter d'un répertoire.

La commande

`help` ⟨*commande*⟩

vous donnera de l'aide sur la commande `ftp` ⟨*commande*⟩ (comme `ls` ou `cd`). Si vous ne spécifiez aucune commande, `ftp` vous affichera la liste de toutes les commandes disponibles.

Si nous tapons `dir` nous verrons le contenu du répertoire initial où nous nous trouvons à l'arrivée sur le système:

```
ftp>dir
200 PORT command successful.
150 Opening ASCII mode data connection for /bin/ls.
total 1337

dr-xr-xr-x  2 root     wheel         512 Aug 13 13:55 bin
drwxr-xr-x  2 root     wheel         512 Aug 13 13:58 dev
drwxr-xr-x  2 root     wheel         512 Jan 25 17:35 etc
drwxr-xr-x 19 root     wheel        1024 Jan 27 21:39 pub
drwxrwx-wx  4 root     ftp-admi     1024 Feb  6 22:10 uploads
drwxr-xr-x  3 root     wheel         512 Mar 11  1992 usr
```

```
226 Transfer complete.
921 bytes received in 0.24 seconds (3.7 Kbytes/s)
ftp>
```

Chacune de ces entrées est un répertoire, comme nous le montre la lettre `d` dans la première colonne. Sur la plupart des sites FTP, le répertoire contenant les fichiers publics est le répertoire `/pub`, aussi allons voir dedans.

```
ftp>cd pub
ftp>dir
200 PORT command successful.
150 ASCII data connection for /bin/ls.
total 846

-rw-r--r--   1 root     staff        1433 Jul 12  1988 README
-r--r--r--   1 3807     staff       15586 May 13  1991 US-DOMAIN.TXT.2
-rw-r--r--   1 539      staff       52664 Feb 20  1991 altenergy.avail
-r--r--r--   1 65534    65534       56456 Dec 17  1990 ataxx.tar.Z
-rw-r--r--   1 root     other     2013041 Jul  3  1991 gesyps.tar.Z
-rw-r--r--   1 432      staff       41831 Jan 30  1989 gnexe.arc
-rw-rw-rw-   1 615      staff       50315 Apr 16  1992 linpack.tar.Z
-r--r--r--   1 root     wheel       12168 Dec 25  1990 localtime.o
-rw-r--r--   1 root     staff        7035 Aug 27  1986 manualslist.tblms
drwxr-xr-x   2 2195     staff         512 Mar 10 00:48 mdw
-rw-r--r--   1 root     staff        5593 Jul 19  1988 t.out.h

226 ASCII Transfer complete.
2443 bytes received in 0.35 seconds (6.8 Kbytes/s)
ftp>
```

Nous voyons déja un certain nombre de fichiers, dont l'un s'appelle `README`, (lisez-moi) qu'il est bon de télécharger (la plupart des sites FTP proposent un tel fichier dans le répertoire `/pub`).

C.4 Télécharger des fichiers

Avant de télécharger, vous devez faire attention à quelques petites choses.

– **Demandez la visualisation du transfert.** Dans ce mode, des « marques » sont affichées à l'écran pendant le transfert, ce qui vous permet de vous faire une idée du temps restant, et de voir que vous n'avez pas été déconnecté du service. En général, les marques consistent en un caractère dièse, (`#`), affiché tous les 1024 ou 8192 octets, selon votre système. Pour mettre ce mode en service, tapez la commande `hash`.

  ```
  ftp>hash
  Hash mark printing on (8192 bytes/hash mark).
  ftp>
  ```

- **Indiquez le type de fichier que vous téléchargez.** Pour FTP, les fichiers peuvent être de deux types: *binary* ou *text* (binaires, ou texte). La plupart de ce que vous allez récupérer consistera en fichiers binaires, toutefois les `README` sont du texte.

  Pourquoi ces deux modes ? Simplement à cause de systèmes comme MS-DOS, où les fichiers texte doivent être modifiés, pour ajouter un caractère retour-chariot à la fin de chaque ligne, afin qu'ils soient lisibles sur ces systèmes. Lors d'un transfert en mode binaire, aucune correction n'est faite bien sûr, les fichiers sont transmis sans aucune modification.

  Les commandes `bin` et `ascii` permettent de positionner le mode de transfert respectivement sur binaire ou texte. *En cas de doute, utilisez toujours le mode binaire.* Si par erreur vous transfériez un fichier binaire en mode texte, vous recevriez un fichier totalement corrompu, inutilisable. (C'est l'une des erreurs les plus courantes lors de l'utilisation de FTP).

  Dans notre exemple, nous voulions prendre le fichier `README`, qui est probablement un fichier texte, nous pouvons donc utiliser la commande

  ```
  ftp>ascii
  200 Type set to A.
  ftp>
  ```

  Notez que si vous transférez depuis un système UNIX vers un système UNIX, ce qui est le cas le plus courant, il y a de grandes chances pour que les fichiers textes disponibles soient normaux, auquel cas vous n'aurez besoin d'aucune correction pour un système particulier et pouvez donc utiliser le mode binaire pour tous les transferts, ce qui vous évitera bien des erreurs.

- **Positionnez votre répertoire local.** Votre *répertoire local* est celui dans lequel vous voulez recevoir les fichiers sur votre système. Tout comme la commande `cd` change le répertoire distant (sur la machine sur laquelle vous êtes connecté en FTP), la commande `lcd` change le répertoire local.

  Par exemple, pour recevoir les fichiers dans `/home/rene/tmp`, tapez la commande

  ```
  ftp>lcd /home/rene/tmp
  Local directory now /home/rene/tmp
  ftp>
  ```

Maintenant, tout est prêt et vous pouvez télécharger le fichier. La commande

`get` ⟨*nom-distant*⟩ ⟨*nom-local*⟩

est utilisée à cet effet, ou ⟨*nom-distant*⟩ est le nom du fichier que vous voulez prendre sur le site FTP et ⟨*nom-local*⟩ est le nom sous lequel vous voulez recevoir ce fichier chez vous. Si vous ne précisez pas ce second argument, le fichier arrivera sous son nom original. Pouvoir changer le nom peut rendre service, si par exemple vous avez déjà un fichier `README`, pour éviter de l'écraser par le nouveau.

Dans notre exemple, pour rapatrier le fichier `README`, nous utiliserons simplement la commande

```
ftp>get README
200 PORT command successful.
150 ASCII data connection for README (128.84.181.1,4527) (1433 bytes).
#
226 ASCII Transfer complete.
local: README remote: README
1493 bytes received in 0.03 seconds (49 Kbytes/s)
ftp>
```

C.5 Quitter FTP

Pour terminer notre session FTP, nous utiliserons la commande

```
quit
```

La commande

```
close
```

peut être utilisée pour couper la connexion avec le site sans sortir du programme, et la commande `open` pourra être utilisée pour aller se connecter ailleurs.

```
ftp>close
221 Goodbye.
ftp>quit
```

C.6 Utilisation de `ftpmail`

`ftpmail` est un service qui permet d'obtenir des fichiers provenant de sites archive FTP, par courrier électronique. Si vous n'avez pas d'accès direct à l'Internet, mais que vous pouvez envoyer et recevoir du courrier, vous pouvez grâce à `ftpmail` télécharger les fichiers que vous convoitez. Malheureusement, `ftpmail` peut être très lent, surtout si vous demandez beaucoup de fichiers. Avant de tenter de rapatrier des archives par `ftpmail`, assurez-vous que vous pourrez recevoir de très gros courriers, et ne serez pas limité pour des raisons techniques, ou financières. En beaucoup d'endroits, le courrier est soumis à des quotas, et si vous les dépassez trop vous pouvez avoir de gros ennuis. Un peu de bon sens, donc.

`sunsite.unc.edu`, l'un des plus importants sites archive Linux, offre un serveur `ftpmail`. Pour l'utiliser, envoyez un courrier électronique à l'adresse

```
ftpmail@sunsite.unc.edu
```

avec dans le corps du message, le simple mot:

```
help
```

Vous recevrez par retour de courrier une liste de commandes `ftpmail` et un petit mode d'emploi pour apprendre à utiliser ce service.

Par exemple, pour avoir une liste des fichiers Linux présents sur `sunsite.unc.edu`, envoyez un courrier à l'adresse citée ci-dessus, contenant le texte suivant :

```
open sunsite.unc.edu
cd /pub/Linux
dir
quit
```

Vous pouvez utiliser le service `ftpmail` pour vous connecter à n'importe quel site FTP dans le monde, néanmoins lisez bien le règlement que vous avez reçu avec le mode d'emploi. Les liaisons internationales peuvent être limitées par l'opérateur pour de nombreuses raisons. [1]

La section suivante donne une liste de différents sites FTP, où vous trouverez toujours les archives de Linux à jour.

C.7 Liste des sites FTP proposant LINUX

Le tableau C.1 est une liste des principaux sites FTP proposant Linux. Notez bien qu'ils ne sont pas les seuls et que beaucoup d'autres sites, peut-être bien plus près de chez vous, proposent probablement aussi le système Linux à jour.

`tsx-11.mit.edu`, `sunsite.unc.edu`, et `nic.funet.fi` sont les sites de référence, où l'essentiel du système est maintenu à jour par les développeurs. `ftp.ibp.fr` est le miroir français de tsx-11. Choisissez toujours un site géographiquement proche de chez vous afin de réduire le coût et de ne pas surcharger les liaisons internationales.

1. **SI VOUS ÊTES EN FRANCE**, préférez un service `ftpmail` français comme `ftpmail@grasp.insa-lyon.fr`. Pour utiliser ce service, écrivez à l'adresse `ftpmail@grasp.insa-lyon.fr`, en mettant comme texte du message : `help french`. Vous recevrez par retour de courrier une aide complète et très claire sur l'utilisation de ce service.

Site name	IP Address	Directory
tsx-11.mit.edu	18.172.1.2	/pub/linux
sunsite.unc.edu	152.2.22.81	/pub/Linux
nic.funet.fi	128.214.6.100	/pub/OS/Linux
ftp.mcc.ac.uk	130.88.200.7	/pub/linux
fgb1.fgb.mw.tu-muenchen.de	129.187.200.1	/pub/linux
ftp.informatik.tu-muenchen.de	131.159.0.110	/pub/Linux
ftp.dfv.rwth-aachen.de	137.226.4.105	/pub/linux
ftp.informatik.rwth-aachen.de	137.226.112.172	/pub/Linux
ftp.ibp.fr	132.227.60.2	/pub/linux
kirk.bu.oz.au	131.244.1.1	/pub/OS/Linux
ftp.uu.net	137.39.1.9	/systems/unix/linux
wuarchive.wustl.edu	128.252.135.4	/systems/linux
ftp.win.tue.nl	131.155.70.100	/pub/linux
ftp.ibr.cs.tu-bs.de	134.169.34.15	/pub/os/linux
ftp.denet.dk	129.142.6.74	/pub/OS/linux

Tab. C.1 - *Sites FTP proposant Linux*

Annexe D

Liste des BBS Linux

Voici ci-dessous une liste de BBS (Bulletin Board Systems) qui diffusent Linux. Cette liste est maintenue par Zane Healy (`healyzh@holonet.net`). Si vous connaissez, ou êtes opérateur d'un BBS qui fournit Linux ne figurant pas sur cette liste, n'hésitez pas à entrer en contact avec lui [1].

La communauté Linux n'est plus uniquement limitée à l'Internet. En fait, certaines statistiques indiquerait que la majorité des utilisateurs de Linux n'ait pas accès à l'Internet. Par conséquent, il est particulièrement important que les BBS continuent d'offrir aide et diffusion du système Linux, dans tous les pays du monde.

D.1 États Unis d'Amérique

Citrus Grove Public Access, 916-381-5822. ZyXEL 16.8/14.4 Sacramento, CA. Internet: `citrus.sac.ca.us`
Higher Powered BBS, 408-737-7040. ? CA. RIME ->HIGHER
hip-hop, 408-773-0768. 19.2k Sunnyvale, CA. USENET access
hip-hop, 408-773-0768. 38.4k Sunnyvale, CA.
Unix Online, 707-765-4631. 9600 Petaluma, CA. USENET access
The Outer Rim, 805-252-6342. Santa Clarita, CA.
Programmer's Exchange, 818-444-3507. El Monte, CA. Fidonet
Programmer's Exchange, 818-579-9711. El Monte, CA.
Micro Oasis, 510-895-5985. 14.4k San Leandro, CA.
Test Engineering, 916-928-0504. Sacramento, CA.

1. En France, il existe un certain nombre de sites, soit BBS soit Linux en accès publics, qui ne sont pas recensés dans cette liste car ce sont des machines de particuliers désirant rester discrets. Vous trouverez les numéros d'appels de ces machines en lisant régulièrement le forum Usenet `fr.comp.os.linux`, qui est également disponible sur les BBS sous le nom de « conférence Linux ».

Slut Club, 813-975-2603. USR/DS 16.8k HST/14.4K Tampa, FL. Fidonet 1:377/42
Lost City Atlantis, 904-727-9334. 14.4k Jacksonville, FL. FidoNet
Aquired Knowledge, 305-720-3669. 14.4k v.32bis Ft. Lauderdale, FL. Internet, UUCP
The Computer Mechanic, 813-544-9345. 14.4k v.32bis St. Petersburg, FL. Fidonet, Sailnet, MXBBSnet
AVSync, 404-320-6202. Atlanta, GA.
Information Overload, 404-471-1549. 19.2k ZyXEL Atlanta, GA. Fidonet 1:133/308
Atlanta Radio Club, 404-850-0546. 9600 Atlanta, GA.
Rebel BBS, 208-887-3937. 9600 Boise, ID.
Rocky Mountain HUB, 208-232-3405. 38.4k Pocatello, ID. Fionet, SLNet, CinemaNet
EchoMania, 618-233-1659. 14.4k HST Belleville, IL. Fidonet 1:2250/1, f'req LINUX
UNIX USER, 708-879-8633. 14.4k Batavia, IL. USENET, Internet mail
PBS BBS, 309-663-7675. 2400 Bloomington, IL.
Third World, 217-356-9512. 9600 v.32 IL.
Digital Underground, 812-941-9427. 14.4k v.32bis IN. USENET
The OA Southern Star, 504-885-5928. New Orleans, LA. Fidonet 1:396/1
Channel One, 617-354-8873. Boston, MA. RIME ->CHANNEL
VWIS Linux Support BBS, 508-793-1570. 9600 Worcester, MA.
WayStar BBS, 508-481-7147. 14.4k V.32bis USR/HST Marlborough, MA. Fidonet 1:333/14
WayStar BBS, 508-481-7293. 14.4k V.32bis USR/HST Marlborough, MA. Fidonet 1:333/15
WayStar BBS, 508-480-8371. 9600 V.32bis or 14.4k USR/HST Marlborough, MA. Fidonet 1:333/16
Programmer's Center, 301-596-1180. 9600 Columbia, MD. RIME
Brodmann's Place, 301-843-5732. 14.4k Waldorf, MD. RIME ->BRODMANN, Fidonet
Main Frame, 301-654-2554. 9600 Gaithersburg, MD. RIME ->MAINFRAME
1 Zero Cybernet BBS, 301-589-4064. MD.
WaterDeep BBS, 410-614-2190. 9600 v.32 Baltimore, MD.
Harbor Heights BBS, 207-663-0391. 14.4k Boothbay Harbor, ME.
Part-Time BBS, 612-544-5552. 14.4k v.32bis Plymouth, MN.
The Sole Survivor, 314-846-2702. 14.4k v.32bis St. Louis, MO. WWIVnet, WWIVlink, etc
MAC's Place, 919-891-1111. 16.8k, DS modem Dunn, NC. RIME ->MAC
Digital Designs, 919-423-4216. 14.4k, 2400 Hope Mills, NC.
Flite Line, 402-421-2434. Lincoln, NE. RIME ->FLITE, DS modem
Legend, 402-438-2433. Lincoln, NE. DS modem
MegaByte Mansion, 402-551-8681. 14.4 V,32bis Omaha, NE.
Mycroft QNX, 201-858-3429. 14.4k NJ.
Steve Leon's, 201-886-8041. 14.4k Cliffside Park, NJ.
Dwight-Englewood BBS, 201-569-3543. 9600 v.42 Englewood, NJ. USENET

The Mothership Cnection, 908-940-1012. 38.4k Franklin Park, NJ.
The Laboratory, 212-927-4980. 16.8k HST, 14.4k v.32bis NY. FidoNet 1:278/707
Valhalla, 516-321-6819. 14.4k HST v.32 Babylon, NY. Fidonet (1:107/255), UseNet (`die.linet.org`)
Intermittent Connection, 503-344-9838. 14.4k HST v.32bis Eugene, OR. 1:152/35
Horizon Systems, 216-899-1086. USR v.32 Westlake, OH.
Horizon Systems, 216-899-1293. 2400 Westlake, OH.
Centre Programmers Unit, 814-353-0566. 14.4k V.32bis/HST Bellefonte, PA.
Allentown Technical, 215-432-5699. 9600 v.32/v.42bis Allentown, PA. WWIV-Net 2578
Tactical-Operations, 814-861-7637. 14.4k V32bis/V42bis State College, PA. Fidonet 1:129/226, `tac_ops.UUCP`
North Shore BBS, 713-251-9757. Houston, TX.
The Annex, 512-575-1188. 9600 HST TX. Fidonet 1:3802/217
The Annex, 512-575-0667. 2400 TX. Fidonet 1:3802/216
Walt Fairs, 713-947-9866. Houston, TX. FidoNet 1:106/18
CyberVille, 817-249-6261. 9600 TX. FidoNet 1:130/78
splat-ooh, 512-578-2720. 14.4k Victoria, TX.
splat-ooh, 512-578-5436. 14.4k Victoria, TX.
alaree, 512-575-5554. 14.4k Victoria, TX.
Ronin BBS, 214-938-2840. 14.4 HST/DS Waxahachie (Dallas), TX. RIME, Intelec, Smartnet, etc.
VTBBS, 703-231-7498. Blacksburg, VA.
MBT, 703-953-0640. Blacksburg, VA.
NOVA, 703-323-3321. 9600 Annandale, VA. Fidonet 1:109/305
Rem-Jem, 703-503-9410. 9600 Fairfax, VA.
Enlightend, 703-370-9528. 14.4k Alexandria, VA. Fidonet 1:109/615
My UnKnown BBS, 703-690-0669. 14.4k V.32bis VA. Fidonet 1:109/370
Georgia Peach BBS, 804-727-0399. 14.4k Newport News, VA.
Top Hat BBS, 206-244-9661. 14.4k WA. Fidonet 1:343/40
victrola.sea.wa.us, 206-838-7456. 19.2k Federal Way, WA. USENET

D.2 Reste du monde

Galaktische Archive, 0043-2228303804. 16.8 ZYX Wien, Austria. Fidonet 2:310/77 (19:00-7:00)
Linux-Support-Oz, +61-2-418-8750. v.32bis 14.4k Sydney, NSW, Austrailia. Internet/Usenet, E-Mail/News
500cc Formula 1 BBS, +61-2-550-4317. V.32bis Sydney, NSW, Australia.
Magic BBS, 403-569-2882. 14.4k HST/Telebit/MNP Calgary, AB, Canada. Internet/Usenet
Logical Solutions, 299-9900 through 9911. 2400 AB, Canada.

Logical Solutions, 299-9912, 299-9913. 14.4k Canada.
Logical Solutions, 299-9914 through 9917. 16.8k v.32bis Canada.
V.A.L.I.S., 403-478-1281. 14.4k v.32bis Edmonton, AB, Canada. USENET
The Windsor Download, (519)-973-9330. v32bis 14.4 ON, Canada.
r-node, 416-249-5366. 2400 Toronto, ON, Canada. USENET
Synapse, 819-246-2344. 819-561-5268 Gatineau, QC, Canada. RIME->SYNAPSE
Radio Free Nyongwa, 514-524-0829. v.32bis ZyXEL Montreal, QC, Canada. USENET, Fidonet
DataComm1, +49.531.132-16. 14.4 HST Braunschweig, NDS, Germany. Fido 2:240/550, LinuxNet
DataComm2, +49.531.132-17. 14.4 HST Braunschweig, NDS, Germany. Fido 2:240/551, LinuxNet
Linux Server /Braukmann, +49.441.592-963. 16.8 ZYX Oldenburg, NDS, Germany. Fido 2:241/2012, LinuxNet
MM's Spielebox, +49.5323.3515. 14.4 ZYX Clausthal-Zfd., NDS, Germany. Fido 2:241/3420
MM's Spielebox, +49.5323.3516. 16.8 ZYX Clausthal-Zfd., NDS, Germany. Fido 2:241/3421
MM's Spielebox, +49.5323.3540. 9600 Clausthal-Zfd., NDS, Germany. Fido 2:241/3422
Bit-Company / J. Bartz, +49.5323.2539. 16.8 ZYX MO Clausthal-Zfd., NDS, Germany. Fido 2:241/3430
Fractal Zone BBS /Maass, +49.721.863-066. 16.8 ZYX Karlsruhe, BW, Germany. Fido 2:241/7462
Hipposoft /M. Junius, +49.241.875-090. 14.4 HST Aachen, NRW, Germany. Fido 2:242/6, 4:30-7,8-23:30
UB-HOFF /A. Hoffmann, +49.203.584-155. 19.2 ZYX+ Duisburg, Germany. Fido 2:242/37
FORMEL-Box, +49.4191.2846. 16.8 ZYX Kaltenkirchen, SHL, Germany. Fido 2:242/329, LinuxNet (6:00-20:00)
BOX/2, +49.89.601-96-77. 16.8 ZYX Muenchen, BAY, Germany. Fido 2:246/147, info magic: LINUX (22-24,0:30-2,5-8)
Die Box Passau 2+1, +49.851.555-96. 14.4 V32b Passau, BAY, Germany. Fido 2:246/200 (8:00-3:30)
Die Box Passau Line 1, +49.851.753-789. 16.8 ZYX Passau, BAY, Germany. Fido 2:246/2000 (8:00-3:30)
Die Box Passau Line 3, +49.851.732-73. 14.4 HST Passau, BAY, Germany. Fido 2:246/202 (5:00-3:30)
Die Box Passau ISDN, +49.851.950-464. 38.4/64k V.110/X.75 Passau, BAY, Germany. Fido 2:246/201 (8:00-24:00,1:00-3:30)
Public Domain Kiste, +49.30.686-62-50. 16.8 ZYX BLN, Germany. Fido 2:2403/17
CS-Port / C. Schmidt, +49.30.491-34-18. 19.2 Z19 Berlin, BLN, Germany. Fido 2:2403/13
BigBrother / R. Gmelch, +49.30.335-63-28. 16.8 Z16 Berlin, BLN, Germany. Fido 2:2403/36.4 (16-23:00)

CRYSTAL BBS, +49.7152.240-86. 14.4 HST Leonberg, BW, Germany. Fido 2:2407/3, LinuxNet
Echoblaster BBS #1, +49.7142.213-92. HST/V32b Bietigheim, BW, Germany. Fido 2:2407/4, LinuxNet (7-19,23-01h
Echoblaster BBS #2, +49.7142.212-35. V32b Bietigheim, BW, Germany. Fido 2:2407/40, LinuxNet (20h-6h)
LinuxServer / P. Berger, +49.711.756-275. 16.8 HST Stuttgart, BW, Germany. Fido 2:2407/34, LinuxNet (8:3-17:5,19-2)
Rising Sun BBS, +49.7147.3845. 16.8 ZYX Sachsenheim, BW, Germany. Fido 2:2407/41, LinuxNet (5:30-2:30)
oytix.north.de, +49.421.396-57-62. ZYX HB, Germany. `mike@oytix.north.de`, login as `gast`
Fiffis Inn BBS, +49-89-5701353. 14.4-19.2 Munich, Germany. FidoNet 2:246/69,Internet,USENET,LinuxNet
The Field of Inverse Chaos, +358 0 506 1836. 14.4k v32bis/HST Helsinki, Finland. USENET; ichaos.nullnet.fi
Modula BBS, +33-1 4043 0124. HST 14.4 v.32bis Paris, France.
Modula BBS, +33-1 4530 1248. HST 14.4 V.32bis Paris, France.
STDIN BBS, +33-72375139. v.32bis Lyon, Laurent Cas, France. FidoNet 2:323/8
Le Lien, +33-72089879. HST 14.4/V32bis Lyon, Pascal Valette, France. FidoNet 2:323/5
Brasil, +33-1-44670844. v.32bis Paris, Laurent Chemla, France.
Cafard Naum, +33-51701632. v.32bis Nantes, Yann Dupont, France.
DUBBS, +353-1-6789000. 19.2 ZyXEL Dublin, Ireland. Fidonet 2:263/167
Galway Online, +353-91-27454. 14.4k v32b Galway, Ireland. RIME, `@iol.ie`
Nemesis' Dungeon, +353-1-324755 or 326900. 14.4k v32bis Dublin, Ireland. Fidonet 2:263/150
nonsolosoftware, +39 51 6140772. v.32bis, v.42bis Italy. Fidonet 2:332/407
nonsolosoftware, +39 51 432904. ZyXEL 19.2k Italy. Fidonet 2:332/417
Advanced Systems, +64-9-379-3365. ZyXEL 16.8k Auckland, New Zealand. Singet, INTLnet, Fidonet
Thunderball Cave, 472567018. Norway. RIME ->CAVE
DownTown BBS Lelystad, +31-3200-48852. 14.4k Lelystad, Netherlands. Fido 2:512/155, UUCP
MUGNET Intl-Cistron BBS, +31-1720-42580. 38.4k Alphen a/d Rijn, Netherlands. UUCP
The Controversy, (65)560-6040. 14.4k V.32bis/HST Singapore. Fidonet 6:600/201
Pats System, +27-12-333-2049. 14.4k v.32bis/HST Pretoria, South Africa. Fidonet 5:71-1/36
Gunship BBS, +46-31-693306. 14.4k HST DS Gothenburg Sweden.
Baboon BBS, +41-62-511726. 19.2k Switzerland. Fido 2:301/580 and /581
The Purple Tentacle, +44-734-590990. HST/V32bis Reading, UK. Fidonet 2:252/305
A6 BBS, +44-582-460273. 14.4k Herts, UK. Fidonet 2:440/111
On the Beach, +444-273-600996. 14.4k/16.8k Brighton, UK. Fidonet 2:441/122

Annexe E

Licence Publique Générale GNU

Voici ci-dessous la Licence Publique Générale GNU (la *GPL* ou *copyleft*), qui protège Linux. Elle est reproduite ici pour éclaircir certaines confusions sur le statut des droits d'auteurs de Linux ; Linux n'est *pas* un partagiciel et n'est *pas* dans le domaine public. Les droits du noyau appartiennent à Linus Torvalds depuis 1993, et le reste des programmes appartient à leurs auteurs respectifs. Donc, Linux *est* protégé, mais vous pouvez toutefois le redistribuer sous les termes de la Licence Publique Générale (GPL) reproduite ici.

ATTENTION: Cette adaptation française n'est pas officielle, elle n'est reproduite ici qu'à titre d'information, mais il n'a été demandé aucune validation de ce texte à la FSF. Pour cette raison, vous trouverez dans l'annexe F page 263 sa version originale en anglais, la seule qui soit réellement reconnue.

LICENCE PUBLIQUE GÉNÉRALE GNU
Version 2, Juin 1991

Copyright ©1989, 1991 Free Software Foundation, Inc. 675 Mass Ave, Cambridge, MA 02139, USA. La copie et la distribution de copies verbatim de ce document est autorisée, mais aucune modification n'est permise.

E.1 Préambule

Les licences d'utilisation de la plupart des éditeurs de logiciels sont destinées à mettre les utilisateurs à la merci de ces éditeurs. À l'opposé, la licence publique générale GNU est destinée à vous garantir la liberté de partager et de modifier les les logiciels librement accessibles, et ainsi de s'assurer que ces programmes sont réellement accessibles sans frais pour tous leurs utilisateurs. Cette Licence Publique Générale s'applique à la majorité des programmes de la Free Software Foundation et à tout autre programme pour lesquels les auteurs ont décidé de l'utiliser. (Quelques autres logiciels sont couverts par la licence publique générale pour bibliothèques GNU à la place). Vous pouvez aussi l'utiliser pour vos propres programmes.

Lorsque nous parlons de « free software », nous entendons « free » dans le sens de « liberté », et non pas de gratuité. Notre licence est conçue pour s'assurer que vous avez la liberté de distribuer des copies des programmes, gratuitement ou non, et que vous recevez ou pouvez obtenir le code source, que vous pouvez modifier les programmes ou en utiliser des parties dans d'autres programmes libres, en sachant que vous pouvez le faire.

Afin de protéger vos droits, nous devons faire des restrictions qui interdisent à quiconque de vous refuser ces droits ou de vous demander d'y renoncer. Ces restrictions vous imposent par conséquent certaines responsabilités si vous distribuez des copies des programmes protégés par la Licence Publique Générale ou si vous les modifiez.

Par exemple, si vous distribuez des copies d'un tel programme, gratuitement ou non, vous devez transmettre aux utilisateurs tous les droits que vous possédez. Vous devez vous assurer qu'ils reçoivent ou qu'il peuvent se procurer le code source. Vous devez leur montrer cette licence afin qu'ils soient eux aussi au courant de leurs droits.

Nous protégeons vos droits en deux étapes : (1) par le copyright du logiciel, et (2) par la délivrance de cette licence qui vous autorise légalement à copier, distribuer et/ou modifier le logiciel.

De plus, pour la protection de chaque auteur et la nôtre, nous voulons nous assurer que chacun comprend bien qu'il n'y a aucune garantie pour ce programme libre. Si le logiciel est modifié par quelqu'un d'autre et redistribué ensuite, nous voulons que tous ceux qui le recevront sachent qu'ils n'ont pas affaire à l'original, de façon que les problèmes introduits par d'autres n'entachent pas la réputation de l'auteur original.

Enfin, tout programme libre est sans cesse menacé par des dépôts de licences. Nous voulons à tout prix éviter que des distributeurs puissent individuellement déposer la licence des logiciels, pour leur propre compte. Pour éviter cela, nous stipulons bien qu'un éventuel dépôt de licence doit prévoir un usage libre pour tous.

Les termes précis et les conditions pour la copie, la distribution et la modification sont les suivants.

E.2 Termes et conditions de copie, distribution et modification

0. Cette licence s'applique à tout programme ou autre travail contenant une notice placée par le possesseur du copyright précisant qu'il peut être distribué selon les termes de cette Licence Publique Générale. Le « programme », désigne soit le programme en lui-même, soit n'importe quel travail qui en est dérivé selon la loi : c'est-à-dire, un ouvrage contenant le programme ou une partie de celui-ci, que ce soit à l'identique ou avec des modifications, et/ou traduit dans une autre langue (à partir de maintenant, nous considérerons donc que le terme « modification » inclut également la « traduction »). Chaque personne à qui s'applique cette licence sera désignée par « Vous ».

   Les activités comme la copie, la distribution et la modification ne sont pas couvertes par cette licence et sortent de son cadre. Le fait d'utiliser le programme n'est pas limité, et les données issues du programme ne sont couvertes que si leur contenu constitue un travail basé sur le logiciel (indépendant ou réalisé en lançant le programme). Tout dépend de ce que le programme est censé faire.

1. Vous pouvez copier et distribuer des copies conformes du code source du programme, tel que vous l'avez reçu, sur n'importe quel support, à condition de placer sur chaque copie un copyright approprié et une limitation de garantie, et de ne pas modifier ou omettre toutes les stipulations se référant à cette licence et à la limitation de garantie, et de fournir avec toute copie du programme un exemplaire de cette Licence Publique Générale GNU.

   Vous pouvez demander une rétribution financière pour l'acte physique de réalisation de la copie, et vous êtes libre de proposer une garantie assurée par vous-même moyennant finances.

2. Vous pouvez modifier votre copie ou vos copies du programme ou toute partie de celui-ci, ou travail basé sur ce programme, et copier et distribuer ces modifications ou ce travail selon les termes de l'article 1, à condition que vous vous conformiez également aux conditions suivantes :

   a. Vous devez rajouter aux fichiers modifiés l'indication très claire de ces modifications, et indiquer la date de chaque changement.

   b. C'est sous les termes de la Licence Publique Générale que vous devez distribuer l'ensemble de toute réalisation contenant tout ou partie du programme, avec ou sans modifications.

   c. Si le programme modifié lit des commandes de manière interactive lors de son exécution, vous devez faire en sorte qu'il affiche, lorsqu'il est lancé normalement, le copyright approprié en indiquant bien la limitation de garantie (ou bien, que vous vous engagez vous-même à fournir une garantie), qu'il stipule que les utilisateurs peuvent librement redistribuer le programme sous ces conditions, et qu'il montre à l'utilisateur comment lire une copie de cette licence. (Exception : si le programme original est interactif mais n'affiche normalement pas un tel message, tout travail dérivé du programme ne sera pas non plus obligé de l'afficher).

Ces conditions s'appliquent à l'ensemble des modifications. Si des éléments identifiables de ce travail ne sont pas dérivés du programme, et peuvent être considérés raisonnablement comme indépendants, alors cette licence ne s'applique pas à ces éléments lorsque vous les distribuez seuls. Mais lorsque vous distribuez ces mêmes éléments comme partie d'un ensemble cohérent dont le reste est basé sur un programme soumis à cette Licence, alors ils sont soumis également á la Licence Publique Générale, qui s'étend ainsi à l'ensemble du produit, quel qu'en soit l'auteur.

Il n'est pas question dans cet article section de s'approprier ou de contester vos droits sur un travail totalement écrit par vous, son but est plutôt de s'accorder le droit de contrôler la libre distribution de tout travail dérivé ou tout travail collectif basé sur le programme.

De plus, toute compilation d'un autre travail avec le programme (ou avec un travail dérivé du programme) sur un support de stockage ou de distribution, ne fait pas tomber cet autre travail sous le contrôle de cette Licence.

3. Vous pouvez copier et distribuer le programme (ou un travail dérivé selon l'article 2) sous forme de code objet ou exécutable, selon les termes des articles 1 et 2 ci-dessus, à condition de respecter les clauses suivantes :

   a. Que la distribution soit accompagnée du code source complet du programme, sous une forme lisible par un ordinateur, et cela selon les termes des articles 1 et 2 ci-dessus, sur un support habituellement utilisé pour l'échange de données ; ou,

   b. Que la distribution contienne une offre écrite, valable pendant au moins les trois prochaines années, de donner à tout tiers qui en fera la demande, une copie sous forme lisible par une machine du code source correspondant, pour un tarif qui ne doit pas être supérieur à ce que vous coûte la copie, selon les termes des articles 1 et 2 ci-dessus, sur un support courant pour l'échange de données informatiques; ou,

   c. Que la distribution soit accompagnée des informations sur l'endroit où le code source peut être obtenu. (Cette alternative n'est autorisée que dans le cas d'une distribution non commerciale, et uniquement si vous avez recu le programme sous forme de code objet ou exécutable avec une telle offre, en accord avec l'alinéa b précédent.)

Le code source d'un travail désigne la forme de cet ouvrage sous laquelle les modifications sont les plus aisées. Ceci désigne la totalité du code source de tous les modules qui composent un programme exécutable, plus tout éventuel fichier de définition associé, ainsi que les scripts utilisés pour effectuer la compilation et l'installation du programme exécutable. Toutefois, il est fait exception de tout ce qui fait partie de l'environnement standard de développement du système d'exploitation utilisé (source ou binaire) comme les compilateurs, bibliothèques, noyau, etc. Sauf si ces éléments sont aussi diffusés avec le programme exécutable.

Si la distribution de l'exécutable ou du code objet consiste à offrir un accès permettant de copier le programme depuis un endroit particulier, alors l'offre d'un accès équivalent pour se procurer le code source au même endroit compte comme une distribution de ce code source, même si l'utilisateur choisit de ne pas profiter de cette offre.

4. Vous ne pouvez pas copier, modifier, céder, déposer, ou distribuer le programme, d'une autre manière que l'autorise la Licence Publique Générale. Toute tentative de copier, modifier, céder, déposer, ou distribuer le programme différemment, annulera immédiatement vos droits d'utilisation du programme sous cette Licence. Toutefois, les tiers ayant reçu de vous des copies du programme ou le droit d'utiliser ces copies, continueront à bénéficier de leur droit d'utilisation tant qu'ils respecteront pleinement les conditions de cette Licence Publique Générale.

5. Vous n'êtes pas obligé d'accepter cette Licence, puisque vous ne l'avez pas signée. Cependant, rien d'autre ne vous autorise à modifier ou distribuer le programme, ou des travaux dérivés. Ces faits sont interdits par la loi, tant que vous n'acceptez pas cette Licence. Par conséquent, en modifiant ou distribuant le programme (ou tout travail basé sur lui), vous indiquez implicitement votre acceptation des termes et conditions de cette Licence.

6. Chaque fois que vous redistribuez le programme (ou tout travail dérivé), le récipiendaire reçoit une licence du détenteur original autorisant la copie, la distribution ou la modification du programme, selon les termes et conditions de la Licence. Vous n'avez pas le droit d'imposer de restriction supplémentaire sur les droits transmis au récipiendaire. Vous n'êtes pas responsable du respect de cette Licence par les tiers.

7. Si, à la suite d'une décision de justice, il vous est imposé d'aller à l'encontre des conditions de cette Licence, cela ne vous dégage pas pour autant des obligations liées à celle-ci. Si vous ne pouvez pas concilier vos obligations légales ou toute autre obligation avec les conditions requises par cette Licence, alors vous ne devez pas distribuer le programme du tout.

   Si une partie quelconque de cet article est rendue invalide ou inapplicable pour quelques raisons que ce soit, le reste de l'article continue à s'appliquer et la totalité de l'article s'appliquera dans toutes autres circonstances.

   Cet article n'a pas pour but de vous pousser à enfreindre des droits ou des dispositions légales ou de contester leur validité, elle n'est là que pour protéger l'intégrité du système de distribution du logiciel en libre accès. De nombreuses personnes ont généreusement contribué à la large gamme de programmes distribuée de cette façon en toute confiance ; il appartient à chaque auteur/contributeur de décider de diffuser ses programmes selon le système de son choix.

8. Si la distribution et/ou l'utilisation du programme est limitée, dans certains pays, soit par des brevets ou des droits sur des interfaces, le propriétaire original des droits qui place le programme sous la Licence Publique Générale peut ajouter explicitement une clause de limitation géographique excluant ces pays particuliers. Dans ce cas, cette clause devient une partie intégrante de cette Licence.

9. La Free Software Foundation peut publier périodiquement des mises à jour ou de nouvelles versions de la Licence Publique Générale. Elles seront écrites dans le même esprit que la présente version, mais pourront différer dans certains détails destinés à clarifier de nouveaux problèmes pouvant survenir.

   Chaque version possède un numéro bien distinct. Si le programme précise un numéro de version de cette Licence et « toute version ultérieure », vous avez le choix de suivre les termes et conditions de cette version ou de toute autre version plus récente publiée par la Free Software Foundation. Si le programme ne spécifie aucun numéro de version, vous pouvez alors choisir d'utiliser n'importe quelle version publiée par la Free Software Foundation.

10. Si vous désirez incorporer des parties du programme dans d'autres programmes libres dont les conditions de distribution diffèrent, écrivez à l'auteur pour lui en demander la permission. Pour les programmes dont le copyright est directement détenu par la Free Software Foundation, écrivez à la Free Software Foundation ; nous faisons quelquefois des exceptions. Notre décision sera guidée à la fois par le but de préserver la liberté de notre programme libre ou de ses dérivés, et par celui de promouvoir le partage et la réutilisation du logiciel en général.

LIMITATION DE GARANTIE

11. PARCE QUE LA LICENCE D'UTILISATION DE CE PROGRAMME EST LIBRE ET GRATUITE, IL N'Y A AUCUNE GARANTIE POUR CE PROGRAMME, DANS LA MESURE PERMISE PAR LA LOI. SAUF SI MENTIONNÉ PAR ÉCRIT, LES DÉTENTEURS DU COPYRIGHT ET/OU LES TIERS FOURNISSENT LE PROGRAMME EN L'ÉTAT, SANS AUCUNE SORTE DE GARANTIE, NI EXPLICITE NI IMPLICITE, Y COMPRIS LES GARANTIES DE COMMERCIALISATION OU D'ADAPTATION DANS UN BUT PARTICULIER. VOUS PRENEZ TOUS LES RISQUES QUANT À LA QUALITÉ ET AUX EFFETS DU PROGRAMME. SI LE PROGRAMME EST DÉFECTUEUX, VOUS ASSUMEZ LE COÛT DE TOUS LES SERVICES, CORRECTIONS OU RÉPARATIONS NÉCESSAIRES.

12. EN AUCUN CAS, À MOINS QUE CE NE SOIT EXPLICITEMENT PRÉVU PAR LA LOI OU ACCEPTÉ PAR ÉCRIT, NI LE PROPRIÉTAIRE DES DROITS, NI TOUTE AUTRE PERSONNE AUTORISÉE À MODIFIER ET/OU REDISTRIBUER LE PROGRAMME COMME IL EST PERMIS CI-DESSUS, NE POURRA ÊTRE TENU POUR RESPONSABLE DE TOUT DOMMAGE DIRECT, INDIRECT, SECONDAIRE OU ACCESSOIRE (Y COMPRIS LES DOMMAGES ENTRAINÉS PAR LA PERTE DE BÉNÉFICE, L'INTERRUPTION D'ACTIVITÉS OU LA PERTE D'INFORMATIONS ET AUTRES DÉCOULANT DE L'UTILISATION OU DE L'IMPOSSIBILITÉ D'UTILISER LE PROGRAMME.

FIN DES TERMES ET CONDITIONS

E.3 Annexe : Comment appliquer ces directives à vos nouveaux programmes

Si vous développez un nouveau programme et désirez qu'il soit utilisé le plus possible par le public, la meilleure méthode est d'en faire un logiciel libre que tout le monde pourra redistribuer et modifier selon ces termes.

Pour cela, introduisez les indications suivantes dans votre programme. Il est préférable et plus sûr de les faire figurer au début de chaque fichier source pour bien préciser les choses ; en tout cas chaque module source devra comporter au minimum la ligne de « copyright » et indiquer où se trouvent toutes les autres indications.

> ⟨*une ligne pour donner le nom du programme et donner une idée de ce qu'il est censé faire.*⟩ Copyright ©19xx ⟨*nom de l'auteur*⟩
>
> Ce programme est libre, vous pouvez le redistribuer et/ou le modifier selon les termes de la Licence Publique Générale GNU telle que publiée par la Free Software Foundation ; soit la version 2 de cette licence, soit (à vous de voir...) toute autre version ultérieure.
>
> Ce programme est distribué dans l'espoir qu'il sera utile, mais SANS AUCUNE GARANTIE, ni explicite ni implicite, y compris les garanties de commercialisation ou d'adaptation dans un but spécifique. Prenez connaissance de la Licence Publique Générale GNU pour plus de détails.
>
> Vous devez avoir reçu une copie de la Licence Publique Générale GNU avec ce programme ; si ce n'est pas le cas, écrivez à la Free Software Foundation, Inc., 675 Mass Ave, Cambridge, MA 02139, USA.

Ajoutez également les informations nécessaires pour vous contacter par courrier électronique ainsi que votre adresse postale.

Si le programme est interactif, faites-lui afficher un court avertissement comme celui-ci lors de son lancement :

```
TroudBall version 69, Copyright (C) 19yy nom de l'auteur
TroudBall est fourni sans AUCUNE GARANTIE ; pour plus de détails tapez 'g'.
Ce programme est libre, et vous êtes encouragés à le redistribuer sous certaines
conditions ; tapez 'c' pour les détails.
```

Les commandes hypothétiques 'g' et 'c' doivent afficher les parties appropriées de la Licence Publique Générale GNU. Bien sûr, vous pouvez appeler les commandes comme bon vous semble, ce peut être aussi des options dans un menu ou des options accessibles par le clic d'une souris, tout dépend de votre programme.

Vous devez aussi demander à votre employeur (si vous travaillez en tant que programmeur) ou votre école, ou autres, de signer une décharge stipulant qu'il renonce aux éventuels droits qu'il pourrait avoir sur le programme, si nécessaire. Voici un petit exemple ; changez simplement les noms :

> Poildur S.A, déclare par la présente ne pas réclamer de copyright sur le programme « TroudBall » écrit par Marcel Dugenou.
>
> ⟨*signature de Monsieur Le Responsable*⟩, 1er Avril 1989
> Monsieur Le Responsable, Directeur Général de Poildur S.A.

Cette Licence Publique Générale ne permet pas d'inclure votre programme dans des logiciels propriétaires. Si votre programme est une fonction de bibliothèque, vous pouvez considérer qu'il est plus utile d'autoriser une édition de liens avec des applications propriétaires et cette bibliothèque. Si c'est ce que vous désirez, utilisez la Licence Générale de Librairie GNU et non celle-ci.

Annexe F

La GPL, texte original

Une traduction française non officielle de cette license se trouve en annexe E, à la page 255.

GNU GENERAL PUBLIC LICENSE
Version 2, June 1991

F.1 Preamble

The licenses for most software are designed to take away your freedom to share and change it. By contrast, the GNU General Public License is intended to guarantee your freedom to share and change free software–to make sure the software is free for all its users. This General Public License applies to most of the Free Software Foundation's software and to any other program whose authors commit to using it. (Some other Free Software Foundation software is covered by the GNU Library General Public License instead.) You can apply it to your programs, too.

When we speak of free software, we are referring to freedom, not price. Our General Public Licenses are designed to make sure that you have the freedom to distribute copies of free software (and charge for this service if you wish), that you receive source code or can get it if you want it, that you can change the software or use pieces of it in new free programs; and that you know you can do these things.

To protect your rights, we need to make restrictions that forbid anyone to deny you these rights or to ask you to surrender the rights. These restrictions translate to certain responsibilities for you if you distribute copies of the software, or if you modify it.

For example, if you distribute copies of such a program, whether gratis or for a fee, you must give the recipients all the rights that you have. You must make sure that they, too, receive or can get the source code. And you must show them these terms so they know their rights.

We protect your rights with two steps: (1) copyright the software, and (2) offer you this license which gives you legal permission to copy, distribute and/or modify the software.

Also, for each author's protection and ours, we want to make certain that everyone understands that there is no warranty for this free software. If the software is modified by someone else and passed on, we want its recipients to know that what they have is not the original, so that any problems introduced by others will not reflect on the original authors' reputations.

Finally, any free program is threatened constantly by software patents. We wish to avoid the danger that redistributors of a free program will individually obtain patent licenses, in effect making the program proprietary. To prevent this, we have made it clear that any patent must be licensed for everyone's free use or not licensed at all.

The precise terms and conditions for copying, distribution and modification follow.

F.2 Terms and Conditions for Copying, Distribution, and Modification

0. This License applies to any program or other work which contains a notice placed by the copyright holder saying it may be distributed under the terms of this General Public License. The "Program", below, refers to any such program or work, and a "work based on the Program" means either the Program or any derivative work under copyright law: that is to say, a work containing the Program or a portion of it, either verbatim or with modifications and/or translated into another language. (Hereinafter, translation is included without limitation in the term "modification".) Each licensee is addressed as "you".

   Activities other than copying, distribution and modification are not covered by this License; they are outside its scope. The act of running the Program is not restricted, and the output from the Program is covered only if its contents constitute a work based on the Program (independent of having been made by running the Program). Whether that is true depends on what the Program does.

1. You may copy and distribute verbatim copies of the Program's source code

as you receive it, in any medium, provided that you conspicuously and appropriately publish on each copy an appropriate copyright notice and disclaimer of warranty; keep intact all the notices that refer to this License and to the absence of any warranty; and give any other recipients of the Program a copy of this License along with the Program.

You may charge a fee for the physical act of transferring a copy, and you may at your option offer warranty protection in exchange for a fee.

2. You may modify your copy or copies of the Program or any portion of it, thus forming a work based on the Program, and copy and distribute such modifications or work under the terms of Section 1 above, provided that you also meet all of these conditions:

    a. You must cause the modified files to carry prominent notices stating that you changed the files and the date of any change.
    b. You must cause any work that you distribute or publish, that in whole or in part contains or is derived from the Program or any part thereof, to be licensed as a whole at no charge to all third parties under the terms of this License.
    c. If the modified program normally reads commands interactively when run, you must cause it, when started running for such interactive use in the most ordinary way, to print or display an announcement including an appropriate copyright notice and a notice that there is no warranty (or else, saying that you provide a warranty) and that users may redistribute the program under these conditions, and telling the user how to view a copy of this License. (Exception: if the Program itself is interactive but does not normally print such an announcement, your work based on the Program is not required to print an announcement.)

These requirements apply to the modified work as a whole. If identifiable sections of that work are not derived from the Program, and can be reasonably considered independent and separate works in themselves, then this License, and its terms, do not apply to those sections when you distribute them as separate works. But when you distribute the same sections as part of a whole which is a work based on the Program, the distribution of the whole must be on the terms of this License, whose permissions for other licensees extend to the entire whole, and thus to each and every part regardless of who wrote it.

Thus, it is not the intent of this section to claim rights or contest your rights to work written entirely by you; rather, the intent is to exercise the right to control the distribution of derivative or collective works based on the Program.

In addition, mere aggregation of another work not based on the Program with the Program (or with a work based on the Program) on a volume of a storage or distribution medium does not bring the other work under the scope of this License.

3. You may copy and distribute the Program (or a work based on it, under Section 2) in object code or executable form under the terms of Sections 1 and 2 above provided that you also do one of the following:

   a. Accompany it with the complete corresponding machine-readable source code, which must be distributed under the terms of Sections 1 and 2 above on a medium customarily used for software interchange; or,
   b. Accompany it with a written offer, valid for at least three years, to give any third party, for a charge no more than your cost of physically performing source distribution, a complete machine-readable copy of the corresponding source code, to be distributed under the terms of Sections 1 and 2 above on a medium customarily used for software interchange; or,
   c. Accompany it with the information you received as to the offer to distribute corresponding source code. (This alternative is allowed only for noncommercial distribution and only if you received the program in object code or executable form with such an offer, in accord with Subsection b above.)

   The source code for a work means the preferred form of the work for making modifications to it. For an executable work, complete source code means all the source code for all modules it contains, plus any associated interface definition files, plus the scripts used to control compilation and installation of the executable. However, as a special exception, the source code distributed need not include anything that is normally distributed (in either source or binary form) with the major components (compiler, kernel, and so on) of the operating system on which the executable runs, unless that component itself accompanies the executable.

   If distribution of executable or object code is made by offering access to copy from a designated place, then offering equivalent access to copy the source code from the same place counts as distribution of the source code, even though third parties are not compelled to copy the source along with the object code.

4. You may not copy, modify, sublicense, or distribute the Program except as expressly provided under this License. Any attempt otherwise to copy, modify, sublicense or distribute the Program is void, and will automatically terminate your rights under this License. However, parties who have received copies, or rights, from you under this License will not have their licenses terminated so long as such parties remain in full compliance.

5. You are not required to accept this License, since you have not signed it. However, nothing else grants you permission to modify or distribute the

Program or its derivative works. These actions are prohibited by law if you do not accept this License. Therefore, by modifying or distributing the Program (or any work based on the Program), you indicate your acceptance of this License to do so, and all its terms and conditions for copying, distributing or modifying the Program or works based on it.

6. Each time you redistribute the Program (or any work based on the Program), the recipient automatically receives a license from the original licensor to copy, distribute or modify the Program subject to these terms and conditions. You may not impose any further restrictions on the recipients' exercise of the rights granted herein. You are not responsible for enforcing compliance by third parties to this License.

7. If, as a consequence of a court judgment or allegation of patent infringement or for any other reason (not limited to patent issues), conditions are imposed on you (whether by court order, agreement or otherwise) that contradict the conditions of this License, they do not excuse you from the conditions of this License. If you cannot distribute so as to satisfy simultaneously your obligations under this License and any other pertinent obligations, then as a consequence you may not distribute the Program at all. For example, if a patent license would not permit royalty-free redistribution of the Program by all those who receive copies directly or indirectly through you, then the only way you could satisfy both it and this License would be to refrain entirely from distribution of the Program.

   If any portion of this section is held invalid or unenforceable under any particular circumstance, the balance of the section is intended to apply and the section as a whole is intended to apply in other circumstances.

   It is not the purpose of this section to induce you to infringe any patents or other property right claims or to contest validity of any such claims; this section has the sole purpose of protecting the integrity of the free software distribution system, which is implemented by public license practices. Many people have made generous contributions to the wide range of software distributed through that system in reliance on consistent application of that system; it is up to the author/donor to decide if he or she is willing to distribute software through any other system and a licensee cannot impose that choice.

   This section is intended to make thoroughly clear what is believed to be a consequence of the rest of this License.

8. If the distribution and/or use of the Program is restricted in certain countries either by patents or by copyrighted interfaces, the original copyright holder who places the Program under this License may add an explicit geographical distribution limitation excluding those countries, so that distribution is permitted only in or among countries not thus excluded. In such case, this License incorporates the limitation as if written in the body of this License.

9. The Free Software Foundation may publish revised and/or new versions of the General Public License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns.

   Each version is given a distinguishing version number. If the Program specifies a version number of this License which applies to it and “any later version”, you have the option of following the terms and conditions either of that version or of any later version published by the Free Software Foundation. If the Program does not specify a version number of this License, you may choose any version ever published by the Free Software Foundation.

10. If you wish to incorporate parts of the Program into other free programs whose distribution conditions are different, write to the author to ask for permission. For software which is copyrighted by the Free Software Foundation, write to the Free Software Foundation; we sometimes make exceptions for this. Our decision will be guided by the two goals of preserving the free status of all derivatives of our free software and of promoting the sharing and reuse of software generally.

NO WARRANTY

11. BECAUSE THE PROGRAM IS LICENSED FREE OF CHARGE, THERE IS NO WARRANTY FOR THE PROGRAM, TO THE EXTENT PERMITTED BY APPLICABLE LAW. EXCEPT WHEN OTHERWISE STATED IN WRITING THE COPYRIGHT HOLDERS AND/OR OTHER PARTIES PROVIDE THE PROGRAM “AS IS” WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. THE ENTIRE RISK AS TO THE QUALITY AND PERFORMANCE OF THE PROGRAM IS WITH YOU. SHOULD THE PROGRAM PROVE DEFECTIVE, YOU ASSUME THE COST OF ALL NECESSARY SERVICING, REPAIR OR CORRECTION.

12. IN NO EVENT UNLESS REQUIRED BY APPLICABLE LAW OR AGREED TO IN WRITING WILL ANY COPYRIGHT HOLDER, OR ANY OTHER PARTY WHO MAY MODIFY AND/OR REDISTRIBUTE THE PROGRAM AS PERMITTED ABOVE, BE LIABLE TO YOU FOR DAMAGES, INCLUDING ANY GENERAL, SPECIAL, INCIDENTAL OR CONSEQUENTIAL DAMAGES ARISING OUT OF THE USE OR INABILITY TO USE THE PROGRAM (INCLUDING BUT NOT LIMITED TO LOSS OF DATA OR DATA BEING RENDERED INACCURATE OR LOSSES SUSTAINED BY YOU OR THIRD PARTIES OR A FAILURE OF THE PROGRAM TO OPERATE WITH ANY OTHER PROGRAMS), EVEN IF SUCH

HOLDER OR OTHER PARTY HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

END OF TERMS AND CONDITIONS

F.3 Appendix: How to Apply These Terms to Your New Programs

If you develop a new program, and you want it to be of the greatest possible use to the public, the best way to achieve this is to make it free software which everyone can redistribute and change under these terms.

To do so, attach the following notices to the program. It is safest to attach them to the start of each source file to most effectively convey the exclusion of warranty; and each file should have at least the "copyright" line and a pointer to where the full notice is found.

> ⟨*one line to give the program's name and a brief idea of what it does.*⟩
> Copyright ©19yy ⟨*name of author*⟩
>
> This program is free software; you can redistribute it and/or modify it under the terms of the GNU General Public License as published by the Free Software Foundation; either version 2 of the License, or (at your option) any later version.
>
> This program is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU General Public License for more details.
>
> You should have received a copy of the GNU General Public License along with this program; if not, write to the Free Software Foundation, Inc., 675 Mass Ave, Cambridge, MA 02139, USA.

Also add information on how to contact you by electronic and paper mail.

If the program is interactive, make it output a short notice like this when it starts in an interactive mode:

```
Gnomovision version 69, Copyright (C) 19yy name of author Gnomovision comes
with ABSOLUTELY NO WARRANTY; for details type 'show w'. This is free software,
and you are welcome to redistribute it under certain conditions; type 'show
c' for details.
```

The hypothetical commands 'show w' and 'show c' should show the appropriate parts of the General Public License. Of course, the commands you use may be called something other than 'show w' and 'show c'; they could even be mouse-clicks or menu items–whatever suits your program.

You should also get your employer (if you work as a programmer) or your school, if any, to sign a "copyright disclaimer" for the program, if necessary. Here is a sample; alter the names:

> Yoyodyne, Inc., hereby disclaims all copyright interest in the program 'Gnomovision' (which makes passes at compilers) written by James Hacker.
>
> ⟨*signature of Ty Coon*⟩, 1 April 1989
> Ty Coon, President of Vice

This General Public License does not permit incorporating your program into proprietary programs. If your program is a subroutine library, you may consider it more useful to permit linking proprietary applications with the library. If this is what you want to do, use the GNU Library General Public License instead of this License.

Index