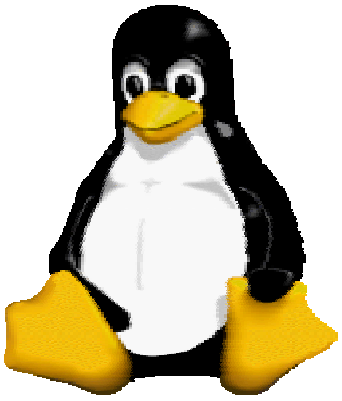

Installation et configuration de linux

1 - Introduction

Linux est une libre implantation totalement gratuite des spécifications POSIX, avec des extensions System V et Berkeley (ce qui signifie qu'il ressemble à Unix, mais ne provient pas du tout des mêmes sources), qui est disponible à la fois sous forme de code source et de binaires. Il est propriété de Linus B. Torvalds (torvalds@transmeta.com) ainsi que des autres contributeurs à ce projet et est librement diffusable selon les termes de la "GNU Public License".
Linux ne relève ni du domaine public, ni du *partagiciel* mais du "*logiciel libre*", appelé couramment "*graticiel*" ; vous pouvez en distribuer des copies mais vous devez fournir le code source correspondant ou au moins le rendre disponible de la même façon. Si vous y faites des modifications que vous diffusez, vous êtes légalement tenu de distribuer le code source de vos modifications. Consultez la "*licence publique générale GNU*" (GPL) pour plus de détails. Vous en trouverez une copie dans l'arborescence des sources de Linux (fichier nommé "COPYING") ou par FTP sur le site prep.ai.mit.edu.
Linux est libre et continuera de l'être. En raison de la nature du copyright GNU qui protège ce système, vouloir tout-à-coup changer cet état de fait serait illégal. Notez bien cependant qu'il est parfaitement autorisé de demander de l'argent pour distribuer Linux, du moment que vous fournissez aussi le code source et ne restreignez pas les droits de l'acheteur. Si vous désirez plus de précisions sur ce sujet, lisez la GPL.
Linux fonctionne sur des machines 386/486/Pentium possédant un bus ISA, EISA ou PCI.
MCA (le bus propriétaire d'IBM) est pour l'instant mal pris en charge car il y a peu de documentation disponible sur ce dinosaure et peu d'intérêt de la part des développeurs Linux, mais il existe malgré tout des "patches" pour certaines machines. Si vous êtes intéressé, voyez : http://www.undergrad.math.uwaterloo.ca/~cpbeaure/mca-linux.html
Un portage vers les plateformes basées sur les processeurs Motorola 680x0 (actuellement des Amiga et Atari) est en cours. Cette version fonctionne d'ores et déjà de manière satisfaisante et nécessite un 68020 avec un gestionnaire de mémoire externe (MMU), un 68030, un 68040, ou un 68060, ainsi qu'un coprocesseur arithmétique. Le réseau et X sont fournis.
Linux fonctionne également bien sur les processeurs Alpha de DEC/Compaq.
Le portage de Linux vers Sparc progresse rapidement. Linux fonctionne sur la plupart des architectures Sparc, et la distribution Red Hat pour Sparc existe déja.
Les portages vers d'autres machines, dont MIPS, PowerPC, et PowerMAC, sont en cours à des stades divers. Ne soyez pas trop impatient, mais si vous êtes intéressé et capable de contribuer à ces portages, vous pouvez trouver d'autres développeurs qui souhaiteraient travailler avec vous.
Pour PowerMac, Apple et OSF ont effectué un portage de Linux au dessus du micro-noyau Mach. Le résultat, appelé Mklinux, est encore en béta-test, mais peut déjà être utilisé.
Linux n'est plus considéré comme un système en béta-test, depuis que la version 1.0 a été rendue disponible le 14 Mars 1994. Il existe toujours des bogues dans le système et de nouveaux bogues apparaîtront et seront corrigés au fur et à mesure du développement. Puisque Linux suit un modèle de développement ouvert, toutes les nouvelles versions seront accessibles au public, qu'elles soient considérées comme suffisament stables ou non. Cependant, afin d'aider les utilisateurs à déterminer si une version donnée est ou non considérée comme stable, une convention de numérotation spéciale a été mise au point. Les versions x.y.z, où y est un nombre pair, sont stables, et seules des corrections de bogues

seront appliquées lorsque z est incrémenté. Par exemple, entre les versions 1.2.2 et 1.2.3, il y a eu uniquement des corrections, et aucun ajout de fonctionnalités. Les versions x.y.z, où y est un nombre impair, sont des versions en béta-test destinées aux développeurs uniquement, peuvent être instables, et contiennent de nouvelles fonctionnalités qui sont ajoutées au cours du développement. De temps en temps, quand le développement du noyau se stabilise, un "gel" intervient pour fournir une nouvelle version "stable" (paire), et le développement continue sur une nouvelle version (impaire).
La version stable actuelle est 2.2.x (x changeant au fur et à mesure que de nouvelles corrections de bogues sont intégrées au noyau), et le développement a déjà commencé sur des noyaux expérimentaux, numérotés 2.3.x. Si 2.2.x est trop récent pour vous, vous pouvez continuer d'utiliser la version 2.0.x (version stable précédente).
Le noyau est très stable depuis longtemps et la majorité des versions récentes peut être utilisée 24H/24 pendant des mois sans aucun réamorçage, plantage ou erreur fatale. Certains sites ne réamorcent leurs machines Linux qu'à l'occasion d'un changement majeur dans le noyau.
Il faut garder à l'esprit que Linux est developpé selon un modèle ouvert et réparti, contrairement à la plupart des logiciels connus qui évoluent souvent selon un modèle fermé et centralisé. Ceci signifie que la version courante de développement est toujours publique (avec une ou deux semaines de retard) afin que tout le monde puisse l'utiliser. Une version apportant de nouvelles fonctionnalités contient par conséquent presque toujours des bogues, mais ceux-ci sont découverts et corrigés rapidement, souvent en quelques heures, car ceux qui y travaillent sont nombreux. Il est donc facile pour un utilisateur final de les éviter.
À l'opposé, le modèle fermé et centralisé signifie que seule une personne ou une équipe travaille sur le projet et qu'elle ne diffuse un programme que lorsqu'elle considère qu'il fonctionne bien. Ceci implique souvent de longs intervalles entre les versions, de longs délais avant correction des bogues et un développement moins rapide. Bien sûr, la version la plus récente d'un programme réalisé ainsi est souvent de bonne qualité, mais le développement en est bien plus lent.

1.1 - Caractéristiques de Linux

– Multi-tâches : exécute plusieurs programmes en pseudo-parallélisme.
– Multi-utilisateurs : plusieurs utilisateurs actifs sur la même machine en même temps (et sans licence multi-utilisateurs !).
– Multi plates-formes : il fonctionne sur différents processeurs, et pas seulement sur Intel.
– Exécution en mode protégé sur les processeurs x86.
– Protection de la mémoire entre les processus, afin qu'un programme ne puisse à lui seul compromettre le fonctionnement de l'ensemble du système.
– Chargement des exécutables à la demande : Linux ne lit sur le disque que les parties effectivement utilisées d'un programme.
– Partage des pages entre exécutables avec copie en écriture. Cela signifie que des processus multiples peuvent partager la même mémoire. Lorsque l'un d'eux tente d'y écrire, cette page (4Ko de mémoire sur processeurs x86) est copiée ailleurs. Ceci accroît les performances et réduit l'occupation mémoire.
– Mémoire virtuelle utilisant la pagination (et non pas le "swap" de processus entiers) sur disque, grâce à une partition, un fichier ou les deux, avec la possibilité d'ajouter et de retirer de nouvelles zones de swap dynamiquement. Jusqu'à 16 zones de 128 Mo peuvent être utilisées simultanément, soit un total de 2 Go de swap disponible.

– Réserve de mémoire commune aux programmes utilisateur et au cache disque (afin que toute la mémoire libre puisse être utilisée pour le cache qui ne sera réduit que lors de l'exécution programmes exigeant de la mémoire) : Linux utilise au mieux à chaque instant TOUTE la mémoire disponible sur la machine.
– Bibliothèques partagées liées dynamiquement (DLL a.out, et ELF) ; des bibliothèques statiques sont bien entendu également disponibles.
– Sauvegarde d'image-mémoire (*core-dump*) lors du plantage d'un programme, à des fins d'analyse post-mortem effectuée grâce à un débogueur.
– Très conforme à POSIX, compatible System V et BSD au niveau programme source. Support des binaires COFF et ELF.
– Bonne compatibilité binaire avec SCO, SVR3 et SVR4, grâce à un module d'émulation conforme iBCS2.
– Tous les sources sont disponibles : le noyau, les pilotes, les outils de développement et tous les programmes utilisateur. De plus, ils sont librement diffusables. Il existe bien quelques applications commerciales pour Linux diffusées sans leurs sources, mais tout ce qui a été libre l'est toujours et le restera.
– Contrôle de processus POSIX (*job control*).
– Pseudo-terminaux (*pty's*).
– Émulation du coprocesseur dans le noyau afin qu'il ne soit pas nécessaire de l'inclure dans les programmes. Tout ordinateur exécutant Linux semble ainsi doté d'un coprocesseur mathématique. Bien entendu, si votre ordinateur comporte déjà un coprocesseur, Linux l'utilisera à la place de l'émulation. Vous pourrez recompiler votre noyau pour éliminer le code de l'émulateur (dont les fonctions seront prises en charge par le FPU), et gagner un peu de mémoire vive.
– De nombreux claviers nationaux ou personnalisés sont définis et il est assez facile d'en ajouter de nouveaux.
– Consoles virtuelles multiples : plusieurs sessions indépendantes accessibles par une combinaison de touches (indépendamment de la carte vidéo utilisée). Elles sont allouées dynamiquement et l'on peut en utiliser jusqu'à 63 simultanément.
– Plusieurs systèmes de fichiers reconnus comme minix-1, *Xenix*, et les plus courants de System V, en plus d'un système très perfectionné (Ext2fs), offrant des noms de 255 caractères et une taille pouvant atteindre 4 To.
– Accès transparent aux partitions MS-DOS (ou aux partitions FAT OS/2) via un système de fichiers spécial : vous n'avez pas besoin de commandes particulières pour utiliser une partition MS-DOS. Elle ressemble a une partition Unix (sauf pour la taille du nom des fichiers, les permissions d'accès, etc). Les partitions compressées de MS-DOS 6 ne sont pas encore reconnues mais il existe un *patch* (dmsdosfs). Le type VFAT (MS-Windows NT, MS-Windows 95) est pris en charge depuis la version 2.0. De plus, il existe un "patch" permettant d'accéder au système de fichiers NTFS (la version 2.1 du noyau intègre directement cette fonctionnalité).
– Système de fichiers spécial nommé UMSDOS, autorisant l'installation de Linux directement sur une partition MS-DOS (les performances sont bien sûr grevées mais c'est très utile pour une installation provisoire).
– Lecture seule du système de fichiers HPFS-2 de OS/2 V 2.1.
– Lecture seule du système de fichiers HFS (Macintosh) disponible sous forme de module pouvant être chargé à la demande.

– Lecture seule du système de fichiers EFS (Silicon Graphics) disponible sous forme de module pouvant être chargé à la demande.
– Système de fichiers ISO 9660, qui permet de lire tous les CD-ROM conformes à ce standard quasi universel. Depuis peu, le format Joliet est également reconnu.
– TCP/IP, incluant tous les outils comme ftp, telnet, NFS, etc.
– Pile de protocoles réseau *Appletalk*.
– Client et serveur *Netware*.
– Client et serveur *Lan Manager* (SMB).
– X Window System (X11R6) sous la forme de XFree86 3.2, gérant l'essentiel des cartes vidéo et des souris disponibles sur PC (y compris la carte monochrome Hercules...).
– Gestion des principales cartes sonores.

1.2 - Matériel

1.2.1 - Configuration minimale

La configuration suivante est probablement la plus petite configuration possible sur laquelle Linux peut fonctionner : 386SX/16, 2 Mo RAM, un lecteur 1.44 Mo ou 1.2 Mo, n'importe quelle carte vidéo reconnue. Elle devrait vous permettre d'amorcer le système et de tester si Linux tourne sur votre machine, mais vous ne serez pas en mesure de faire quoi que ce soit d'utile. (NDR : 4 Mo sont quasiment indispensables.)
Pour y parvenir, vous aurez aussi besoin d'espace disque : 5 à 10 Mo devraient suffire pour un système minimal ne comportant que les commandes les plus importantes et peut-être une ou deux petites applications, comme par exemple un programme d'émulation de terminal. Ceci reste très, très limité et très inconfortable, puisqu'il ne reste pas assez de place pour faire quoi que ce soit, sauf grâce à de minuscules applications. Ceci n'est généralement pas recommandé, sauf pour tester si le noyau fonctionne et bien sûr pour pouvoir se vanter des faibles ressources matérielles nécessaires.

1.2.2 - Configuration recommandée

Si vous souhaitez utiliser des programmes travaillant intensivement, comme gcc, X et TeX, vous voudrez probablement un processeur plus rapide qu'un 386SX/16, qui devrait cependant suffire si vous êtes patient.
En pratique, vous avez besoin d'au moins 4 Mo si vous n'utilisez pas X11 et d'au moins 8 Mo si vous l'utilisez. De plus, si vous voulez que soient simultanément actifs plusieurs utilisateurs ou plusieurs gros programmes (comme des compilations), plus de 4 Mo sont souhaitables. Tout fonctionnera avec moins de RAM (même avec 2 Mo), mais Linux utilisera alors la mémoire virtuelle (le disque dur servant de mémoire lente) et l'ensemble s'en trouvera considérablement ralenti, au point de s'avérer inutilisable.
La quantité d'espace disque nécessaire dépend des programmes que vous voulez installer. La série de base des utilitaires, shells et programmes d'administration devrait tenir sur un peu moins de 10 Mo, plus un peu de place pour les fichiers utilisateurs. Pour un système plus complet, Il est courant d'obtenir des valeurs de l'ordre de 20/30 Mo si vous n'utilisez pas X Window, et 40 Mo sinon (il n'est ici question que des binaires, les sources demeurent "nettement" plus encombrants). Ajoutez l'espace disque que vous voulez réserver aux utilisateurs. Étant donné le prix de la mémoire de masse de nos jours, si vous achetez une nouvelle machine, il est ridicule de l'équiper d'un disque dur de faible capacité.

Ajoutez plus de mémoire vive, de disque dur, un processeur plus rapide selon vos besoins, vos souhaits et votre budget pour aller au-delà d'un système à peine utilisable. En général, la grosse différence entre Linux et MS-DOS est qu'ajouter de la mémoire à Linux entraîne un gros gain de performances, ce qui n'est pas vraiment le cas avec MS-DOS. Ceci est bien sûr en rapport avec la limite des 640 Ko directement exploitables sous MS-DOS, qui n'a aucun sens sous Linux.
Si vous trouvez votre système trop lent, AJOUTEZ DE LA MEMOIRE avant de dépenser des sommes folles pour un processeur plus puissant. Sur un 386, passer de 4 à 8 Mo de mémoire peut multiplier la vitesse d'exécution par un facteur de 10 ou 20...

1.2.3 - Liste incomplète de programmes disponibles

La plupart des outils et des programmes classiques d'Unix ont été portés vers Linux, ce qui comprend presque tous les programmes GNU et de nombreux clients X provenant de diverses sources. En fait, le mot "portage" est souvent exagéré, puisqu'un grand nombre d'applications ont été compilées sans modification ou presque en raison de la forte conformité à POSIX de Linux. Malheureusement, il n'existe pour le moment qu'assez peu d'applications pour l'utilisateur final, mais cela a récemment commencé à changer. Voici néanmoins une liste très réduite de programmes qui fonctionnent sous Linux.
Commandes Unix de base :
ls, tr, sed, awk, etc ... Pratiquement tout y est.
Outils de développement :
gcc, gdb, make, bison, flex, perl, rcs, cvs, prof.
Langages et environnements :
C, C++, Objective C, Modula-3, Modula-2, Oberon, Ada95, Pascal, Fortran, ML, scheme, Tcl/tk, Perl, Python, Common Lisp, et bien d'autres.
Environnements graphiques :
X11R5 (XFree86 2.x), X11R6 (XFree86 3.x), MGR.
Editeurs :
GNU Emacs, XEmacs, MicroEmacs, jove, ez, epoch, elvis (GNU vi), vim (clone vi), vile, joe, pico, jed et bien d'autres.
Shells :
Bash (Bourne-shell compatible POSIX), zsh (incluant la compatibilité ksh), pdksh, tcsh, csh, rc, es, ash (Bourne-shell utilisé par BSD) et bien d'autres.
Télécommunications :
Taylor (compatible BNU) UUCP, SLIP, CSLIP, PPP, kermit, szrz (Zmodem), minicom, pcomm, xcomm, term/slap (exécute plusieurs shells, redirige les connexions réseau, et permet les affichages de fenêtres X-Window, sur une seule ligne téléphonique) Seyon (programme de communication sous X-Window) et de nombreux programmes de télécopie et de transmission de la voix. Bien entendu, les accès distants en mode terminal par ligne série sont supportés en standard.
News et mail :
C-news, INN, trn, nn, tin, sendmail, smail, elm, mh, pine, etc.
Formateurs de texte :
TeX, LaTeX, groff, doc, ez, Linuxdoc-SGML, etc.
Jeux :
Nethack, de nombreux jeux sous X11, dont DOOM. L'un des jeux les plus passionnants consiste à rechercher tous ceux qui sont disponibles sur les sites diffusant Linux...

Tous ces programmes (qui ne représentent pas le centième de ce qui existe) sont bien entendu disponibles gratuitement. Les applications commerciales commencent à exister, demandez à votre fournisseur si votre logiciel préféré a été porté sous Linux.
(NDR : une liste des programmes portés pour Linux est disponible. Elle est contenue dans le fichier LSM*.*, ou *Linux Software Map*)

1.3 - Qui utilise Linux ?

Linux est disponible gratuitement et il n'est demandé à personne d'enregistrer ses copies à une autorité quelconque ; aussi est-il très difficile de savoir combien de personnes dans le monde utilisent ce système d'exploitation. Plusieurs sociétés commerciales se consacrent exclusivement à la vente et au support de Linux (ce n'est pas interdit) mais leur clientèle ne constitue qu'une minorité des utilisateurs de Linux.
Les groupes Linux sur Usenet sont parmi les plus lus selon les statistiques, on pourrait donc penser que plusieurs centaines de milliers d'utilisateurs ou de personnes intéressées.
Toutefois, Harald T. Alvestrand a décidé d'essayer de compter tant bien que mal les utilisateurs de Linux. Il demande que vous postiez un courrier électronique à l'adresse linux-counter@uninett.no, avec l'une des phrases suivantes dans le SUJET du message : "I use Linux at home", "I use Linux at work", ou "I use Linux at home and at work". Il compte aussi les votes contenant "I don't use Linux", au passage. Il diffuse périodiquement ces statistiques dans comp.os.linux.misc. Il est possible de remplir un formulaire plus détaillé, pour enregistrer des amis n'ayant pas la possibilité de poster du courrier électronique.
Ce compteur est fatalement assez pessimiste. Lors de la conférence Linux de Berlin de 1995, il a été annoncé une estimation à plus d'un million d'utilisateurs actifs dans le monde ; sans autre précision quant au mode de détermination de ce nombre.

1.4 - Comment débuter

Comme il est expliqué ci-dessus Linux n'est pas géré de manière centralisée. De par ce fait, il n'en existe pas de "distribution officielle" (certains y travaillent toutefois). A la place, il existe différentes "distributions", qui sont des collections plus ou moins complètes de programmes Linux, destinées à installer un système en état de fonctionnement.
Il faut commencer par récupérer et LIRE la liste des "Frequently Asked Questions" (FAQ) pris sur l'un des sites archivant les FAQ'S Usenet en général (par exemple, rtfm.mit.edu ou l'un de ses miroirs français comme ftp.lip6.fr dans le répertoire /pub/doc/faqs). Ces documents contiennent des centaines de renseignements permettant de résoudre la plupart de vos problèmes.
Les "HOWTO", petits guides traitant chacun d'un domaine de Linux, sont une lecture indispensable. La plupart d'entre eux est disponible en version française dans le répertoire /pub/linux/french/docs/HOWTO sur le site ftp.lip6.fr.
Si vous avez accès au Web vous pouvez également trouver de précieux renseignements sur les URL :
– http://www.freenix.org/linux
– http://echo-linux.alienor.fr/
– http://www.linux-france.com/article/
– http://www.linux-kheops.com/line/
– http://uhp.u-nancy.fr/linux/
Si le monde UNIX est tout nouveau pour vous, et que Linux est votre premier contact avec ce domaine passionnant, suivez absolument les conseils suivants :

– Ne vous ruez pas sur les forums de discussion Linux pour poser mille questions auxquelles tout le monde est lassé de répondre. Lisez les quelques semaines auparavant, voyez ce qui s'y passe, consultez-en les archives mises à disposition sur certains sites, et il y a fort à parier que vous trouverez ce que vous cherchez sans déranger personne.
– Linux est un système UNIX parmi d'autres, et par conséquent toute la littérature spécifique à UNIX s'y applique. On n'apprend pas à maîtriser un nouveau système d'exploitation sans un minimum d'effort de documentation, souvenez-vous du temps qu'il vous a fallu pour maîtriser celui avec lequel vous travailliez auparavant, et des nombreux ouvrages que vous avez du consulter pour arriver à vos fins.
– Procurez-vous et lisez attentivement tous les ouvrages gratuits rédigés sur Linux. Ils contiennent les bases indispensables à votre apprentissage du système. Il ne vous sera pas pardonné de poser une question dont la réponse est contenue dans l'un de ces livres, ou dans l'un des documents complémentaires "HOWTO" disponibles tout aussi librement sur les sites diffusant Linux.
– Expérimentez, tentez de comprendre et de résoudre votre problème vous-même : c'est le meilleur apprentissage. Vous n'arriverez jamais à rien si vous faites tout faire par les autres. Linux et ses différentes documentations représentent des milliers d'heures de travail bénévole de centaines de développeurs, rédacteurs, traducteurs, testeurs ; ne condamnez pas le système si vous n'arrivez pas à en tirer parti parce que vous refusez de faire à votre tour un tout petit effort.

1.5 - Statut légal de Linux

Bien que Linux soit diffusé avec son code source au complet, le système est "copyrighté" et ne relève donc pas du domaine public. Toutefois, il est disponible gratuitement selon les termes de la "GPL" (Gnu Public License). Lisez cette "GPL" pour obtenir davantage d'informations. (En quelques mots, cette licence assure que nul ne pourra retirer certains droits à l'utilisateur du *logiciel libre,* comme par exemple l'accès gratuit à son code source. Mais elle n'interdit nullement la vente des programmes auxquels elle s'applique).
Les programmes qui tournent sous Linux ont chacun leur copyright propre, bien que beaucoup d'entre eux utilisent également la GPL. X Window dépend du "MIT X Copyright", et de nombreux utilitaires, de celui de BSD. Dans tous les cas, tous les programmes disponibles sur les sites diffusant Linux sont librement diffusables (sinon ils ne s'y trouveraient pas !).

2 - Installation de linux

2.1 - Les différentes possibilités d'installations

Il existe en fait plusieurs possibilités pour installer Linux... En voici un court résumé :
– Installation classique : vous avez une partition libre pour y mettre Linux. Dans ce cas, pas de problème : il suffit de suivre les instructions des sections suivantes.
– Aïe : je n'avais pas prévu d'installer Linux et n'ai qu'une partition MS-DOS. Dans ce cas, il existe deux solutions :

– FIPS : toutes les distributions fournissent un programme DOS qui permet en fait de repartitionner votre partition MS-Windows sans avoir à tout réinstaller. Comment ça marche ? C'est simple. Il faut dans un premier temps passer un coup de scandisk puis un coup de defrag (optimisation complète : tous les fichiers doivent être remontés). Ensuite, rebootez avec une disquette DOS (avec le moins de programmes résidents que possible). Il suffit ensuite de lancer FIPS et de suivre les instructions. Cela permet de réduire la taille de votre partition DOS, et ainsi d'installer Linux sur la place restante.

– Installer Linux en utilisant *umsdos*. Ce système est très pratique. Cela permet d'installer un système Linux, sans toucher les partitions : Linux est installé en fait sur la partition DOS. Toutefois, cette installation est très limitée car le système de fichiers umsdos est très très lent. Cette solution peut être envisagée dans des cas extrèmes... mais qui peuvent être très pratiques.
– Une machine sans disque ! Il peut arriver que vous ayez déjà une machine sur le réseau avec Linux parfaitement installé. On peut alors faire en sorte que la machine utilise l'autre machine comme serveur. Dans ce cas, il suffit d'une disquette de boot ! Le mini Howto *nfs-root* détaille complètement cette opération.

2.2 - Distributions

2.2.1 - Définition et comment choisir une distribution

Tous les autres systèmes d'exploitation (MS-DOS, MS-Windows, Mac-OS, les Unix commerciaux, même FreeBSD) sont livrés (distribués) sous forme d'un ensemble comprenant le système d'exploitation proprement dit (un Unixien dit "le noyau") plus tous les logiciels qui servent à en tirer quelque chose, comme /bin/ls, /usr/bin/gcc, /usr/games/tetris plus l'éventuelle interface graphique, etc. On peut toujours ajouter ou (parfois) enlever des trucs, mais c'est souvent peu pratique.
Linux, au contraire, tire profit de la modularité de Unix : le mot Linux, stricto-sensu, ne désigne que le noyau. Chacun est libre de prendre le noyau et de l'emballer joliment avec d'autres logiciels, formant ainsi une *distribution*. Une distribution, c'est le noyau (le même pour toutes) plus certains logiciels (toutes les distributions Linux utilisent les logiciels GNU pour les commandes de base comme ls, grep ou tar et toutes utilisent X11 pour le graphique) plus un programme d'installation, plus un système de gestion des logiciels, pour en ajouter ou en enlever proprement et facilement.
Les deux premiers élements (le noyau et les logiciels de base) sont communs à toutes les distributions. La concurrence se fait sur le programme d'installation et sur la gestion des

paquetages (un paquetage est un ensemble cohérent de logiciels, c'est l'unité d'installation, de désinstallation, ou de configuration).
La Slackware ne fournit quasiment rien pour l'installation et rien du tout pour la désinstallation. Une fois qu'elle est installée, tout se fait à la main. Les mises à jour sont pénibles (par exemple, elle ne respecte pas les fichiers de configuration déjà installés). Toutefois, c'est la distribution idéale si vous souhaitez gérer et mettre à jour vous-même les différents produits (en les recompilant).
La Debian fut, sauf erreur, la première à avoir un système de paquetages, et la première à avoir un système de dépendance entre paquetages (du genre, "gimp a besoin de gtk").
La RedHat a une interface graphique de configuration des paquetages et gère également les dépendances.
Après avoir choisi Linux, il faut donc choisir une distribution particulière.

2.2.2 - Description rapide

On appelle "distribution" un ensemble de supports (disquettes, CD-ROM ...) contenant de quoi installer un logiciel sur une machine. Il n'existe pas de distribution officielle de Linux. Seules des distributions circulent. Ici, seules la Slackware et la Red Hat seront détaillées. Toutefois, voici quelques mots sur les principales distributions existantes :
– Slackware : c'est la distribution qui faisait historiquement référence dans la mesure où c'était la seule distribution à peu près correctement faite au début de la généralisation de Linux. De nos jours, elle est fortement concurrencée par de nouvelles distributions comme Red Hat/Debian et on peut craindre qu'elle ne disparaisse en raison des difficultés engendrées par son évolution. De plus, elle ne gère qu'une seule architecture (Intel) et n'a pas de notion de paquetage.
– Red Hat : c'est une distribution initialement commerciale, mais qui est également disponible par téléchargement sur les sites FTP. Elle est bien plus professionnelle que Slackware. De plus, elle est disponible pour d'autres architectures processeurs "non Intel" telles que Alpha et Sparc. Cette distribution se distingue par son sérieux, sa simplicité d'installation, et surtout par son extrême convivialité. De plus, la mise à jour du système est gérée par un système de paquetages. On peut effectuer l'installation par ftp. Site ftp officiel : ftp://ftp.redhat.com/pub/redhat - Site Web : http://www.redhat.com/
– Debian : c'est la version de Linux soutenue par la FSF (Free Software Foundation).
– SLS : Première grande distribution de Linux, mais elle est maintenant obsolète.
– Suse : distribution Allemande assez stable. Elle a une assez bonne réputation.
Site Web : http://www.suse.com/

2.2.3 - Disquettes d'amorçage

Avec la Red Hat ou la Slackware, il est nécessaire d'utiliser une disquette de boot (et une disquette root pour la Slackware). La manière de copier ces fichiers sur disquette est commune aux deux distributions. Notons que Red Hat offre un moyen, compatible avec la plupart des systèmes, d'installer à partir de MS-DOS sans disquette ou à partir d'un CD-ROM bootable (à condition que le BIOS de la machine le permette).
Après avoir sélectionné la ou les images à copier sur disquette pour permettre l'amorçage de la machine, il est nécessaire de la copier sur disquette. Pour copier cette disquette, plusieurs solutions existent :

1. copier le fichier sur la disquette avec un cat image > /dev/fd0 (/dev/fd0 correspondant au périphérique de votre lecteur de disquettes.). Il est possible que vous deviez décompresser le fichier. Pour cela, utilisez le programmes gunzip : gunzip bare.i.gz.
2. si jamais vous êtes sous MS-DOS, il faut se servir du programme *RAWRITE*. Pour cela : décompacter le fichier si nécessaire (le nom se termine par ".gz") grâce à *GZIP.EXE* (commande : gzip -d NOM_DU_FICHIER.gz), formater la disquette MS-DOS qui ne doit pas présenter le moindre défaut de surface ("secteurs défectueux") puis utiliser *RAWRITE* pour écrire le (ou les fichiers) sur les disquettes.

2.3 - Installation de la Red Hat

2.3.1 - Disquette de boot

La Red Hat utilise un système très différent de la Slackware : elle met à profit, lors de l'amorçage de la machine, les modules chargeables du noyau. Toutefois, il est à noter qu'elle ne permet pas d'être installée uniquement grâce à des disquettes (la Slackware s'impose dans ce cas). Enfin, sa dernière limitation est qu'elle impose d'avoir une partition Linux native (ext2fs).
Il n'existe en fait qu'une seule disquette de boot : images/boot.img. Si vous installez un PC utilisant du PCMCIA ou si vous effectuez une installation par réseau depuis un site FTP, vous devez utiliser une disquette suppémentaire : images/supp.img.
Remarque : il existe une disquette supplémentaire (supp.img) qui est utilisée lorsque vous utilisez l'option rescue lors du boot.
Une fois que vous avez copié la disquette de boot (et éventuellement la seconde si nécessaire), amorcer la machine... et vous accéderez directement au système d'installation. Difficile de faire plus simple ! De plus, ce système de paquetage permet de gérer les dépendances entre les bibliothèques et les outils.

2.3.2 - La Red Hat sans disquette

Il existe deux possibilités pour installer la Red Hat sans disquette de démarrage. Si le PC est équippé d'un lecteur CD "bootable" (lecteur SCSI ou ATAPI El Torito et les bonnes options activées dans le SETUP), on peut démarrer directement sur le CD.
Si la machine fonctionne sous DOS (gestionnaires du lecteur CD chargés), on peut essayer d'employer le programme autoboot, placé dans le répertoire dosutils, qui permet sur la plupart des machines d'installer sans même avoir à créer une disquette de démarrage !

2.3.3 - Distribution

La distribution Red Hat est composée d'un ensemble de paquetages (un peu plus de 560) regroupés sous une forme de centres d'intérêts (en fait une arborescence de groupes).
La manipulation "directe" des paquetages (archives rpm) est détaillée dans une prochaine section.
Il est à noter qu'il existe bien souvent un répertoire du nom de updates dans lequel se trouvent des fichiers qui permettent de mettre à jour (évolution et correction de bogues éventuelles) la distribution. Une fois que vous avez installé votre machine, pensez à installer l'ensemble des fichiers situés dans ce répertoire.
De même, notons la présence d'un répertoire contrib regroupant un ensemble d'outils n'appartenant pas à la distribution officielle mais qui sont fort utiles.

2.4 - Commencer Installation

Après ces préliminaires, on passe aux choses sérieuses. Dans un premier temps, faites une sauvegarde du disque dur, c'est beaucoup plus prudent, et mettez de côté une disquette formatée.
Ensuite, éteindre la machine, et insérer la disquette de boot.
Avertissement préliminaire : ne jamais éteindre brutalement la machine sous Linux ! En effet, Linux comme tout système Unix, utilise des caches disques en mémoire vive : vous perdriez des informations en éteignant brutalement la machine. Les seules manières adéquates d'éteindre Linux sont soit de faire un *Ctrl-Alt-Del*, soit l'une de ces trois commandes à lancer en tant que *root* : halt, reboot, shutdown (Consultez le manuel pour connaître les différences).

2.4.1 - Installation initiale

Allumer la machine. Linux boote sur la disquette *boot*. Dans le cas de la Slackware, il va demander d'insérer une seconde disquette, dite *root*. Remplacer la disquette, puis appuyer sur Entrée.
Remarque à propos du message "BIOS32" que certains peuvent voir lors du boot : ce message signifie que votre bus est PCI. Si vous obtenez le message suivant : "BIOS32 extended no supported", cela signifie donc que la machine n'est pas PCI. Il ne faut pas s'inquiéter et utiliser malgré cela la disquette de boot standard.
Avant de se lancer dans l'installation du système, voici quelques trucs. Linux utilise un procédé qui installe votre système d'une manière quasi automatique. Dans le cas de la Red Hat, l'installation est vraiment toute simple et progressive. Il suffit de suivre les indications.

2.4.2 - Partitionnement du disque

Il va falloir dans un premier temps effectuer le formatage du disque. Vous allez créer au moins une partition Linux avec le système de fichiers *ext2fs*, et au moins une partition de swap (le swap est un système de mémoire virtuelle, c'est-à-dire que si vous n'avez pas assez de mémoire pour faire fonctionner tel ou tel programme, il va alors utiliser l'espace disque comme de la mémoire... Cette explication est un peu simplifiée mais, en gros, c'est ça.)
Il est beaucoup plus facile de supprimer une partition de swap que d'en créer une après coup donc prévoir large. Si vous voulez faire tourner X Window, 8 Mo de mémoire vive sont absolument nécessaires.
Les disques sous Linux sont des *devices*. Par exemple, le premier disque dur est /dev/hdax, le second /dev/hdbx pour un contrôleur IDE. Le x correspond au numéro de la partition du disque. Si vous avez déjà un système MS-DOS installé, il se trouve sûrement sur /dev/hda1.
Linux va alors vous guider dans l'installation. Il vous permet d'installer votre système d'une manière complète d'un seul coup. Le seul problème, c'est que si jamais vous ratez quelque chose, il est fort probable que vous serez obligé de tout recommencer.
Si le disque n'est pas partitionné, c'est le programme fdisk qui va permettre de réaliser l'opération. Sa syntaxe est : fdisk /dev/disque. Dans le cas de la Red Hat, il se lance tout seul.
Par exemple, si vous souhaitez installer Linux sur le second disque IDE, tapez fdisk /dev/hdb.
Les commandes de fdisk (dans l'ordre d'utilité) sont :
– m : la liste de toutes les commandes ;
– p : affiche les partitions ;
– n : ajouter une nouvelle partition ;
– d : détruire une partition ;

– t : changer le type de la partition ;
– q : quitter sans sauvegarder les changements ;
– w : sauvegarder et quitter.
Pour ajouter une partition, tapez n puis spécifiez le numéro de la partition (principale ou étendue), le numéro du bloc de début et sa taille (une taille paire est conseillée). Ensuite, donnez-lui son type (avec t) :
– 83 : Linux
– 82 : Swap.
Bon, sauvegardez (w).
La partition est créée. Il est toutefois possible que vous ayez besoin de rebooter Linux pour qu'elle soit prise en compte par le noyau.
La Red Hat s'occupe du formatage automatique des partitions. Pour les formater à la main, faites : mke2fs -cv /dev/hdax où *x* correspond au numéro de la partition.

2.4.3 - Installation avec la Red Hat

L'installation avec la Red Hat est très intuitive. À partir de la version 4.2, la procédure d'installation demande à l'utilisateur de préciser le type de clavier employé. (attention : les premières versions des versions 5.0 françaises proposaient une mauvaise valeur par défaut, ce problème a été corrigé depuis).
– L'installation des paquetages proprement dits survient, après le partitionnement des disques, et leur formatage, un menu propose les paquetages à installer sur le disque.
– Paramétrage de la souris : pas de problème (la redhat 5.* reconnait automatiquement ce type de périphérique). Penser à dire "yes" à la question "emuler le 3ème boutton" lorsqu'on a une souris à 2 boutons. Cette option permet d'obtenir le 3ème boutton de la souris en appuyant simultanément sur les 2 autres. Pour re-configurer la souris apres avoir installé un système (ou si on change de souris) on pourra utiliser /usr/sbin/mouseconfig
– Paramétrage Xwindow :

* Choix de carte graphique : A partir de la redhat 5.0 la reconnaissance du matériel pour un bon nombre de cartes vidéo est automatique.... Si la carte n'est pas reconnue, il va falloir mettre les mains dans le camboui, c'est à dire dans le fichier de configuration /etc/X11/XF86Config ou /usr/X11R6/lib/X11/XF86Config

* Choix du display (écran) = Custom (i.e pas de modèle reconnu signifie qu'il faudra par la suite faire un paramétrage explicite dans le fichier /usr/X11R6/lib/X11/XF86Config

* Paramétrage réseau : (dans le cas ou on a une connexion réseau permanente), il faut renseigner l'adresse IP du PC, ainsi que les serveurs de noms, et masques de broadcast ....(aucun problème, ca marche du 1er coup)

* Choix d'installation de LILO (Linux Loader) : ce programme se loge sur une partition disque bootable permettra d'avoir le choix de booter à l'allumage de la machine soit sous W95 soit sous Linux

Sur le MBR (master boot record)
Au debut de la partition "root" / de Linux (dans mon cas /dev/hda3)

On préconise d'installer LILO sur le Master Boot Record si on n'a pas d'autres système OS à gérer.

* etc.....choix du mot de passe de root.....

L'install est fini. (on pourra voir /tmp/install.* pour le compte rendu d'installation).....il faut maintenant rebooter la machine!

2.4.4 - Problemes possibles lors du boot

2.4.4.1 - "LI" sans "LO"

LILO a été installé sur le MBR, mais lors du boot, seul "LI" apparait à l'écran (au lieu de LILO) et le PC se bloque! Ce problème provient du fait que seule la première partie de LILO commence à s'exécuter puis lilo ne parvient pas a charger l'image du noyau linux sur le disque, ceci parcequ'il ne le trouve pas. On s'en tirera dans 95% des cas en mettant l'option "linear" dans le fichier de conf de lilo: /etc/lilo.conf cette option permet de s'affranchir de la géométrie du disque (CHS) (cylinder/head/sector) en faisant un adressage linéaire du disque...

```
boot=/dev/hda
delay=900
map=/boot/map
install=/boot/boot.b
prompt
timeout=50
linear          <---- là...
other=/dev/hda1
        label=dos
        table=/dev/hda
image=/boot/vmlinuz
        label=linux
        root=/dev/hdc3
        read-only
        append="mem=80M"
```

2.4.4.2 - message "Kernet panic unable to mount VFS system"

(ou message "unable to open initial console")
Dans ce cas LILO a pu s'exécuter et a chargé le noyau linux, le noyau commence à s'éxécuter et reconnaitre tous les périphériques du PC, mais le noyau n'arrive pas à accéder au disque qui contient la partition système "/" (root). Le noyau doit d'une façon ou d'une autre savoir quelle partition de quel disque utiliser comme racine. On peut fournir cette information par la commande rdev. rdev permet de "patcher" (modifier) le noyau afin qu'il pointe sur le bon périphérique root "/" . ex: rdev /vmlinuz /dev/hda6

3 - Le gestionnaire de packages : RPM

3.1 - Objectifs de RPM

Avant de chercher à utiliser RPM, il est préférable de savoir à quoi il peut servir.

Mise à jour : vous pouvez avec RPM effectuer une mise à jour des différents composants de votre système sans avoir à tout réinstaller. Lorsque vous avez à votre disposition une nouvelle version d'un système d'exploitation basée sur RPM (ex : Red Hat Linux), vous n'avez pas besoin de réinstaller votre machine (comme vous le feriez pour d'autres systèmes). RPM effectue une mise à jour automatisée et intelligente de votre système. Vos fichiers de configuration seront conservés lors de la procédure de mise à jour.

Interrogation : RPM possède des fonctions d'interrogation très poussées. Vous pouvez effectuer des recherches, grâce à la base de données, sur des packages ou sur un simple fichier. Ainsi, vous pouvez facilement savoir à quel package appartient tel ou tel fichier et donc savoir d'où il vient. Les fichiers contenus dans un package RPM se trouvent dans une archive compressée, avec un en-tête binaire contenant des informations sur le package et son contenu. Vous pouvez ainsi interroger rapidement et individuellement les packages.

Vérification du système : RPM possède aussi des fonctions de vérification des packages. Si vous pensez avoir effacé un fichier contenu dans un package, il vous suffit de vérifier ce package et toute anomalie vous sera rapportée. Si nécessaire, vous pouvez alors réinstaller le package, sans que vos fichiers de configuration soient supprimés.

Sources premières : un des objectifs est de permettre l'utilisation des sources premières, telles qu'elles sont distribuées par les auteurs du logiciel. Avec RPM, vous aurez à votre disposition les sources premières, les mises à jour (patch) utilisés et les instructions nécessaires à la compilation. C'est un grand avantage, et pour plusieurs raisons. Par exemple, vous pouvez facilement évoluer vers une nouvelle version de package. Il vous suffit de jeter un simple coup d'oeil aux mises à jour (patch) pour savoir ce qu'il vous suffit de faire.

Tout ceci est d'une grande importance pour les programmeurs, mais au final les utilisateurs en seront aussi très satisfaits. La distribution BOGUS sont à l'origine de ce concept.
RPM possède donc cinq fonctions différentes (sans compter la création de package) : installation, suppression, mise à jour, interrogation et vérification. Voici une présentation détaillée de ces cinq fonctions. Pour plus d'informations et pour une description des options, utilisez la commande rpm --help ou consultez les pages du man (ainsi que l'ouvrage consacré à RPM).

3.2 - Installation

Les packages RPM ont généralement des noms du type foo-1.0-1.i386.rpm, (foo) est le nom du package, (1.0) la version, (1) le numéro de mise à jour, et (i386) l'architecture. Pour installer un package, utilisez simplement la commande :

```
$ rpm -ivh foo-1.0-1.i386.rpm
foo ########################################
```

Comme vous pouvez le constater, RPM affiche le nom du package (qui ne correspond pas forcément au nom du fichier, qui aurait pu être 1.rpm) et une succession de caractères # qui représentent une jauge de progression.
L'installation de package est très simple, mais vous pouvez tout de même rencontrer quelques erreurs :

Package déjà installé
Si le package est déjà installé, vous verrez apparaître :

```
$ rpm -ivh foo-1.0-1.i386.rpm
foo package foo-1.0-1 is already installed
error : foo-1.0-1.i386.rpm cannot be installed
```

Si vous voulez vraiment installer ce package, utilisez l'option --replacepkgs qui permet d'ignorer l'erreur.

Fichiers en conflit
Si vous tentez d'installer un package qui contient un fichier déjà installé sur le système, vous verrez apparaître :

```
$ rpm -ivh foo-1.0-1.i386.rpm
foo /usr/bin/foo conflicts with file from bar-1.0-1
error : foo-1.0-1.i386.rpm cannot be installed
```

Pour ignorer l'erreur, utilisez l'option --replacefiles.

Dépendances
Des packages RPM peuvent dépendre d'autres packages, ce qui signifie qu'ils doivent au préalable être installés pour que tout fonctionne correctement. Si vous tentez d'installer un package dépendant d'un autre package, vous verrez apparaître :

```
$ rpm -ivh bar-1.0-1.i386.rpm
failed dependencies :
foo is needed by bar-1.0-1
```

Pour éviter ce problème, vous devez d'abord installer les packages requis. Vous pouvez tout de même forcer l'installation (mais le package risque fort de ne pas fonctionner) en utilisant l'option --nodeps.

3.3 - Suppression

La suppression d'un package est aussi simple que son installation :

```
$ rpm -e foo
```

Note : nous utilisons ici le nom foo, qui n'est pas le nom original du package (ici foo-1.0-1.i386.rpm).

Lors d'une suppression , vous pouvez rencontrer des problèmes de dépendances dans le cas où le package que vous tentez de supprimer est requis par d'autres packages. Vous verrez apparaître :

```
$ rpm -e foo
removing these packages would break dependencies :
foo is needed by bar-1.0-1
```

Vous pouvez tout de même forcer la suppression (mais les packages dépendants risquent fort de ne pas fonctionner) en utilisant l'option --nodeps.

3.4 - Mise à jour

La mise à jour d'un package est aussi simple que son installation :

```
$ rpm -Uvh foo-2.0-1.i386.rpm
foo ##################################
```

Vous ne vous en apercevrez pas, mais RPM supprime automatiquement les anciennes versions du package foo. Vous pouvez toujours utiliser l'option -U pour installer les packages, puisqu'elle fonctionne même si le système ne possède pas d'ancienne version du package.
RPM effectue une mise à jour intelligente du package avec une sauvegarde des fichiers de configuration, vous verrez apparaître un message du type :

```
saving /etc/foo.conf as /etc/foo.conf.rpmsave
```

Cela signifie que les modifications que vous avez apportées au fichier de configuration risquent de ne plus être compatibles avec le nouveau fichier de configuration du package, RPM effectue donc une sauvegarde du fichier d'origine et installe le nouveau. Observez les différences entre les deux fichiers, afin que le système puisse continuer à fonctionner correctement.
Une mise à jour combine en fait les fonctions d'installation et de suppression de package, vous pouvez donc rencontrer une des erreurs décrites plus haut, plus une. Si RPM pense que vous êtes en train d'effectuer une mise à jour vers une version plus ancienne, vous verrez apparaître :

```
$ rpm -Uvh foo-1.0-1.i386.rpm
foo package foo-2.0-1 (which is newer) is already installed
error : foo-1.0-1.i386.rpm cannot be installed
```

Pour ignorer cette erreur et donc pour forcer la mise à jour, utilisez l'option --oldpackage.

3.5 - Interrogation

Pour interroger la base de données sur les packages installés, utilisez la commande rpm -q. Utilisez simplement rpm -q foo pour afficher le nom, la version et le numéro de mise à jour du package foo :

```
$ rpm -q foo
rpm-2.0-1
```

Vous pouvez remplacer le nom du package par une des ces options (Package Specification Options) :
* -a interroge tous les packages installés
* -f <file> interroge tous les packages qui contiennent le fichier <file>
* -p <packagefile> interroge le package <packagefile>

Il existe plusieurs façons de spécifier les informations à afficher lors de l'interrogation des packages. Vous pouvez utiliser les options suivantes (Information Selection Options) :
* -i affiche un certain nombre d'informations sur le package (nom, description, numéro de mise à jour, taille, date de création, date d'installation, fournisseur, ...)
* -l affiche la liste de fichiers contenus dans le package
* -s affiche l'état de tous les fichiers du package (état : normal ou absent)
* -d affiche une liste des fichiers de documentation (pages du man, pages info, README, ...)
* -c affiche une liste des fichiers de configuration. Ces fichiers sont ceux que vous modifierez après l'installation pour les adapter à votre système (sendmail.cf, passwd, inittab, ...)

Pour les options qui affichent une liste de fichiers, vous pouvez ajouter -v pour un affichage avec un format similaire à celui de la commande ls -l.
Par exemple, pour savoir les paquetages installés sur votre machine, il suffit de faire :

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

On peut aussi grouper les options : -qa est équivalent à -q -a. Pour avoir plus de renseignements sur les différents paquetages installés (numéro de version, quelle distribution, ...,), il suffit de rajouter en plus l'option -l et -i. Par exemple :

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

Ces options peuvent être utilisées sur un fichier rpm que vous souhaitez installer pour savoir ce qu'il contient : il suffit d'ajouter l'option -p :

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

Si vous ajoutez l'option -i, vous obtiendez les renseignements sur le paquetage.

3.6 - Vérification

La vérification permet de comparer des informations relatives aux fichiers installés sur le système avec celles du package d'origine.
Entre autres, la vérification se fait sur la taille, sur la somme MD5, sur les permissions, sur le type, sur le propriétaire et sur le groupe de chaque fichier.
rpm -V vérifie un package. Vous pouvez aussi combiner d'autres options décrites plus haut (Package Selection Options). La commande rpm -V foo vérifie que tous les fichiers du package foo sont tels qu'ils étaient lors de l'installation d'origine. Exemples :

* Pour vérifier qu'un package contient un fichier particulier :
  rpm -Vf /bin/vi
* Pour vérifier tous les packages installés :
  rpm -Va
* Pour vérifier un package installé par rapport au package RPM :
  rpm -Vp foo-1.0-1.i386.rpm
* Ceci peut être utile si vous pensez que votre base de données est corrompue.

Si la vérification s'est correctement déroulée, vous ne verrez rien apparaître. Si la vérification met à jour des différences, une chaîne de huit caractères apparaît. Le caractère c suivi du nom du fichier indique qu'il s'agit d'un fichier de configuration. Chacun des huit caractères

représente une comparaison, si un point (.) apparaît cela signifie que la comparaison n'a pas échoué. Voici les caractères utilisés lorsque la comparaison a échoué :

5	Somme MD5
S	Taille du fichier
L	Lien symbolique
T	Date de modification du fichier
D	Périphérique
U	Utilisateur
G	Groupe
M	Mode (permissions et type de fichier)

Suivant le résultat, c'est à vous de déterminer la marche à suivre (supprimer le package, le réinstaller, corriger le problème, ...).

4 - Configuration de Linux

4.1 - Clavier Français

Bien, commençons par la chose utile : le clavier français. Normalement, les distributions configurent ça tout seul... mais on ne sait jamais !
Une fois loggé en *root*, tapez la commande suivante :
/usr/bin/loadkeys /usr/lib/kbd/keytables/fr-latin1.map
Si vous avez un clavier suisse-romand, utilisez sf-latin1.map.
Maintenant, le clavier est français ! Attention, cela reste temporaire. La solution la plus simple est de le rajouter dans le fichier /etc/rc.local avec la Slackware mais vous pouvez aussi utiliser le programme /sbin/setup ou directement /usr/lib/setup/SeTkeymap. Pour la Red Hat, réaliser l'opération avec le panneau de configuration kbdconfig.
Avec ces deux distributions, la procédure d'installation aura normalement procédé à cette configuration et vous n'aurez probablement rien à faire, sauf si vous voulez changer de type de clavier...

4.2 - LILO

LILO (LInux LOader) permet d'avoir un système multiple d'amorçage : MS-Windows, MS-Dos, Linux, etc. Ce programme va se loger dans le secteur de boot de votre disque pour vous permettre de sélectionner la partition sur laquelle booter.
Si vous avez installé *LILO* sur le disque dur et que cela plante, bootez avec une disquette MS-DOS et tapez FDISK /MBR pour restaurer les paramètres de boot du DOS, et enlever *LILO* par la même occasion. Il suffira de relancer *LILO* en ayant rebooté sous Linux avec une disquette de boot.
Le fichier de configuration LILO se trouve généralement dans le fichier /etc/lilo.conf. Les distributions permettent la génération d'un fichier d'une manière automatique.
Voici un exemple de fichier :

```
boot = /dev/hda    # disque sur lequel on met le fichier lilo:
delay = 300        # temps d'attente avant le lancement du boot.
                   # permet d'avoir le temps de selectionner la partition
                   # a amorcer.
vga = normal       # mode de l'écran a choisir
ramdisk = 0        # paranoia setting
```

```
# Linux : dernière version : le Pc boote ici par défaut
image = /vmlinuz    # Noyau Linux 1 (chemin complet et nom du fichier)
  root = /dev/hda2
  append="no-hlt, aha1542=0x230"
  label = linux
  read-only
other = /dev/hda1   # Partition 1 :
  label = dos       # texte à taper pour booter dessus : dos
  table = /dev/hda  # disque : /dev/hda
image = /zimage     # Noyau Linux 2
  root = /dev/hda2  # Racine (Partition 2 de mon disque)
  append="aha1542=0x230" # append... voir explication plus bas
  label = old       # Texte a taper : old
  read-only         #
```

Plusieurs questions peuvent vous effleurer l'esprit :
1. read-only : lorsque vous amorcez Linux, il va faire une vérification de la cohérence des informations des partitions Linux. C'est une vérification des systèmes de fichiers. Pour réaliser cette opération, il est indispensable que la partition soit en lecture seule. En effet, vérifier un disque en cours d'utilisation pourrait tout détruire. Cette vérification n'a pas lieu à tous les coups si vous éteignez bien la machine (appuyez sur les touches *Ctrl-Alt-Del* ou utilisez les commandes halt, reboot et shudown). Cette vérification n'est de toute manière lancée que sur les partitions Linux. C'est le programme fsck qui effectue cette opération. Je vous déconseille de le lancer "pour voir" : une utilisation erronée peut provoquer des pertes de données sur le disque !
2. pourquoi deux noyaux ? C'est assez simple : Linux change, bouge. Les noyaux sont modifiés assez souvent. Par exemple, on peut amorcer sa machine soit avec un noyau stable, soit avec un noyau de développement.
3. append : Cette option est très importante : elle permet de donner un ou plusieurs paramètres au noyau Linux. Par exemple, ici on lui indique qu'une carte SCSI se trouve à l'adresse 230. Dans le cas contraire, elle ne serait pas reconnue. Cette option est utilisée également si vous avez plusieurs interfaces réseau, SCSI, etc.
Pour faire prendre en compte au noyau une augmentation de mémoire centrale, il faut rajouter la ligne d'option append=80M dans le fichier /etc/lilo.conf, puis relancer le programme lilo.
Ex de syntaxe : append="mem=80M aha152x=0x340,10,5" ou bien (plus ennuyeux) au prompt de lilo, taper la commande linux mem=32M
4. Pour installer le fichier, exécutez /sbin/lilo et c'est tout !
En cas de problème avec LILO il faut d'abord essayer de supprimer l'option "compact" qui peut se trouver dans son fichier de configuration.
Ne surtout pas oublier de relancer LILO après chaque modification de son fichier de configuration !

4.3 - Ajouter un utilisateur, un groupe

Si vous possèdez la Red Hat, vous pouvez utiliser le panneau de configuration (control-panel ou bien directement usercfg, utilisables par root sous X Window) : c'est vraiment très bien fait mais CELA N'EMPECHE PAS DE CONNAITRE LES VRAIS COMMANDES.

Ajouter un utilisateur se fait en deux temps et trois mouvements. Deux fichiers sont importants : /etc/passwd et /etc/group.
Une ligne du fichier passwd est composée de cette manière :
gandalf::501:100:Eric Dumas:/home/gandalf:/bin/bash
sgandalf::0:0:Super Eric Dumas:/:/bin/sh
Une ligne est composée de 7 colonnes :
1. Nom de l'utilisateur : 8 caractères max ;
2. Mot de passe chiffré ;
3. Numéro d'identificateur unique de l'utilisateur (uid). Rq : 0 = root ;
4. Numéro du groupe (gid) ;
5. Nom "officiel" de l'utilisateur (dit GECOS) ;
6. Répertoire racine de l'utilisateur ;
7. Interpréteur de commandes.
Si le champ mot de passe contient le caractère x, c'est que vous utilisez les shadow. Le mot de passe chiffré se trouve dans le fichier /etc/shadow. Cette configuration permet en plus de définir la durée de validité des mots de passe.
Le fichier /etc/group contient la liste des groupes de la machine. Par exemple, le groupe 100 est :
users::100:games,gandalf
Pour ajouter un utilisateur *toto*, faire
1. ajoutez-le dans /etc/passwd ;
2. si besoin est, créez le groupe dans lequel il va se trouver, en modifiant /etc/group ;
3. créez son home directory :

```
mkdir /home/toto
cp /etc/skel/* /home/toto
chown -R toto.le_groupe_de_toto /home/toto
```

4. Enfin, donnez-lui un mot de passe en tapant passwd toto .
Vous n'avez plus qu'à essayer.
Je vous conseille très fortement de travailler sous un utilisateur commun. En effet, il est très dangereux de travailler en root. Une fausse manipulation et vous devez tout réinstaller, voire perdre de précieux fichiers personnels.

4.4 - Montage automatique de partitions : /etc/fstab

Le fichier /etc/fstab contient toutes les informations concernant le montage de vos partitions.
Voici un exemple :

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

```
none            /proc       proc        defaults        0       0
# Partition de swap
/dev/hda3       none        swap        defaults        0       0
# Fichier de swap
/usr2/swap_file /usr2       swap        defaults        0       0
# NFS
zeppelin:/users /users  nfs     rw,rsize=8192,wsize=8192,noauto,user,intr,exec
```

Voici une description des six colonnes :

1. device (périphérique) de la partition. Dans le cas d'un fichier de swap, c'est le nom du fichier.
2. point de montage de la partition ;
3. type de la partition ;
4. options (vous pouvez spécifier une partition en lecture seule, etc).
5. fréquence correspond au nombre de jours entre deux traitements du fichier par la commande dump. Cette commande n'existe que pour ext2fs (c'est un portage de la version 4.4BSD) mais n'est pas encore incluse dans toutes les distributions.
6. Ordre de tests des partitions (fsck). Si vous mettez 0, aucune vérification automatique n'est effectuée lors du démarrage. Les partitions situées sur un même disque seront testées d'une manière séquentielle mais si elles sont situées sur deux disques différents, c'est fait en parallèle.

Ajouter ici les partitions dont vous avez besoin.

Les volumes NFS profiteront des paramètres suivants : rsize=8192,wsize=8192, surtout si vous possèdez une carte réseau rapide.

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

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

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

la RedHat fournit également l'utilitaire fstool sous X qui permet de rajouter et supprimer les différents périphériques montés à l'aide de quelques clics de souris et autres boîtes de dialogues.

4.5 - Montage manuel de partitions

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

Le -t indique le type du support. Parmi les plus courants, on trouve

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

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

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

Pour démonter, tapez umount /mnt.

4.6 - Le fichier /etc/syslog.conf

Le démon syslogd permet de récupérer différents messages du noyau, qui concernent des messages de lancement de certains programmes, des messages de loggin, et également des messages d'erreur.
Il est parfois intéressant de pouvoir consulter ces messages pour détecter une éventuelle erreur ou le dysfonctionnement d'un programme ou d'un service.
Pour cela le fichier syslog.conf permet avec des règles de choisir où ces messages doivent parvenir.
Il peut donc etre utile de rajouter en première ligne du fichier /etc/syslog.conf :

```
*.*                     /dev/tty8
```

Elle va donc rediriger tout ces messages dans la console tty8.
Il faut rappeller que pour changer de console, il suffit de faire Alt+Fx, Fx correspondant à la touche de fonction comportant le numéro de la console que vous voulez atteindre. Sous X, il faut faire Ctrl+Alt+Fn.

4.7 - Mise en place des quota

4.7.1 - Logiciels nécessaires

Le support des quota disque a été intégré dans le noyau Linux depuis la version 1.3.46. Il vous faut donc utiliser un noyau postérieur pour pouvoir bénéficier des quota.
Vous avez également besoin des outils de gestion des quota. On peut les trouver sur :
ftp://ftp.funet.fi/pub/Linux/PEOPLE/Linus/subsystems/quota
ou, plus près, sur :
ftp://ftp.lip6.fr/pub/linux/kernel/sources/subsystems/quota

4.7.2 - Compilation des utilitaires

Après avoir extrait le contenu de l'archive , il vous faut compiler ces utilitaires s'il s'agit d'un fichier *.tar.gz*. Cela se fait en tapant tout simplement make. Dans certains cas (selon la version de bibliothèque C utilisée), il peut être nécessaire d'ajouter -I. à la variable CFLAGS définie dans le Makefile.
Si vous utilisez le système de fichiers ext2 et que vous avez installé les bibliothèques contenues dans la distribution e2fsprogs (utilitaires de gestion du système de fichiers ext2), vous pouvez inclure le support ext2fs dans les utilitaires des quota, afin d'accélérer leur exécution. Pour cela, éditez le fichier Makefile afin que la variable CFLAGS contienne -DEXT2_DIRECT et que la variable EXT2LIBS contienne -lext2fs -lcom_err avant de compiler les programmes.
Après la compilation, installez les utilitaires par la commande make install. Attention, la commande quota est installée dans le répertoire /usr/ucb qui n'existe pas sur la plupart des installations Linux. Il vous faut donc modifier le Makefile pour installer cette commande dans un autre répertoire (/usr/bin est un bon choix). Il peut également être nécessaire de définir la variable LN à la valeur ln -sf.
Si il s'agit d'un package rpm, la commande est decrite précedemment.

4.7.3 - Activation de la gestion des quota dans le noyau

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

4.7.4 - Activation des quota sur les systèmes de fichiers

Il existe deux types de quota : les quota liés aux utilisateurs et les quota liés aux groupes d'utilisateur. Les premiers définissent le nombre maximal de fichiers et de blocs disque associés à chaque utilisateur, les seconds définissent le nombre maximal de fichiers et de blocs disque associés à chaque groupe d'utilisateurs. Les deux types de quota peuvent être activés séparément.
Pour activer les quota pour les utilisateurs, il faut indiquer l'option usrquota pour les systèmes de fichiers concernés dans /etc/fstab. Les quota concernant les groupes sont régis par l'option grpquota. Les fichiers de définition des quota s'appellent respectivement quota.user et quota.group et sont situés dans la racine de chaque système de fichiers concerné.
Il est possible de modifier les noms de fichiers de gestion des quota en utilisant les syntaxes suivantes :
usrquota=nom_de_fichier
grpquota=nom_de_fichier
Voici un exemple de fichier /etc/fstab :

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

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

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

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

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

Cette commande doit également être exécutée la première fois pour créer les fichiers deux fichiers de gestion des quotas : quota.user et quota.group. En effet, ils ne sont pas créés automatiquement.
Si les utilitaires ont été compilés avec -DEXT2_DIRECT, la commande quotacheck doit être relativement rapide. Dans le cas contraire, elle peut être très lente car elle doit explorer tous les répertoires du système de fichiers de manière hiérarchique. Il est, de toutes façons, conseillé de la lancer automatiquement lors du démarrage avant l'activation des quota par quotaon :

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

4.7.5 - Attribution d'un quota à un utilisateur

La commande edquota est utilisée pour affecter un quota à un utilisateur ou à un groupe d'utilisateurs. Sa syntaxe est edquota -u utilisateur ou edquota -g groupe. Cette commande

lance un éditeur de texte contenant la définition des quota attribués à l'utilisateur ou au groupe et prend en compte leur nouvelle valeur lorsque le fichier est ré-écrit.
Pour chaque utilisateur ou groupe, il existe deux limitations : le nombre de fichiers et le nombre de blocs disque (exprimés en blocs de 1024 octets). Pour chacune, il existe deux limites :
1. la limite "douce" : lorsque cette limite est atteinte ou dépassée, un message d'avertissement est affiché lors de chaque nouvelle allocation de fichier ou de bloc ;
2. la limite "dure" : lorsque cette limite est atteinte, il est impossible à l'utilisateur de créer de nouveau fichier ou d'allouer de nouveau bloc.
La limite "douce" se transforme en limite "dure" quand elle a été atteinte ou dépassée depuis un certain temps (sept jours par défaut).

4.7.6 - Statistiques sur les quota

Tout utilisateur peut obtenir l'état des quota qui lui sont attribués (les limites sur le nombre de fichiers et de blocs qui lui sont alloués) grâce à la commande quota.
Le super-utilisateur peut obtenir les mêmes informations sur tout utilisateur ou groupe avec la même commande : quota -u utilisateur ou quota -g groupe. Il peut de plus utiliser la commande repquota pour obtenir une liste des quota associés à un ou plusieurs systèmes de fichiers.

4.7.7 - Autres documentations sur les quota

L'implémentation des quota dans Linux est compatible avec l'implémentation originale dans 4.2BSD Unix. Toutes les documentations concernant les quota dans BSD, que ce soit au niveau des commandes ou au niveau de l'interface de programmation, sont donc utilisables.
Les pages de manuel des utilitaires sont assez claires et très instructives. Enfin, les documentations diffusées en format HTML forment une bonne introduction à la gestion des quota.

4.8 - mtools

Plutôt que de monter vos partitions MS-DOS, je vous conseille d'utiliser les mtools. Il s'agit de l'équivalent des commandes MS-DOS, mais avec un m avant !
Exemple : mcopy toto.c a: . Si vous souhaitez copier un fichier texte pour qu'il soit exploitable sous MS-DOS, ou l'inverse, n'oubliez pas d'ajouter l'option -t : cela convertit le caractère de fin de ligne qui est différent entre les deux systèmes.
En principe, ils sont déjà à peu près configurés. Au cas où, vous devez modifier le fichier /etc/mtools.conf.

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

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

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

mtools_lower_case=1
```

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

4.9 - Partition de swap

4.9.1 - Swap : création de la partition

Pour rajouter une partition de swap de 8 Mo (8192 Ko), il faut d'abord créer la partition à l'aide du programme fdisk, puis soit vous passez par le programme *setup* de la Slackware, soit vous le faites à la main (ATTENTION : commande DANGEREUSE, destruction du contenu de la partition concernée ASSURÉE) : mkswap /dev/hda3 8192
Rajoutez la ligne /dev/hda3 swap swap defaults 0 0 dans /etc/fstab.
Enfin, pour que ça fonctionne, réamorcez la machine ou lancez swapon -a.

4.9.2 - Swap : fichier

Pour le fichier, c'est plus compliqué. Invoquer :
dd if=/dev/zero of=/usr2/swap_file bs=1024 count=8192
mkswap /usr2/swap_file 8192
Ajoutez dans /etc/fstab la ligne : /usr2/swap_file /usr2 swap defaults 0 0.
Faites bien attention à mettre cette ligne après le montage de la partition /usr2. Sinon, ce n'est pas près de fonctionner.
Désormais, vous êtes obligé pour quitter Linux de faire un
swapoff -a
reboot #Ou autre
En effet, si vous ne désactivez pas le fichier d'échange, Linux ne va pas pouvoir démonter la partition, et donc, il fera un fsck dessus à chaque fois que vous relancerez votre machine. Il est aussi possible, avec certains systèmes, de placer la commande swapoff dans /etc/rc.d/rc.K ou bien /etc/rc.d/init.d/halt afin d'automatiser cela.

4.10 - Enchainement des scripts de démarrage

Le processus init lit ses paramètres dans /etc/inittab. Ce dernier donne les scripts à lancer en fonction d'un niveau de démarrage. Le niveau 3 est le niveau par défaut. « init » lance ensuite le scripts /etc/rc n qui va par la suite lancer les scripts présents dans le répertoire /etc/rc.d/rcn.d ou " n " est le niveau de démarrage choisi... exemple : /etc/rc 3 -> /etc/rc.d/rc3.d

Dans ce répertoire on trouve un ensemble de scripts commencant par S ou K et qui sont des liens vers /etc/rc.d/init.d. Pour un niveau donné, afin de lancer ou ne pas lancer des daemons au démarrage il suffit de mettre ou d'enlever les scripts correspondant dans le répertoire rc3.d.
On trouve des fichiers commencant par S s'il s'agit de lancer le script ou K s'il s'agit de tuer un process... Par exemple, pour ne pas démarrer en niveau 3 le serveur NIS , il suffit d'invalider le script /etc/rc3.d/S65ypserv , simplement en renommant S65ypserv en K65ypserv.

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

C'est dans ce fichier que vous allez mettre tous les programmes personnels que vous souhaitez lancer au boot. En voici une version :

```
#! /bin/sh

# du menage en v'la,...
echo "Cleaning /tmp..."
find /tmp -type f -mtime +1 | xargs -r rm
find /tmp -type d -mtime +1 | xargs -r rmdir

# Vive l'azertyuiop :-)

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

Ce qui est important, c'est le clavier français. Si vous ajoutez cette ligne, le clavier sera français dès le boot.
En cas de problème il faut modifier ainsi fr-latin1.map :

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

4.12 - Ordonnancement de travaux : la crontab

La crontab est une table qui contient des commandes qui doivent être lancées à intervalle régulier. Vous pouvez y mettre la sauvegarde journalière, etc.
Le format est assez simple :

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

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

Ensuite vient la commande. Cette crontab est très utile si vous utilisez UUCP (pour envoyer le courrier, récupérer les news, etc).
Tout utilisateur peut se créer une crontab grâce à la commande crontab.

4.13 - Imprimer !

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

Le démon lpd doit être lancé au boot. Il est lancé soit dans rc.M soit dans rc.inetd2 (vous devez le voir lorsqu'il lance tous les démons de la machine. Si ce n'est pas le cas, il faut décommenter quelques lignes dans l'un des fichiers cités).
Supposons que l'imprimante s'appelle betty. Dans un premier temps, dans le fichier /etc/rc.d/rc.local, ajoutez
echo "Setting up printer betty"
/usr/sbin/lpc up betty
Passons maintenant aux choses sérieuses. Le fichier dans lequel va être décrit l'imprimante est le fichier /etc/printcap. Exemple :
betty|lp:lp=/dev/lp1:sd=/var/spool/lp1:sh:\
lf=/var/adm/lpd-errs:of=/etc/start-dj500:
La première colonne représente le nom de l'imprimante. J'ai mis betty et lp car de cette manière, c'est l'imprimante par défaut. Ensuite, la configuration se fait via des mots-clefs. Pour plus d'information, faites man printcap. Voici quand même une description de ceux utilisés.
– lp : device sur lequel est branché l'imprimante (LPT1 ici) ;
– sd : répertoire du spool d'impression ;
– sh : supprime la page d'entête ;
– lf : fichier d'erreurs ;
– of : programme de filtrage.
– rm : remote machine qui gère l'imprimante
– rp : nom de l'imprimante sur la machine distante
Comment ça marche ? Bon, supposons que j'ai un fichier toto.ps. Je fais donc lpr toto.ps (ou lpr -Pbetty toto.ps). Le fichier va être déposé dans le répertoire /var/spool/lp1.
Le démon va ensuite envoyer le fichier sur le programme de filtrage. Ce système permet d'imprimer des fichiers ayant des formats assez variés. Utiliser lprm pour retirer une requête d'impression active.

Exemples :

- accès à des imprimantes distantes accessibles en réseau par d'autres stations Unix.

```
##PRINTTOOL1## REMOTE
lp1|deskjet:\                          <== nom local de l'imprimante (coté linux)
        :sd=/var/spool/lpd/lp1:\       <== zone de spool local pour cette imprimante
        :mx#0:\                     <== maximum file size [0=illimité]
        :rm=lpserv.com.univ-plm.fr:\              <== remote machine qui gère l'imprimante
        :rp=deskjet:              <== nom   de l'imprimante sur la machine distante
##PRINTTOOL1## REMOTE
lp|laser4m:\
        :sd=/var/spool/lpd/lp:\
        :mx#0:\
        :rm=lpserv.com.univ-plm.fr:\
        :rp=laser4m:
```

- accès sur des imprimantes réseau (qui ont une interface ethernet) et postscript. Dans ce cas on peut directement adresser l'imprimante, sans passer par une machine gérant un spool distant, pour soumettre les travaux d'impression

```
lp|laserlum5eme:\
:lp=/dev/null:sh:\        <=== nom du device d'impression local (/dev/lp normalement)
:sd=/var/spool/lpd/lp:\        <=== zone de spool locale pour cette imprimante
:mx#0:\
:rm=imprr5.com.univ-plm.fr:rp=raw:      <== nom reseau IP de la machine distante gérant son spool
#      :rp=laserlum5eme:\                <=== remote printer : nom de l'imprimante
#     :if=/var/spool/lpd/lp/filter:          <=== filtre éventuel de conversion des données
```

- Accès à une imprimante sur port parallèle : "personnelle"
On peut utiliser le programme de configuration printtool de la distribution RedHat. IL n'y a pas de probleme pour faire reconnaitre l'imprimante sur un port parallèle.

4.14 - Lancer des " R "(emote) commandes

Pour autoriser un utilisateur sur une machine distante à lancer des commandes sur la machine locale, il faut l'autoriser en le répertoriant dans le fichier ~/.rhosts
Ceci n'est peut être pas suffisant...il arrive que " root " ne puisse pas se rloguer ou faire des remote commande (comme rsh) depuis une autre machine. Ceci est dû au système de sécurité de /etc/securetty visant à n'autoriser des connexions que depuis un terminal " sûr ". On sécurise les accès depuis une console authentifiée.
Pour invalider cette sécurité il faut commenter la ligne suivante dans le fichier /etc/pam.conf.

```
#rlogin auth required
```

et laisser

```
rlogin auth sufficient
```

La commande /sbin/pamconfig permet de paramétrer ce fichier. Ce fichier /etc/pam.conf régit la possibilité d'utiliser ou pas la plupart des rcommandes à distance...depuis certains comptes ou certaines consoles tty.

4.15 - locate

Lorsque vous installez le package GNU "find", en plus du programme de recherche se trouve un programme locate qui permet de retrouver rapidement l'emplacement d'un fichier. Ce programme se trouve dans certaines distributions dans /usr/lib/locate.
Lancez, en tant que root, /usr/lib/locate/updatedb. Cela invoquera un find / sur les disques montés et placera les noms de tous les fichiers (sous forme d'une table de hachage) dans /usr/lib/locate/find.codes. locate permettra ensuite de localiser l'un d'eux.
Remarque : lorsque vous lancez ce programme, pensez à démonter les partitions MS-DOS et votre CD-ROM !
Une autre solution assez agréable consiste à insérer dans la *crontab* la ligne :

```
updatedb --prunepaths='/tmp /usr/tmp /var/tmp /proc /users /root /dos
/mnt /var/adm /var/spool /var/catman /home'
```

Cette commande réalise la mise à jour de l'arborescence sans les fichiers temporaires, l'arborescence utilisateur, une partition MS-DOS, etc.

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

4.16 - Mettre une console externe comme console

Certains utilisateurs peuvent vouloir mettre comme console un minitel ou un vieux vt100 qui ne passe pas sur la carte vidéo mais sur le port série. Il suffit d'ajouter dans le fichier lilo.conf la ligne suivante :
append = "scon=0x03f8 serial=0,9600n8"

4.17 - Les fichiers d'initialisation des interpréteurs de commandes

Voici la description des fichiers qui sont chargés lors de la connexion d'un utilisateur. Le nom du ou des fichiers est spécifique à l'interpréteur de commandes utilisé.
Dans le cas de bash, c'est un peu compliqué. D'après la page de manuel de bash, lors d'une connexion interactive, il exécute :
– /etc/profile s'il existe ;
– ~/.bash_profile s'il existe, sinon ~/.bash_login s'il existe, sinon ~/.profile s'il existe ;
– ~/.bashrc s'il existe.
Dans le cas où bash est invoqué en tant que sh, il n'exécute que les fichiers /etc/profile et ~/.profile s'ils existent.
Dans le cas de ksh, la page de manuel est moins précise mais une étude rapide du code source montre que :
– /etc/profile est exécuté s'il existe ;
– ~/.profile est exécuté s'il existe, si le numéro d'utilisateur effectif est identique au numéro d'utilisateur réel et si le numéro de groupe effectif est égal au numéro de groupe réel ;
– /etc/suid_profile est exécuté s'il existe, si les numéros d'utilisateur effectif et réel diffèrent ou si les numéros de groupe effectif et réel diffèrent ;
– enfin, si la variable d'environnement ENV est positionnée, le fichier dont le nom est contenu dans $ENV est exécuté s'il existe.
Enfin, pour être complet sur les interpréteurs de commandes compatibles avec le Bourne Shell, citons le cas de zsh qui exécute :
– /etc/zshenv s'il existe ;
– ~/.zshenv (ou $ZDOTDIR/.zshenv si ZDOTDIR est positionné) s'il existe ;
– /etc/zprofile s'il existe ;
– ~/.zprofile (ou $ZDOTDIR/.zprofile) s'il existe ;
– /etc/zshrc s'il existe ;
– ~/.zshrc (ou $ZDOTDIR/.zshrc) s'il existe.
Dans le cas de csh (enfin plutôt tcsh) :
– /etc/csh.cshrc est exécuté s'il existe ;
– /etc/csh.login est exécuté s'il existe ;
– ~/.tcshrc est exécuté s'il existe, sinon ~/.cshrc est exécuté s'il existe ;
– ~/.history (ou le contenu de la variable histfile) est chargé en mémoire s'il existe ;
– ~/.login est exécuté s'il existe ;
– ~/.cshdirs (ou le contenu de la variable dirsfile) est chargé en mémoire s'il existe.

4.18 - Les sauvegardes sous Linux

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

4.18.1 - dump et restore

Les outils dump et restore ont été adaptés pour Linux par Rémy Card. Ils sont livrés dans les distributions.
Une fois les sources compilés et installés, l'utilisation de dump et restore est relativement simple. Pour effectuer la sauvegarde d'une partition /dev/sda1 sur /dev/rmt0, il suffit de faire par exemple :
gandalf# dump 0sfu 3600 /dev/rmt0 /dev/sda1
gandalf# dump 0sfu ondee:/dev/rmt0 /dev/sda1
La deuxième commande permet de sauvegarder un disque sur un périphérique distant (par exemple situé ici sur la machine ondee). Les options de dump peuvent sembler complexes. En voici une courte description :
– 0 à 9 : niveau de sauvegarde. 0 correspond à une sauvegarde complète, alors que les autres niveaux n correspondent à la sauvegarde des fichiers qui ont été modifiés depuis la nième sauvegarde ;
– s : taille de la bande en pieds ;
– f : fichier - peut être composé de machine:fichier ;
– u : écriture de la date et du niveau de sauvegarde dans le fichier /etc/dumpdates.
Il existe d'autres options. Pour plus renseignements, consultez les pages de manuel.
Il existe deux modes pour effectuer une restauration : en ligne de commande ou en mode dit "interactif". Le deuxième mode est plus simple pour des restaurations partielles. Le premier est surtout utilisé pour des restaurations complètes.
Pour restaurer la bande en mode interactif il suffit de faire
gandalf# restore -if /dev/rmt0
gandalf# restore -if ondee:/dev/rmt0
Dans ce cas, un mini-interpréteur de commandes est lancé. Utiliser la commande help pour plus de détails.
Pour restaurer une bande complètement, lancez :
gandalf# restore rf /dev/rmt0
Note importante : pour l'utilisation de dump et restore à travers un réseau (sauvegarde sur des périphériques distants), vous devez utiliser des fichiers .rhosts. Dans l'exemple de sauvegarde ci-dessus, la machine ondee doit avoir
#cat ~root/.rhosts
gandalf
#
Attention toutefois aux failles de sécurité engendrées par les fichiers .rhosts.
L'utilisation de périphériques distants nécessite également la présence du programme rmt sur la machine gérant les périphériques de sauvegarde. Ce programme est inclus dans la distribution source de dump pour Linux.

4.18.2 - tar

A la différence de dump ou restore, tar permet de sauvegarder les fichiers désirés, d'exclure certains répertoires, etc. Il est à noter que le tar utilisé sous Linux est le tar GNU. Il possède certaines options particulières.
Pour connaître toutes les options possibles, je vous conseille de faire tar --help . Une utilisation simple de tar peut être illustrée avec la sauvegarde d'une partition d'utilisateurs :
tar cvf /dev/rmt0 /users | mail backup-user
La liste des fichiers sera ainsi envoyée à l'utilisateur *backup-user*.
Certains sites utilisent exclusivement tar pour effectuer leurs sauvegardes, c'est un choix.

4.19 - Num Lock au démarrage

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

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

4.20 - Les fichiers core

Certains développeurs aiment exploiter le contenu des fichiers core. Or, sous Linux, ils sont désactivés par défaut. Pour pouvoir les générer, il faut introduire : ulimit -c unlimited sous bash ou limit coredumpsize unlimited

4.21 - SMP, autres architectures

SMP, abréviation de *Symetric Multi-Processors*, désigne la capacité des noyaux Linux 2.0 et versions suivantes de fonctionner sur des machines à plusieurs processeurs. Sur architecture ix86, le support SMP de Linux suit les spécifications Intel MP (v1.1, v1.4) qui vont théoriquement jusqu'à 16 processeurs. En fait, le SMP n'est pas spécifique aux plates-formes Intel : il existe également sur les plates-formes Alpha, Sparc, PowerPC. La version 2.0 ne gère le SMP que pour les plates-formes de type Intel.
Pour compiler un noyau avec le support SMP, il faut *décommenter* la ligne SMP = 1 dans le fichier Makefile. Il est en outre recommandé de compiler le support pour l'*Enhanced Real Time Clock* (CONFIG_RTC).
Lors du démarrage du noyau, Linux boot d'abord sur un processeur, puis active le(s) autre(s).
Extraits des informations données par la commande dmesg :

```
Intel MultiProcessor Specification v1.1
   Virtual Wire compatibility mode.
OEM ID: INTEL    Product ID: Nightshade   APIC at: 0xFEE00000
Processor #1 Pentium(tm) Pro APIC version 17
Processor #0 Pentium(tm) Pro APIC version 17
I/O APIC #2 Version 17 at 0xFEC00000.
Processors: 2
...
Calibrating delay loop.. ok - 396.49 BogoMIPS
...
Booting processor 0 stack 00002000: Calibrating delay loop.. ok - 396.49 BogoMIPS
```

Total of 2 processors activated (792.99 BogoMIPS).
Ce qui précède suffit pour avoir un système multiprocesseur qui fonctionne. Ce qui suit va le rendre plus agréable à utiliser et plus efficace.
Le package procps contient entre autre les commandes top et ps qui peuvent être modifiées pour afficher le numéro du processeur sur lequel une commande tourne. La marche à suivre ainsi que les patches sont disponibles (en anglais) à :
http://www.cs.inf.ethz.ch/~rauch/procps.html.
GNU make accepte l'option -j qui spécifie un nombre maximal de tâches pouvant être lancées simultanément. Il peut donc être intéressant d'utiliser cette option (make -j 5 par exemple) sur un système multi-processeurs.
Dans la majorité des cas, un système SMP ne se distingue pas parce qu'une même tâche se déroule plus vite, mais bien parce que l'on peut faire tourner plusieurs grosse tâches simultanément (par exemple compiler un programme C et un fichier LaTeX en même temps) ou bien parce que le système répond instantanément malgré la présence d'une grosse tâche (p.ex. lire son mail tout en compilant le noyau).
Pour vraiment faire en sorte qu'un même programme utilise toute la puissance CPU de la machine, il faut investir dans le parallélisme. Lire à ce sujet l'excellentissime *Linux Parallel Processing HOWTO* (que l'on trouve aux endroits habituels).
Pour en savoir plus, vous pouvez consulter la Linux SMP FAQ de David Mentré, http://www.irisa.fr/prive/mentre/smp-faq/.

4.22 - Opérations sur disques et file systems

Pour créer un file system de type ext2 sur une disquette
```
mke2fs -c /dev/fd0
```
Pour contrôler l'etat d'un file system
```
e2fsck /dev/fd0
```
Formatter une disquette de densité 1.4
```
fdformat /dev/fd0H1440
```

4.23 - Comment générer une disquette de boot

Il peut être très utile d'avoir sur disquette un noyau pour booter dessus. Pour faire cela, prenez une disquette formatée et allez à la racine de votre disque.
Identifiez votre noyau. Il doit s'appeler zImage ou vmlinuz.
Vous le copiez d'abord sur disquette :
```
cat /zImage > /dev/fd0
```
puis faîtes (exemple) :
```
rdev /dev/fd0 /dev/hda2
rdev -R /dev/fd0 1
```
Le deuxième paramètre de la première commande doit être le nom de la partition racine Linux de votre système.
Cette disquette vous permet alors de booter (tester !).
Si vous utilisez la Slackware, vous pouvez aussi utiliser la commande /sbin/makebootdisk qui fera tout cela à votre place.

4.24 - Mon mot de passe, où est mon mot de passe ?

En bref : vous avez perdu le mot de passe de root... Bon, il y a plusieurs solutions.

La première :
1. bootez avec les disquettes boot et root de votre distribution.
2. loguez-vous en root (là, pas de mot de passe).
3. montez la racine du disque dur : mount -t ext2 /dev/hda1 /mnt
4. allez dans le répertoire /mnt. Il ne reste plus qu'à éditer le fichier /mnt/etc/passwd, y supprimer le mot de passe de root : root::0:0:root:/root:/bin/bash.

Vous rebootez Linux normalement, et en principe, vous n'avez plus besoin de mot de passe pour vous loguer en root.
Bon, passons à d'autres solutions : rebooter le pc en mode dit *single-user*. Pour cela, lors du boot avec LILO, fournir LILO: linux single (remplacer ici "linux" par le nom sous lequel LILO connaît votre noyau). Un shell root va apparaître. Attention : le clavier est en qwerty et la partition en lecture seule. Pour y remédier :

```
loadkeys /usr/lib/kbd/keytables/fr.map
mount -w -n -o remount /
```

Soit vous utilisez la commande passwd, soit vous éditez le fichier /etc/passwd
Si vous n'avez pas de disquette de boot, vous pouvez souvent vous en sortir en passant init=/bin/sh lors de l'amorçage. Ensuite, monter la racine et éditer à la main le fichier /etc/passwd.

4.25 - Consoles virtuelles

4.25.1 - Aspects pratiques

Linux vous permet de travailler en même temps sur plusieurs fenêtres textes. Pour passer de l'une à l'autre, fais Alt + Fn ou n est le numéro de la fenêtre virtuelle et Fn une touche de fonction (par exemple F1, F2 ...). Vous pouvez configurer le nombre de fenêtres dans le fichier /etc/inittab.
Quand vous êtes sous X, la combinaison devient Ctrl + Alt + Fn. Il est alors possible de retourner sous X en accédant ainsi à la console qu'il emploie (la première "libre", c'est-à-dire non gérée par un getty).
Utiliser maj + pages (touches SHIFT et page précédente ou page suivante) pour "remonter". Ceci fonctionne aussi sous xterm.

4.25.2 - Aspects théoriques

4.25.2.1 - Principe

Les consoles virtuelles sont des périphériques. Elles sont au nombre de 63 (ou plus, ou moins, si l'on change ça lors de la compilation du noyau). Elles correspondent aux fichiers "spéciaux" tty1 ... tty63 du répertoire /dev.
Maintenant, pour pouvoir les utiliser, il faut pouvoir attacher un processus à ce terminal. C'est le rôle des utilitaires tels que agetty. Il est d'ailleurs intéressant de suivre (à l'aide de top) l'évolution des processus lors du login :
Tout d'abord, agetty (de propriétaire root) vous demande votre login. Dès que vous appuyez sur entrée, agetty exécute le programme login (toujours de propriétaire root) par recouvrement.
login (attaché à la VC puisqu'héritant de agetty) vous demande votre mot de passe. Il lance alors votre shell de login (de propriétaire "vous").

Vous pouvez noter que quand vous terminez votre session (i.e. votre shell de login), un agetty est relancé.

4.25.2.2 - Configuration

C'est là qu'intervient le fichier inittab. init (de PID 1), est le processus ancêtre de tous les processus. Comme c'est le premier processus, c'est lui qui se charge de lancer les agetty. Il se sert pour cela du fichier de configuration /etc/inittab.
Entre autres choses, ce fichier contient des lignes du type :

```
c1:1235:respawn:/sbin/agetty 38400 tty1 linux
```

Pour une explication, se référer à la page de manuel d'inittab(5). Ca veut dire en gros que init lance un agetty sur /dev/tty1 qu'il relance chaque fois qu'il meurt (respawn) avec un type de terminal Linux pour les cas où on est dans les "runlevels" 1,2,3 ou 5.
Donc, il faut mettre une ligne de ce type pour chaque console virtuelle.
On voit ici qu'il y a autant de agetty lancés que de consoles virtuelles utilisables, et qu'on est limité par ce nombre des lignes dans inittab (sauf si l'on lance 63 agetty...).
Il existe une autre méthode, c'est l'allocation de VC dynamiquement. Cela se fait grâce à un petit démon (cspawnd) (ftp://ftp.lip6.fr/pub/linux/sunsite/utils/console/dynamic-vc-1.1.tar.gz) : « cspawnd is normally started in rc.local as a daemon that is signaled by the kernel when the console user presses the Spawn_Console key (see below). Upon receipt of a signal from the kernel, cspawnd activates the newly allocated VC and spawns a getty process to enable the user to login. In its default configuration, it also attempts to periodically deallocate unused VCs ».
Il convient alors de laisser une VC gérée par la première méthode, les autres étant alouées dynamiquement.

4.25.2.3 - Les touches

Maintenant, il faut pouvoir passer d'une VC à l'autre. C'est le noyau qui s'en charge. Quand une touche "ConsoleN" (N=1...63) est frappée, Le noyau fait correspondre la VC numéro N à la console (la vrai).
Il y a aussi "Decr_Console", "Incr_Console" qui respectivement, incrémente et décrémente le numéro de la console ; ainsi que Last_Console qui fait passer à la dernière console utilisée. cspawnd rajoute la touche "Spawn_Console" (voir plus haut).
Il ne reste plus qu'a faire correspondre ceci à des touches (ou des combinaisons de touches) du clavier. Cela est fait grâce à l'utilitaire loadkeys (lancé par init au boot [dans un fichier /etc/rcXXX]). Cet utilitaire prend en paramètre un fichier .map (sûrement /usr/lib/kbd/keytables/fr-lat1.map pour vous) qui contient des lignes du style :

```
            keycode  87 = F11 F11 Console_23
control     keycode  87 = F11
alt         keycode  87 = Console_11
control alt keycode  87 = Console_11
```

Bien sûr, vous pouvez retrouver tout ceci dans les pages de man et toutes les aides en ligne mises à votre disposition. Citons loadkeys(8), init(8), inittab(5), cspawnd(8) ainsi que les fichiers du répertoire /usr/src/linux/Documentation.

4.26 - X Window

Maintenant que tout fonctionne bien, on passe à un autre gros morceau : l'installation de X Window.
Pour pouvoir utiliser X Window, il faut posséder une configuration raisonnable (au moins 8Mo de mémoire). Dans le cas contraire, la machine sera très lente : elle va passer son temps à swapper.
Dans un premier temps, installez à l'aide de votre distribution le serveur correspondant à la carte graphique.

4.26.1 - XF86Config

Le problème avec X Window, c'est de configurer le fichier XF86Config. Il s'agit ici du serveur XFree86 3.2, basé sur X11R6. Avant de commencer à construire votre fichier, lancez le programme SuperProbe. Ceci donne par exemple les renseignements suivants :

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

Ensuite, le plus dur reste à faire : construire le fichier de configuration. Attention : dans certaines distributions, le fichier XF86config est placé dans le répertoire /etc/X11, ce qui est une bonne idée dans le cas d'un réseau de stations Linux partageant /usr par NFS. Si vous utilisez une ou plusieurs machines indépendantes, vous pouvez supprimer /etc/X11/XF86config et utiliser /usr/X11R6/lib/X11/XF86Config.
Les nouvelles versions de distributions sont livrées avec certains outils permettant d'automatiser l'installation et la configuration du serveur. Essayez dans un premier temps avec cet outil : il est fort probable que cela fonctionne directement. (Xconfigurator pour redhat.)
Ce fichier se présente sous la forme de sections. En voici une description :
– Fichiers

```
Section "Files"
   RgbPath     "/usr/X11R6/lib/X11/rgb"

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

Dans cette section, il n'y a pas grand chose à modifier. Ajoutez ou supprimez les chemins des polices de caractères, et la liste des couleurs RGB.
– Options pour le serveur :

```
Section "ServerFlags"
EndSection
```

– Clavier

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

```
    ServerNumLock

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

– La souris :

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

# Peripherique a utiliser
    Device      "/dev/mouse"
    BaudRate    1200

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

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

– Le moniteur. Il s'agit de détailler les caractéristiques du moniteur.. Les fichiers situés dans le répertoire doc contiennent les caractéristiques de tout un tas d'écrans. Il est fort possible que vous y trouviez votre bonheur.

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

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

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

– la carte graphique.

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

```
   # Nom de la carte
   BoardName   "clgd5428"

   #memoire video
   VideoRam    1024

   # nom du chip
   Chipset     "clgd5428"

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

– Enfin, on réunit tout : la section écran :

```
Section "Screen"
   Driver      "svga"

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

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

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

Si votre carte le permet, vous pouvez ajouter d'autres sous-sections pour le display :

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

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

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

1. lire la documentation se trouvant dans ./doc. Un bon nombre d'écrans et de cartes sont décrites avec les horloges à utiliser.
2. commencer avec une résolution faible (640x480) : c'est plus simple ;
3. si vous avez besoin des clocks de la carte, lancez X -probeonly 2>/tmp/err (si vous utilisez comme interpréteur de commandes csh ou tcsh, remplacez 2> par >&. Dans le fichier /tmp/err

vont se trouver tous les messages du serveur, dont le chip de la carte ainsi que ses clocks. La seule chose qu'il va rester à configurer, c'est l'écran.
4. lire le "Video HOWTO", écrit par J.M. Vansteene et disponible sur ftp.lip6.fr dans le répertoire /pub/linux/french/docs/HOWTO ;
5. utiliser le programme xvidtune : il permet "d'affiner" la mise au point de l'image (centrage, etc). Par exemple, lors d'un tremblement de l'image, on peut résoudre le problème avec ce programme.
Si jamais l'anglais vous rebute, vous pouvez lire une version française de toute la documentation XFree86 3.2, au format HTML. Pour cela, allez sur le serveur http://www.kheops.com.

4.26.2 - Clavier français sous X, ancienne méthode : Xmodmap

Comme XFree86 a été conçu par des gens se servant de claviers américains (qwertyiop), la configuration du clavier peut sembler être assez affolante. Je vous conseille de suivre ce qui suit...
Tout d'abord, vous devez avoir dans le fichier XF86Config les options suivantes :

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

Ensuite, placez le fichier suivant dans le répertoire /usr/X11R6/lib/X11/xinit. Il doit s'appeler .Xmodmap.
Le fichier qui suit a été fait par René Cougnenc.. Il fonctionne sur XFree86 3.2.

```
! clavier Français AZERTYUIOP pour XFree86 3.2
!
!
! Les accents circonflexes des principales voyelles sont obtenus avec
! la touche Mode_switch (ALT_GR).
!
! Les tremas sont obtenus par ALT_GR + SHIFT, sur 'e', 'i', 'o', et 'u'.
!
! Esperons qu'un jour on pourra obtenir le clavier Francais standard sur
! lequel tout le monde apprend a taper...
! ------------------------------------------------------------------
!
keycode  9 = Escape
keycode  10 = ampersand 1
keycode  11 = eacute 2 asciitilde
keycode  12 = quotedbl 3 numbersign
keycode  13 = apostrophe 4 braceleft
keycode  14 = parenleft 5 bracketleft
keycode  15 = minus 6 bar
keycode  16 = egrave 7 grave
keycode  17 = underscore 8 backslash
keycode  18 = ccedilla 9 asciicircum
keycode  19 = agrave 0 at
keycode  20 = parenright degree bracketright
keycode  21 = equal plus braceright
keycode  22 = BackSpace
keycode  23 = Tab
keycode  24 = a A acircumflex
keycode  25 = Z
keycode  26 = e E ecircumflex ediaeresis
keycode  27 = R
keycode  28 = T
keycode  29 = Y
keycode  30 = u U ucircumflex udiaeresis
keycode  31 = i I icircumflex idiaeresis
keycode  32 = o O ocircumflex
keycode  33 = P
keycode  34 =  dead_circumflex dead_diaeresis
keycode  35 = dollar sterling
keycode  36 = Return
keycode  37 = Control_L
keycode  38 = Q
keycode  39 = S
keycode  40 = D
keycode  41 = F
keycode  42 = G
keycode  43 = H
keycode  44 = J
keycode  45 = K
keycode  46 = L
keycode  47 = M
keycode  48 = ugrave percent
```

```
keycode  49 = guillemotleft guillemotright
keycode  50 = Shift_L
keycode  51 = asterisk mu
keycode  52 = W
keycode  53 = X
keycode  54 = C
keycode  55 = V
keycode  56 = B
keycode  57 = N
keycode  58 = comma question
keycode  59 = semicolon period
keycode  60 = colon slash
keycode  61 = exclam slash
keycode  62 = Shift_R
keycode  63 = asterisk
keycode  64 = Alt_L
keycode  65 = space
keycode  66 = Caps_Lock
keycode  67 = F1
keycode  68 = F2
keycode  69 = F3
keycode  70 = F4
keycode  71 = F5
keycode  72 = F6
keycode  73 = F7
keycode  74 = F8
keycode  75 = F9
keycode  76 = F10
keycode  77 = Num_Lock
keycode  78 = Scroll_Lock
keycode  79 = 7
keycode  80 = 8
keycode  81 = 9
keycode  82 = KP_Subtract
keycode  83 = 4
keycode  84 = 5
keycode  85 = 6
keycode  86 = KP_Add
keycode  87 = 1
keycode  88 = 2
keycode  89 = 3
keycode  90 = 0
keycode  91 = period
keycode  92 = 0x1007ff00
keycode  93 =
keycode  94 = less greater
keycode  95 = F11
keycode  96 = F12
keycode  97 = Home
keycode  98 = Up
keycode  99 = Prior
keycode 100 = Left
keycode 101 = Begin
keycode 102 = Right
keycode 103 = End
keycode 104 = Down
keycode 105 = Next
keycode 106 = Insert
keycode 107 = Delete
keycode 108 = KP_Enter
keycode 109 = Control_R
keycode 110 = Pause
keycode 111 = Print
keycode 112 = KP_Divide
keycode 113 = Mode_switch
keycode 114 = Break

! Nouvelles valeurs pour le pave numerique a partir
de XFree86 3.1.1
!keycode 147 = Home
!keycode 148 = Up
!keycode 149 = Prior
!keycode 150 = Left
!keycode 152 = Right
!keycode 153 = End
!keycode 154 = Down
!keycode 155 = Next
!keycode 156 = Insert
!keycode 157 = Delete

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

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

4.26.3 - Clavier français sous X... 2ième possibilité

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

Donc, il faut que ALT 4 et ALT 7 produisent de vrais accents aigus et apostrophe. Il nous faut quand même des touches muettes (en plus). Il va falloir modifier la "map" du clavier.
On va donc faire :
– ALT 4 = accent aigu en dur
– ALT 7 = accent grave en dur
– ALT 9 = circonflexe en dur
– ALT ^ = accent aigu
– ALT ù = accent grave
Dans le répertoire /usr/lib/kbd/keytables.map, copier fr-lat1.map sous un autre nom (comme par exemple monclavier.map). Lancer showkey pour voir le numéro des touches concernées dans l'ordre : 5 8 10 26 et 40. Puis éditer monclavier.map pour faire les modifications.
Pour tester, faire loadkeys monclavier. Lorsque tout semble fonctionner, modifier /etc/rc.d/rc.keymap pour qu'il charge *monclavier.map* et gribouiller le clavier avec un crayon, pas un feutre... :-).
On peut aussi mettre un "tilde muet" sur le 2 supérieur et un "tilde dur" (indispensable au shelliste) sur ALT-2.

4.26.4 - Clavier français sous X... utiliser xkb

Il est nécessaire d'utiliser un serveur XFree86 3.2 minimum pour mettre en place cela. Dans la section "Keyboard" du fichier de configuration, ajouter :

```
XkbKeycodes     "xfree86"
XkbTypes        "default"
XkbCompat       "default"
XkbSymbols      "us(pc101)"
XkbGeometry     "pc"
XkbRules        "xfree86"
XkbModel        "pc102"
XkbLayout       "fr"
```

Si le clavier n'offre pas 101 touches il faudra remplacer le "102" par "101", "pc102", "104" ou "105" afin de pouvoir d'utiliser les touches spéciales (MS-Windows).
Pour plus d'informations, regardez le contenu du répertoire /usr/X11R6/lib/X11/xkb.

4.26.5 - xdm

Pour démarrer X Window directement au boot (xdm), vous pouvez modifier le fichier /etc/inittab en changeant la ligne contenant initdefault. Par exemple, il suffit de remplacer :
id:3:initdefault: par id:5:initdefault:
Mais attention ! Certaines distributions n'utilisent pas les mêmes niveaux. Le plus simple est de regarder dans le fichier /etc/inittab : celui-ci contient généralement une description des niveaux d'exécution et de ce à quoi ils correspondent.
Pour lancer xdm en 16 ou 32 bits, il faut modifier le fichier /usr/X11/lib/X11/xdm et mettre :
:0 local /usr/X11/bin/X -bpp 16 (ou 32)

4.27 - Sécurité

4.27.1 - X Window : le fichier .Xauthority

Le fichier .Xauthority est généré par le programme xauth. Il s'agit d'un système d'authentification pour les applications graphiques. Cela permet d'éviter que d'autres personnes envoient des images, des fenêtres sur votre écran - mais également que des personnes puissent "voir" ce qu'il y a sur votre écran.
Le principe est de donner une clef d'identification, en hexadécimal avec un nombre pair de caractères.
Lancer xauth et faire :
add MaMachine:0 MIT-MAGIC-COOKIE-1 MonCode
add MaMachine/unix:0 MIT-MAGIC-COOKIE-1 MonCode
Pour la machine locale, c'est en fait "hostname:NoDisplay".
Un fois lancé, le serveur X interdit toute connexion, sauf si l'application :
– est exécutée sur une machine référencée ;
– possède le code.
Vous pouvez désactiver le système pour certaines machines avec un xhost +toto par exemple.
Rq: certaines versions obligent à lancer le serveur X avec la commande
xinit -- -auth $HOME/.Xauthority

4.27.2 - Protection de la machine contre l'extérieur

Une solution pour éviter les connexions externes est d'utiliser TCP/Wrappers. Il est très fortement conseillé de le recompiler !
L'installation est assez intuitive. En bref, il vous suffit d'indiquer le nom des machines autorisées dans le fichier /etc/hosts.allow et les machines interdites dans /etc/hosts.deny. On peut permettre l'envoi de courrier lorsqu'une machine tente de se connecter alors qu'elle est interdite en mettant par exemple dans le fichier /etc/hosts.deny :
wu.ftpd: ALL: twist = /usr/sbin/real-daemon-dir/safe_finger -l @%h |
/bin/mail -s %d-%h root
(Sur une seule ligne :-)).
Si vous voulez plus de détails, lisez le document suivant : ftp://ftp.win.tue.nl:/pub/security/tcp_wrapper.ps.Z.

5 - Recompiler le noyau

5.1 - Cyrix

Il existe des patchs non officiels pour le noyau 2.0.xx. Toutefois, le mieux est d'utiliser l'utilitaire set6x86 qui positionne certains registres du cyrix pour optimiser son fonctionnement. Le site incontournable pour tout ce qui concerne linux/cyrix : est http://wauug.erols.com/~balsa/linux/cyrix/index.html.

5.2 - Pentium

Linux gère sans aucun problème les machines basées sur Pentium. Pour optimiser un peu la compilation, il suffit de spécifier Pentium lors de la recompilation du noyau. Il existe toutefois

un gcc spécialisé pour Pentium, il suffit de consulter la page http://www.goof.com/pcg pour plus de renseignements.

5.3 - Compilation

Les nouveaux sources du noyau Linux peuvent être récupérés sur le site ftp.lip6.fr dans le répertoire /pub/linux/kernel/sources. Il s'agit d'un miroir du site ftp.kernel.org.
Installez donc les sources (normalement dans le répertoire /usr/src/linux). La compilation est très simple à réaliser mais vous devez suivre quelques points très scrupuleusement :
1. conserver à portée de la main une disquette permettant de démarrer le système et de monter la partition root, afin de pouvoir reprendre le contrôle en cas d'erreur et tester qu'elle permet bien de démarrer.
2. make config va demander la configuration que vous souhaitez. Certains points peuvent vous sembler assez obscurs. Dans ce cas, ne les changez pas. Vous pouvez lancer make menuconfig en mode texte pour avoir le programme en couleur ou make xconfig sous X à condition d'avoir installé Tcl/Tk. Remarque importante : il s'agit du noyau 2.0.26, donc certaines options ou certaines indications peuvent ne pas exister sur d'autres versions.
3. make dep se charge de créer les dépendances.
4. make clean va nettoyer toute l'arborescence.
5. Lance make zImage pour compiler votre noyau. Vous pouvez aller boire un bon café ou plusieurs suivant la machine (avec un 486 DX2/66 et 16 Mo, la compilation exige un peu plus de 40 minutes). Cela dépend surtout de la quantité de mémoire vive disponible.
6. si vous avez des modules, lancer leur compilation par : make modules puis les installer par make modules_install. Ils seront copiés dans le répertoire /lib/modules/2.0.26.
7. Quand c'est terminé, aller dans /usr/src/linux/arch/i386/boot. Le nouveau noyau flambant neuf est le fichier *zImage*, qui est déjà compacté. Le copier dans la racine : / en lui attribuant un nouveau nom (par exemple "vmlinuz2"). Modifier la configuration de LILO mais ne rien retirer : copier/coller en tête du fichier une section existante puis la modifier afin qu'elle concerne le nouveau noyau. Puis relancer lilo pour que le nouveau noyau soit pris en compte et reboote.
Remarque : si vous ne parvenez pas à modifier la configuration de lilo il faudra installer le nouveau fichier du noyau en lieu et place de l'ancien. Je vous conseille fortement de laisser l'ancien noyau dans un coin et surtout de pouvoir booter dessus, par exemple grâce à une disquette, afin de ne pas tout perdre.
Réamorcer et ça doit fonctionner.

5.4 - Modules Chargeables

5.4.1 - Structure du noyau

Le noyau d'un système Unix peut être représenté sous la forme d'un objet monolithique. Toutefois, un tel objet possède l'inconvénient d'être gros et statique. A chaque fois que l'on désire rajouter un nouveau périphérique, il est nécessaire de recompiler le noyau. De plus, si l'on n'utilise certains gestionnaires particuliers que rarement, on est obligé de l'avoir dans le noyau, ce qui a tendance à consommer de la mémoire.

5.4.2 - Qu'est-ce qu'un module chargeable ?

Les modules chargeables permettent de joindre l'utile à l'agréable en ayant un noyau le plus petit que possible, chargeant à la demande ce dont il a besoin, soit d'une manière manuelle par le super utilisateur de la machine, soit d'une manière automatique. De cette manière, le gain de ressources est non négligeable.
La première question que l'on peut se poser, c'est : " Pourquoi deux techniques de chargement ?"
La première technique est manuelle : il faut charger ou décharger les modules à la main. La deuxième est automatique, grâce à l'utilisation d'un démon spécialisé qui est l'esclave du noyau et qui charge et décharge les modules pour lui. En fait, la version 1.2 de Linux n'offrait que la possibilité d'un chargement manuel qui est limité au super-utilisateur de la machine et qui est assez lourd à manipuler. Au fil du développement de la version 2.0, un nouveau système implémenté par Bjorn Ekwall permit d'effectuer un chargement dynamique et automatique des modules.

5.4.3 - Compilation du noyau

Lors de la compilation du noyau, il est nécessaire de spécifier des options particulières pour activer l'utilisation des modules chargeables :

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

Voici le détail de ces trois options :
– CONFIG_MODULES : active le mécanisme de modules chargeab les. Sans cette option, ils ne fonctionneront pas sur votre système.
– CONFIG_MODVERSIONS : si vous laissez cette option à N (par défaut), vous devrez recompiler les modules à chaque mise à jour du noyau de la machine. Cela signifie que des modules compilés pour un noyau 2.0.n ne fonctionneront pas avec un module compilé pour 2.0.n+1. Si vous activez cette option, les modules pourront être lancés... Cette option est particulièrement intéressante dans le cas de modules externes au noyau livrés sous forme binaire (ils sont accompagnés d'un patch à appliquer sur les sources du noyau). Il est fortement déconseillé de l'utiliser dans le cadre de modules intégrés au noyau (il est plus simple de les recompiler et de les installer que d'aller regarder si les anciens modules sont compatibles avec les nouveaux).
– CONFIG_KERNELD : cette option active le chargement automatique des modules. Vous devez également activer les IPC Système V car le noyau et le démon communiquent via des files de messages.
Une fois configuré, il vous suffit de lancer la compilation ainsi que l'installation :

```
gandalf# make dep ; make clean
gandalf# make zImage
gandalf# make modules ; make modules_install
```

Une fois ces opérations effectuées, les modules se trouvent alors dans le répertoire /lib/modules/x.y.z où x.y.z correspond au numéro de version du noyau. Il ne nous reste plus qu'à voir le chargement.
Note importante : tous les outils de manipulation des modules se trouvent dans modules-2.0.0.tar.gz situé dans le répertoire v2.0. Ils sont installés par défaut avec les distributions standards, mais il est important d'utiliser la bonne version.

5.4.4 - Chargement : méthode manuelle

Le chargement manuel est basé sur trois commandes :
– insmod : insère un module dans le noyau ;
– rmmod : décharge un module, si plus aucun processus ne l'utilise ;
– lsmod : affiche la liste des modules chargés.
Leur utilisation oblige d'être en super-utilisateur. Voici un exemple d'utilisation :

```
gandalf# insmod nfs.o
gandalf# lsmod
Module:       #pages:  Used by:
nfs           12       4
gandalf# mount -t nfs /truc /mnt
gandalf# lsmod
Module:       #pages:  Used by:
nfs           12       5
gandalf# cd /mnt
...
gandalf# cd /
gandalf# umount /mnt
Module:       #pages:  Used by:
nfs           12       4
gandalf# ps axu | grep nfs
root     5535  0.0  0.0     0     0  q2 SW  17:15   0:00 (nfsiod)
root     5536  0.0  0.0     0     0  q2 SW  17:15   0:00 (nfsiod)
root     5537  0.0  0.0     0     0  q2 SW  17:15   0:00 (nfsiod)
root     5538  0.0  0.0     0     0  q2 SW  17:15   0:00 (nfsiod)
root     5557  0.0  0.4   864   300  q2 S   17:16   0:00 grep nfs
gandalf# kill -9 5535 5536 5537 5538
gandalf# lsmod
gandalf# rmmod nfs.o
```

Il est nécessaire de "tuer" les 4 démons nfsiod car ils sont lancés dès que NFS est activé. Comme vous pouvez le voir, ces opérations deviennent relativement pénibles. C'est pour cette raison que le système de chargement automatique a été créé.

5.4.5 - Chargement automatique : kerneld

5.4.5.1 - Introduction

Le système de chargement automatique de modules permet de réduire au minimum la taille de son noyau. Le principe de fonctionnement est particulièrement simple : un démon en mode

utilisateur est à l'écoute des ordres du noyau (via une file de messages de type IPC Système V). Lorsque un processus essaye d'accéder à une ressource système (via un appel système open, etc...), le noyau envoie l'ordre de chargement du module à kerneld. Une fois le message reçu, kerneld exécute modprobe pour charger les modules nécessaires.

5.4.5.2 - Conseils pour la compilation du noyau

Par contre, lors de la compilation du noyau, il est nécessaire d'y mettre au moins le support pour permettre l'amorçage de la machine et le montage de la racine de votre système de fichiers (par exemple, support IDE + ext2fs). Vous pouvez y mettre tout le reste en module (carte son, systèmes de fichiers, carte SCSI, etc).

5.4.5.3 - Mise en place

Cette partie de modification n'est valable que si la machine n'était pas pourvue de kerneld. Les nouvelles distributions effectuent une installation tout à fait correcte.
Pour réaliser la mise en place du système de chargement de modules, il est nécessaire d'effectuer certaines modifications au niveau de votre configuration. En effet, il est nécessaire lors de l'amorçage de la machine de lancer le démon kerneld et de réaliser une espèce de liste des dépendances des modules : certains modules ne peuvent être lancés avant que d'autres ne le soient. Dans un premier temps, il faut créer le fichier /etc/rc.d/rc.modules dans lequel, il faut mettre :

```
# Modules

#
# Creation d'un lien logique pour le noyau courant
#
#
/bin/rm -f /lib/modules/current
ln -sf /lib/modules/`uname -r` /lib/modules/current
#
# Creation des dependances
if [ ! -r /lib/modules/current/modules.dep ]
then
        echo "Creating module dependencies"
        /sbin/depmod -a
fi
#
# Chargement des modules d'amorçage...
#
if [ -x /sbin/kerneld ]
then
        if find /lib/modules/boot -type f -o type l > /dev/null 2>&1
        then
                echo "Loading boot-time modules"
                /sbin/modprobe -a -t boot \*
        fi
else
        echo "Loading modules"
```

```
        /sbin/modprobe -a \*
fi
#
# Si vous possédez d'autres types de démons kerneld à lancer...
#
if [ -x /sbin/kdsound ]
then
        echo "Starting sound daemon"
        /sbin/kdsound &
fi
```

Cela permet de générer la dépendance de vos modules à chaque fois que vous amorcez votre machine. Ensuite, dans le fichier /etc/rd.d/rc.S (peut dépendre de la distribution...), il convient d'ajouter :

```
# Start update.
/sbin/update &

# *** A AJOUTER ***
# Lancement de kerneld le plus tôt possible, de telle manière
# que les modules de systèmes de fichiers puissent être chargés
if [ -x /sbin/kerneld ]
then
        echo "kerneld running"
        /sbin/kerneld
fi
...

# Un peu de ménage
cat /dev/null > /var/adm/utmp

# Lancement du script des modules
if [ -f /etc/rc.d/rc.modules ]; then
        /etc/rc.d/rc.modules
fi
```

Une fois ces modifications effectuées et la machine réamorcée, tout doit être en place. Si kerneld permet de charger automatiquement les modules, il permet également de les décharger au bout d'un certain temps de non utilisation. Par défaut, si aucun processus n'accède au module pendant plus de 60 secondes, il est automatiquement déchargé. Il est possible de modifier cette valeur en rajoutant le paramètre delay=Nb_Secondes à kerneld où Nb_Secondes est le délai en nombre de secondes.

5.4.5.4 - Le fichier /etc/conf.modules

Il peut arriver qu'il soit nécessaire de configurer un dernier fichier : le fichier /etc/conf.modules. Ce fichier contient les chemins où se trouvent les modules devant être chargés et ensuite des alias pour les modules. Si vous n'avez pas ce fichier, vous pouvez le créer avec :

gandalf# /sbin/modprobe -c | grep -v '^path' >/etc/conf.modules

Il peut arriver que lors du premier amorçage vous obteniez ce message :
Cannot locate module for net-pf-3
Cannot locate module for net-pf-4
Cannot locate module for net-pf-5
Pas de panique ! Ce message n'est pas méchant et pour ne plus l'avoir, ajouter les lignes suivantes dans le fichier /etc/conf.modules :

alias net-pf-3 off
alias net-pf-4 off
alias net-pf-5 off

Il peut arriver que certains périphériques aient besoin de certains paramètres particulier. Consultez le document Kernel HowTo.

Au boot, des messages d'erreurs concernant les modules créés peuvent apparaître, par exemple :

Unresolved symbol in module /lib/module/2.0.30/ppa.o
insmod : <symbole> wrong verion or undefined
insmod : loading failed the modules symbol (from your 2.0.30à don't match your linux 2.0.30

C'est parceque les modules compilés présents dans /lib/modules/2.0.* sont ceux correspondant à l'ancien noyau et pas au nouveau que l'on vient de recompiler (si tant est qu'on ait modifié les modules à compiler lors de la compilation du nouveau noyau, par "make modules ; make modules_install")!
Après avoir déplacé /usr/src/linux/System.map vers /boot/System.map, la liste des fonctions définies pour les modules est donc adaptée au nouveau noyau mais les modules trouvables dans /lib/modules/2.0.30/ sont les anciens correspondant au noyau précédent. Ils risquent d'etre trop nombreux par rapport à la version précédente, si lors de la recompilation du nouveau noyau, on a demandé moins de modules que le nombre pre-installe).

On peut alors déplacer /lib/modules/2.0.30 en /lib/modules/2.0.30.old (par exemple) et refaire un 'make modules; make modules_install'. On aura un nouveau /lib/modules/2.0.30/ avec juste les modules déclarés pour le nouveau noyau, donc dans le System.map, et lors du test des dépendances de modules au boot, le noyau ne trouvera plus de fichiers .o sous modules/2.0.30/ avec des fonctions non déclarées.

6 - Configuration des autres services.

6.1 - Le support réseau sous Linux

6.1.1 - Installation ou changement de carte éthernet (3c509,3C590)

- Voir si la carte éthernet en question possède un driver sous Linux en lisant soit le Ethernet-Howto, soit en allant chercher dans /lib/modules/2.0.30/net/ si il y a un module objet.o correspondant à la carte à installer.... exemple 3c59x.o
Les sources peuvent etre trouvés dans /usr/src/linux/drivers/net ...on peut recompiler le module à partir de là aussi...
- Vérifier les modules réseau installés par le noyau en faisant :

```
$ /sbin/lsmod
Module:      #pages:  Used by:
nfs          12       10 (autoclean)
3c59x         3        1 (autoclean)
```

- Si le module chargé n'est pas le bon (ou pour mettre à jour une dernière version de pilote):
booter en single-user ,
enlever le module existant : rmmod <nom du module> (ex. rmmod 3c509)
tester et réinstaler dynamiquement le module, sans rebooter, : $ insmod 3c59x.o
si tout va bien , placer le module nouveau au bon endroit dans /lib/modules/2.0.30/net/
modifier le fichier /etc/conf.modules :

```
$ more /etc/conf.modules
alias eth0 3c59x
alias cdrom isp16
$ vi /etc/conf.modules
```

et modifier l'alias pour l'interface eth0 par le bon module driver
- Enfin, la bonne page pour ce thème est là : http://cesdis.gsfc.nasa.gov/linux/misc/

6.1.2 - Support des interfaces réseau

Si vous avez spécifié des paramètres corrects lors de l'installation de la machine, le support des interfaces réseau doit déjà fonctionner.
Si ce n'est pas le cas, il n'est pas trop tard pour le faire. Tout d'abord le noyau doit supporter les interfaces réseau. Ensuite vérifier que les interfaces sont bien supportées avec la commande ifconfig, si vous avez les interfaces loopback et Ethernet c'est ok.
Si vous avez des problèmes à faire reconnaître votre carte Ethernet essayez d'utiliser la fonction append dans le fichier /etc/lilo.conf
Les interfaces activées automatiquement au boot sont définies dans les fichiers du répertoire /etc/sysconfig/network-scripts/
Si vous ne voulez pas configurer les interfaces de façon automatique, vous pouvez bien sûr utiliser la commande ifconfig en faisant par exemple (cas d'une machine d'adresse IP 192.168.1.202) :

```
$ifconfig eth0 192.168.1.202
```

6.1.3 - Ajout des routes

Une fois vos interfaces configurées correctement, il faut maintenant ajouter les routes, c'est à dire les adresses IP des ordinateurs avec lesquels vous voulez vous connecter. La commande netstat -r affiche les routes déjà configurées, il faut au moins avoir la route vers loopback.
Pour les ajouter de nouvelles routes faire :

```
$route add 127.0.0.0
$route add 192.168.1.202
```

De la même manière, ajouter les routes vers les ordinateurs vers lesquels vous voulez avoir accès en spécifiant leurs adresses IP. Cependant, vous n'êtes pas obligé de configurer une route pour chaque ordinateur. En ajoutant une route vers le sous-réseau (par exemple 192.168.1.0) tous les ordinateurs connectés au brin local qui ont une adresse commençant par 192.168.1 seront accessibles.
Pour vérifier que le réseau marche bien, il faut maintenant tester grâce à un petit ping en spécifiant l'adresse d'un ordinateur de votre réseau.

6.1.4 - Spécifier le nom de la machine, puis des hôtes de votre réseau

Si vous voulez modifier le nom de votre ordinateur, il faudra modifier :

- la ligne HOSTNAME du fichier /etc/sysconfig/network
- le contenu du fichier /etc/HOSTNAME
- la ligne correspondante de votre fichier /etc/hosts
- et tous les autres endroits où vous l'avez spécifié dans un fichier de configuration...

Spécifier les adresses IP de chaque machine n'est évidemment pas très humain, on peut donc associer le nom des machines aux adresses IP, cela se fait dans le
fichier /etc/hosts

```
127.0.0.1       localhost
192.168.1.200    zeppelin
192.168.1.201    pantera
192.168.1.202    fresne
192.168.1.203    trixie
192.168.1.209    flupke
```

Ensuite toutes les manipulations qui concernent l'adresse IP pourront se faire en spécifiant le nom de la machine.

6.1.5 - Regrouper des machines par sous-réseau

À partir d'un certain nombre de machine, le fichier hosts peut prendre une taille plutôt conséquente, c'est pourquoi il est possible de regrouper les machines par leurs adresses IP dans un sous-réseau auquel on donnera alors un nom.
Les sous-réseaux sont définis dans le fichier /etc/networks comme suit :

```
zep-net    192.168.1.0
autre-net  198.163.8.0
```

Le sous-réseau zep-net regroupe en fait toutes les machines commençant par 192.168.1, c'est-à-dire les machines définies plus haut dans le fichier /etc/hosts.

Cela simplifie ainsi beaucoup l'ajout des routes ou les autres manipulations que l'on fera alors pour le sous-réseau entier plutôt que machine par machine.

6.2 - Paramétrage NFS

Le service NFS (Network File System) permet de monter des périphériques d'hôtes distants et de les utiliser comme si ils faisaient partie de sa propre machine. Ce système est pratique et efficace. Il a juste un inconvénient, c'est qu'il est relativement lent. Sa mise en place est des plus simple.

6.2.1 - Coté client

Sur la machine cliente (pc linux) il faut dire quels sont les partitions à monter
De façon automatique :

```
vi /etc/fstab (et faire les modifications nécessaires)
mount -av
```

Ou manuelle :

```
mount -t nfs nom_du_serv:/rep_exporte /rep_local
```

Il faut tout de même que le noyau supporte le système de fichier nfs, vérifier en faisant un cat /proc/filesystems, une ligne nodev nfs doit apparaître, si ça n'est pas le cas, vous devez recompiler votre noyau.

6.2.2 - Coté serveur

Sur les machines hôtes (celles qui ont le disque à monter) , il faut autoriser les machines extérieures à monter une partition. Pour cela, il faut éditer le fichier /etc/exports et rajouter le nom de la machine Linux autorisée à faire un montage nfs puis lancer exportfs -va pour mettre a jour.
Exemple :

```
/users      fresne(rw) pantera(rw) flupke(rw)
/           fresne(ro)
/usr/local
```

Évidement, il est souhaitable pour des raisons de sécurité de ne pas permettre à tout le monde de monter des fichiers en lecture-écriture(rw).
Vous pouvez maintenant lancer sur le serveur les démons NFS qui sont rpc.mountd et rpc.nfsd.

6.3 - Paramétrage NIS

Le service NYS permet de distribuer sur un certain nombre de machines des cartes. Ces cartes sont en fait les fichiers de configuration correspondant aux comptes utilisateurs, aux adresses des différentes machines...
Pratiquement, NYS permet en fait d'utiliser son compte utilisateur avec le même mot de passe et les mêmes paramètres sur n'importe quelle machine appartenant au réseau NYS. Pour cela, on exportera les comptes utilisateur par NFS, et on utilisera NYS pour partager les données et les paramètres sur tous les clients du réseau.
Une fois installé, tous les fichiers de configuration sont centralisés sur le serveur. Cela simplifie grandement les mises à jour, la maintenance, et les ordinateurs clients ont ainsi en local des fichiers minimum.

Les cartes sont créées à partir des fichiers du serveur :
/etc/passwd -> carte des mots de passe
/etc/group -> groupe des utilisateurs
/etc/hosts -> les machines du réseau
/etc/networks -> liste des sous-réseaux

6.3.1 - Coté client

6.3.1.1 - Avec les distributions récentes

Il y a encore quelque temps, il était nécessaire de lancer un démon sur le client NIS. Ce client s'appelait ypbind, mais il est maintenant devenu inutile avec la distribution RedHat. En effet les fonctions de NIS ont été intégrées directement dans les librairies, qui ont maintenant le nom de NYS.
La documentation n'est malheureusement pas très claire à ce sujet, et il est parfois difficile de trouver la bonne façon de configurer son client NYS car les différentes docs se mélangeaient allègrement les pieds entre les 2 implémentations.
Ce sont les fonctions NYS qui sont maintenant utilisés et dont nous allons parler.
Il faut tout d'abord configurer comme pour le serveur le nom de domaine. Vous pouvez mettre dans le fichier /etc/sysconfig/network la ligne :

```
NIS_DOMAIN=mon_domaine
```

mais comme le script ypserv n'est évidemment pas lancé, il faudra définir domainname dans un des scripts de lancement.Il faut donc mettre dans /etc/rc.d/rc.sysinit à la suite du hostname la ligne domainname {$NIS_DOMAIN}
Il faut ensuite créer le fichier /etc/yp.conf

```
domainname mon_domaine_nis
ypserver mon_serveur_yp
ypsever  mon_deuxieme_serveur
```

Il permet d'indiquer le nom du serveur yp que l'on utilisera. Il est tout à fait possible de spécifier plusieurs serveurs avec la ligne ypserver. Celui qui sera utilisé est celui qui répondra en premier. Cela sert essentiellement pour des machines qui sont souvent déplacées entre des réseaux différents.
Éditer également le fichier /etc/nsswitch.conf

```
passwd:     files nis
group:      files nis
hosts:      files nis dns
networks:   files nis
```

Ce fichier permet de déterminer par où commencera la recherche d'une information dans les différentes cartes.
À partir de là, les différentes fonctions de NYS doivent fonctionner. Pour le tester, il suffit d'essayer un :

```
$ypcat passwd
```

6.3.1.2 - Avec ypbind

Les étapes sont sensiblement les mêmes :
1. Dire a quel domaine NIS le PC client veut accéder par la commande domainname, si la commande ne retourne rien, on peut et doit assigner le nom du domaine NIS par la commande :

domainname <nomdudomaine_NIS>
2. Lancer /usr/sbin/rpc.portmap
3. Créer le répertoire /var/yp (il est en général déjà créé avec la distribution redhat)
4. Lancer le daemon ypbind (après avoir lu la man page de ypbind) et après avoir paramétré le fichier /etc/yp.conf de la manière suivante :
$ cat /etc/yp.conf
ypserver <adresse_IP_du_serveur_NIS_pour_le_domaine_considere>
(dans ce cas on force l'adresse du serveur)
ou
$ cat /etc/yp.conf
domain <nom_du_domaine_NIS> server <adresse_IP_du_serveur_NIS >
ou
$ cat /etc/yp.conf
domain <nom_du_domaine_NIS> broadcast
5. On peut vérifier le handshake entre ypbind (coté client) et ypserv (coté serveur) en lancant ypbind en mode debug
ypbind -debug
6. vérifier le fonctionnement de ypbind par
rpcinfo -p localhost
rpcinfo -u localhost ypbind
7. Modification du fichier /etc/passwd
Rajouter la ligne + :* :0 :0 : : : à la fin du fichier passwd ... pour indiquer que les noms de login doivent être obtenus depuis la map "passwd" du serveur NIS. Pour vérifier l'accès et la récupération de la map passwd du serveur NIS , lancer
ypcat passwd

6.3.2 - Coté serveur

Avant tout, il faut spécifier à quel domaine NIS appartient la machine. Cela se fait par la commande domainname.
$domainname mon_domaine
Ce domaine est totalement différent du domaine DNS. Vous pouvez donc donner un nom de domaine différent ou identique à votre domaine DNS. Pour éviter les confusions, il est parfois préférable de distinguer ces 2 noms.
Pour ne pas avoir à le faire à chaque fois que vous redémarrez votre machine il suffit d'ajouter dans le fichier /etc/sysconfig/network la ligne :
NIS_DOMAIN=mon_domaine
Le nom de domaine sera initialisé automatiquement lors du lancement de ypserv.
Aller ensuite dans le répertoire /var/yp et éditer le Makefile. Aller alors à la ligne all:, elle contient la liste des cartes possibles. Recopiez-là et commentez-là pour en garder une trace.
Laissez une ligne contenant simplement : all: passwd hosts group
Ce sont les trois cartes généralement suffisantes pour la majorité des utilisations, en tout cas elles suffisent largement pour tester. Il sera ensuite toujours temps de rajouter les autres cartes dont vous avez besoin, comme par exemple la carte de /etc/networks.
Ensuite il n'y a plus qu'à faire un make, les cartes vont alors être créés dans le répertoire /var/yp/mon_domaine.
Il faut également éditer le fichier ypserv.conf
sunos_kludge: no

```
tryresolve: no
dns: no
# Host          : Map          : Security :Passwd_mangle
#
192.168.1.      : passwd.byname   : port     : yes
192.168.1.      : passwd.byuid    : port     : yes
# Not everybody should see shadow password, not secure, since
# under MSDOG everybody is root and can access ports < 1024 !!! *
                : shadow.byname   : port     : yes
```

Enfin, il faut autoriser les clients à se connecter, pour cela il faut éditer le fichier hosts.allow si ce n'est pas encore fait pour qu'il ressemble à quelque chose comme cela :

```
ALL: 192.168.1.          ou bien
ALL: fresne, pantera
```

Il ne reste plus qu'à lancer le démon ypserv.
Le serveur NYS à proprement parler est prêt. Pour que les utilisateurs puissent se connecter de façon transparente sur les clients, il faut maintenant exporter le répertoire de leurs comptes par NFS.
Vous pouvez alors exporter les répertoires /home/un_utilisateur de chaque utilisateur par NFS et les monter sur le client, de préférence en lecture-écriture.
Toutefois, si vous avez un grand nombre d'utilisateurs avec NYS, il parait judicieux de les regrouper dans un seul et même répertoire, /nisusers par exemple que vous n'aurez plus qu'à exporter. Dans ce cas, il faudra également modifier le champ home_directory du fichier /etc/passwd, et le script adduser pour qu'il soit compatible.

6.3.3 - Le démon yppasswd

Lorsque l'on utilise NYS et les mots de passe distribués, la commande passwd sur un client risque de ne pas avoir le comportement attendu puisque qu'elle va éditer le fichier local /etc/passwd.
C'est donc le démon yppasswd du serveur qui doit se charger de cela. En fait lorsqu'un utilisateur voudra changer son mot de passe, il utilisera la commande yppasswd, qui ira modifier le fichier /etc/passwd du serveur NYS, et qui également mettra à jour les cartes, en faisant appel aux fonctions de notre bon démon.
Pour que l'utilisation de yppasswd soit transparente pour les utilisateurs, vous pouvez renommer le fichier /usr/bin/passwd en lpasswd par exemple, et ensuite faire un lien passwd vers yppasswd avec la commande :

```
ln -sf yppasswd passwd
```

Les utilisateurs pourront ainsi changer leur mot de passe sans se rendre compte qu'ils utilisent un compte NYS.

6.4 - Sendmail : envoi et lecture de mail

Ceci est le cas où le PC Linux n'a aucun rôle de serveur de messagerie. On est sur un réseau local éthernet, on a un serveur de messagerie avec un compte dessus, et on veut juste lire le mail présent sur le serveur à partir de son pc linux.
On souhaite :
1. Lors de la réception de courrier : renvoyer à un serveur de messagerie tout le courrier que le PC reçoit (macro DH de sendmail.cf)

2. Lors de l'émission de courrier depuis le PC, modifier le champ From : pour qu'apparaissent le nom du PC soitmasqué par le nom du serveur de messagerie (macro DM de sendmail.cf)

Pour cela :

éditer /etc/sendmail.cf

modifier la macro DM et DH

DM com.univ-mrs.fr

définit le nom de domaine réseau qui " masquera " la machine depuis laquelle on écrit un mail.

Exemple : si mon domaine est com.univ-mrs.fr et que j'écrit de la machine m1.com.univ-mrs.fr, a l'émission le champ From : du mail sera labellé avec com.univ-mrs.fr au lieu de m1. m1.com.univ-mrs.fr

DH com.univ-mrs.fr

définit le nom IP du serveur de messagerie pour le domaine en question... ainsi un mail adressé à compc32.com.univ-mrs.fr aboutira sur le serveur de messagerie du domaine "com.univ-mrs.fr", c'est à dire sur la machine "M" pointée par le MX record du DNS du domaine "com.univ-mrs.fr"

Pour lire le mail situé sur le serveur de messagerie, sur ce réseau éthernet local, le plus simple est de monter par NFS le répertoire de réception des messages à travers le réseau :

- dans /etc/fstab :

```
mailhost:/usr/mail      /usr/mail       nfs   soft,intr,rw
```

- ou à la main:

```
mount  mailhost:/usr/mail     /usr/mail
```

- ne pas oublier de positionner la variable MAIL dans ~/.bash_profile ou /etc/profile ou /etc/bashrc :

```
export MAIL=/usr/mail/$USER
```

- ne pas oublier de permettre au PC client de monter la partition de mail sur le serveur de messagerie, inscrire le nom du PC client dans le fichier /etc/exports et relancer /usr/etc/exportfs -av

Si on a pas la chance d'etre connecté directement à un réseau éthernet local, et qu'on veut accéder au mail d'un serveur de messagerie d'un fournisseur d'accès internet, à travers une liaison téléphonique , il faut lire : http://w1.neuronnexion.fr/~delepine/linux/sendmail.rtc.html

6.5 - Samba et les réseaux MS-Windows

Samba est une collections d'outils disponibles de façon libre sur les systèmes Unix qui permettent d'accéder à un réseau MS-Windows. Samba permet ainsi à une machine Linux de devenir serveur de fichier pour tout un réseau sous MS-Windows, ou encore d'accéder aux différentes ressources qu'un serveur MS-Windows aura partagé.

6.5.1 - Configuration de Samba sur la machine Linux

Si vous avez installé correctement samba, les 2 démons smbd et nmbd doivent êtres lancés automatiquement au boot de votre ordinateur. Si ça n'est pas le cas installez les packages.
Les fichiers de configurations par défaut de samba se sont grandement améliorés récemment, en tout cas ceux fournis dans la RedHat permettent d'accéder à des répertoires homes assez

facilement. Du moins, ils étaient suffisamment commentés pour permettre de servir de modèle à des modifications personnelles.
Exemple de fichier /etc/smb.conf :

```
[global]
   workgroup = mongroup
   comment = Mon serveur samba
   volume = RedHat

   printing = bsd
   printcap name = /etc/printcap
   load printers = yes

   log file = /var/log/samba-log.%m
   max log size = 50
   lock directory = /var/lock/samba
   locking = yes
   strict locking = no
   share modes = yes

   security = SHARE
   null passwords = yes
   socket options = TCP_NODELAY

; Permet au serveur Samba de devenir serveur du domaine
   os level = 33
   domain master = yes

; Facilite la gestion des noms longs
   preserve case = yes
   short preserve case = yes

; Chaque client accède à son répertoire
[homes]
   comment = Home Directories
   browseable = yes
   read only = no
   create mode = 0750

; Service d'impression pour partager les imprimantes
[printers]
   comment = All Printers
   path = /var/spool/samba
   browseable = no
   printable = yes
   public = no
   writable = no
   create mode = 0700
```

```
; Répertoire public /home/samba, accessible en lecture/écriture pour
; tout le monde
[public]
  comment = Repertoire Public
  path = /home/samba
  public = yes
  writable = yes
  browsable = yes
  printable = no
```

Une fois le fichier de configuration modifié, il faut relancer Samba avec :

```
/etc/rc.d/init.d/smb stop
/etc/rc.d/init.d/smb start
```

Enfin, si vous modifiez le fichier smb.conf, utilisez l'utilitaire testparm qui permet de vérifier la cohérence des informations de votre fichier. Il permet également d'afficher la totalité de la configuration de votre serveur. Ce qui permet de vérifier la validité de vos modifications.

6.5.2 - Configuration de la machine MS-Windows

Pour le moment, Samba ne permet d'accéder à la machine MS-Windows que si elle utilise une pile TCP/IP. En attendant la version qui devrait offrir un service direct avec NetBIOS. Il est donc nécessaire de configurer sous MS-Windows le support TCP/IP pour votre carte Ethernet, et d'affecter une adresse IP à votre machine. Ceci étant fait après quelques reboots, devrait apparaître dans votre Voisinage Réseau votre nouveau serveur Samba sous Linux.
Vous allez normalement alors pouvoir accéder au compte de l'utilisateur avec lequel vous êtes connectés sous MS-Windows, ainsi qu'au répertoire public /home/samba

6.5.3 - Les outils smbclient et smbmount

Smbclient et smbmount permettent d'accéder aux différentes ressources offertes par le serveur MS-Windows.

smbclient : le ftp pour MS-Windows. Il permet de tester, répertorier et accéder aux ressources offertes par le serveur MS-Windows.
Tout d'abord, il est possible de vérifier si le serveur MS-Windows est bien disponible, et de lister les différentes ressources qu'il partage grâce à la commande :

```
$smbclient -L nom_serveur_windows
```

nom_serveur_windows représentant le nom que vous avez donné à votre machine sous MS-Windows.
Ensuite, il est possible d'accéder au différents fichiers de la ressource partagée en s'y connectant avec smbclient, puis de la parcourir et de récupérer les fichier exactement comme avec un ftp :

```
$smbclient \\\\nom_serveur_windows\\nom_ressource
```

Avec la majorité des Shell, il est nécessaire de doubler les backslash (\) pour qu'ils soient pris en compte. nom_ressource représente le nom de la ressource partagé que l'on peut obtenir avec un smbclient -L
Après cela, il devrait vous être demandé un mot de passe (qui correspond à celui spécifié pour la ressource sous MS-Windows). Une fois le mot de passe entré, il n'y a plus qu'à parcourir les fichiers exactement comme avec ftp.

Smbmount : monter sous Linux des répertoires MS-Windows. Il permet de monter (comme par NFS) des répertoires MS-Windows sur l'arborescence Linux et de les manipuler le plus naturellement du monde.

```
$smbmount //nom_serveur_windows/nom_ressource /mnt -f 777
```

smbmount utilise les '/' à la différence de smbclient.
Il est nécessaire de rajouter une entrée dans le fichier /etc/hosts pour nom_serveur_windows avec son adresse IP, pour que smbmount puisse marcher correctement.
Enfin, il est nécessaire de passer quelques paramètres pour assurer la compatibilité des droits de MS-Windows avec les droits Unix. La clause -f 777 met tous les fichiers en lecture-écriture pour tout le monde.
On peut bien sûr la combiner avec l'option -u uid et -g gid qui permettrons de définir les droits précis des fichiers de la ressource pour pallier les carences de MS-Windows dans ce domaine.

6.6 - APACHE / httpd : un serveur Web

Le démon httpd fourni avec la Redhat est celui écrit par Apache. Celui-ci permet à votre machine de devenir un serveur Web (aussi appelé WWW).
Sa configuration est des plus simple. Tout d'abord, il faut installer apache sur votre machine si ça n'est pas encore fait. À vous de choisir votre solution préférée : un bon rpm en ligne ou la version graphique glint (qui ne présente pas toujours correctement les messages d'erreur éventuellement engendrés lors d'une installation avortée).

6.6.1 - Configuration d'Apache

Tous les fichiers de configuration de httpd se trouvent dans le répertoire /etc/httpd/conf/. Le fichier principal est le fichier httpd.conf. Il suffit de réactiver la ligne ServerName qui est normalement commentée, et de spécifier le nom auquel le serveur répondra, par exemple :

```
ServerName nom_de_ma_machine
```

Bien évidemment, ce nom doit être un nom valide de la machine, c'est-à-dire un nom auquel elle répondra soit directement, soit après une résolution par NYS ou DNS. Pour le moment, mettre tout simplement le nom habituel de la machine.
Il ne reste plus qu'à tester le bon fonctionnement du serveur Web, en lançant un browser quelconque à l'adresse http://nom_de_ma_machine/ ou http://localhost/ si vous êtes directement sur le serveur. Cela devrait afficher la page de présentation de apache. Celle-ci est installée dans le répertoire /home/httpd/ où vous pourrez bien entendu placer vos pages Web.

Les utilisateurs peuvent maintenant insérer leurs pages Web dans un répertoire public_html sur leur compte qui sera accessible par http://nom_de_ma_machine/~user_name/. Le nom de ce répertoire ainsi que les pages chargées par défaut (style index.html) sont paramètrés et modifiables dans le fichier srm.conf.

6.6.2 - Accès aux fichiers et gestion des droits d'accès

Avant tout, pour que le démon httpd puisse accéder aux fichiers, ceux-ci doivent avoir le droit de lecture pour tous. Pour l'appliquer à tous les fichiers d'un répertoire, nous lancerons ainsi la commande chmod a+r *. Par ailleurs, Apache fournit un système permettant de définir les

droits d'accès des différents répertoires. Les droits d'accès par défaut sont définis dans le fichier /etc/httpd/conf/access.conf.
Il est toutefois possible de préciser un droit d'accès pour chaque répertoire, ce qui peut être utile si l'on installe des pages qui doivent être accessibles uniquement de façon interne à votre entreprise/association/réseau, et non à tous les internautes.
Cela se fait dans un fichier appelé par défaut .htaccess qui contient par exemple :

```
order deny,allow
deny from all
allow from .votre_domaine.fr
```

Ce qui permet d'autoriser uniquement les utilisateurs de .votre_domaine.fr à accéder aux fichiers du répertoire où est placé ce fichier.

6.6.3 - Configuration et utilisation des scripts CGI

L'un des principaux intérêts d'un serveur Web est de permettre d'exécuter des programmes sur le serveur. Ces programmes sont généralement dénommés des scripts cgi.
Apache permet bien évidemment d'exécuter de tels scripts. Par défaut, ces scripts sont autorisés à être exécuté uniquement si ils sont stockés dans le répertoire : /home/httpd/cgi-bin.
Mettons donc dans ce répertoire un petit script perl test.pl :

```
#! /usr/bin/perl
print "Content-type: text/html\n\n";
print "Tout baigne\n";
```

Et à l'affichage par votre browser préféré de http://nom_de_ma_machine/cgi-bin/test.pl vous lirez le message "Tout baigne".
Vous pouvez également autoriser le lancement de scripts CGI dans d'autres répertoires ajoutant l'option AddHandler cgi-script.cgi dans votre fichier srm.conf. Cependant, il est préférable de limiter au cas par cas cette possibilité pour des raisons de sécurité grâce au fichier access.conf.
La configuration fournie ici est suffisante pour la majorité des utilisations, mais les fichiers de configuration sont suffisamment bien commentés pour ne pas mériter de plus amples détails.