

Glossaire

par Jean-Christophe, Fred et d'autres contributeurs

Trouvez ici les définitions des mots que vous ne connaissez pas.

Vous ne savez pas ce qu'est une compilation ? A quoi sert un module ? Ce qu'est un serveur X ? Cette rubrique est faite pour vous ! Et si vous êtes vraiment perdus, allez ici. :-)

Vous trouverez ici les termes souvent utilisés sous Linux et qui peuvent être inconnus du débutant. Je suppose que vous vous y connaissez un minimum en informatique : je ne vais pas vous expliquer ici ce qu'est un microprocesseur, une carte ethernet et autres joyeusetés non spécifiques à Linux/Unix.

Avertissement : ce glossaire est loin d'être complet. S'il vous manque un terme, cliquez sur le bouton ci-dessous !

bibliothèque

Une bibliothèque est un ensemble de fonctions regroupées entre-elles. Les bibliothèques permettent de partager des fonctions entre différents programmes ; ces programmes n'ont donc pas besoin d'inclure directement ces fonctions dans leur code. Par exemple, les bibliothèques nommées *.so de Linux sont des ensembles de fonctions partagées par les programmes et chargées une seule fois en mémoire (ce qui implique des économies de mémoire !). La fameuse bibliothèque `libc` contient toutes les routines de base du langage C, autrement dit du système puisque quasiment tous les programmes l'utilisent.

Par mauvaise traduction (c'est un faux-ami ;), on parle souvent de *librairie* (en anglais, une bibliothèque se dit *library*).

binaire

Le mot binaire peut recouvrir un ensemble de choses très variées. La signification première de ce mot est un système de numération n'utilisant que 2 chiffres (0 et 1) au lieu des 10 (0, 1, 2, 3, ... 9) qui nous sont habituellement nécessaires. C'est avec ce système de numération que l'ordinateur conserve ses informations. Par extension, on qualifiera de binaire toute information qui n'est compréhensible que par un ordinateur.

On dit qu'un programme est un binaire lorsqu'il a été compilé. Il est alors incompréhensible pour nous, mais exécutable par un ordinateur.

booter

On dit qu'un ordinateur boote, ou plutôt démarre en bon français, lorsqu'il charge le BIOS puis le système d'exploitation.

code source

C'est un fichier texte, éditable dans n'importe quel éditeur de texte (emacs, vi, etc.), et qui dans un certain langage, décrit un programme. Le noyau de Linux est disponible sous forme de source, ainsi que tous les logiciels dits "libres", ce qui permet à tout un chacun de modifier et d'améliorer ces logiciels, en diffusant les améliorations qui ont été apportées.

compiler

Compiler, c'est transformer un code source en fichier binaire, directement exécutable par le noyau de Linux. Le compilateur est le programme qui réalise cette opération. La "compilation croisée" (*cross compilation* en anglais) est le fait de compiler pour une architecture différente (i.e. microprocesseur différent par exemple) que celle de la machine qui compile.

Compiler c'est aussi regrouper par exemple des morceaux de musique, des mp3/ogg sur un même CD, etc.

console

Un shell, autrement nommé xterm, console, terminal, etc. est une fenêtre dans laquelle vous disposez d'une invite de commandes (ou *prompt*) et où vous pouvez taper... des commandes. Cela ressemble au mode DOS des systèmes d'exploitation Microsoft, mais en bien plus puissant. (Voir l'article sur le [shell](#)).

Le shell est constitué d'un programme, qui attend vos commandes, et les interprète et exécute. Plusieurs shells existent : bash, tcsh, ksh, ash, sh, csh... le plus utilisé étant le bash.

On parle également de consoles virtuelles : dans une installation Linux classique, vous disposez en général de 12 consoles virtuelles, accessibles par la combinaison de touches [Control]-[Alt]-[Fn] où [Fn] est l'une des 12 touches de fonction. Les 6 premières consoles (F1 à F6) sont en général des consoles texte, sur lesquelles vous pouvez vous connecter, et disposer alors d'un shell ; la console n°7 (F7) correspond en général à la console graphique X Window. Les autres consoles virtuelles peuvent être utilisées pour d'autres besoins, comme afficher des messages système, pour le démarrage/arrêt du système, etc.

déboguage

phase dans le développement d'une application où l'on va chercher à résoudre les problèmes d'exécution, c'est en fait une phase de résolution des problèmes d'un programme.

distribution

Une distribution (distro ou distrib en langage courant) c'est l'ensemble de Linux (qui à proprement parler n'est que le noyau du système d'exploitation), et de nombreux programmes qui forment un système d'exploitation complet, de la ligne de commande à l'environnement graphique. Les distributions les plus connues sont Red Hat, Mandrake, SuSE, Debian, Slackware, etc.

driver

Un 'driver' ou 'pilote' est un logiciel particulier qui ne sert qu'à une seule chose : permettre à votre système de 'communiquer' (ie: échanger des informations) avec un périphérique particulier. Avec Linux il existe deux types de drivers :

◇ les 'drivers' modules que l'on doit très souvent recompilier avec le noyau.

◇ les 'drivers' logiciels comme par exemple le driver de votre imprimante (c'est souvent ghostscript)

éditeur de texte

Un éditeur de texte est un logiciel vous permettant de visualiser, modifier, enregistrer tout fichier au format texte. Par exemple `vi`, `emacs`, `xedit`, `kedit`, `gnotepad` sont de tels programmes.

Remarque : la plupart des fichiers de configuration sous Linux sont des fichiers texte. Avouez que ça simplifie la vie !

émulateur

Un émulateur est un logiciel qui permet de faire tourner un programme sur une autre plate-forme que celle pour laquelle il a été prévu. Par exemple, avec les émulateurs adéquats, vous pouvez faire tourner des programmes d'Amiga, C64, TO7... sur un PC. Des émulateurs existent qui vous permettent de faire tourner presque n'importe quoi sur n'importe quelle plate-forme (voir par exemple ce site). FreeMWare (maintenant Plex86), clone du produit commercial VMWare, est un émulateur un peu particulier qui émule un PC sur un PC. Cela permet par exemple de faire tourner MS Windows sous Linux. Sur Macintosh, VirtualPC (commercial), permet lui aussi de faire tourner MS Windows sous MacOS. Mac On Linux (MOL) permet de lancer des applications MacOS depuis Linux sur un Apple. Bochs est un émulateur sous Linux permettant de faire tourner du code x86 sur d'autres architectures (PowerPC, etc.). Enfin, Wine, dont l'acronyme signifie "Wine Is Not an Emulator", est particulier, car il n'émule pas une machine comme le fait Plex86, mais est une réimplémentation des appels système de Windows sous Linux. C'est un émulateur en ce sens qu'il permet bien de faire tourner des programmes Windows sous Linux, mais ce n'est pas que ça, car il fournit aussi une librairie, la `wine1ib`, qui facilite le portage d'applications Windows sous Linux. Certains émulateurs utilisent aussi cette approche, comme UltraHLE (émulateur de la console Nintendo 64).

environnement de bureau

La frontière entre gestionnaire de fenêtres (ou window manager) et environnement de bureau (ou desktop environment) n'est pas toujours très précise.

Les gestionnaires de fenêtres sont des environnements simples, alors que les DE comme KDE, Gnome et récemment Xfce sont à l'autre bout de l'échelle de la taille et de la complexité. Ils peuvent fournir leur propre display manager et window manager, et peuvent parfois utiliser un WM alternatif (comme Gnome, qui livré avec Enlightenment s'accommode très bien de le remplacer par WindowMaker). Les DE fournissent des tas de logiciels qui s'intègrent bien à l'environnement, au premier rang desquels les gestionnaires de fichiers, les panneaux de configuration, les navigateurs internet, les éditeurs de texte et d'images, les jeux (ah ! Une petite partie de démineur sous KDE, ça me manquait :), et même des suites bureautiques, comme KOffice de KDE.

En bref, les DE essaient de vous donner au moins la même chose que Windows lorsque vous venez de l'installer (et beaucoup plus puisqu'il existe des tas d'applications pour chaque DE).

firewall

Un firewall (mur pare feu) est un équipement placé en entrée d'un réseau permettant de le protéger (lorsqu'il est bien configuré ;) d'intrusions extérieures. Pour des configurations 'domestiques' le firewall est souvent un logiciel tournant sur le même ordinateur que celui qui sert à se connecter à internet.

firmware

Le firmware est le programme qui sert à faire le lien entre un périphérique évolué (disque dur, graveur, scanner, modem adsl) et votre ordinateur. C'est un programme écrit dans le langage que comprend l'électronique du périphérique. On peut comparer le firmware d'un périphérique au BIOS d'un ordinateur. C'est lui qui se charge de l'initialisation du périphérique et qui reçoit les ordres des programmes plus évolués.

front-end

Un front-end est une interface graphique permettant d'utiliser "à la souris" des programmes en mode console (ou *back-end*) : le front-end définit, d'après les informations que vous avez entrées dans l'interface, la ligne de commande à lancer afin d'effectuer l'action voulue.

Quelques exemples : `gcombust` est un front end à `mkisofs/cdrecord` pour graver des CD ; `grip` est un frontend permettant d'encoder des CD audio en mp3 ou ogg, etc.

gestionnaire d'écran

Lorsque vous lancez votre ordinateur sous Linux, si vous avez choisi le mode de démarrage graphique, vous arrivez sous X sur un écran vous permettant de saisir votre username et votre mot de passe. C'est le gestionnaire d'écran (en anglais display manager, ou DM). Le DM de base s'appelle XDM ou X Display Manager. Il est livré avec le serveur X. KDE ou Gnome proposent leurs propres DM, respectivement KDM et GDM, qui ont le même look que l'environnement d'où ils sont tirés. On l'appelle aussi parfois "bannière de login" ou "bannière de connexion".

Lorsque vous vous connectez, le DM "disparaît" pour laisser la place à un gestionnaire de fenêtre (`fvwm2`, etc.) ou à un environnement de bureau (Gnome, KDE...). Lorsque vous quitterez ce client X, le système redémarrera le DM.

gestionnaire de fenêtres

La frontière entre gestionnaire de fenêtres (ou window manager) et environnement de bureau (ou desktop environment) n'est pas toujours très précise. En général, le gestionnaire de fenêtres est le client X qui possède les fenêtres et les gère (barre de titre, encadrement, déplacement, changement de taille, etc.). Les applications, quant à elles, possèdent l'intérieur de la fenêtre ; c'est elles qui créent les fenêtres, en gèrent le contenu ainsi que l'interaction avec ce contenu.

Il est ainsi possible de changer de WM, et de voir toute l'apparence de tout l'écran changer complètement. Quelques gestionnaires de fenêtres : KWM le WM de KDE, Enlightenment, WindowMaker, `fvwm` et toutes ses déclinaisons, `twm` l'ancêtre, `olwm`, etc.

IDS

Un détecteur d'intrusion est un programme qui suit le trafic de votre réseau à la recherche de tentatives d'intrusions. Il peut prévenir l'administrateur du système dès qu'une telle intrusion a lieu. On peut ou pas le coupler à un mur pare-feu.

langage

En informatique, le langage c'est la façon dont vous allez parler à la machine pour qu'elle fasse ce que vous souhaitez. Les langages de bas niveau sont le binaire (directement compris par le microprocesseur, mais qui peut vous conduire à l'hôpital psychiatrique), et l'assembleur, qui se compose de mnémoniques qui se traduisent directement en binaire. Sous Linux, la plupart des logiciels, et le noyau lui-même sont programmés dans un langage de plus haut niveau très populaire aujourd'hui : le langage C. L'évolution du C vers l'objet (manière différente de représenter les choses, où données et programme ne font plus qu'un) est le C++, utilisé pour programmer KDE. D'autres langages utilisés aujourd'hui sont Java, Pascal, Fortran, Cobol. Ces langages, ainsi que le C, doivent passer par une phase de compilation avant d'être compris par la machine.

Il existe également des langages qui n'ont pas besoin de compilation, car ils sont en réalité composés d'instructions lues et interprétées par un logiciel. Ce sont les langages interprétés ou langages de scripts. On compte parmi eux le célèbre PHP, Tcl/Tk, et les langages des shells (sh, csh, etc.).

Il existe un troisième type de langage : ceux qui compilent le code juste avant de l'exécuter. Ces langages combinent en parti le meilleur des deux mondes : rapidité et souplesse. Perl et Java peuvent fonctionner de cette manière.

libre

Un logiciel libre est distribué avec son code source, que vous pouvez modifier et/ou redistribuer selon les termes d'une licence comme la Gnu Public License (GPL), ou une licence de type BSD. C'est le principe de développement de Linux et de la majorité (sinon toutes) des applications livrées dans les distributions de Linux.

A noter que libre ne signifie pas forcément gratuit, et vice versa.

Pour plus d'informations sur les concepts du logiciel libre, lisez [cet article](#).

lien symbolique

1. Le lien symbolique est un concept Unix. Un fichier ou un répertoire se situe dans un et un seul répertoire. Grâce au lien symbolique, vous pouvez donner le don d'ubiquité à un fichier ou un répertoire. Le fichier/répertoire peut ainsi apparaître à différents endroits dans l'arborescence. Attention, ce n'est pas une copie, mais bel et bien un même fichier qui apparaît à plusieurs endroits. On peut comparer le lien symbolique à un *synonyme*.

Par exemple, le lien `/usr/include/X11` pointe vers `./X11R6/include/X11`, ce qui signifie que lorsque vous vous rendez dans le "répertoire" `/usr/include/X11`, vous êtes en fait dans `/usr/X11R6/include/X11`. De même, le lien `/dev/mouse` pointe vers le "vrai" périphérique de votre souris, par exemple `/dev/ttyS0`.

Pour créer un lien symbolique, utilisez la commande `ln -s`. Par exemple, `ln -s /dev/ttyS1 /dev/modem` vous permettra d'utiliser le périphérique `/dev/modem` dans vos applications et de ne plus vous soucier de savoir sur quel port série il est connecté.

2. Autre signification : on parle aussi de "lien hypertexte" (ou lien tout court) pour désigner une zone active sur une page web, qui renvoie sur une autre page.

login

Le login est l'action de se connecter (en anglais : to log in) à un système, le logout celle de se déconnecter.

modem

Un modem est un périphérique particulier dont le l'objet est de transformer (on devrait dire moduler) le signal numérique en provenance de l'ordinateur auquel il se trouve branché, en un signal analogique capable de se déplacer sur une ligne téléphonique. Il fait aussi l'opération inverse (on dit démoduler). En utilisant un tel périphérique deux ordinateurs distants de plusieurs milliers de kilomètres peuvent communiquer entre eux.

On distingue actuellement trois catégories principales de modems : ceux qui permettent de se connecter via RTC (téléphone classique), via ISDN (ou Numéris) ou via ADSL.

Remarque : vous entendrez peut-être parler de winmodems ou linmodems. Ce sont des modems particuliers nécessitant un driver spécifique pour fonctionner (voir l'entrée winmodem du glossaire).

noyau

Le noyau (ou kernel) de Linux est le logiciel de base qui permet de démarrer l'ordinateur, de gérer les périphériques, directement ou par l'intermédiaire de modules, de gérer les tâches et de leur partager le temps machine afin de réaliser ce qu'on appelle un fonctionnement *multitâche*, ou plusieurs applications s'exécutent simultanément. Le noyau comprend toute la couche réseau de Linux, les interfaces POSIX (norme unifiant les appels système, permettant ainsi à des programmes d'être facilement portés sur d'autres systèmes d'exploitation). Le noyau sait gérer de nombreux systèmes de fichiers, comprend de nombreux drivers, et bien d'autres choses encore.

Le noyau est disponible sous forme de source, et on peut le compiler après l'avoir configuré pour répondre à ses besoins.

Étymologiquement parlant, "Linux" est juste le nom du noyau du système d'exploitation, mais on a tendance par abus de langage à utiliser "Linux" pour désigner l'ensemble de Linux, du serveur X et des nombreuses applications. La majorité des applications de base livrées avec Linux font partie du projet [GNU](#) (acronyme récursif signifiant GNU is Not Unix). On devrait donc parler de GNU/Linux.

page de man

man est le manuel de Linux (et des Unixes en général). Il se consulte en mode texte, par la commande `man` suivie du nom de la commande sur laquelle on veut avoir de l'aide. Par exemple, pour savoir comment fonctionne le `man`, "`man man`" est la commande appropriée. "`man ps`" vous donnera toutes les informations sur le fonctionnement de `ps`, etc. Vous pouvez aussi consulter le man en mode graphique, avec `xman`, ou dans les environnements de bureau comme KDE en tapant l'URL `man:la_commande` (ex : `man:ps`).

paquet

Un paquet (package en anglais) est un fichier qui contient un programme à installer ainsi que les informations nécessaires à son fonctionnement, son installation, sa désinstallation éventuelle. Il existe plusieurs types de paquets : ceux de la distribution **Debian** sont terminés par `.deb`, ceux de la **Red Hat**, de la **Mandrake** ou de la **SuSE** sont terminés par `.rpm`, ceux de la **Slackware** par `.tgz`. Certains logiciels sont fournis sous la forme de paquets source : ils se terminent soit par `.src.rpm`, soit par `.tar.gz`, il vous faudra compiler de tels paquets avant de pouvoir les utiliser.

partition

Pour utiliser certains media comme les disques durs, on a besoin de les structurer, de les découper en gros sous-ensembles : les partitions.

Chaque partition est ensuite elle-même structurée par un système de fichiers. Découper un disque dur en plusieurs partitions permet par exemple d'y faire coexister deux systèmes d'exploitation différents, comme Windows et Linux, qui utilisent des systèmes de fichiers différents (voir la [rubrique installation](#) pour le partitionnement pour Linux).

path

Le *path* ou "chemin de recherche" est l'ensemble des répertoires dans lesquels va rechercher un exécutable (`/bin`, `/usr/bin`, etc.). Lorsque vous lancez un programme (nommé par exemple `test`), le système va parcourir ces répertoires, jusqu'à trouver `/usr/bin/test` par exemple, puis lire le fichier et l'exécuter. Le *path* se matérialise sous la forme d'une variable d'environnement nommée `$PATH` contenant la liste des différents répertoires, séparés par `:`. Par exemple : `$PATH=/bin:/usr/bin`.

périphérique

Un périphérique est un matériel qui doit communiquer avec votre ordinateur, par exemple : une imprimante, un modem, un scanner, une souris, un clavier sont des périphériques.

point de montage

Sous Unix, les systèmes de fichiers, qu'ils se trouvent sur une partition d'un disque dur, une disquette, un CD-ROM, un ZIP... sont attachés à des répertoires particuliers de l'arborescence appelés "points de montage". C'est ce qu'on appelle "*monter* une partition, un disque, un CD-ROM, etc.". Le point de montage est un simple répertoire, vide avant le montage, et qui après le montage, représente le contenu de la partition qui y est montée. On n'accède pas à la disquette via une "lettre" désignant un "lecteur", comme le `A:` de DOS, mais à travers un répertoire, comme `/floppy`.

Par exemple, supposons que nous ayons créé une partition `/dev/hda2` (premier disque IDE, 2^{ème} partition) pour contenir le système. On *montera* cette partition (automatiquement, heureusement !) dans le répertoire racine `/`. Si la partition `/dev/hda3` est destinée à contenir les données utilisateur, on la montera dans le point de montage (=répertoire) `/home`. On verra le contenu de cette partition dans le répertoire `/home` comme si c'était n'importe quel autre répertoire, alors que physiquement les données sont sur une autre partition voire un autre disque.

De même, pour accéder à la disquette, on montera le périphérique `/dev/fd0` dans le point de montage `/mnt/floppy` ou `/floppy` (selon la distribution) et pour accéder au CD-ROM, on montera le périphérique `/dev/cdrom` dans le point de montage `/mnt/cdrom` ou `/cdrom`.

process id

Linux est un système d'exploitation multitâches, ce qui signifie que plusieurs programmes (qui peuvent être à la fois des applications utilisateur ou des tâches système) peuvent tourner en même temps. Les PID sont des numéros entiers assignés par le noyau qui permettent d'identifier et de suivre et de communiquer avec ces programmes. Chaque PID est unique.

Afin de trouver le PID d'un programme, vous pouvez utiliser des applications graphiques, comme **ktop** ou **kpm** sous KDE. En mode texte, la commande **ps** vous permettra de lister les informations sur les processus, dont le PID. La commande **top** présente une vue permettant de classer les processus, par exemple par consommation CPU décroissante. Consultez les pages de man de **ps** et de **top**.

processus

Tout programme tournant sous Linux est un processus, qu'il s'agisse d'une application utilisateur ou de tâches système.

programme

Un logiciel (ou programme, ou application...) est un ensemble d'instructions élémentaires (afficher quelque chose à l'écran, faire une opération mathématique – addition par exemple, imprimer un mot, afficher un dessin, etc.) réunies de manière relativement complexe (on parle d'algorithme) capable de faire des choses complexes (mettre en page un document, vous permettre de tuer des petits hommes verts virtuels, etc.).

root

L'utilisateur **root** est le super utilisateur, l'administrateur d'un système Linux (et Unix en général).

Habituellement, on ne travaille sous cet utilisateur que pour les tâches d'administration de la machine. Il est recommandé de créer au moins un utilisateur normal, sous lequel on fera tout le reste (bureautique, internet, jeux, etc.).

Pour se loguer en **root**, la façon recommandée, pour des questions de sécurité, est de se loguer d'abord sous un 'user' normal, puis dans une console texte, de taper la commande `'su -'`, puis le mot de passe du **root**. On obtient ainsi un 'prompt' appartenant au **root**.

script

Un script est un (souvent petit) programme écrit dans un langage interprété. A la différence des programmes écrits en C ou en C++ (ou autre langage compilé), les scripts sont traduits en langage machine au fur et à mesure de leur exécution. Ils sont en général plus lents à exécuter, mais aussi plus faciles à écrire.

Les langages utilisés pour écrire un script sont (liste non exhaustive) : `bash`, `perl`, `php`, `python`...

Un script s'exécute comme tous les programmes en lui donnant les droits d'exécution et on l'appelle par son nom, ou en lançant l'interpréteur avec le script en paramètre.

serveur de fonte

C'est un programme à qui X s'adresse pour calculer l'image des fontes demandée par les clients X (les applications tournant sous X Window). Le serveur de fontes, comme les programmes sous X, bénéficie d'une interface réseau : sur votre réseau, il suffit de faire tourner un seul serveur de fontes sur un seul ordinateur, les autres pouvant s'en servir comme point central pour récupérer les fontes.

serveur de noms

Le DNS (Domain Name Server ou Serveur de noms de domaine) est un serveur qui offre le service permettant d'associer une adresse IP (adresse d'un ordinateur, numérique de la forme 123.456.144.12) à un nom plus facilement mémorisable, comme `lea-linux.org`.

swap

Le *swap* est la mémoire virtuelle de Linux : lorsque toute la mémoire vive (RAM) est remplie, Linux, comme tout système d'exploitation, simule de la mémoire vive sur le disque dur : on dit qu'il *swappe* car il échange (*to swap* en anglais) des informations entre la mémoire vive et le disque dur.

système d'exploitation

Un système d'exploitation (en anglais *Operating System* ou *OS*) est une interface entre le matériel et les logiciels applicatifs. Par extension, on considère souvent que c'est aussi ensemble de logiciels permettant la communisation entre un ordinateur et les utilisateurs de cet ordinateur (cette définition a été introduite par Windows qui comporte toutes les couches, jusqu'à la couche applicative). Il doit permettre un certain nombre d'opérations de base (comme : fournir une 'abstraction' du concept de fichier, permettre de créer des fichiers, de les copier, de les effacer, etc., permettre de lancer un logiciel, de le terminer pour passer à autre chose...). Avec l'arrivée d'ordinateurs de plus en plus puissants, les systèmes d'exploitation comme Windows sont devenus des usines à gaz capable de remplir les fonctions les plus hétéroclites (comme afficher des fenêtres ;-). Mais ce faisant ils ne remplissent plus très bien leurs fonctions de bases de manière satisfaisante (sans en occasionner leur perte par exemple), et cela nuit à leur stabilité. Linux est un système d'exploitation relativement basique (capable de multi-tâche quand même) et très robuste. Il n'accomplit que des fonctions basiques mais il le fait bien et laisse les tâches plus complexes à d'autres logiciels (X Window pour l'environnement de bureau, Apache pour le serveur Web etc.). La division des tâches rendant la recherche et l'élimination des problèmes beaucoup plus simples.

système de fichier

C'est la façon dont les données sont organisées sur une partition, une disquette, un ZIP, etc. Linux possède ses propres FS, comme **ext2**, **ext3**, **reiser fs**, **minix**... et il gère une multitude de systèmes de fichiers provenant d'autres architectures, dont FAT, VFAT (i.e. FAT32) (les systèmes de fichiers de DOS/Windows), Amiga Fast File System, OS/2 HPFS, ISO 9660 (CDROM) avec les extensions "noms longs" Joliet de Microsoft et Rock Ridge, NTFS (Windows NT, uniquement en lecture à l'heure actuelle), NFS (file system réseau), SMBFS (disques Windows partagés), etc.

Un système de fichiers est dit « journalisé » lorsqu'il conserve sur le disque la trace (le journal) des opérations à réaliser, et les écrit ensuite sur le disque. De tels systèmes de fichiers (ex : ext3 ou reiserfs mais aussi ntfs) sont plus tolérants aux crashes du système, car ils conservent ainsi l'intégrité du système de fichiers (plus besoin de fsck ou de scandisk lors du redémarrage). Certains de ces systèmes de fichiers sont aussi capables de conserver l'intégrité des fichiers eux-même (support des « transactions »), au prix d'une dégradation des performances (chaque écriture doit se faire plusieurs fois).

Unix

Unix est une famille de systèmes d'exploitation, dont Linux s'inspire et reprend la philosophie. D'autres Unixes sont : BSD et ses déclinaisons, Digital Unix, AIX d'IBM, Solaris de Sun, HP/UX de Hewlett Packard, etc. Les Unix sont faits de telle manière que les programmes sont censés être "portables" entre deux unixes différents, c'est à dire qu'il suffit de compiler le code source d'un programme pour le faire tourner sur n'importe quelle architecture utilisant un système d'exploitation Unix. Dans la réalité, chaque constructeur a ajouté ses propres extensions (tiens comme c'est bizarre :) et il faut toujours adapter les logiciels. Cependant, grâce à la norme POSIX – que respecte Linux, les adaptations sont aujourd'hui relativement mineures pour porter un logiciel d'un Unix conforme à POSIX à un autre.

winmodem

On appelle "winmodems" les modems qui ont besoin d'un driver spécifique fonctionnant sous Windows afin de fonctionner. Ces modems n'implémentent pas toute la logique de communication dans le modem, c'est le micro-processeur de l'ordinateur qui fait le travail via le driver spécifique. Ils se présentent en général sous la forme de modems internes au format carte PCI. Ces modems ne sont généralement pas supportés sous Linux car les fabricants refusent souvent de donner les spécifications du matériel, voire empêchent le développement de tels drivers par des programmeurs tiers. Cependant, des efforts ont récemment été faits afin de faire fonctionner de tels modems sous Linux : voir le site www.linmodems.org.

Remarque : le terme "winmodem" est une marque déposée de USRobotics (3Com). On utilise ce terme par abus de langage pour désigner tous les modems fonctionnant sur le même principe, décrit ci-dessus. On parle également de "linmodem" par analogie avec **Linux**.

X Window

C'est le sous-système graphique de Linux. X Window n'est pas seulement un driver pour la carte vidéo, c'est aussi une interface (API) pour les applications, afin qu'elles affichent à l'écran et reçoivent l'entrée du clavier et de la souris.

X est aussi un serveur réseau, c'est à dire qu'il peut également offrir ses services à travers un réseau, ce qui permet d'afficher sur un écran une application qui s'exécute sur une autre machine, même si les deux architectures sont complètement différentes. C'est pourquoi on parle de *serveur X* pour désigner le sous-système graphique. Le système X Window tourne sur quasiment tous les Unix, et a même été porté sous Windows ou OS/2. Quasiment tous les logiciels graphiques sous Linux utilisent X, vous pouvez donc les faire fonctionner à travers un réseau comme décrit ci-dessus.

L'utilisateur n'interagit pas directement avec X, mais plutôt avec ce qu'on appelle des clients X (par opposition au serveur X). Vous utilisez sans doute des clients comme un Gestionnaire de fenêtres (window manager), ou un Environnement de bureau (desktop environment) comme KDE ou Gnome. Pour vous connecter, peut-être utilisez vous aussi un Display manager, comme KDM ou XDM. Au dessus de ces clients se trouvent les applications.

Le système X Window (ou X Window system, ou X Window, ou encore X) est une marque déposée du X Consortium. Les serveurs X libres distribués avec Linux proviennent du projet XFree86.

Remarque :

En comparaison à Windows, Macintosh and so on... Dans Windows, tout est compris dans la même boîte, le sous-système graphique, l'interface graphique et des applications de base. De même pour le Mac, BeOS, etc. On ne fait pas facilement la différence entre les composants.

Dans le monde X, ce sont des composants séparés, qui pris ensemble forment l'interface graphique utilisateur. Cela semble peut-être compliqué, mais cela a un grand avantage : vous pouvez choisir chacun des composants selon votre goût et vous construire l'interface graphique qui vous plaît le plus. c'est ce qui vous permet par exemple, de remplacer le gestionnaire de fenêtre Enlightenment par WindowMaker dans l'environnement de bureau Gnome.

