Cours Unix

A - Introduction à Unix

1 . L'utilisation d'Unix dans l'entreprise

Serveur	Base de données (Oracle, Informix, DB2, Ingres…) et applications de gestion / gestion de Production etc… Services Internet (DNS, WEB, Messagerie, FTP, Proxy, Firewall…) Serveur de fichiers et d'imprimantes (pour Unix, Mac, ou Windows)
Stations de travail	Bureautique CAO / DAO / FAO

2 . Unix : un système d'exploitation

a) Des concepts

- Processus
  Démarrage du système
  Démarrage d'une session utilisateur
  Communication inter-processus (IPC Inter Process Communication)

- Systèmes de fichiers
  Organisation des disques et des fichiers
  Procédures de sauvegarde

- Utilisateurs et groupes
  Droits d'accès
  Courrier électronique et messagerie

- Imprimantes et files d'attente

- Protocoles et applications de communication

- Interfaces utilisateurs

b) Des points de vue spécifiques

- Utilisateur
- Administrateur
- Développeur

3 . Quelques repères historiques

a) Généalogie d'Unix

Le développement d'UNIX commença en 1968 dans les BELL Laboratories, filiale de Western Electric Company, elle-même filiale de AT&T. Une première version expérimentale sur PDP/7 voyait le jour, en 1969.

La première version officielle fut installée sur un PDP 11/20 de DEC (Digital Equipment Corporation). Elle était signée de Ken THOMSON et Dennis M. RITCHIE.

La seconde édition d'UNIX disponible en 1972, fut installée sur une centaine de machines de type PDP 11.

* UNIX et langage C

Ken THOMSON travaillait sur un langage interprété élémentaire : le langage B

Ce langage fut amélioré par D. RITCHIE pour donner le langage C, compilé.

En 1973, le code d'UNIX fut réécrit en langage C par Ken THOMSON et Dennis M. RITCHIE.

* Temps Réel et Génie logiciel

A partir de 1979 vont être développés deux systèmes d'exploitation dérivés d'UNIX

MERT qui est une version temps réel d'UNIX. Elle ne sera utilisée qu'en interne chez AT&T.

PWB (Programmers Work bench) qui accentuera les caractéristiques d'UNIX permettant le développement des projets logiciels. Un des produits en est l'utilitaire SCCS (SCCS = Source Code Control System).

* UNIX et «UNIX like»

La version 6 fut la première commercialisée d'UNIX en 1975.

A partir de 1979, commencent à apparaître des versions hors d'ATT d'UNIX : certaines sont issues d'une licence UNIX

-UNIX B.S.D (de l'Université de Berkeley)

-XENIX (développé par Microsoft)

-Sun os (développé par sun)

D'autres sont des imitations qui en reprennent les fonctionnalités, comme

- EDRIS de whitesmith

* Un système indépendant du matériel

A partir de 1981, AT&T adopte une nouvelle terminologie pour les versions d'UNIX

On verra successivement:

- En 1981 UNIX system III
- En 1983 System V (UNIX système IV ne fut pas commercialisé)

A ce changement de dénomination correspond un nouvel engagement d'AT&T : le portage d'UNIX sur les microprocesseurs les plus puissants du marché INTEL IAPX X286, Motorola 68000 : Ce qui va ouvrir considérablement l'éventail des machines supportant UNIX.

Les versions suivantes d'UNIX vont s'appeler **system V R2** (1986), **system V R3** (1987).

La version **R4** d'UNIX system V propose une synthèse de 4 systèmes majeurs **:**

UNIX system V, Berkeley, XENIX et SUN OS.

Le système d'ATT est devenue la propriété de SCO Unix, après un cours passage chez NOVELL. Puis SCO Unix a ensuite été racheté par Caldera .

* Linux ...

Le dernier né des systèmes Unix est Linux, (Linuz Thorwald Unix) dont on connaît la rapide diffusion grâce à Internet, et à l'engouement pour "le logiciel libre".

> Linus B.Torvalds est l'inventeur de ce système d'exploitation entièrement gratuit. Au début des années 90, il voulait mettre au point son propre système d'exploitation pendant ses loisirs. Linus Torvalds avait pour intention de développer une version d'UNIX pouvant être utilisé sur une architecture de type 80386. Le premier clone d'UNIX fonctionnant sur PC a été Minix, écrit par Andrew Tanenbaum, un système d'exploitation minimal pouvant être utilisé sur PC. Linus Torvalds décida donc d'étendre les possibilités de Minix, en créant ce qui allait devenir Linux. Intéressés par cette initiative, de nombreuses personnes ont contribué à aider Linus Torvalds à réaliser ce système, si bien qu'en 1991 une première version du système a vu le jour. C'est en mars 1992 qu'a été diffusée la première version.
>
> Avec le nombre croissant de développeurs travaillant sur ce système, celui-ci a rapidement pu intégrer tous les outils présents sous UNIX. De nouveaux outils pour Linux apparaissent désormais a une vitesse vertigineuse.
>
> L'originalité de ce système réside dans le fait que Linux n'a pas été développé dans un but commercial. En effet aucune ligne de code n'a été copiée des systèmes UNIX originaux (en effet Linux s'inspire de nombreuses versions d'UNIX commerciales: BSD UNIX, System V. BSD UNIX). Ainsi, tout le monde, depuis sa création, est libre de l'utiliser mais aussi de l'améliorer.
>
> extrait de http://www.commentcamarche.net/linux/linintro.php3

* Les distributions Linux

Pour les différentes versions commerciales d'Unix, il existe une entreprise qui est responsable du développement du logiciel Unix et des différents outils logiciels associés. Cette entreprise livre à son client les logiciels que celui-ci a acheté, ainsi que les différentes mises à jour de logiciel qui interviendront plus tard, selon les termes du contrat de maintenance logiciel conclu entre l'entreprise et son client.
Dans le domaine de l'Open Source qui est celui de Linux les choses sont un peu différentes, dans la mesure où il n'y a pas d'entreprise unique de développement du logiciel Linux mais une foule de sites internet où sont entreposées les différentes versions des briques logicielles du système Linux. Il est donc possible de reconstruire un système Linux "sur mesure" en collectant sur ces différents sites les briques de base que l'on souhaite. Cependant cette reconstruction est fastidieuse et demande un certain savoir-faire.
Certaines sociétés se sont donc spécialisées dans ce genre d'activité et proposent des systèmes "clés en mains", disponibles sur DVD ou en téléchargement sur Internet, avec des procédures de mise à jour plus ou moins sophistiquées et automatisées.
Le site Internet http://distrowatch.com/ propose une sorte de vue panoramique des différentes distributions Linux et leur plus ou moins grande popularité, sachant que les distributions de type poste de travail graphique seront naturellement plus populaires que des distributions spécialisées pour le temps réel ou les pare-feu réseau par exemple.

Les distributions les plus populaires (avec leur numéro de version, en Novembre 2007) sont les suivantes:

1. Ubuntu 7.10
2. openSUSE 10.3
3. Fedora 7
4. Mandriva 2008
5. PCLinuxOS 2007
6. Knoppix 5.1.1
7. SimplyMEPIS 6.5
8. Kubuntu 7.10
9. Debian 4.0r1
10. Freespire 2.0.6
11. Slackware 12.0

Autres liens pour Linux :

http://www.linux-france.org/article/materiel/mac/bonnesadresses.html
http://www.linux.org/info/index.html

b) La Normalisation

La diffusion considérable d'UNIX a permis le développement d'un grand nombre d'applications dans tous les domaines: bureautique, gestion commerciale et comptable, gestion de production, CAO etc...

Garantir la pérennité de ces applications sur les machines futures, c'est assurer à la fois la compatibilité ascendante des versions successives d'UNIX et l'exploitation officielle par UNIX des nouvelles possibilités du matériel comme la mémoire virtuelle, les communications entre machines ou les interfaces graphiques etc...

Ceci a conduit les constructeurs d'informatique d'une part, les utilisateurs de systèmes UNIX d'autre part, à se regrouper pour définir des normes régissant les différents aspects du système d'exploitation.

* Du coté des constructeurs

- 1984: X/OPEN groupe constitué des constructeurs
- 1988 : association AT&T et SUN autour d'UNIX system V
- 1988: OSF «Open software foundation» où l'on retrouve
  IBM, BULL, DEC, HP, APOLLO, et NIXDORF.
  L'UNIX de base choisi est l'AIX D'IBM.

Les principaux documents produits sont:

- Pour X OPEN: X/OPEN portability guide basé sur les
  caractéristiques d'UNIX V/R2 de AT&T
- Pour AT&T : SVID (system V Interface Définition - 1989)

* Du coté des utilisateurs

On trouve principalement l'association «**/User/Group**» fondée en 1981 et qui réalise un travail de normalisation d'UNIX au sein de l'IEEE (Institute of technical and electronics engineers).

Cette association a publié en 1986, les premiers résultats de ses travaux dans un document appelé POSIX («Portable operating system for computer environment»).

Le projet POSIX[1] (norme IEEE/1003) est relativement indépendant des constructeurs.

Il tente de normaliser à la fois le langage C et l'interface système dans le but de garantir la portabilité des applications au niveau du langage Source.

1 Bien qu'il soit principalement implémenté sur des systèmes de type UNIX, le standard POSIX peut être utilisé par n'importe quel autre système d'exploitation. Par exemple, Microsoft Windows NT est conforme à POSIX.1, ce qui est suffisant pour des programmes POSIX relativement simples. Cependant, des programmes plus complexes ont besoin d'une compatibilité plus grande. Des logiciels supplémentaires tels que Windows Services for UNIX ou Cygwin peuvent apporter à Windows ce niveau de compatibilité.

source : http://fr.wikipedia.org/wiki/POSIX

B - Le système d'exploitation Unix en mode utilisateur

1 . Déroulement d'une session utilisateur

a) La connexion d'un utilisateur sous UNIX

sur un système Windows : démarrer/exécuter/**telnet**[2] **Nom_hôte**

login : taper son nom d'utilisateur
password : taper son mot de passe

en démarrant sous Linux

login : taper son nom d'utilisateur
password : taper son mot de passe

A partir de ce moment il est possible d'utiliser l'une quelconque des commandes UNIX: par exemple:

- pwd : pour connaître le répertoire courant
- cd <nom de répertoire> : pour changer de répertoire courant
- ls -l : pour connaître la liste des fichiers du répertoire courant
- cat <nom de Fichier> : pour afficher le contenu d'un fichier
- who ou finger : pour connaître la liste des utilisateurs connectés
- passwd : pour changer son mot de passe (yppasswd si annuaire NIS)

La fin de session UNIX se fera en tapant **Ctrl+D** ou **exit**

b) Modifier le fichier .profile

Le fichier **~/.profile** ou **~/.bash_profile** (~ désignant le répertoire d'accueil de l'utilisateur) est un fichier contenant des commandes interprétées par le shell au moment de la connexion de l'utilisateur: c'est un fichier de texte modifiable par tout éditeur de texte (vi, pico, joe sous Linux, kedit sous Kde)

Ce fichier joue un rôle comparable au rôle joué par le fichier autoexec.bat au niveau de MS DOS. Cependant pour MS DOS, mono-utilisateur, ce fichier est unique, alors que pour UNIX, il peut exister autant de fichiers .profile que d'utilisateurs

Pour être pris en compte au moment de la connexion UNIX, ce fichier doit se trouver dans le répertoire d'accueil de l'utilisateur (Home directory).

2 A la place de la commande telnet de Windows on peut utiliser d'autres logiciels comme putty, offrant aussi le choix du protocole crypté ssh
http://www.chiark.greenend.org.uk/~sgtatham/putty/download.html

Exemple de fichier **. profile**

clear	pour effacer l'écran
echo Bonjour $LOGNAME	pour afficher un message d'accueil
pwd	pour afficher le répertoire courant
PS1 = "$LOGNAME > "	pour définir le «prompt» primaire
PS2 = " suite : "	pour définir le «prompt» secondaire

Remarques :

$LOGNAME est une variable d'environnement qui contient le nom d'utilisateur donné au "login"
$PS 1 est le message d'attente de commande du shell(voir **man bash**)
$PS2 est le message d'attente secondaire qui sera affiché
lorsque l'utilisateur utilisera <Entrée> au milieu d'une commande;

Exemple:

```
demo@forum > echo "Bienvenue sur
suite : Unix "
Bienvenue sur
Unix
demo@forum >
```

c) Fichiers textes et Redirection des Entrées/Sorties standard

De nombreuses commandes UNIX permettent de traiter des fichiers de texte.
Elles sont souvent utilisées avec la redirection des entrées ou des sorties standard.
Rappelons que

>	redirige les sorties sur un fichier
>>	redirige les sorties en complétant un fichier qui contient déjà des données
<	redirige les entrées : ce qui permet de prendre dans un fichier de texte les réponses normalement demandées à l'opérateur
|	(AltGr + 6) définit un filtre pour un enchaînement de commandes Les sorties d'une commande deviennent les entrées de la commande suivante

Application :

Pour connaître tous les noms de fichiers et répertoires, on pourra utiliser la commande **find**, mais il sera difficile de lire le texte sans utiliser le filtre **more** qui permet d'afficher page par page les lignes de texte produites par la commande **find**.

La syntaxe exacte sera la suivante: **find / | more**

d) Arborescence des répertoires[3]

Le système UNIX définit une organisation typique des répertoires : 1'utilisateur peut créer d'autres répertoires mais un certain nombre sont nécessaires au fonctionnement du système, citons en particulier

/tmp	répertoire de fichiers temporaires, accessible en écriture par tous les utilisateurs
/bin et /usr/bin	répertoires des commandes utilisateur
/sbin et /usr/sbin	répertoires des commandes superviseur
/usr/include	répertoire des fichiers .h du langage C
/usr/lib	répertoire des bibliothèques utilisées par les compilateurs et l'éditeur de liens
/var/log	répertoire des messages système
/var/spool	répertoire des files d'attentes d'impression
/var/spool/mail	répertoire du courrier électronique
/dev	répertoire des pilotes de périphériques

Tous les utilisateurs ont un **répertoire personnel** (répertoire d'accueil, en anglais home directory) dans le répertoire **/home** (ou à l'ISAIP-ESAIP dans /serveur) ou dans un de ses répertoires subordonnés.

- La variable d'environnement **$HOME** contient le chemin d'accès à ce répertoire.
- La commande **cd**, (change directory) sans paramètre, permettra très simplement de revenir à ce répertoire après avoir activé un autre répertoire de travail.
- La commande **pwd** (print working directory) quant à elle, affiche le nom du répertoire de travail

Un nom de fichier peut être défini de façon absolue ou relative:

.	désigne le répertoire courant
. .	désigne le répertoire de niveau immédiatement supérieur. **../texte** désigne ainsi le fichier (ou le répertoire) **texte** situé dans le répertoire **parent du répertoire courant**

La variable **$PATH** défini le chemin d'accès aux commandes; si plusieurs répertoires sont susceptibles de contenir les commandes utilisées, on séparera les noms de ces répertoires par ' : ' dans la chaîne de caractères affectée à **$PATH.**

3 On pourra par exemple utiliser la commande **tree / -d -L 2 | less** pour visualiser les 2 premiers niveaux de répertoires du systèmes sur lequel on travaille (voir **tree --help** pour les paramètres de cette commande)

Quelques commandes relatives aux fichiers et répertoires :

mkdir xy	permet de créer le répertoire xy
rmdir xy	permet de supprimer le répertoire xy
rm -r xy	permet de supprimer (récursivement) tous les fichier du répertoire xy et des répertoires subordonnés.
Il -R xy	permet d'afficher (récursivement) tous les fichier du répertoire xy et des répertoires subordonnés.
file *	permet de connaître le type de tous les fichiers du répertoire courant.

* La commande find

Cette commande permet de rechercher des fichiers ou répertoires dans une arborescence, à partir d'un niveau donné.
Les paramètres de **find** sont expliqués ci-après; un certain nombre font appel à des notions sur le système de fichiers ou les utilitaires de sauvegarde qui seront expliquées plus tard.

Paramètres de la commande find :

1. le répertoire de départ pour la recherche
2. des critères de sélection de fichiers
3. des commandes à exécuter sur les fichiers trouvés.

Critères de sélection de fichiers

-name "*.c"	recherche par nom (* et ? autorisés, utiliser alors des guillemets)
-perm 0755	recherche par droits d'accès (en octal)
-type [c,b,d]	recherche par type de fichier (c : mode caractère; b: mode bloc; d:répertoire)
-links n	fichiers ayant n liens
-user nom	fichiers dont le propriétaire est nom
-inum n	fichiers dont le n° d'inode est n
-size n	fichiers dont la taille est égale à n blocs +300c signifie > 300 caractères (octets) -3 signifie < 3 blocs
-atime n	fichiers dont le dernier accès date de n jours +n signifie > n jours -n signifie < n jours
-mtime n	dernière modification du contenu du fichier il y a **n jours** (modified)
-ctime n	dernière modification des attributs du fichier il y a **n jours** (changed)
-newer **fichier1**	fichiers modifiés plus récemment que **fichier1**

\(expression\) fichiers vérifiant l'expression parenthésée :
combinaison à l'aide des opérateurs -o, -a, et !
des conditions élémentaires ci-dessus (Or, And et Not).

Commandes à exécuter sur les fichiers trouvés

-exec **cmd** \;	exécution de la commande **cmd** pour tout fichier trouvé; dans l'expression de la commande, le nom du fichier sera remplacé par **{}** Ex.: **-exec pg {} \;** fera afficher par **pg** le contenu du fichier trouvé.
-ok **cmd** \;	identique à **exec** avec demande de confirmation de commande à chaque nom de fichier. Exemple : **-ok pg {} \;**
-print	affichage du chemin d'accès aux fichiers trouvés, donc correspondant aux critères
-cpio dev	sauvegarde du contenu des fichiers trouvés sur le périphérique **dev** dans le format **cpio**

Exemples

find . -name "*.c" -print 2>/dev/null

Affiche tous les noms de fichiers ayant l'extension .c , dans toute l'arborescence rattachée au répertoire courant , répertoire ' . '
Les messages d'erreurs éventuels sont redirigés sur /dev/null, donc détruits.

find \(-user toto -a -size +1024c \) -ok file {} \;

Recherche tous les fichiers appartenant à toto et de taille supérieure à 1024 Octets.
Pour chaque fichier trouvé, la commande demandera à l'opérateur s'il veut voir afficher par file le type des informations contenues dans ce fichier.

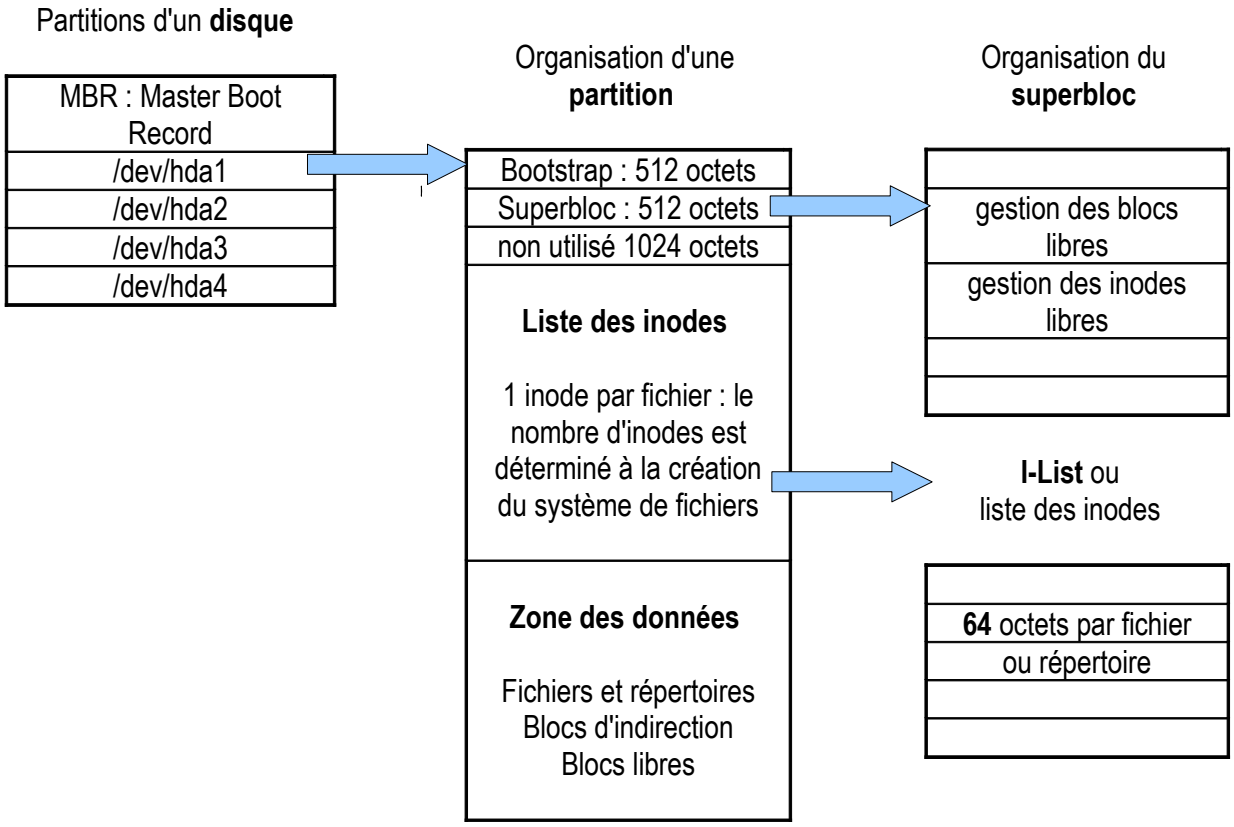
2 . Les concepts de base

- ➔ Le système de fichiers d'UNIX
- ➔ Les utilisateurs et les droits d'accès
- ➔ Les processus
- ➔ La gestion des imprimantes

a) Le système de fichiers UNIX

* Introduction

L'utilisateur d'un système d'exploitation n'a pas en général à se préoccuper de la manière dont les informations sont rangées sur les disques : c'est au contraire un des rôles des systèmes d'exploitation de décharger l'utilisateur de ce souci.
Mais pour mieux comprendre le fonctionnement de certaines commandes, il est utile de connaître quelques informations essentielles décrivant le système de fichiers.
Le schéma suivant montre l'architecture générale du système de fichiers UNIX.



* Partitions et montage des volumes

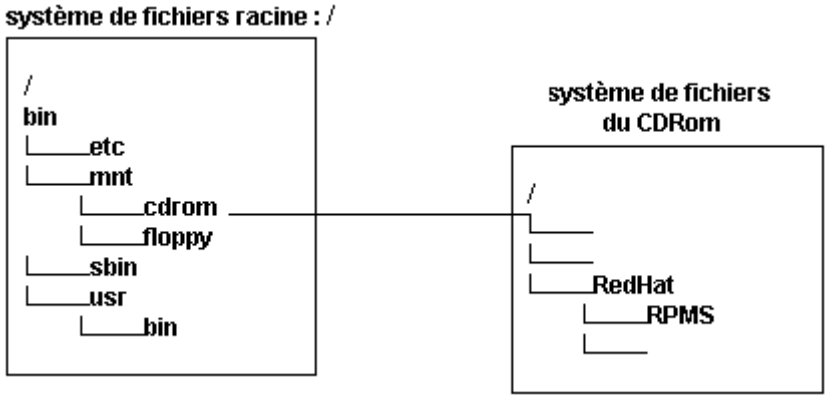
Un disque physique est en général «découpé» en un certain nombre de partitions qui seront vues par le systèmes comme autant de disques physiques distincts.
Sur chaque partition sera définie un organisation spécifique des données qui s'appelle un «**système de fichiers**» ou «**file system**».
Cette organisation repose sur une gestion de blocs : blocs libres et blocs alloués aux fichiers ; On distinguera des blocs physiques et des blocs logiques qui sur un PC 386 sous Linux ont respectivement pour taille 512 et 1024 octets (ces valeurs peuvent être différentes sur d'autres configurations UNIX)

Une des partitions est privilégiée - c'est la partition système qui contient le programme d'amorçage (bootstrap) dans son premier secteur physique. Mais ce Bootstrap (pour Linux, "LiLo" = Linux Loader) peut aussi être logé dans le MBR, indépendant des partitions.

La partition système (/) contient le noyau du système UNIX (localisé pour Linux, dans /boot) et toutes les commandes de base, répertoriés dans /bin et /sbin.

Au démarrage du système, seule la partition système est est connue d'UNIX : les autres partitions seront rattachées à un répertoire de cette partition par une opération de montage :

> si l'on considère qu'un système de fichiers est une arborescence de répertoires,
> le **montage d'un volume** consiste à rattacher à un répertoire connu du système
> l'arborescence contenue dans un autre volume (ou partition),
> comme l'indique le schéma ci-après: montage d'un CDRom



La commande **mount** permet d'afficher la liste des volumes montés. Elle permet aussi de monter un volume non monté; **umount** permettant de démonter un volume..
Unix définit aussi un fichier **/etc/fstab** (4) ou **/etc/mnttab** (4) contenant la liste des volumes à monter automatiquement à l'initialisation du système.

Certaines partitions peuvent être utilisées par le système sous un mode non structuré, c'est à dire sans supporter la structure des systèmes de fichiers que nous décrirons ci-après. En général ceci est réalisé pour garantir un accès plus rapide aux données : c'est le cas pour la partition de **SWAP** (mémoire d'échange) utilisée par le système pour recopier une partie de la mémoire centrale lorsque les demandes cumulées des utilisateurs excèdent la taille de la mémoire RAM disponible.

* Structure d'un système de fichiers UNIX

Le système de fichiers UNIX est basé sur une structure particulière appelée **inode** ou noeud d'index (index node).

A chaque fichier correspond un inode et un seul, qui est en fait une table des numéros de blocs logiques (1024 octets) alloués au fichier. Mais cette table serait trop volumineuse si on devait y référencer tous les blocs d'un fichier important. : l'inode ne contient en fait de référence directe qu'aux 10 premiers blocs du fichiers (10240 octets ou 10 Ko).

Le 1lème numéro de bloc est le numéro d'un bloc contenant lui même 256 numéros de blocs de données supplémentaires (simple indirection)

Le 12ème et le 13ème numéro de bloc donnent accès à des listes de double et de triple indirection, comme indiqué sur le schéma ci-après.

N° d'octet	taille en octet	Contenu du champ
0	2	type et mode du fichier (droits d'accès)
2	2	nombre de liens (synonymes)
4	2	UID numéro du propriétaire
6	2	GID numéro du groupe propriétaire
8	4	taille du fichier en nombre d'octets
12	39	13 pointeurs sur 3 octets des blocs de données
51	1	non utilisé
52	4	date du dernier accès
56	4	date de la dernière modification des données
60	4	date du dernier changement (dernière modification de l'inode)

Description des 13 pointeurs de blocs de données	taille maximum des données pointées
1 à 10 : adressage direct des données	1024 * 10 = 10240 octets
11 : simple indirection adressage d'une table de 256 pointeurs chacun pointant sur des blocs de données	1024 * 256 = 262 144 octets
12 : double indirection adressage d'une table de 256 pointeurs chacun pointant une table de 256 pointeurs de données	1024 * 256 * 256 = 67 108 864 octets
13 : triple indirection	1024 * 256 * 256 *256 = 17 179 869 000 octets

Il apparaît immédiatement que l'accès aux petits fichiers sera très rapide, comparativement à l'accès aux données d'un grand fichier qui pour une lecture de données pourra nécessiter le lecture supplémentaire de 3 blocs d'index. Notons cependant que cette indirection est moins pénalisante si l'on traite séquentiellement les données que dans le cas d'un accès direct.

La compréhension de cette indirection peut suggérer des procédures d'optimisation; mais c'est aussi

la raison pour laquelle des gestionnaire de bases de données sous UNIX utilisent des partitions non structurées (ou structurées selon une méthode non UNIX) pour mémoriser les données sur disque. L'inode contient encore d'autres informations qui apparaissent dans la liste ci-dessus.

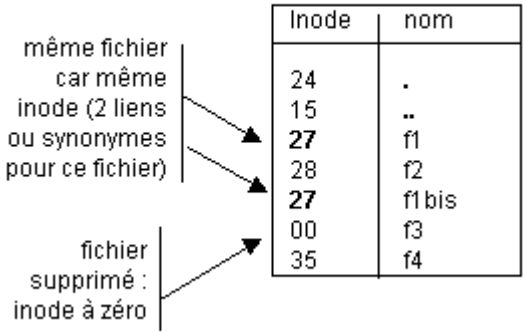
La commande **ls -il** affiche la plupart de ces informations : l'option **-i** donnant le numéro de l'inode associé au fichier.

* Répertoires

Un répertoire pour UNIX est un fichier ordinaire, ayant aussi son numéro d'inode, mais dont le contenu est une table associant un nom symbolique (14 caractères) à un numéro d'inode.

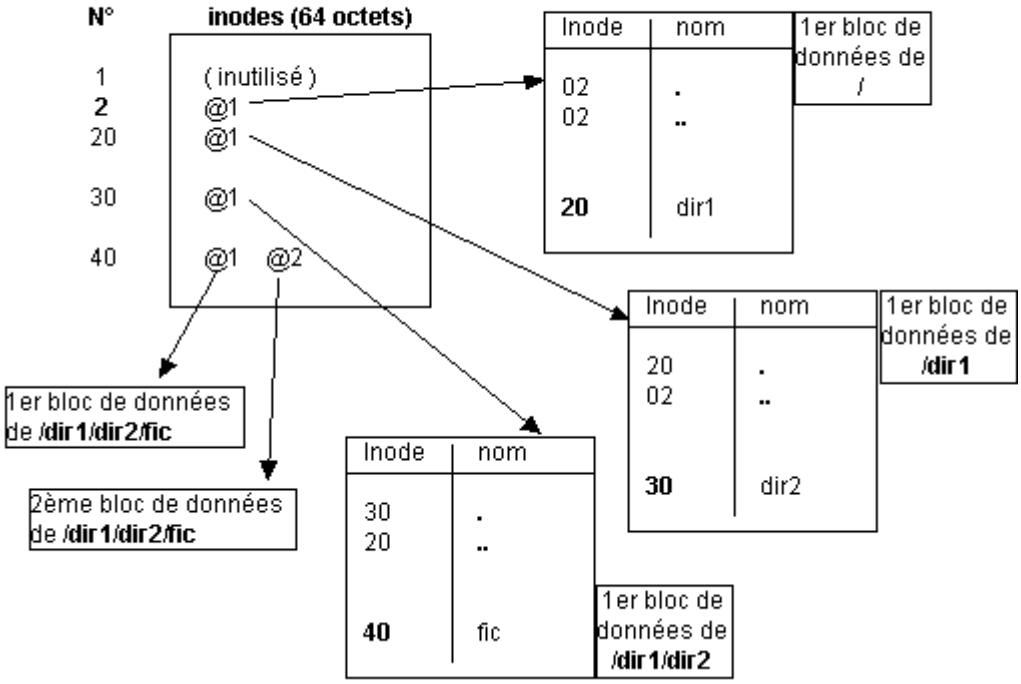
On voit ici apparaître ce qu'est un lien (créé par la commande ln) : c'est l'association d'un nouveau nom à un inode déjà référencé dans un répertoire, comme le montre le schéma ci-contre.

Exemple de répertoire Unix



Accès aux données de /dir1/dir2/fic à partir de l'inode 2

Le répertoire racine (/) de chaque volume correspond toujours à l'inode 2



* Liste des inodes (i-liste)

La liste des inodes commence au 3ème bloc physique du système de fichiers : sa taille est fixée à l'initialisation du système de fichiers (utilitaire **mkfs** : make file system)

Un inode libre aura tous ses champs mis à zéro.

Lorsque l'on supprime un fichier par la commande **rm**, le fichier n'est effectivement supprimé que si le nombre de liens était égal à 1; dans le cas contraire, c'est seulement un lien qui disparaît.

* Superbloc et gestion des blocs libres et des inodes libres

Le superbloc est la structure de plus haut niveau dans la description d'un système de fichiers, en plus de l'identification du système de fichiers, il contient les informations utilisées par le système pour la gestion des blocs libres et la gestion des inodes libres.

Le superbloc est le 2ème bloc physique (512 octets) de tout système de fichiers.

Il est chargé en mémoire centrale dès que le système de fichiers est «monté».

Le schéma suivant donne la structure du superbloc.

@Hexa	octets	Contenu des champs
0	200	Bootstrap (secteur d'amorçage)
200	2	N° du premier bloc allouable après la i-list
202	4	Taille du système de fichiers
206		Gestion des blocs libres
2d0		Gestion des inodes libres
39a	4	Flags / Verrous
39e	4	Date dernière mise à jour du superbloc
3a2	4	Nombre total de blocs libres (en blocs logiques de 1024 octets)
3a6	2	Nombre total d'inodes libres
3a8		informations "device"
3b0	6	nom du système de fichiers
3b6		nom du volume
3bc		type de système de fichiers
400		bloc inutilisé
800		début de la 1-liste

* Applications.

Un certain nombre d'utilitaires permettent de faire apparaître la structure du système de fichiers on pourra par exemple utiliser les commandes suivantes :

1) Partitions du disque	**fdisk** -l	(nécessite une session **root**)
	cfdisk	
	gestionnaire de partitions de **webmin** ou **yast**	
2) volumes montés	**mount**	
3) superbloc	**df**	(disk free blocs)
4) répertoires	**ls** -ial	
	du <rep>	(disk usage)
5) inodes	**ls** -ial	(noms + inodes)

b) Les utilisateurs et les droits d'accès

* Utilisateurs et groupes

Les utilisateurs définis sur un système Linux peuvent être listés à partir de **webmin** (ou **yast** ou ...).

> Les utilisateurs locaux (n'appartenant pas à un annuaire réseau comme LDAP, NIS, ActiveDirectory ou eDirectory) peuvent être connus par la commande **less /etc/paswd**

La commande **who -uH** ou **finger** permet d'obtenir la liste des utilisateurs actuellement connectés, avec le numéro de processus associé au **shell** initial de chacun d'eux.

En cours de session, il est possible de changer de nom d'utilisateur (à condition de connaître le mot de passe du nouvel utilisateur) pour acquérir les droits qui s'y rattachent: ceci est réalisé par la commande **su** (su = shift user).

Exemple :

su demo
su - demo

"**su - demo**" permet de changer en même temps d'environnement, alors que "**su demo** " change les droits sans changer l'environnement de travail (répertoire actif, variables d'environnement etc...).

Application

La commande **id** permet à tout moment de vérifier son identité (mon nom **d'utilisateur** et les groupes auxquels j'appartiens)

id demo donne les caractéristiques de l'utilisateur **demo.**

* Droits d'accès

Dans un système multi-utilisateur il est nécessaire de protéger à la fois le système et les différents utilisateurs contre les possibles altérations des données provoquées par des actions volontaires ou fortuites des autres utilisateurs.

UNIX définit la notion de **super-utilisateur** : protégé par un mot de passe, le super-utilisateur (nommé **root**) a tous les droits sur le système : fichiers, processus etc... Le numéro d'utilisateur de **root** est zéro.

Droits d'accès "rwx" aux fichiers sous UNIX : Principe

Tout fichier appartient à un utilisateur et à un groupe d'utilisateur

Pour chaque fichier sont définis les droits r w x concernant le propriétaire du fichier, le groupe propriétaire et enfin les autres utilisateurs

Le droit **r**	permet de **lire** les données du fichier
Le droit **w**	permet de **modifier** les données du fichier
Le droit **x**	permet **d'exécuter** le programme lorsque le contenu du fichier est un programme et **d'utiliser ce nom de répertoire dans un chemin d'accès** lorsque le fichier est un répertoire.

Lorsqu'un nouveau fichier est créé : utilisation de umask

- le propriétaire est l'utilisateur qui le crée
- le groupe propriétaire est le groupe dont fait partie l'utilisateur qui crée ce fichier,
- les droits d'accès sont les droits implicites que l'utilisateur a défini par une commande **umask** une commande **umask** existe dans **/etc/profile,** exécutée à l'ouverture de session par tout utilisateur.

Les droits révoqués par umask, sont exprimés sous forme octale, (**1** indique la révocation du droit et **0** sa non-révocation donc son attribution) par exemple

umask 007 attribuera les droits **111 111 000** soit "**rwx rwx ---"** à tout nouveau fichier

on peut aussi définir, de manière équivalente, les droits implicite en mode symbolique :

umask -S u=rwx,g=rwx,o=

Remarquons cependant que dans ce cas, un nouveau fichier créé par un éditeur de texte n'aura pas automatiquement le **droit x** pour le propriétaire et le groupe; ce qui est normal car le système ne peut supposer que le fichier contient un programme exécutable, le **droit x** devra donc être spécifiquement ajouté par la commande **chmod** , s'il s'agit par exemple d'une procédure script **shell** (voir commande **chmod** ci-après).

Par contre si la création du fichier résulte de la **copie** d'un fichier ayant le droit **x,** ce droit sera attribué automatiquement au nouveau fichier sous contrôle des autorisation définies par **umask**. C'est la cas aussi lorsque le nouveau fichier est un exécutable créé par un compilateur.

Attention : la création d'un fichier par un utilitaire comme le gestionnaire de fichiers (comme **kfm** ou **nautilus**) peut affecter des droits différents de ceux que l'on a définis par **umask**.

Changer les droits d'accès : chown, chgrp, chmod

Pour changer les droits d'accès attribués à un fichier ou un répertoire, il faut en être le **propriétaire**.

Pour changer l'identificateur du propriétaire il faut être **superviseur (root)**

```
chown proprietaire2 fichierl fichier2 ...
chown proprietaire3 *.c
```

Pour changer l'identificateur du **groupe propriétaire** il faut faire partie du nouveau groupe

```
chgrp proprietaire2 fichierl fichier2
chgrp proprietaire3 *.c
```

Pour changer les droits d'accès on utilisera **chmod** en définissant les nouveaux droits en **octal** ou sous une forme **symbolique** dans laquelle

u=user g=group o=other a=all

\+ ajoute les droits, - retire les droits, = attribue les droits

exemples :

```
chmod 740 fichierl fichier2
chmod u=rwx,g=r,o-rwx fichier3
chmod u+rw *.c
chmod a=-,u+rw,g+r *.txt
chmod u=rwx,g=rx,o=r fichier4
etc ...
```

La commande **ls -l** permet d'afficher en même temps que les noms de fichiers d'un répertoire, les droits d'accès qui y sont attachés:

Utiliser **ls -ld** si l'on précise un nom de répertoire, par exemple:

ls -ld /etc pour connaître l'affectation des droits d'accès relatifs à **/etc**,
ls -ld . pour afficher les droits du répertoire courant)

* Création et suppression des utilisateurs et des groupes

Les utilisateurs sont créés par la commande **useradd** qui vient modifier le fichier **/etc/passwd et /etc/shadow.**

voir man **useradd** pour les paramètres associés à cette commande, qui peuvent varier selon les différentes versions Unix, et même les différentes distribution Linux.

Exemple de liste d'utilisateurs Unix :

USER	**UID**	**GID**	**HOMEDIR**	**SHELL**
root	0	3	/	
lp	71	2	/usr/spool/lp	
charles	1101	110	/users/profs/charles	
tregouet	1102	110	/users/profs/tregouet	
demo	3013	30	/users/demo	
lecointr	1106	110	/users/profs/lecointre	
bouchez	2001	200	/users/stagiaires/bouchez	
rnis	2002	200	/users/stagiaires/rnis	

Le fichier /etc/passwd a lui même la structure suivante

Utilisateur : Mot de passe crypté : UID : GID : * : Répertoire : shell

root:2R3UfuIBcvxf2:0:3:0000-Admin(0000):/:
lp:**NO LOGIN**:71:2:0000-lp(0000):/usr/spool/lp:
charles:JBOOGDB776hM6:1101:110::/users/profs/charles:
tregouet:DPGio9VwAVznk:1102:110::/users/profs/tregouet:
demo:FpElc4TqVOIbA:3013:30::/users/demo:
lecointr::1106:110::/users/profs/lecointre:
bouchez:Yk6qXaGEcaQcM:2001:200::/users/stagiaires/bouchez:
rnis::2002:200::/users/stagiaires/rnis:

Le champ noté (*) est un champ d'information.
Remarquons aussi que les mots de passe sont cryptés et ne peuvent donc être reconnus directement. Si un utilisateur oublie son mot de passe, le seul remède est, en mode super-utilisateur, de définir un nouveau mot de passe.

Cependant si le fichier /etc/passwd est accessible en lecture, il deviendra possible de lancer un programme de "crack" permettant d'essayer un grand nombre de mots de passe jusqu'à trouver celui qui donnera le mot de passe crypté enregistré dans /etc/passwd.

Pour une meilleure sécurité, il sera donc **recommandé** d'appliquer les règles suivantes :

- ✔ utiliser une modification du système d'authentification faisant appel au fichier **/etc/shadow** dans lequel seront mémorisés les mots de passe, le fichier **/etc/shadow** ne sera accessible en lecture qu'au seul superviseur, le fichier **/etc/passwd** quant à lui restera accessible en lecture pour tous les utilisateurs, mais il ne contiendra plus les mots de passe.

- ✔ ne pas utiliser de mots du dictionnaire pour les mots de passe, ni de mots de passe constitués à partir de données connues de sa vie privée comme les prénoms de ses proches ou sa propre date de naissance, ou son lieu de résidence...

Les groupes sont créés par la commande **groupadd** qui vient modifier le fichier **/etc/group**
La syntaxe de cette commande est donné par man **groupadd**

Un exemple de liste de groupes est donné ci-après :

GRP	**GID**	**USERS**
root	0	root
other	1	
bin	2	root, bin,daemon
sys	3	root,bin,sys,adm
étudiants	130	albert,arnaudeau,borowczy

Le fichier **/etc/group** a quant à lui la structure suivante

Nom du groupe : Mot de passe : GID : Liste des membres du groupe

root::0:root

other::l:

bin::2:root,bin,daemon

sys::3:root,bin,sys,adm

etudiants::130:albert,arnaudeau,borowczy

Le mot de passe défini sur les groupes est normalement exploité par la commande **newgrp** cependant cette possibilité de changer de groupe introduit une faiblesse dans le système de sécurité d'UNIX, elle est **déconseillée**, et aucun utilitaire n'est fourni qui permette de changer facilement le mot de passe associé à un groupe.

En général, un utilisateur pourra changer de groupe principal en utilisant **newgrp** pour un groupe dont il fait déjà partie, et il ne lui sera demandé aucun mot de passe. C'est l'administrateur **root** qui, seul, a le pouvoir de changer les membres d'un groupe.

c) Les processus

Dans un système multitâche, le processeur unique partage son activité entre plusieurs programmes qui s'exécutent ainsi dans une **apparente simultanéité**. L'allocation du processeur aux différentes tâches actives (ou processus actifs) est une des activités principales du système d'exploitation.

Pour mener à bien cette gestion des processus (qui nécessite aussi du temps processeur ...), le système d'exploitation définit des informations spécifiques sur l'état de chaque processus: ces informations sont affichées par la commande **ps** décrite ci-après. Nous considérerons ici que la notion de tâche est équivalente à la notion de processus.

Sous UNIX, à chaque utilisateur connecté est associé un processus qui exécute en général le programme **shell**, qui est l'interpréteur de commandes. Lorsqu'un utilisateur tape le nom d'une commande, le **shell** crée un nouveau processus qui exécute cette commande. Ce nouveau processus disparaît à la fin de l'exécution de cette commande.

La liste des processus actifs à un moment donné est fournie par la commande **ps**. Sans paramètres, **ps** ne donne que les processus créés par l'utilisateur qui invoque la commande.

ps admet différentes options, qui peuvent varier d'un système Unix à l'autre. Utiliser man **ps** pour connaître la syntaxe exacte de **ps** sur le système utilisé.

Sous Linux on utilisera en particulier
ps axu affiche tous les processus de la machine, et le propriétaire de chaque processus.
ps axf (f=forest)fait apparaître l'arborescence des processus.
Résultat obtenu aussi par **pstree**.
ps u -uwwwrun donne tous les processus de l'utilisateur **wwwrun** (serveur web)

Enfin la commande **top** fait apparaître les processus classés d'après leur taux d'utilisation du processeur. Ces commandes se retrouvent dans une présentation graphique sous **Xwindow**; en particulier, avec le gestionnaire de fenêtres **kde**, on pourra utiliser **kpm** et **ktop** (ou **ksysguard**). Voir l'aide de **kpm** ((ou **ksysguard**) pour la description des différents champs d'information décrivant un processus

* Processus en arrière plan

Lorsque le **shell** lance une commande, il crée un nouveau processus exécutant cette commande, et reste en attente de la fin de ce processus fils. Il est possible, cependant de lancer une commande en arrière plan, en faisant suivre le nom de cette commande du caractère &.

Ainsi **cc x.c -o x &** va compiler **x.c** sans que l'utilisateur ait à attendre la fin de la compilation pour lancer d'autres commandes.

Notons que dans ce cas, il est intéressant de se faire avertir de la fin de la compilation ; la commande suivante le réalise: (**cc x.c -o x ; echo fin de compilation**) **&**

La variable **shell $!** donne par ailleurs le numéro du dernier processus lancé en arrière plan.

* Exemple de commande ps :

```
ptregouet@linux:~> ps lf

F   UID   PID  PPID PRI  NI    VSZ  RSS WCHAN  STAT TTY        TIME COMMAND
0  1000  4567  4563  15   0   4024 1724 wait4  Ss   pts/1      0:00 /bin/bash
0  1000  4622  4567  16   0   2148  632 -      R+   pts/1      0:00  \_ ps lf
5  1000  4615     1  16   0 32556 22572 schedu S   pts/1      0:02 kate
```

F Flags

UID Numéro d'utilisateur du propriétaire du processus
PID Numéro du processus
PPID Numéro du processus parent

PRI priorité du processus
NI courtoisie[4] du processus (seul root peut la diminuer)

VSZ Virtual Size : espace de mémoire virtuelle allouée au processus

RSS Resident Set Size = portion de la mémoire actuellement en mémoire
VSZ-RSS est la portion de mémoire actuellement en mémoire de swap
La commande TOP utilise VIRT et RES pour désigner VSZ et RSS

voir man ps :
The SIZE and RSS fields don't count the page tables and the task_struct of a proc; this is at least 12k of memory that is always resident. SIZE is the virtual size of the proc (code+data+stack).

WCHAN Indique les événements attendus par un processus endormi

STAT Status : état du processus

voir man ps :
PROCESS STATE CODES

D uninterruptible sleep (usually IO)
R runnable (on run queue)
S sleeping
T traced or stopped
W paging (2.4 kernels and older only)
X dead
Z a defunct ("zombie") process

TTY identification du terminal associé au processus
le shell 4567 lit et écrit sur /dev/pts/1
voir ce qu'affiche la commande tty

TIME temps processeur consomme par le processus
pour avoir une mesure plus precise,
taper time suivi de la commande (avec ses paramètres)

COMMAND La commande exécutée par le processus

4 la courtoisie est le contraire de la priorité :
- Si la priorité est élevée, le processus s'exécute **plus** rapidement
- Si la courtoisie est élevée, le processus s'exécute **moins** rapidement

d) La gestion des imprimantes

Dans un système multi-utilisateur, il n'est pas possible de laisser aux utilisateurs un accès immédiat à l'imprimante: ceci les obligerait à prendre en charge leur synchronisation mutuelle , pour éviter le mélange des textes sur l'imprimante ! Le mécanisme universellement utilisé pour gérer ce genre de problèmes est celui des files d'attentes d'impression ou "**spool**"

Un processus système, **lpd**, ordonnanceur des éditions, prend en charge la gestion des files d'attente des demandes d'impression. Les utilisateurs adresseront à cet ordonnanceur, des requêtes comportant le nom du fichier à imprimer (en particulier par la commande **lpr**).

C'est cet ordonnanceur, qui par un accès exclusif à l'imprimante, éditera les documents, les uns après les autres, en gérant éventuellement des priorités.

Il sera possible d'accéder à la file d'attente des éditions en cours par la commande **lpq**, et de supprimer un document de cette file d'attente par la commande **lprm**. En fin il est possible de contrôler le travail de **lpd** par la commande **lpc**.

Les distribution Linux fournissent aujourd'hui un bon support des gestionnaires d'imprimantes. Les outils d'installation de la distribution sont en général bien appropriés pour la plupart des imprimantes.

Le système de gestion d'imprimantes actuellement le plus utilisé sous Linux est **CUPS Common Unix Printing System.**

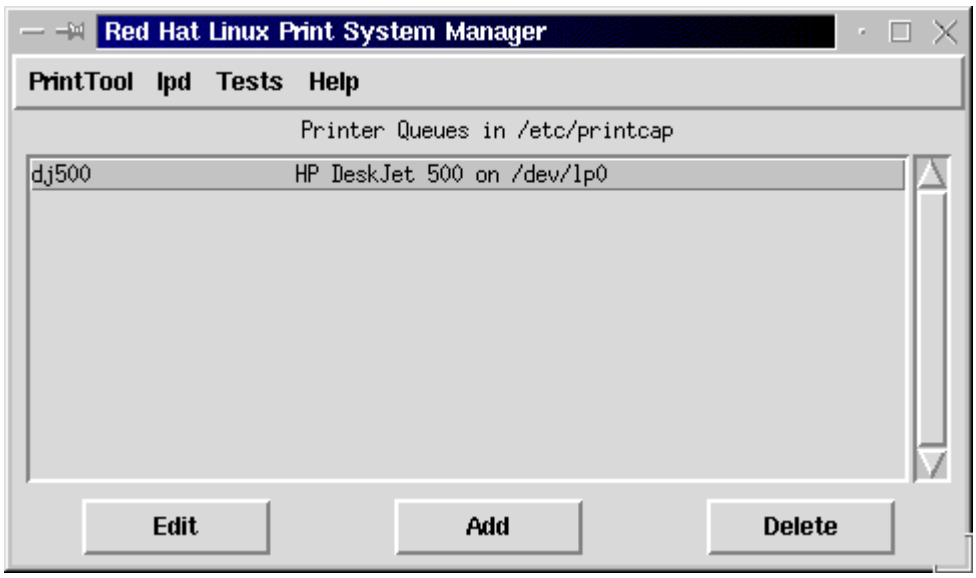
CUPS permet à la fois de gérer les imprimantes locales, de partager les imprimantes locales et d'accéder à des imprimantes partagées.

Si l'imprimante connaît le langage de description de page PostScript c'est sans doute le **pilote PostScript** qui donnera le plus de satisfaction aux utilisateurs, et le maximum de compatibilité avec des application Linux, Windows, ou Mac utilisant une imprimante partagée par Linux.

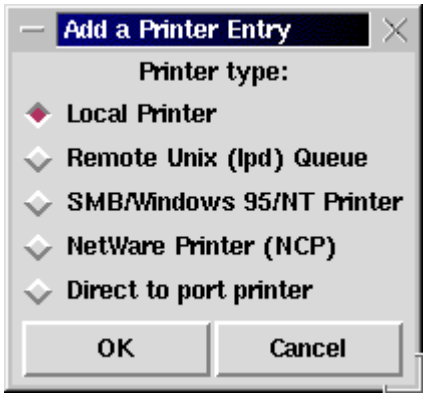
Pour compléter, voir sur Internet les articles suivants à propos de CUPS (mais pour une installation simple, utiliser l'outil de configuration des imprimantes intégré à la distribution que l'on utilise).

- http://www.lea-linux.org/cached/index/Admin-admin_imp-cups.html
- http://fr.wikipedia.org/wiki/Common_Unix_Printing_System#Red_Hat_Linux.2FFedora

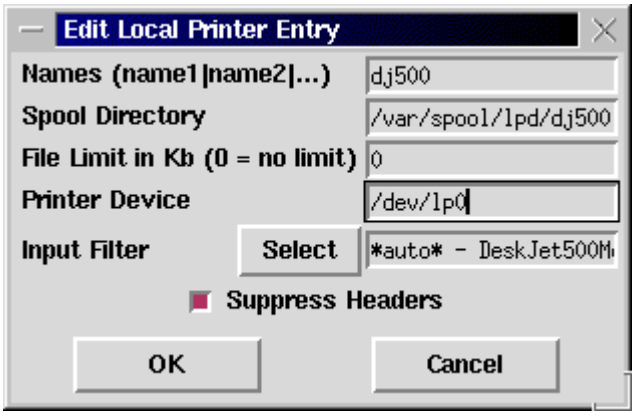
* Exemple (non CUPS) définition d'une imprimante sous Linux à l'aide de **printtool** :



L'imprimante pourra être locale ou distante...



Il sera nécessaire de disposer du "pilote" adéquat ...



C - Aide-mémoire des Commandes utilisateur (Linux)

Pour démarrer l'interface graphique : **startx**
Pour une fenêtre de connexion en mode graphique : **init 5** (en mode superviseur)

mc : Midnight Commander (Gestionnaire de Fichiers en mode texte)

Ctrl + Alt +Fn : Nouvelle Console**, Ctrl + Alt + F7 :** Retour à l'interface Graphique
._ **:** Fichier caché (tout fichier dont le nom commence par .)
. **:** Répertoire courant
.. **:** Répertoire parent
/ **:** Racine du système de fichiers
une seule racine et non une par unité comme sous Windows
~ **:** répertoire d'accueil de l'utilisateur

./configure : Exécute la commande "configure" située dans le répertoire courant.

En effet, pour des raisons de sécurité, le répertoire courant "**.**" *n'est pas implicitement dans PATH*, comme il l'est sous Windows : Un pirate aurait pu remplacer, à votre insu, une commande standard par une commande du même nom se trouvant dans le répertoire courant.

m + [touche TAB] **:** Recherche dans le répertoire courant les fichiers commençant par m
less fichier **:** affiche le fichier
joe fichier **:** édite le fichier
pico –r 256 fichier **:** édite le fichier (lignes de 256 colonnes sans coupure : rows)
gedit fichier & : éditeur graphique Gnome (lancé en arrière plan par &)
kedit fichier & : éditeur graphique KDE (lancé en arrière plan par &)

Obtenir de L'aide :
man : Aide (q pour quitter)
-h --help : Aide associée à une commande. Exemple : **cp --help** ou **cp -h**

Commandes de Base :

ls –alR /	**:** Affiche tous les fichiers d'un système
ls –alR / | grep doc	**:** Affiche tous les fichiers contenant doc
cd ..	**:** Remonte d'un niveau (change directory)
cd /	**:** Retourne à la racine
cd -	**:** Retourne au répertoire précédent
cd	**:** Retourne au répertoire utilisateur
rm	**:** Supprime un fichier (remove)
cp	**:** copie un fichier (copy)
mv	**:** Déplace un fichier , ou le renomme (move)
mkdir rep	**:** Crée le répertoire rep
mkdir –p rep1/rep2/rep3	**:** Crée un répertoire et ses sous répertoires associés
rmdir –p rep1/rep2/rep3	**:** Supprime le répertoire et ses sous répertoire associés
who ou **finger**	**:** liste des utilisateurs connectés
pwd	**:** Affiche le chemin d'accès au répertoire courant (print working directory)

Fin de session :

bye ou **exit** ou **^D**

Arrêt de la machine :

shutdown –h now
halt
poweroff

Redémarrage de la machine :

Ctrl + Alt + Suppr
shutdown –r now
reboot

Montage / Démontage :

mount –t iso9660 /dev/cdrom /mnt/cdrom
mount /mnt/cdrom (si paramétrage correct de /etc/fstab, ou cliquer sur l'icône cdrom du bureau kde)
umount /mnt/cdrom (ou cliquer avec le bouton droit sur l'icone cdrom du bureau kde et démonter)

mount –t msdos /dev/fd0 /mnt/floppy
mount /mnt/floppy (si paramétrage correct de /etc/fstab, ou cliquer sur l'icone disquette du bureau kde)
umount /mnt/floppy (ou cliquer avec le bouton droit sur l'icone disquette du bureau kde et démonter)

mount forum:/serveur /local (monte le volume réseau NFS nommé /**serveur** du serveur **Forum** sous le nom /**local**)

df, du : Utilisation disque
free : Mémoire libre et mémoire utilisée

"Recherche de fichier " :
find / -name "fichier.*" **:** Chercher les fichiers dont le nom est fichier.*

Processus :

& : à la suite d'une commande, lance cette commande en arrière plan, et affiche le n° (PID) du processus créé.

. : le . suivi d'un espace, précédant une commande script shell fait exécuter ce script par le processus shell courant, donc dans le même environnement, au lieu de créer un processus shell fils exécutant le script.

ps auxf : Liste des processus actifs
pstree : arborescence des processus
kill –9 PID : Tuer un processus de numéro PID
sous kde utiliser kpm ou ktop pour avoir la listre des processus et les tuer si nécessaire

Lancer l'interface Graphique :

startx : Lance l'interface graphique (recopier un fichier .Xclients correct sur ~ pour lancer kde)
kdm : Lance K Desktop Manager : demande de login en mode graphique (ceci est réalisé par **init 5**)

Utilisateurs / groupes et droits d'accès aux fichiers et répertoires :

su - **:** devenir super utilisateur (root)
su - demo **:** devenir l'utilisateur demo (en adoptant l'environnement initial de demo)
su - demo **:** devenir l'utilisateur demo (sans changer d'environnement)
passwd **:** Change le mot de passe de l'utilisateur (yppasswd si annuaire NIS)
useradd paul **:** Ajout d'un utilisateur
userdel paul **:** Supprime un utilisateur
groupadd etudiants **:** Ajoute un nouveau Groupe d'utilisateurs nommé étudiants
chgrp user /dev/hd* **:** attribue au groupe user tous les fichiers du répertoire /dev dont le nom commence par hd
groups **:** affiche les groupes auquel appartient l'utilisateur actif
groupmod **:** modifie les propriétés d'un groupe
groupdel **:** supprime un groupe

chown nobody /shared/book.tex : Changer le propriétaire du fichier **/shared/book.tex** en **nobody**.

chown -Rc jean.musique *.mid concerts/ : Donner la propriété de tous les fichiers dans le répertoire courant se terminant par **.mid** et de tous les fichiers et sous-répertoires du répertoire **concerts/** à **jean** et au groupe **musique**, en ne rapportant que les fichiers affectés par la commande.

Droit de lecture (r pour Read, « lire ») : Pour un fichier, cela autorise à en lire le contenu. Pour un répertoire, cela autorise à lister les fichiers contenus dans ce répertoire, si et seulement si le droit d'exécution sur ce répertoire est positionné également;

Droit d'écriture (w pour Write, « écrire »): Pour un fichier, cela autorise à en modifier le contenu. Pour un répertoire, cela autorise à créer des fichiers et à en effacer, même si l'on n'est pas le propriétaire de ces fichiers;

Droit d'exécution (x pour exécute, « exécuter »): Pour un fichier, cela en autorise l'exécution (par conséquent, seuls les fichiers exécutables ont normalement ce droit positionné). Pour un répertoire, cela autorise l'utilisateur à le traverser (c'est-à-dire de s'y rendre ou de se rendre dans l'un de ses répertoires fils).

```
$ ls -l

-rw-r-------   1 francis  users            0 jui  8 14:11 un_fichier
drwxr-xr--    2 gael     users         1024 jui  8 14:11 un_répertoire/
```

d indique un **répertoire**, **-** un **fichier ordinaire**, **l** un **lien** symbolique (raccourci), **b** ou **c** un driver...

- les trois premiers (r w -) sont les droits de l'utilisateur propriétaire de ce fichier, en l'occurrence francis. L'utilisateur francis a donc le droit de lire le fichier (r), de le modifier (w) mais pas de l'exécuter (-);
- les trois suivants (r - -) s'appliquent à tout utilisateur qui n'est pas francis mais qui appartient au groupe users: il pourra lire le fichier (r), mais ne pourra ni écrire dedans ni l'exécuter (--);
- les trois derniers (- - -) s'appliquent à tout utilisateur qui n'est pas francis et qui n'appartient pas au groupe users: un tel utilisateur n'a tout simplement aucun droit sur ce fichier.

chmod XXX fichier : Change les droits d'un fichier

chmod -c 644 divers/fichier* : Changer les droits d'accès de tous les fichiers du répertoire divers/ dont les noms commencent par fichiers en rw-r--r-- (droit d'accès en lecture pour tout le monde et droit d'accès en écriture pour le propriétaire du fichier seulement), et ne rapporter que les fichiers affectés par l'opération.

chmod -R o-w /shared/docs : Enlever de façon récursive le droit d'accès en écriture aux "autres" utilisateurs sur tous les fichiers et sous-répertoires du répertoire /shared/docs/.

chmod -R og-w,o-x prive/ : Enlever de façon récursive le droit d'accès en écriture pour le groupe et les autres sur tout le répertoire prive/, et retirer le droit d'accès en exécution pour les autres.

Réseau :

/sbin/ifconfig	Semblable à ipconfig sur windows, affiche les caractéristiques des cartes réseau
ping yahoo.fr	Demande d'écho au serveur Yahoo.fr
traceroute yahoo.fr	Recherche du chemin de routeurs **(** ou **tracepath yahoo.fr/80)**
/sbin/route	Affiche les routes IP (liste des routeurs traversés pour joindre le destinataire)
nslookup ou **host**	Interroge le serveur DNS (noms de domaine de l'Internet) (**nslookup -sil**)
netconf	Utilitaire de configuration Réseau de **Linuxconf**

Imprimantes :

lpr fichier : Imprime le contenu de fichier
lpq : affiche la queue d'impression
lpc : diverses commandes de contrôle des imprimantes et des files d'attente

sous Xwindow : utiliser **printtool**, **lprng** pour configurer les imprimantes

Samba client :

smbclient //serveur/repertoire –U utilisateur –I 192.168.1.1

Accède au répertoire partagé nommé **partage** d'un serveur SMB (Linux + samba ou Windows). Le nom Netbios du serveur est **serveur**.

mount -t smbfs -o username=tridge,password=foobar //fjall/test /data/test

Permet de « monter » sous le nom local **/data/test** le répertoire partégé test du serveur Windows (ou Samba) nommé **fjall**. Les droits d'accès sur le serveur **fjall** seront ceux de l'utilisateur **tridge**, identifié par le mot de passe **foobar**.

On peut ensuite utiliser diverses commandes (de type ftp) et en particulier les 2 suivantes :

- **get fichier : obtenir un fichier**
- **put fichier : mettre un fichier**

D - Bibliographie et aide en ligne

1 . Bibliographie

Editions Masson Sciences

Les systèmes d'exploitation, structure et concepts fondamentaux
de CL. Lhermitte

Conception du système UNIX
de M.J. Bach

UNIX, Système et environnement
de A.B. Fontaine et Ph Hammes

Le langage C
de B.W. Kernighan et D.M. Ritchie

UNIX, programmation avancée
de M.J. Rochkind

Editions "Interédition"

Les systèmes d'exploitation
de Tanenbaum

Le système UNIX
de Bourne

L'environnement de programmation UNIX
de Kernighan / Pike

Editions Sybex

Unix - Micro Référence
de Brent D. Heslop et David Angel

2 . Liens Internet

http://www.linux-france.org/article/debutant/debutant-linux.html : commandes fondamentales Linux

http://lea-linux.org/ : Beaucoup de documents en français dont **le Lea Book**

3 . Le Manuel (man) d'UNIX

man Unix en ligne sur Internet :

- http://man.linuxquestions.org/
- http://pwet.fr/man/linux
- http://ldsol.com/doc/man/manfr/man-html-0.9/

La documentation d'UNIX est divisée en 8 sections:

1) **Commandes usuelles**
2) **Appels système (fonctions du langage C)**
3) **Sous programmes**
4) **Format des fichiers systèmes**
5) **Macros et conventions**
6) **Jeux**
7) **Fichiers spéciaux**
8) **Maintenance**

Il est habituel, quand on cite une commande UNIX de la faire suivre du numéro de section où elle est documentée, par exemple: **ps (1).**
La documentation est accessible sous forme d'aide en ligne par la commande **man,** suivie du nom de la commande (ou du mot clé) pour laquelle on recherche des renseignements.

Exemples :

- man kill
- man **2** ls (on précise la section 2, listes des fonctions du langage C)
- man **-k** kill (toutes les commandes ayant rapport avec le mot clé, **keyword**, **kill**)
- voir **man man** pour une aide sur l'utilisation de man

Sous kde, l'**aide KDE** donne aussi accès au manuel présenté dans un format HTML. On peut aussi utiliser **xman** sur la plupart des systèmes supportant Xwindow.

Chaque sujet présenté est décrit dans un format standard comportant un certain nombre de paragraphes qui peuvent, pour une commande donnée, ne pas être tous présents.

Ces paragraphes sont les suivants :

Name	explication du nom de la commande (ou du mot clé).
Synopsis	syntaxe générale d'utilisation de la commande
Description	explique les fonctions réalisées par la commande
Files	donne la liste des fichiers systèmes concernés par cette commande (leur description se trouve en général dans la section 4)
examples	exemples d'utilisation de la commande
See also	renvoi à d'autres entrées dans la documentation
Exit status	valeurs du code de retour de la commande
Diagnostic	explication des messages d'erreurs, qui peuvent apparaître lors de l'exécution de la commande
Application usage	précise le contexte normal d'utilisation de la commande
Bugs	explique les anomalies de fonctionnement de la commande : voir par exemple ce qu'il est dit dans ce paragraphe à propos de la commande **cp**(1)

E - Commandes Windows et Linux. Comparaison

	Linux	**Windows**
désignation des volumes de stockage	**une seule arborescence: /** les autres volumes locaux ou distants, sont "montés" au niveau d'un sous répertoire déjà connu du système (par exmple /usr, /serveur … voir commande df)	Chaque volume, local ou distant est accessible à partir d'une **lettre d'unité** C:, D:, J: etc…
Désignation des répertoires	exemple : /etc/rc.d/init.d	exmple : d:\winnt\system
Chemins d'accès aux répertoires des commandes	echo $PATH /usr/local/bin : /bin : /usr/bin	echo %PATH% Z:. ; C:\WINDOWS\COMMAND ; Y:.
changer d'unité ou de répertoire actif	cd / cd ..	cd d:\ cd ..
créer un répertoire	mkdir ~/essai	md j:\essai
supprimer un répertoire	rmdir ~/essai	rd j:\essai
lister un répertoire	dir ~ ls -al ~	dir /w j:\ dir j:\
copier un fichier	cp ~/*.doc /tmp	copy j:*.doc d:\temp
obtenir de l'aide sur une commande	cp --help man cp	copy /? help copy
config réseau	/sbin/ifconfig	ipconfig
trace route IP	/usr/sbin/traceroute www **-I**	tracert www
interrogation DNS	nslookup -sil www host www	nslookup www
test de connexion IP	ping www	ping www
Redirection	ls -al ~ > /tmp/liste.txt ls -al ~ | **grep** "*.doc" |sort | more	dir j:\ > j:\temp\liste.txt ls -al ~ | **find** "*.doc" | sort | more