

UNIX

Utilisateur

Débuter sur Unix (Installation / arrêt / démarrage)

Installation d'un système Unix

Lequel choisir

2 cas se présentent :

- Possesseur d'une station de travail \Rightarrow demander un système Unix auprès du constructeur de la station, ou voir sur le Net s'il existe un Linux pour votre station.
- Possesseur d'un PC \Rightarrow C'est vous qui avez un problème car plusieurs types d'Unix sont disponibles !!!
 - ◆ SCO: Il commence à être supplanté par les autres. Quelques entreprises continuent à l'utiliser.
 - ◆ Solaris : L'UNIX de SUN, c'est un bon système avec lequel beaucoup d'entreprises travaillent.
 - ◆ BSD : Un Unix Freeware qui provient de l'université de *Berkley*. C'est une référence.
 - ◆ Linux : Un Unix en Freeware que tous les étudiants qui sortent de l'école doivent connaître. Il offre énormément de possibilités car tout le monde travaille pour le faire évoluer.

Les versions disponibles sont :

- ♣ Slackware : Ancêtre des linux, elle souffrait d'une installation difficile.
- ♣ Suse : Nos voisins Allemands ont conçus une bonne version bien robuste.
- ♣ Debian : Très intéressante version de Linux, bien fournie.
- ♣ Ubuntu : Basée sur la Debian, c'est la version qui semble devenir un standard.
- ♣ Red Hat : C'est la version la plus répandue.
- ♣ Mandrake : Basé sur la Red Hat, elle était plus simple à manipulée.

note 1 : Les versions se caractérisent par la diversité des packages (programmes) fournis. De base toutes ont : le noyau , le réseau TCP/IP , L'interface graphique X-window , et les outils GNU. Les packages supplémentaires peuvent être de type : interface graphique "look and feel" *KDE* , des émulateurs DOS , des broser WEB , etc.

note 2 : En fait quelque soit la version, Linux se compose d'un noyau et de packages que l'on ajoute. C'est ce que fournissent les éditeurs susnommés dans leurs CD. Ils ont rajoutés une interface d'installation. Lorsque l'on a un système Linux et un accès Internet rapide, il est plus intéressant de charger les packages par le net et de se construire son système "taillé sur mesure".

nota : Le système *Next* développé par l'équipe de *Steve Job* a été porté sur PC et doit être le futur des Mac. C'est à la base un Unix qui se nomme *Mach* et qui est issu de *Carneridge Melldown*. Sa propriété est d'être construit à base de micros-noyaux ce qui est un nouveaux concept des systèmes.

Où se le procurer

- Divers supports permettent d'installer Unix sur une machine.
 - ◆ Les disquettes (vieille manière qui n'a plus d'intérêt en 1997)
 - ◆ Les bandes (cartouche de streamer). Encore utilisé par certain pour diffuser des informations.
 - ◆ **Le CD-Rom**. C'est le média actuel qu'utilisent tous les éditeurs qui fournissent Unix.
 - ◆ Le réseau Internet. Il permet d'avoir la dernière version la plus à jour.

Les journaux fournissent des CD de démonstrations. Il peut être intéressant de tester les versions fournies dans le cas ou l'on souhaite acquérir par la suite une version complète. C'est le meilleur rapport qualité/prix pour un système. La liste (non exhaustive) est : Dream , 100% Linux , Info_PC , PC Expert , etc.

Nota : Depuis que Linux est "à la mode", les revues qui ne juraient que par Windows commencent à en parler, et de nouvelles se créent autour de ce thème ("business is business")

Installation

Le plus classique est un système fourni sur CD plus un fascicule, ou article, d'installation.

On parlera d'une installation de Linux sur un PC. Pour une station de travail voir le fournisseur.

Partitions

Nécessité

- Découpage logique et/ou physique des disques
 - ♦ Formatage du disque
 - Partition physique: notion de "tranche" disque (slice)
 - =
 - Partition logique : FS = *File System* ou non FS (swap)
 - allocation des blocs
 - Volumes logiques (JFS sous OSF/1)

Quantité de partitions

- Une partition pour le système : Elle contiendra le "noyau"
 - Une partition pour le swap (voir ci-dessous)
 - Une partition utilisateur
- Lorsqu'un système Unix est installé dans une entreprise il est souvent employé par plusieurs utilisateurs.
- ♦ Créer une partition qui isole les utilisateurs du noyau simplifie le travail de l'administrateur.
 - Il découple physiquement (sur 2 supports s'il le faut) le système et les utilisateurs.
 - La sauvegarde et l'archivage s'en trouvent simplifié.

nota : Ce découpage est le plus "sain" et le plus courant.

Il est possible d'avoir un regroupement des 2 partitions noyau et utilisateurs en 1 seule partitions. C'est ce qui est souvent fait lorsqu'une seule personne est sur un poste (station de travail), ou que le système est sur un PC pour faire une évaluation du système.

Nous conseillons pour débiter la configuration suivante : (les tailles de partitions sont à adapter suivant les systèmes utilisés).

Prenons un disque de 2 Go. Nous allons tailler 3 partitions :

- 1 DOS de \approx 1 Go
 - 1 Linux de \approx 1 Go
 - 1 swap de \approx 2 fois la taille mémoire (Unix exige une partition swap pour fonctionner)
- ex. 1 Go Dos + 1Go Linux + 50 Mo swap (on suppose qu'il y a 32 Mo de Ram)

Préparation des partitions

2 cas se présentent :

Vous avez déjà installé DOS sur tout le disque :

2 solutions :

- Vous supprimez tout et vous réinstallez en créant une partition DOS plus petite.
- Vous utilisez un re-partitionneur DOS de type Partition-Magic qui vous permettra de retailer votre partition , ou vous utilisez celui qui est fourni sur le CD des version Linux et qui s'appelle ***fips*** il se trouve dans la directory du CD ***dosutils***.

Attention fips ne sait que tasser les partitions il ne sait pas les agrandir, et si vous le laissez faire tout seul il va tasser au maximum et ne plus vous laisser de place pour DOS. A manier avec tact.

Vous avez un disque vierge : (il faut faire dans l'ordre)

- 1) créer une partition principale pour DOS , et l'installer complètement.

Il ne faut pas : - créer la partition DOS, - créer celles de Linux et l'installer - puis installer le DOS en dernier, car Windows 95/98 supprime les multiboots lorsqu'il s'installe.

- 2) Installer Linux : Lors de l'installation il va demander de créer ses partitions et à la fin il installera le multiboot

Manipulation des partitions

L'utilitaire `fdisk` permet de créer, supprimer, modifier, etc. les partitions. Il est appelé directement quand on installe le système sur un disque, mais il est possible de l'appeler lorsque le système Unix est en fonctionnement (seul l'administrateur en a le droit).

nota: Les utilisateurs de DOS et d'Unix auront tout avantage à utiliser le `fdisk` d'Unix en lieu et place de celui de DOS.

Attention : Ne pas confondre une partition "normale" et l'ersatz qu'est une partition étendue sous DOS.

Lorsqu'une partition étendue est créée, elle porte le n° 2 qui est physique, et elle réfère une autre partition, n° 5, qui elle est logique et qui peut être subdivisée. Détruire la partition étendue c'est détruire celle qui est physique.

Partition SWAP

- La partition swap sert :
 - A conserver les pages lors de basculement entre processus.
 - Lorsqu'un processus fait des demandes de mémoires physique et qu'il dépasse la taille limite de pages qui lui sont alloué.
- Sa taille est fonction de la mémoire physique présente.
 - pas assez \Rightarrow saturation du swap \Rightarrow le système devra échanger les pages du fichier directement.
 - trop \Rightarrow il existe un risque de perdre des pages dans le swap lors d'un arrêt brutal.
- ♦ La formule *empirique* de calcul de la taille du swap est :
 - \Rightarrow Pour de la RAM < 8 Mo taille de swap $\approx 2 * \text{la taille de la RAM}$
 - $\Rightarrow 8$ Mo $< \text{RAM} < 16$ Mo taille de swap $\approx 1,7 * \text{la taille de la RAM}$
 - $\Rightarrow 16$ Mo $< \text{RAM} < 32$ Mo taille de swap $\approx 1,3 * \text{la taille de la RAM}$
 - $\Rightarrow 32$ Mo $< \text{RAM}$ taille de swap $\approx 1 * \text{la taille de la RAM}$

Installation du swap

Lors de l'installation du système : créer une partition de swap en plus de la ou des partitions Unix.

Le cas particulier de Linux

Il est possible de travailler avec un fichier de swap et non une partition.

- Avantage
 - Lorsque Unix est déjà installé, nul besoin de refaire une partition.
 - s'il y a déjà 3 partitions existantes et qu'Unix prend la 4^{ième}, il est donc impossible d'avoir une 5^{ième} pour le swap.
- Inconvénients
 - Cela ralentit les échanges de pages car l'échange se fait entre mémoire et fichier.

Procédure à suivre pour créer un fichier de swap

- | | |
|--|---|
| 1) créer un fichier de swap | <code>dd if=/dev/zero of=/swap bs=1024 count=8000</code>
(la taille en page, donc en Ko \uparrow) |
| 2) initialiser le fichier comme un espace de swap | <code>mkswap /swap 8000</code>
(le même \uparrow nombre que ci-dessus, s.v.p.) |
| 3) sûr que le fichier est bien écrit sur le disque | <code>/etc/sync</code> |
| 4) "swapper" sur le fichier que l'on a créé | <code>swapon /swap</code> |

Détruire un fichier de swap

- | | |
|-------------------------|----------------------------|
| 1) supprimer le swap | <code>swapoff /swap</code> |
| 2) supprimer le fichier | <code>rm /swap</code> |

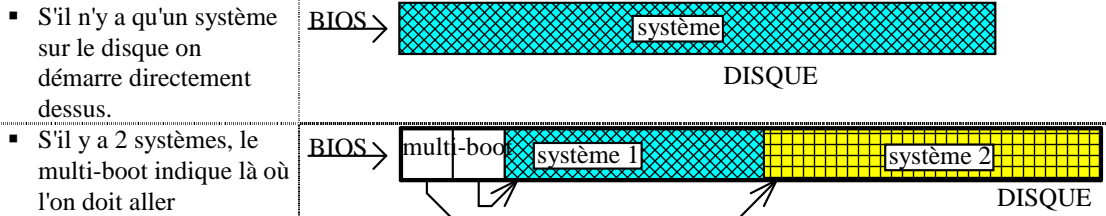
Démarrage

Lorsque l'on est sur une station de travail, le démarrage se fait sur Unix et l'on arrive souvent directement sur l'interface graphique X window.

Multi-boot

Lorsque l'on est sur un PC on a très souvent 2, voir 3, systèmes qui coexistent sur le disque. Il faut choisir lequel on désire au démarrage. C'est le rôle du multiboot.

Le BIOS fait démarrer le PC et va lire la premières pistes du disque. C'est à cet endroit qu'il y a indiqué où le système est placé sur le disque (sur quelle piste démarre le système).



Le multi-boot est un petit programme qui demande pendant un certain temps sur quel système on désire aller. On choisi sur lequel on veut travailler, sinon après une tempo il va de lui-même sur un système défini comme celui où il doit aller par défaut.

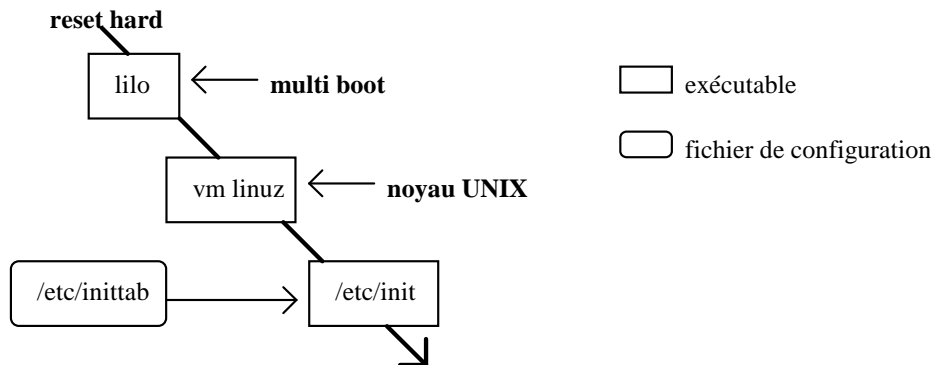
Lorsque l'on a installé Linux, le multi-boot qui s'est installé est **lilo**. En général il affiche après le BIOS :

LILO boot :

- S'il n'a pas de réponse il va démarrer sur le système qu'il a par défaut.
- Pour lui dire lequel on désire, il faut lui donner un nom de système suivi de ↵.
- Si l'on ne connaît pas les noms des systèmes installés taper sur la touche tabulation et la liste des noms s'affichera, puis il y aura à nouveau l'invite **boot :**.

nota : par défaut à l'installation, et s'il y a un système DOS déjà installé, Linux propose 2 noms qui sont *dos* et *linux* (attention aux minuscules !!!). Cela nous semble de bons noms, qui sont simples et sans ambiguïté, il paraît souhaitable de les conservés.

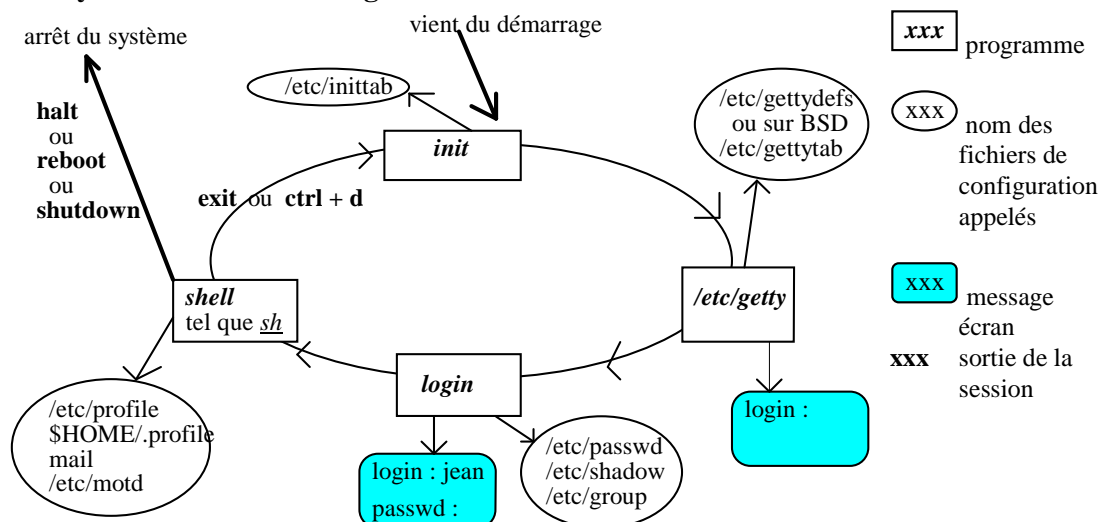
Démarrage d'un système Unix (ex. Linux).



Lorsqu'un système Unix démarre il charge en mémoire une partie de son noyau (sous Linux c'est le fichier vmlinuz qui est chargé). Une fois que cela est fait il exécute la 1^{ère} tâche parente de toutes les autres qui se nomme *init*. *init* est chargée d'exécuter le script de démarrage qui est contenu dans le fichier *inittab*.

nota : Un exemple du fichier *inittab* sur un système Linux Mandrake 5.1 est donné en annexe. Ce fichier appelle le fichier *rc.sysinit* qui configure le système. Ce fichier est lui aussi est donné en annexe.

Cycle de la session de login



Pour travailler avec Unix il faut se connecter, c.a.d. se faire reconnaître par le système. Le moyen utiliser pour cela est le login. Il consiste à rentrer son *nom de login* et son *mot de passe* qui sont donnés par l'administrateur. Il existe un *nom de login* par défaut qui est celui de l'administrateur, ce nom est **root**, tous les autres noms ont été créés par l'administrateur. Les mots de passes sont créés par les utilisateurs eux-mêmes. L'administrateur ne peut pas les retrouver, il peut seulement les supprimer.

La séquence de travail sur Unix est :

- 1) à la demande **login** : donner son nom de login
- 2) s'il y a une demande **password** : donner son mot de passe (on peut ne pas avoir de mot de passe)
- 3) travailler sur le système grâce aux commandes shell
- 4) arrêter sa session par un **exit** ou un **Ctrl + d** (ou arrêter le système si on est administrateur)

Nom et mot de passe

- Nom de login
 - ♦ Il permet de positionner l'utilisateur dans sa directory et de lui donner des droits.
 - ♦ C'est le moyen pour le système de rajouter un utilisateur dans la liste des utilisateurs connectés.
 - ♦ Si on est l'administrateur le nom de login est **root**, sinon l'administrateur nous en a donné un
- Mot de passe
 - ♦ Il permet au système de vérifier si l'utilisateur est bien celui qui demande la connexion.
 - ♦ En cas d'erreur le système redemande le nom et le mot de passe. (Certains systèmes sécurisés autorisent 3 erreurs, au-delà le poste est déconnecté et seul l'administrateur peut le reconnecter. Ceci afin de protéger le système d'accès non autorisés)
 - ♦ Si on est l'administrateur le mot de passe est celui de l'installation ou celui que l'on a refait, sinon c'est le mot de passe que l'on a donné à la 1^{ère} connexion.

nota : Lors de l'installation d'un Unix, "l'installateur", considéré comme l'administrateur, doit donner un mot de passe. Pour simplifier les opérations suivantes il est conseillé de donner **6 espaces** comme mot de passe. Il sera toujours possible de le changer par la suite mais si on n'a pas le bon mot de passe on ne peut pas rentrer dans le système.

Arrêt d'un système Unix (ex. Linux)

Seul l'administrateur à le droit d'arrêter le système Unix.

- **sync** : mise à jour du système de fichiers sur le disque et termine les accès disque qui sont en attente.

La commande *sync* appelle le *daemon* /etc/update.

- ♦ La commande doit toujours être faite 2 fois afin d'assurer une mise à jour complète.

sync ; sync

ou

sync
sync

- ordre d'arrêt :

⇒ **shutdown <paramètre> <temps>**

- ♦ paramètre : **-f** = fast shutdown ; **-r** = reboot ; **-q** = sans affichage de message ; **-h** = halt

- ♦ temps : **<heure> : <minute>** ; (si on veut tout de suite ⇔ **now** = immédiatement)

⇒ autres commandes : **halt** = arrêt rapide
reboot ≡ shutdown -r now
fasthalt ≡ shutdown -f now

nota 1 : Pour Linux les touches Ctrl + Alt + Del font office de reboot.

nota 2 : Il est préférable d'utiliser *shutdown -h now* qui termine proprement le système.

Présentation du système Unix

Caractéristiques du système.

C'est un système multi utilisateurs et multi tâches.

Unix est surtout un système de développement et propose beaucoup d'utilitaires puissants et standards.

Composantes d'UNIX

- Le noyau
 - ◆ Gestion mémoire
 - ◆ Entrées / Sorties de "bas niveau"
 - ◆ Enchaînement des tâches
- Un (ou plusieurs) interpréteur(s) de commandes
 - ◆ Bourne-shell ou shell
 - ◆ C-shell
 - ◆ Korn-shell
 - ◆ autres (shell constructeur)
- De nombreux outils
 - ◆ Compilateur C
 - ◆ Editeurs
 - ◆ Traitement de texte
 - ◆ Analyseurs ou "filtres"
 - ◆ Messagerie (directe et différée)
 - ◆ etc...
- Logiciels de communication avec d'autres systèmes UNIX
 - ◆ Réseau TCP/IP et protocole Internet
 - ◆ Système de fichiers distribués NFS
 - ◆ Réseau d'information de service NIS
 - ◆ Serveur de Nom DNS
 - ◆ UUCP
- Des interfaces graphiques
 - ◆ X-window/Motif pour la plupart¹

Les caractéristiques

- Système de fichiers hiérarchisé
- Réalisation des Entrées / Sorties des processus (redirections, mécanisme du "pipe")
- Les langages de commandes (shells) sont aussi des langages de programmation
- "Appels systèmes" depuis le langage C
- Taille réduite du noyau (portable et efficace)
- Création aisée de nouvelles commandes (en shell ou en C)
- Indépendance vis-à-vis des constructeurs (pérennité des solutions informatiques)
- Connexion "aisée" de nouveaux périphériques.

Les points faibles

- Manque de convivialité, aspect parfois "rustique" (cela tend à disparaître).
- Rigueur nécessaire dans l'administration du système
- Absence de méthode d'accès aux fichiers (rien n'est géré par le noyau, tout doit être fait par programme)
- Offre logicielle très abondante mais souvent mal référencée (problème du choix d'un outil sous UNIX)
- Manque de compétences : spécialistes UNIX, SHELL et langage C encore relativement peu nombreux (bien qu'elles soient en constante progression avec les nouvelles générations qui sortent de l'école)
- absence de fonctionnalités "temps réel".

¹ Sun possède une interface propriétaire sur X-window, *Open-View*, qui tend à disparaître au profit de Motif. Sur Linux KDE et GNOME sont les interfaces les plus courantes.

L'administrateur système

□ Qualités requises

- Rigueur, sens de l'organisation (méthode)
- Sens du dialogue, disponibilité
- Sens des responsabilités
- Adaptabilité, patience...

□ Son rôle

➤ Documenter le parc (cahier de bord)

- Schéma du parc (parc homogène, hétérogène)
- Configurations systèmes (réseau, listing fichiers systèmes)
- Schéma de répartition des ressources :
 - ◆ espace disque
 - ◆ CPU
 - ◆ périphériques de sauvegardes
 - ◆ imprimantes
 - ◆ applications
 - ◆ comptes utilisateurs
- Information sur l'emplacement des documentations
- Information sur l'emplacement des supports magnétiques
- Liste des mots de passe "root"

➤ Gérer le matériel

- Matériels informatiques
- Fournitures

➤ Administratif

- Négociations avec les fournisseurs :
 - ◆ commandes/achats
 - ◆ contrats de maintenance
- Contrôle des commandes à la livraison

➤ Technique

- Aide aux choix techniques (analyse des besoins)
- Installations/configurations du matériel
- Maintenance :
 - ◆ Suivi et/ou réalisation des interventions de maintenance

➤ Gérer les données

- Logiciels
- Données utilisateurs et systèmes

➤ Administratif

- Négociations avec les fournisseurs :
 - ◆ Commandes de logiciels
 - ◆ Contrats de maintenance
- Contrôle des commandes à la livraison
- Suivi des mises à jour des logiciels

➤ Technique

- Aide aux choix techniques (analyse des besoins)
- Installations des logiciels
- Sauvegardes périodiques

➤ Gérer les utilisateurs

- Ajout/suppression des comptes

➤ Assistance

- Dialogue avec les utilisateurs (hot-line)
- Dépannage
- Rédaction de notices explicatives

➤ Surveiller les systèmes: tableau de bord

- Occupation espace disque et état physique des disques
- Performances (usage CPU = exécution des processus)
- Charge réseau
- Sécurité

□ Comment administrer ?

➤ Savoir récupérer la bonne information au bon moment

- "man" Unix
- Documentations "papier" et "CD" hypertexte
- Usenet et Internet

➤ Garder trace des moindres faits et gestes(cahier de bord)

- Rédaction systématique et détaillée des procédures d'installation
- Datation des interventions pour conservation de l'historique

➤ Planifier le "planifiable"...

- Gestion des priorités
- Estimation du temps à consacrer à chaque tâche

➤ Se former aux évolutions techniques

- autant que possible...

L'éditeur de texte vi

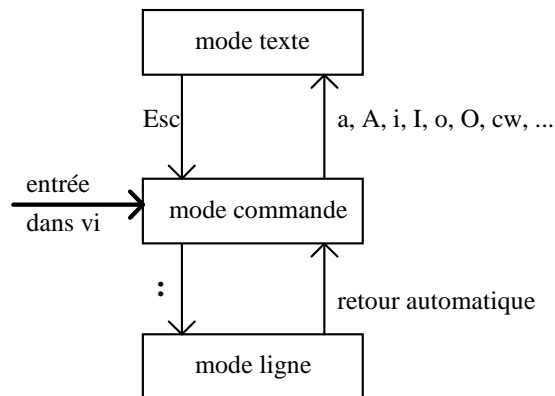
L'éditeur livré par défaut sur tous les Unix se nomme *vi*. C'est le 1^{ier} éditeur plein écran qui fut créé pour Unix. Il offre beaucoup de fonctions mais comme il est ancien il possède une interface que l'on a perdu l'habitude d'utiliser.

Le plus gros problème des utilisateurs est de se faire à son interface assez fruste. C'est l'éditeur des développeurs, il est fait pour, et par, eux.

vi possède 3 modes de fonctionnement.

- le mode commande
- le mode insertion de texte
- le mode ligne (ou expression)

On passe d'un mode à l'autre en tapant sur des touches spécifiques. *vi* étant un éditeur de texte qui devait fonctionner sur tous les types de consoles, dont certaines n'avaient pas de touche de fonction, les commandes sont faites par des touches caractères, ce qui déconcerte le débutant étant habitué à taper du texte directement.



- Appel de l'éditeur

On édite un fichier en tapant : **vi** <nom du fichier texte>

- Sortie de l'éditeur

On peut sortir de 2 manières :

- ♣ sauvegarder le texte : **ZZ**
 - :w** pour sauvegarder les données
 - :w N** pour sauvegarder les données dans un fichier de nom N
 - :q** pour quitter
 - :wq** pour écrire et quitter
 - :x** idem :wq
- ♣ ne pas sauvegarder : **:q!**

Commandes

Les commandes dans les lignes grisées sont très souvent utilisées

^ = Ctrl

◆ Insertion

caractère	description	mnémonique
a	ajout après le curseur	add
A	ajout à la fin de la ligne	"
i	ajout avant le curseur	insert
I	ajout en début de ligne	"
o	ajoute une ligne après le curseur	
O	ajoute une ligne avant le curseur	

◆ déplacement des lignes à l'écran

→ ou l	déplacement d'1 caractère à droite	
← ou h	déplacement d'1 caractère à gauche	
↑ ou k	déplacement d'1 ligne vers le haut	
↓ ou j	déplacement d'1 ligne vers le bas	
+	positionnement au début de la ligne suivante	
-	positionnement au début de la ligne précédente	
^d	descend d'1/2 écran	
^u	monte d'1/2 écran	
^f	déplace d'1 page vers la fin du fichier	
^b	déplace d'1 page vers le début du fichier	
z	met le texte de manière à ce que le curseur soit au milieu de l'écran	

◆ déplacement dans le texte

\$	va à la fin de la ligne	
0 ou ^	va au début de la ligne	
w	avance de 1 mots	word
λw	avance de λ mots ex. 3w avance de 3 mots	
b	recule de 1 mots	back
λb	recule de λ mots ex. 3b recule de 3 mots	
e	à la fin du mots	end
:λ	va à la ligne λ ex. :31 va à la ligne 31 (<i>fréquent en compil</i>)	

◆ remplacement de texte

cw	modifie un mot (arrêt avec <i>Esc</i>)	change word
λcw	modifie λ mots ex. 3cw modifie les 3 mots suivants	"
C ou c\$	supprime et remplace du curseur à la fin de ligne (arrêt avec <i>Esc</i>)	"
r	remplace 1 caractère saisi au clavier	replace
R	remplace jusqu'au prochain <i>Esc</i>	"

◆ recherche de texte

/chaîne	recherche de chaîne vers la fin du texte	
?chaîne	recherche de chaîne vers le début du texte	
n	recherche l'occurrence de chaîne suivante	
N	recherche l'occurrence de chaîne précédente	

♦ suppression de texte

x	supprime un caractère	
λx	supprime λ caractères ex. $3x$ supprime 3 caractères	
dw	supprime un mot	delete word
λdw	supprime λ mots ex. $3dw$ supprime 3 mots	"
d\$ ou D	supprime jusqu'à la fin de ligne	
d0 (zéro)	supprime du début de ligne jusqu'au curseur	
dd	supprime 1 ligne	
λdd	supprime λ lignes ex. $3dd$ supprime 3 lignes	
J	concatène 2 lignes (suppression du retour chariot)	

♦ copie de texte

yw	copie un mot	yank word
λyw	copie λ mots ex. $3yw$ copie 3 mots	"
yy	copie 1 ligne	yank yank
λyy	copie λ lignes ex. $3yy$ copie 3 lignes	"

♦ coller le texte copié

p	après le curseur	put
P	avant le curseur	"

♦ défaire le résultat d'une commande

u		undo
----------	--	------

♦ permuter 2 caractères

^		
----------	--	--

♦ permutation Majuscule \leftrightarrow minuscule

~		
----------	--	--

Ce dont il faut absolument se souvenir comme commande ou radical de commande:

commande		radicale	
a, i, o, O	insérer du texte	\$	la fin de ligne
ZZ ou :x	sortir	0	le début de ligne
r	remplacer 1 caractère	d	supprimer une ligne ou un mot
x	supprimer 1 caractère	y	copier
/token	recherche de token	p	coller
:n°	aller à la ligne n°	w	un mot

Ligne ou expressions

Il faut être en mode commande pour aller en mode expression.

- On rentre en mode expression par un `:`.
- Dès que l'on a taper le `:` la dernière ligne sur l'écran s'affiche avec ce `:` l'expression frapper au clavier s'inscrira à la suite.
- On la valide par un retour chariot `↵`, ce qui l'exécute (si elle est correcte), nous fait automatiquement sortir du mode expression, et revenir en mode commande.

♦ fin d'édition

expression	description
<code>:wq</code>	écrit le fichier sur disque et sort
<code>:x</code>	idem <code>:wq</code>
<code>:q</code>	quitte sans sauvegarder le texte
<code>:q!</code>	quitte <i>absolument</i> sans sauvegarder le texte

♦ commande d'édition

<code>:e <nom></code>	édite un nouveau fichier <nom>
<code>:w</code>	sauvegarde le texte sur le disque
<code>:w <nom></code>	sauvegarde le texte sur le disque dans le fichier nommé <nom>
<code>:r <nom></code>	ajoute le fichier <nom> après le curseur
<code>:! <cmd></code>	exécute la commande unix <i>cmd</i> et retourne à vi
<code>:r! <cmd></code>	exécute la commande unix <i>cmd</i> et insertion de son résultat après la ligne courante
<code>:λ</code>	positionne le curseur à la ligne λ

♦ substitution de texte

<code>:λ,γs/old/new/</code>	de la ligne λ à la ligne γ change le 1 ^{er} <i>old</i> par <i>new</i>
<code>:λ,γs/old/new/g</code>	de la ligne λ à la ligne γ change tous les <i>old</i> de la ligne par <i>new</i>

nota : pour λ et γ certains caractères ont une signification particulière

I début de texte , *\$* fin de texte , *.* ligne courante , *%* remplace 1,\$

♦ déplacement de texte

<code>:λ,γm/n°_de_ligne</code>	déplace le bloc de lignes de λ à γ en position n°_de_ligne
ex. <code>:30m/50</code>	déplace la ligne 30 en 50

♦ suppression de ligne

<code>:λ,γd</code>	supprime de la ligne λ à la ligne γ
ex. <code>:30,50d</code>	supprime de la ligne 30 à 50

♦ commande globale

<code>:λ,γg/chaîne/commande</code>	de la ligne λ à la ligne γ si on trouve chaîne on exécute la commande
ex. <code>:20,30g/toto/d</code>	supprime entre les lignes 20 et 30 toutes celles qui contiennent <i>toto</i>

♦ commande de personnalisation

<code>:set</code>	affiche tous les paramètres positionnés
<code>:set all</code>	affiche tous les paramètres possibles et leurs valeurs actuelles
<code>:set var [= valeur]</code>	positionne le paramètre <i>var</i> à la valeur <i>value</i>
<code>:set novar</code>	supprime une initialisation

ex. `:set nu` affiche les n° de ligne devant chaque ligne

`:set nonu` enlève les n° de ligne devant chaque ligne

♦ les abréviations

On abrège une chaîne de caractère en une chaîne restreinte. On pourra alors, en mode insertion, taper l'abréviation pour avoir la chaîne complète.

:ab <restreinte> <longue>	abréviations d'une chaîne longue de caractère en une chaîne restreinte
:unab <restreinte>	supprime l'abréviations

ex. :ab sys système abréviation de système par sys

♦ les macros

Cela permet de simplifier les commandes et de programmer les touches de fonctions.

:map caractère commande	le caractère remplacera la commande
:unmap caractères	supprime la macro

nota : pour les touches de fonctions la séquence est :

Ctrl + v suivie de la touche de fonction (cela se traduira par ^[[λ~ , avec λ = nombre)

Ctrl + v suivi de ↵ = ^M (cela indique un retour chariot)

ex. :map ^[[20~ :set nu^M programme la touche F9 pour afficher les numéros de lignes

☞ : Si on ne termine pas les macros par ^M, elles ne seront exécutées que quand on aura fait un ↵ après leurs appel. Alors que si on les termine par ^M, les appeler les fera s'exécuter directement.

Personalisation permanente de vi

Le fichier **.exrc** situé dans le repertoire d'accueil (home directory), permet de personnaliser vi. On peut y placer les personnalisations permanentes , les abréviation, et les macros les plus usuelles, etc.

ex.

le fichier .exrc	commentaire
:set ai	auto-indentation
:set tabstop=5	les tabulations seront sur 5 caractères
:set nonu	pas de numéro de ligne
:ab sys système	abréviation de système en sys
:map ^[[20~ :set nu^M	F9 permet de placer les numéros de ligne
:map ^[[20~ :set nonu^M	F10 permet de les enlever

Les commandes les plus utiles

♦ les caractères

ajouter d'1 caractère avant le curseur	i
après le curseur	a
le changer	r
le supprimer	x
le permuter avec celui d'avant	^

♦ les mots

se déplacer de 1 mot à droite	w
se déplacer de λ mot à droite	λ w
se déplacer de 1 mot à gauche	b
se déplacer de λ mot à gauche	λ b
changer 1 mot (Esc quand on a fini)	cw
le supprimer	dw
le copier (p ou P pour le placer avant ou après le curseur)	yw
les copier (p ou P pour les placer avant ou après le curseur)	λ yw

♦ les lignes

changer du curseur à la fin de la ligne (Esc quand on a fini)	C ou c\$
supprimer 1 ligne	dd
supprimer λ lignes	λ dd
supprimer du curseur au début de la ligne	d0 (zéro)
supprimer du curseur à la fin de la ligne	d\$
copier 1 ligne (p ou P pour le placer avant ou après le curseur)	yy
copier λ lignes (p ou P pour les placer avant ou après le curseur)	λ yy
concaténer 2 ligne	J

♦ déplacer le curseur

de 1 ligne en bas	↓ ou j
de 1 ligne en haut	↑ ou k
de 1 caractère à gauche	← ou h
de 1 caractère à droite	→ ou l
de 1 mot à droite	w
de λ mot à droite	λ w
de 1 mot à gauche	b
de λ mot à gauche	λ b
à la fin de la ligne	\$
au début de la ligne	0 (zéro)
à la fin du texte	:\$
au début du texte	:0 (zéro)

Le système de fichiers Unix

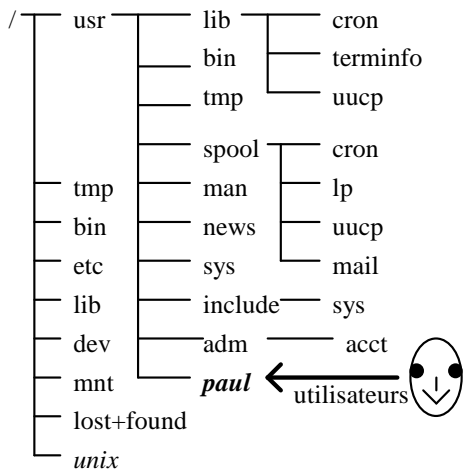
Organisation arborescente

Notion de systèmes de fichiers (File System=FS)

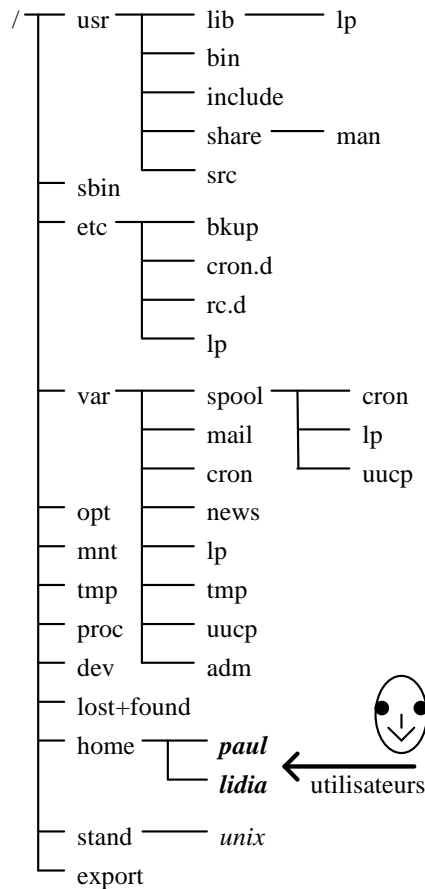
- Un système de fichier représente un espace disque correspondant à tout ou partie (partition) d'un disque physique.
- Il est initialisé par la commande `mkfs`, puis attaché à un répertoire du système de fichiers principal (désigné par /) par la commande `mount`.
- Il possède une organisation arborescente composée de répertoires et de fichiers, dont la structure varie selon la version Unix utilisée.

Arborescence pré-SVR4

Elle a été adoptée jusqu'aux versions BSD 4.3 et SYS V R3.

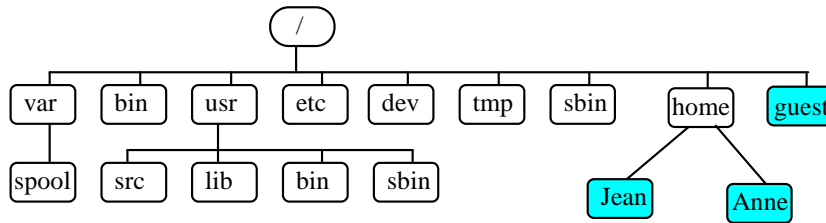


Arborescence SVR4



Arborescence de Linux

- Les principaux répertoires et leurs usages



/	racine	tmp	répertoire temporaire
dev	les fichiers de périphériques	etc	pour l'administration du système (<i>passwd</i> , <i>group</i> , <i>inittab</i> , etc.)
bin	utilisés par tous (<i>ls</i> , <i>rm</i> , <i>mv</i> , etc.)	sbin	gestion du système (<i>adduser</i> , etc.)
usr	tous les utilitaires de root	home	le début des répertoires utilisateurs
usr/bin	commandes les moins utilisées (communication entre utilisateurs)	guest	répertoire pour les usagés qui se connectent de l'extérieur par le réseau
usr/man	le manuel en ligne	var/spool	contient des fichiers à imprimer
usr/spool	gestion de l'imprimante	var	fichiers de "variables" (<i>mail</i> , <i>log</i> , etc.)

Les types de fichiers

Types de fichiers Unix et création

Type de fichier		Symbole (ls)	Créé par
ordinaire (texte ou binaire)		-	vi (editeurs), cp, cat, touch, cc, ld, ...
répertoire		d	mkdir
fichiers spéciaux	périphérique mode blocs	b	mknod
	périphérique mode caractères	c	mknod
	FIFO (tubes nommés)	p	mknod
	socket	s	appel fonction système (socket(2))
Liens (symboliques et physiques)		l	ln

n.b.: Tous ces fichiers peuvent être supprimés par la commande **rm**, y compris les répertoires (il faudra préciser l'option "-r" pour ces derniers).

Un répertoire VIDE peut être supprimé par la commande **rmdir**.

Les liens

Certains fichiers/répertoires sont des liens. Ce sont des redirections vers d'autres fichiers/répertoires. On peut voir un lien comme un alias.

- La commande pour créer un lien est `ln <nom> <cible>` où nom est le nom de l'alias que l'on créé et cible le nom du fichier sur lequel on créé le lien.
- On ne peut créer de liens que sur les fichiers appartenant au disque, et pas sur des médias temporaires tel que : disquette, réseau, etc.
- Cela induit que la structure des fichiers d'Unix n'est plus un arbre mais un D.A.G.² (il existe des raccourcis pour aller d'un point à l'autre de l'arbre). Cette technique de liens complique singulièrement le système pour la gestion des fichiers, mais facilite le travail de l'utilisateur.

² Diagramme Acyclic Graphe. C'est un arbre qui contient des liens entre ses éléments en dehors des attaches normales de parent/enfants

Détermination du type

- La commande `ls -l` renseigne sur les caractéristiques d'un fichier.
 - Le premier caractère de la ligne décrit le type du fichier.
 - Sept types possibles: - **d l b c p s** (Cf. tableau ci-dessous).

-	rw	----	1	pierre	progteam	27684	Feb 29	10:42	/home/pierre/main.c
↑	↑		↑	↑	↑	↑	↑	↑	↑
1	2		3	4	5	6	7	8	9

- | | |
|---|--|
| 1- Type du fichier | 6- Taille en octets |
| 2- Permissions du fichier | 7- Date de dernières modifications |
| 3- Nombre de liens physiques | 8- Heure de dernières modifications (ou année) |
| 4- Nom de l'utilisateur (propriétaire du fichier) | 9- Nom du fichier |
| 5- Nom du groupe d'appartenance | |

- Les permissions du fichier se décomposent en :

	proprio	groupe	autre
permissions symboliques	rw x	-- x	r - x
valeur binaire	111	001	101
valeur octale	7	1	5

- La commande `chmod <valeur> <fichier>` permet de changer les permissions du fichier. Seul le propriétaire ou l'administrateur ont la possibilité de le faire.
- La commande `chgrp <valeur> <fichier>` permet de changer les permissions de groupe du fichier. Seul le propriétaire ou l'administrateur ont la possibilité de le faire.
- La commande : `file <nom>`, lit les deux premiers octets du descripteur de fichier où est inscrit le nombre magique et les compare avec les chaînes de caractères contenues dans le fichier `/etc/magic`. La correspondance est alors effectuée entre ce "nombre magique" et le type de fichier associé. La fiabilité de la commande dépend de l'exhaustivité du contenu de `/etc/magic`. On a ainsi un affinement (par comparaison avec le résultat de la commande `ls -l`) sur la détermination du type de fichier.

Exemple de fichier `/etc/magic`:

0	short	070707	cpio archive
0	short	0143561	byte-swapped cpio archive
0	string	070707	ASCII cpio archive
0	long	0177555	very old archive
0	short	0177545	old archive
0	long	0100554	APL workspace (Ken's original?)
0	long	0101555	PDP-11 single precision APL workspace
0	long	0101554	PDP-11 double precision APL workspace
0	long	0101557	VAX single precision APL workspace
0	long	0101556	VAX double precision APL workspace
0	short	017437	old packed data
0	string	\037\036	packed data
0	string	\377\037	compacted data

- La commande `find <paramètre>` permet rechercher un fichier dans l'arborescence du FS.
- ◊ Les paramètres sont organisés en 2 parties :
 - le répertoire de départ :
 - • pour le répertoire courant
 - / pour la racine du FS
 - /n₁/n₂/.../n_i pour le répertoire n_i
 - une expression qui peut être très compliquée, en voici quelques ex.
 - recherche le fichier main.c à partir de la racine. C'est la forme la plus simple.
`find / -name main.c -print`
 - recherche le fichier main.c à partir /usr et /etc
`find /etc /usr -name try.c -print`
 - changer les permissions de tout les fichiers qui sont sous /home/paul
`find /home/paul -type f -exec chmod u=rwx,g=rx,o=rx {} \;` (ne pas oublier le ;)
 - afficher tous les répertoires dont la taille est supérieure à 10 Ko
`find / -type d -size +10 -print`
 - afficher tous les fichiers du répertoires /home/paul dont la taille est supérieure à 1 Mo
`find /home/paul -type f -size +1000 -print`

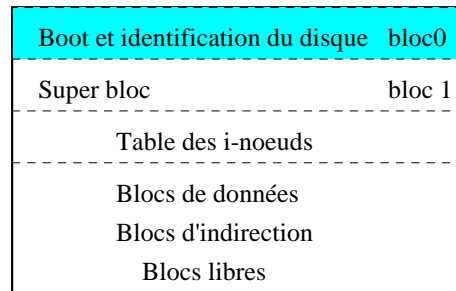
Extensions courantes

domaine	extension	type	contenu du fichier
LANGAGES	.s	<i>texte</i>	assembleur
	.c	<i>t</i>	C
	.cc .c++ .C	<i>t</i>	C++
	.h	<i>t</i>	header de C ou C++
	.f	<i>t</i>	Fortran
	.e	<i>t</i>	Elf
	.p	<i>t</i>	Pascal
	.el .elc	<i>t</i>	Lisp Lisp compilé
	.l	<i>t</i>	Lex
	.y	<i>t</i>	Yacc
	.sh .csh .ksh	<i>t</i>	Bourne Shell C-Shell Korn Shell
DOCUMENTATION	.o	<i>binaire</i>	binaire Objet
	.a	<i>b</i>	librairie de binaires objets
	.1 à .8	<i>t</i>	Nroff (pages du manuel)
	.ms .me .mm	<i>t</i>	Nroff et macros <i>ms</i> et macros <i>me</i> et macros <i>mm</i>
	.ps .epsf	<i>t</i>	PostScript PostScript en capsulé
DIVERS	.tex	<i>t</i>	TeX, LaTeX ou SliTeX
	.C .z .Z .gz	<i>b</i>	fichier compressé par <i>compress</i> , <i>zip</i> , <i>gzip</i>
	.tar	<i>t/b</i>	fichier archive <i>tar</i>
	.xwd	<i>b</i>	dump X-Windows <i>image</i>
	.gif	<i>b</i>	image format GIF
	.jpg	<i>b</i>	image format JPEG

n.b.: Les fichiers exécutables n'ont en général pas d'extension.

Le contenu des fichiers de type "texte" est visualisable à l'écran et éditable.

Structure d'un FS



(Ce schéma est une simplification du vrai F.S. d'Unix).

- Le Boot est situé au début de la partition Unix et sert à charger le système (Ne pas confondre avec le multi-boot qui est en piste 0 et qui sert à indiquer quel système l'on choisi lorsque l'on démarre).
- Le super bloc contient les informations suivantes :
 - la date de dernière mise à jour du système de fichiers
 - la taille de la table des i-noeuds (le nombre de fichiers stockable par le système de fichiers)
 - la taille du système de fichiers
 - un pointeur sur la liste des blocs libres
 - l'indication read-only et d'autres informations
- la table des i-noeuds est constituée de blocs dont le nombre est déterminé au moment de l'initialisation du système de fichiers par la commande *mkfs*. un i-noeud caractérise un fichier , donc le nombre maximum de fichiers est égal au nombre d'i-noeuds. Un i-noeud contient les informations suivantes :
 - le type de fichiers
 - le nombre de liens
 - l'*uid* du propriétaire
 - le *gid* du propriétaire
 - la taille en octet du fichier
 - l'adresse physique des blocs (13 adresses)
 - la date du dernier accès
 - la date de la dernière modification
 - la date de création

Les informations sur les i-noeuds sont obtenu par les commandes *ls -il* , ou la commande *stat*, qui est aussi un appel système dans un programme.

Un i-noeud contient 13 adresses de blocs de données, les 10 premières pointent directement, la 11^{ième} pointe sur un bloc contenant 256 adresses, la 12^{ième} a 2 indirections du même type que la 11^{ième}, et la dernière en a 3.

Si on considère qu'un bloc contient 1024 octets on a :

les 10 premiers blocs $\Rightarrow 10 \times 1\text{Ko} = 10\text{ Ko}$

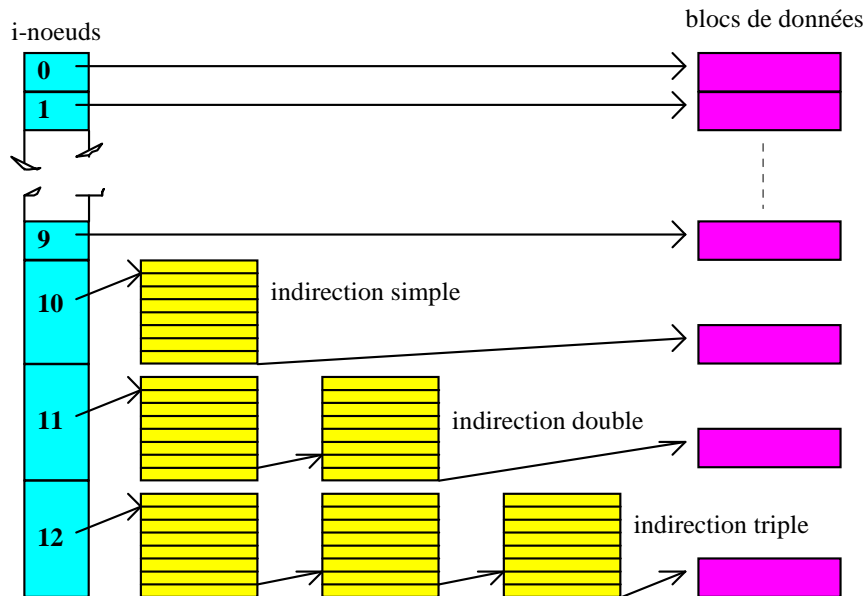
le 11^{ième} $\Rightarrow 256 \times 1\text{ Ko} = 256\text{ Ko}$

le 12^{ième} $\Rightarrow (256)^2 \times 1\text{Ko} = 64\text{ Mo}$

le 13^{ième} $\Rightarrow (256)^3 \times 1\text{ Ko} = 16\text{ Go}$

La taille totale d'un fichier peut dépasser 16 Go ce qui est largement suffisant dans la majorité des cas.

La structure d'un i-noeud est ainsi faite :



Cette structure a 2 avantages :

- Les fichiers petits ($< 10\text{ Ko}$) ont leurs données accédées en un temps très courts.
- Les i-noeuds sont des structures petites et donc très commodément manipulable.

Nota : Si l'on désire traiter des fichiers plus gros que 16Go, il faut alors revoir la taille des blocs ce qui peut se définir lors de l'installation du système. (c'est néanmoins assez rare)

Ne pas confondre système et base de données, qui elles gère des fichiers très gros mais en employant des techniques d'accès à des pistes entières afin d'accélérer les échanges et les recherches.

Vérification de l'intégrité d'un FS

En réalité un FS Unix n'est pas constitué d'un seul super bloc. Il en existe plusieurs répartis sur différents cylindres du disque. Cela permet d'accélérer les accès aux fichiers par une économie de déplacement du bras. Mais cela impose en contrepartie une vérification de la cohérence des différents super blocs.

- La commande de vérification de cohérence de système de fichiers est `fck`
 - ◇ Linux se singularise en utilisant la commande `e2fck` à la place de `fck`. Cependant l'appel à `fck` "marche" en général car c'est qu'un lien sur `e2fck` qui est fait.
 - * Il faut avoir le filesystem `/proc`, qui est un filesystem en mémoire, pour faire `e2fck` sinon cela ne fonctionne pas.
 - * Ne jamais faire de `fck` sur `/proc` car c'est un filesystem en mémoire géré directement par le noyau Unix.

Gestion de l'espace disque

- L'espace disque libre se fait par la commande `df` initiale de *disk free*. Sans paramètre cette commande donne la place libre restante en nbr. de blocs libres.

♦ paramètres

option	signification
-t	affiche en plus le nbr. de blocs de données et d'i-nodes libres dans ce système
-v	(<i>verbose</i>) permet d'avoir l'occupation mémoire en pourcentage.
-i	(<i>inode</i>) permet d'avoir le calcul du pourcentage d'occupation des i-nodes

- La commande `du` permet de connaître la place en Ko occupée par les fichiers et répertoires donnés en argument.

♦ paramètres

option	signification
-s	affiche la taille totale sans autre indication
-a	affiche le détail des fichiers et sous répertoires
-r	génère un message pour les répertoires qu'il ne peut pas lire (pas de droit)

Savoir dans quel répertoire on est

- La commande `pwd` permet de savoir où l'on se trouve dans le FS. Elle est fort utile !!!

Commandes relatives à un système de fichiers

- Toutes les commandes et leurs paramètres sont décrites dans : `man <nom de la commande>` Ce texte n'est pas toujours très facile à lire mais c'est un manuel fortement utile dans certain cas. Il nécessite néanmoins de connaître la commande que l'on veut utiliser.

Gestion des fichiers

* ls	donne des indications sur les fichiers contenus dans le répertoire
* cat <nom>	affiche le fichier à l'écran (attention aux fichiers binaires) peut aussi les concaténer ex. <i>cat n1 n2 > n3</i>
* split	permet le découpage d'un fichier trop long en plusieurs plus petits
* more <nom>	affiche le fichier par page
* cp <src> <dest>	copie le fichier spécifié dans la src vers la destination dest
* rm <nom>	supprime le(s) les fichier(s)
* mv <nom1> <nom2>	renomme le fichier de nom1 en nom2
* cmp <nom1> <nom2>	compare 2 fichiers (si identiques alors aucun message)
* diff	compare des fichiers (l'option -b permet d'ignorer les espaces en fin de ligne)
* file <nom>	reconnaître la nature du fichier
* chmod <val> <nom>	change les droits d'un fichier
* find	recherche éventuellement d'un fichier
* grep	recherche d'expressions régulières dans un fichier

ex d'utilisation

- ◆ \$ split -300 toto toto. découpe le fichier toto en fichier toto.aa, toto.ab, etc., de 300 lignes chacun.
\$ cat toto.* > toto concatène les fichiers toto.aa, toto.ab, etc. en un fichier toto

- ◆ les fichiers f1 :

essai de la commande diff
voici le 1er fichier

 et f2 :

essai de la commande diff
voilà le 2ième fichier il est un peu plus gros

\$ diff f1 f2
2c4 (indique la ligne et la position du caractère qui diffère)
< voici le 1er fichier

> voilà le 2ième fichier il est un peu plus gros

- ◆ Le répertoire /home/paul contient un répertoire /exo et un fichier toto. On est sous /home/paul/exo et on désire copier le fichier.

\$ cp ../toto . ← très important, ne pas oublier le . qui indique le répertoire courant

On veut ensuite renommer ce fichier en tata

\$ mv toto tata

⇒ On pouvait faire ces 2 commandes sur une même ligne séparée par un ;

\$ cp ../toto . ; mv toto tata

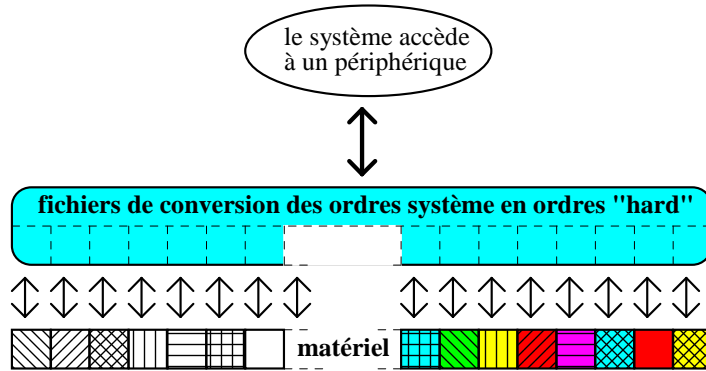
Gestion des répertoires

* cd <chemin>	se déplacer dans le FS. / donne la racine et .. le parent du répertoire courant
* pwd	affiche le chemin du répertoire courant
* mkdir <nom>	créer un répertoire
* rmdir <nom>	détruire un répertoire (il doit être vide)
* rm -r <nom>	supprime le(s) les sous répertoires et les fichiers qui y sont contenus
* mv <nom1> <nom2>	renomme le répertoire de nom1 en nom2

Nota : Souvent la commande rm "demande" si oui ou non on veut le faire. Pour éviter cette demande il faut la forcer par la syntaxe : rm -f ... ou rm -fr ...

Les périphériques, le noyau

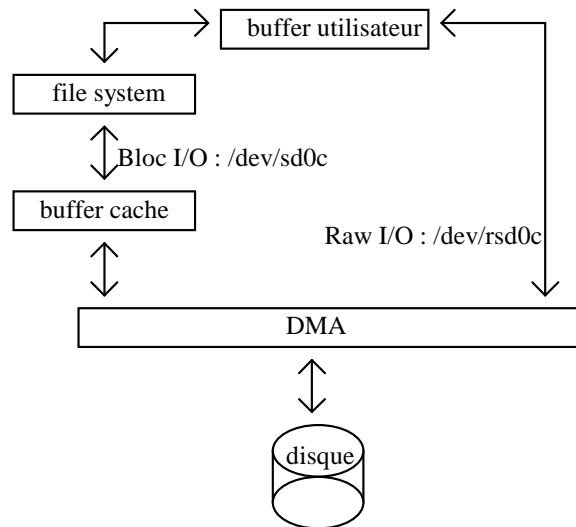
Architecture matériel



- Les fichiers situés dans le répertoire `/dev` permettent à Unix d'accéder aux périphériques.
- Ces fichiers sont spéciaux et sont créés à l'aide de la commande `mknod` (voir ci-dessous).
- Ils ne contiennent pas de données mais leur i-node détermine l'accès au pilote d'I/O par 2 numéros :
 - ♦ le numéro de périphérique majeur (major device number)
 - ♦ le numéro de périphérique mineur (minor device number)

Les 2 nombres, majeur & mineur, correspondent à un index dans une structure du noyau appelée *cdevsw* pour les périphériques de type caractère ou *bdevsw* pour ceux de type bloc ou *idevsw* pour ceux de type ligne.

ex. d'accès au disque



- On peut accéder aux informations caractère par caractère ou en bloc.
- Lorsqu'on liste (`ls -l`) les fichiers dans `/dev`, on voit le type d'accès par la première lettre: si c'est un **b** c'est un device **bloc**, si c'est un **c** c'est un device **caractère**.
 - ♦ Le système de fichier est interfacé au dessus de l'interface bloc qui utilise les *buffer cache* d'Unix. Le rôle du cache étant de différer les écritures et d'anticiper les lectures. Le daemon *update* a pour fonction de vider le cache toutes les 30s. *sync* fait explicitement la même chose.

Les périphériques

- Ils sont associés à des fichiers dans le répertoire `/dev`.

fichier	périphérique
<code>/dev/hda</code>	premier disque dur IDE
<code>/dev/hda1</code>	première partition du premier disque dur IDE
<code>/dev/hda2</code>	deuxième partition du premier disque dur IDE , etc.
<code>/dev/hda5</code>	premier lecteur logique d'une partition étendue DOS
<code>/dev/hda6</code>	deuxième lecteur logique d'une partition étendue DOS , etc.
<code>/dev/hdb</code>	deuxième disque dur IDE (éventuellement CD-ROM IDE/ATAPI)
<code>/dev/hdb1</code>	première partition du deuxième disque dur IDE
<code>/dev/sda</code>	premier disque dur SCSI
<code>/dev/sda1</code>	première partition du premier disque dur SCSI
<code>/dev/fd0</code>	premier lecteur de disquette
<code>/dev/fd1</code>	deuxième lecteur de disquette
<code>/dev/scd0</code>	premier lecteur CD-ROM sur SCSI
<code>/dev/ttys0</code>	premier port série ≈ COM1
<code>/dev/ttys1</code>	deuxième port série ≈ COM2
<code>/dev/lp0</code>	premier port parallèle ≈ LPT1
<code>/dev/lp1</code>	deuxième port parallèle ≈ LPT2
<code>/dev/console</code>	écran
<code>/dev/mouse</code>	la souris (c'est généralement un lien sur <code>/dev/ttyS0</code> ou <code>/dev/cua0</code>)
<code>/dev/null</code>	le périphérique null (celui qui sert de "poubelle" !!!)
<code>/dev/swap</code>	l'espace de disque swap
<code>/dev/mem</code>	la mémoire vive physique
<code>/dev/kmem</code>	l'espace mémoire du noyau Unix
<code>/dev/drum</code>	l'espace disque de pagination

- ♦ La science de ces devices, bien que n'étant pas indispensable, est tellement fréquent à l'administrateur qu'il fini toujours par les connaître par cœur.

L'ajout de périphériques

- Les *devices* présentent une interface standard d'accès au noyau. Chaque drivers offrent les routines :
 - * attach , open , close , read , write, reset , stop , select , timeout , etc.
 - * plus certaines autres qui traitent les spécificités dues au matériel.
- A l'intérieur du noyau l'adresse de ces fonctions est placée dans une structure qui est une table de saut. Actuellement il existe 2 tables : une pour les blocs et une pour les caractères. La table est indexée par le *major*. Le *minor* permet d'adresser le périphérique parmi plusieurs de même type.
- Pour ajouter un périphérique, il faut connaître le type de périphérique *bloc/raw* et les 2 numéros *majeur* & *mineur*, à associer au niveau *device*.

ex. `mknod /dev/ttyc c 12 4` permet d'ajouter un device, nommé *tytc*, de type caractère, dont le major est 12 et dont le minor indique qu'il est le 4^{ième} périphérique de ce type.

Recompilation du noyau

- Après avoir rajouté les *devices* voulus, il faut recompiler le noyau afin de mettre à jour les tables.
sous Linux. Pour cela, il faut avoir installé les sources.

0)	<code>cd /usr/src/linux</code>	passer dans le répertoire sources de Linux
0 bis)	Editer le fichier <code>/usr/src/linux/Makefile</code>	vérifier que la définition de ROOT_DEV est correcte ROOT_DEV = CURRENT
1)	<code>make config</code>	lancer la commande de compilation de la configuration
2)	<code>make dep</code>	lancer la commande de compilation des dépendances
3)	<code>make clean</code>	lancer la commande de nettoyage des anciens fichiers
4)	<code>make zlilo ou</code>	la plus complète, elle génère un noyau et l'intègre à Linux Loader (lilo)
	<code>make zImage ou</code>	Génère un noyau mais ne l'enregistre pas à la racine mais dans <code>/usr/src/linux/arch/i386/boot</code> et ne l'incorpore pas à lilo
	<code>make disk</code>	installe directement le noyau sur une disquette

- *make Image* compile un noyau et place le nouveau noyau dans `/usr/src/linux/Image`
- *make zImage* compile un noyau sous forme compressée, celui-ci se décompressera de lui-même lors d'un démarrage du système. Cela permet d'occuper moins de place

nota : Une fois que l'on a une nouvelle version du noyau il est important de la sauvegardée sur disquette
`cp Image /dev/fd0`

Ajout d'une imprimante

Le périphérique d'imprimante n'est pas accessible aux utilisateurs, seul l'administrateur a les droits permettant de le manipuler.

Les utilisateurs doivent passer par une file d'attente pour imprimer. Ils envoient le fichier à un processus de gestion (*line printer scheduler* ou **lpsched**), une requête d'impression. Le processus enregistre la demande dans une file d'attente (*queue*). La première requête est traitée ensuite par le programme d'impression, puis la suivante jusqu'à ce que la file soit vide.

Pour chaque imprimante connectée au serveur, l'administrateur doit installer une file d'attente. Pour ces files l'administrateur doit donner les 3 points suivants :

- le nom de la file
- le nom du fichier de périphérique par lequel l'impression se fera
- le nom du programme d'impression qui exécutera la demande

Ces 3 informations sont placées dans le fichier */etc/printcap* qui est lu lors du démarrage du démon **lpd**.
ex.

ligne	signification
# HP laser jet plus	<i>commentaire</i>
lp hpj:\	<i>nom de(s) imprimante(s) , séparés par /</i>
:lp=/dev/lp1:\	<i>nom du fichier périphérique</i>
:sd=/usr/spool/lp1:\	<i>dir. où sont stockées les demandes d'impression (spooling dir.)</i>
:mx#0:\	<i>pas de limite de taille. On peut aussi fixer une valeur max.</i>
:of=/usr/spool/lp1/hpjlp:\	<i>programme de sortie</i>
:lf=/usr/spool/lp1/hp-log:	<i>fichier de report des erreurs</i>

nota Les \ sont là pour indiquer que les informations continuent sur la ligne suivante (toutes les informations doivent être sur une même ligne pour **lpd**). Ils doivent figurer dans le fichiers comme continueurs de ligne, à chaque fois que l'on désire passer à la ligne suivante pour une plus grande clarté de lecture.

Lancer une impression

La commande **lpr** permet à l'utilisateur de faire une demande d'impression au programme **lpd** (*line printer daemon*), qui est le démon gérant l'impression. On obtient en retour un numéro de requête d'impression, qui est composé du nom de la file d'attente et du rang dans la file.

Etat de l'impression

lpstat fournit les informations sur l'état des fichiers qui sont en attente d'impression.

option	signification
-t	(total) affiche un état complet du "spooler"
-u <i>Nom</i>	(user) ne sont données que les demandes de l'utilisateur <i>Nom</i>
-v	affiche la liste des imprimantes disponibles. Pour chaque imprimante donne le gestionnaire.

Arrêter une demande d'impression

Afin de supprimer le fichier en attente dans la file on peut utiliser la commande **cancel**.

Si l'on fait suivre par un numéro de demande c'est la demande qui sera supprimée de la file d'attente. Si c'est un nom d'imprimante c'est la requête en cours qui est supprimée.

ex.

ligne	signification
\$ lp livre1	<i>demande l'impression de livre1</i>
Ptr_HP-23	<i>il s'imprimera sur Ptr_HP et il a le numéro 23</i>
\$ lp -n2 lettre23-5	<i>demande l'impression, 2 fois, de lettre23-5</i>
Ptr_HP-34	<i>il s'imprimera sur Ptr_HP et il a le numéro 34 (il y a eu 11 demandes entre temps</i>
\$ cancel Ptr_HP-34	<i>arrête la demande d'impression des 2 lettres23-5 sur Ptr_HP_34</i>

Pilotage de l'imprimante

Les utilisateurs peuvent intervenir sur le fonctionnement de l'imprimante.

Si un problème survient, ils peuvent arrêter l'impression par **disable** <nom de l'imprimante>, dès que le problème est réglé ils peuvent remettre en route les impressions par **enable** <nom de l'imprimante>.

Les demande dans la file ne sont pas touchées par l'arrêt et la libération de l'imprimante.

Comme ces 2 commandes peuvent perturbées le travail de tous les utilisateurs, les administrateurs choisissent souvent de ne pas l'autoriser aux utilisateurs.

Montage/Démontage d'un FS

Il est possible de rajouter/soustraire un autre FS à celui du système.

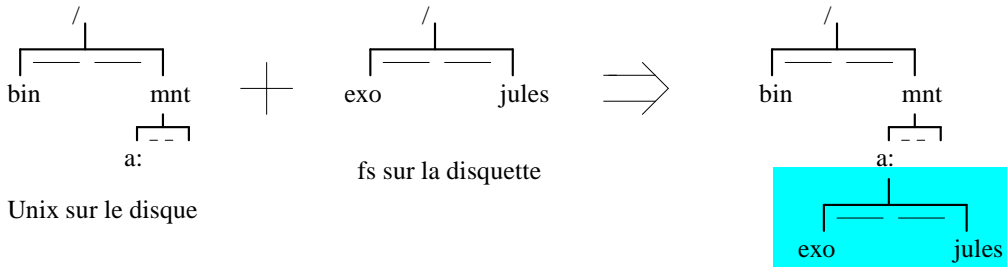
- L'ajout se fait par la commande `mount <device> <répertoire>`

Commentaire [a1] :

nota : Les devices sont des noms de périphériques situés dans le répertoire `/dev`.

ex. on suppose qu'il existe un répertoire `/mnt/a`:

`mount /dev/fd0 /mnt/a` : monte le FS de la disquette sur le FS Unix en `/mnt/a`:



⇒ Une bonne habitude consiste à toujours utiliser `/mnt` comme répertoire de départ pour monter un nouveau FS.

- Cas d'une partition disque, ou d'un autre disque, `/usr`

Il semble assez naturel de monter cette partition au démarrage du système, afin que les utilisateurs puissent travailler sur leurs fichiers.

Linux

- Linux sait reconnaître les systèmes suivants : (donc accéder aux fichiers qui y sont stockés)

nom	système de fichier	commentaire
ext2	secondaire étendu	la plupart des systèmes Linux
ext	étendu	remplacé par ext2
minix	Minix	<i>Minix</i> est un Unix de Tanenbaum de l'université d'Amsterdam
xia	Xia	idem ext2 mais rarement utilisé
umdos	UMSDOS	utilisé quand on installe Linux sur une partition DOS
msdos	MS-DOS	permet d'accéder à la partition DOS
proc	/proc	permet d'avoir des informations par des processus tel que : ps , etc.
iso9660	ISO 9660	format des CD ROM
xenix	Xenix	<i>Xenix</i> est le système Unix de Microsoft (abandonné)
sysv	System V	<i>System V</i> est une variante de Xenix pour µp x86
coherent	Coherent	<i>Coherent</i> est un Unix pour x86
hpfs	HPFS	pour accéder à une partition HPFS en read only

- Le fichier `/etc/fstab` contient les principaux systèmes de fichiers montés au démarrage d'Unix. On peut visualiser les informations de ce fichier en appelant `mount -av`. C'est par ailleurs cette commande qui est appelée au début dans `/etc/rc` (c'est le fichier d'initialisation système exécuté au boot)

Démonter un device

- Pour démonter un device il suffit d'utiliser la commande

`umount <répertoire où est monté le device>`

Attention : Pour démonter il faut être dans un répertoire qui n'est pas la racine de celui que l'on va démonter sinon le système répond "process busy". Dans notre ex. il ne faut pas être dans `a:`, ou au dessous, pour démonter. Un bon réflexe consiste à se placer dans `/mnt` pour démonter.

nota : Tous les systèmes sont démontés lorsque l'on quitte Unix avec shutdown.

Exemple de mount et umount

Nota : Il faut auparavant vérifier avec la commande mount les système qui sont montés, car bien évidemment s'ils le sont déjà on ne pourra les monter encore une fois.

La véritable commande, pour monter un périphérique, s'écrit ainsi :

`mount [-t type] <device> <directory>` Le type est facultatif. Par défaut c'est celui de Linux : ext2

- ❖ Monter une disquette au format MSDOS sur /mnt/floppy.

mount	-t msdos	/dev/fd0	/mnt/floppy
↑	↑	↑	↑
Cmd	type	device	directory

Démonter la disquette

`umount /mnt/floppy`

- ❖ Monter un CD sur /mnt/cdrom.

`mount -t iso9660 /dev/fd0 /mnt/cdrom` (Un cdrom est codé suivant le type iso9660)

Démonter le CD

`umount /mnt/cdrom`

- ❖ Monter un 2^{ème} disque IDE au format DOS sur /mnt/disque2. (il faut que la dir. /mnt/disque2 existe)

`mount -t msdos /dev/hdb1 /mnt/disque2`

Démonter le disque

`umount /mnt/disque2`

- ❖ Monter un 2^{ème} disque au format DOS sur /mnt/disque2. (il faut que la dir. /mnt/disque2 existe)

IDE	SCSI
<code>mount -t msdos /dev/hdb1 /mnt/disque2</code>	<code>mount -t msdos /dev/sdb1 /mnt/disque2</code>

Démonter le disque

`umount /mnt/disque2`

Le fichier fstab

Ce fichier est lu au démarrage de Linux et monte les FS qui y sont décrits.

Ex. de fichier fstab

#/dev/sda1	/mnt/DOS_sda1	vfat	user,exec,conv=binary	0 0
/dev/sda1	/dos	msdos		0 0
/dev/sda5	/boot	ext2	defaults	1 2
/dev/sda6	/	ext2	defaults	1 1
/dev/sda7	swap	swap	defaults	0 0
/mnt/floppy	/mnt/floppy	supermount	fs=vfat,dev=/dev/fd0	0 0
none	/proc	proc	defaults	0 0
none	/dev/pts	devpts	mode=0620	0 0
/mnt/cdrom	/mnt/cdrom	supermount	fs=iso9660,dev=/dev/cdrom	0 0

La syntaxe simplifiée des directives du fichier est :

`<device> <directory> <type> 0 0`

nota : On peut en voir une expression en 2^{ième} ligne pour monter la partition dos en /dos plutôt qu'en /mnt/DOS_sda1. Il est monté en tant que msdos car nous désirons compiler avec le GNU Linux des sources situées dans cette partition DOS, ce qui n'est pas possible en vfat.

Annexes

Fichiers de démarrage

inittab

```
#
# inittab      This file describes how the INIT process should set up the system in a certain run-level.
#
# Author:      Miquel van Smoorenburg, <miquels@drinkel.nl.mugnet.org>
#              Modified for RHS Linux by Marc Ewing and Donnie Barnes
#
# Modified June 1998 - Gael Duval for Linux Mandrake
#
# Default runlevel. The runlevels used by RHS are:
# 0 - halt (Do NOT set initdefault to this)
# 1 - Single user mode
# 2 - Multiuser, without NFS (The same as 3, if you do not have networking)
# 3 - Full multiuser mode
# 4 - unused
# 5 - X11
# 6 - reboot (Do NOT set initdefault to this)
#
id:3:initdefault:                # C'est le démarrage : graphique = 5 / texte = 3 voir ci-dessus

# System initialization.
si::sysinit:/etc/rc.d/rc.sysinit    # appel des scripts d'init

10:0:wait:/etc/rc.d/rc 0             #
11:1:wait:/etc/rc.d/rc 1             # "
12:2:wait:/etc/rc.d/rc 2             #
13:3:wait:/etc/rc.d/rc 3             #
14:4:wait:/etc/rc.d/rc 4             # "
15:5:wait:/etc/rc.d/rc 5             #
16:6:wait:/etc/rc.d/rc 6             #

# Things to run in every runlevel.
ud::once:/sbin/update

# Trap CTRL-ALT-DELETE
ca::ctrlaltdel:/sbin/shutdown -t3 -r now    # comment arrêter Linux avec Ctrl + Alt + Del

# When our UPS tells us power has failed, assume we have a few minutes of power left. Schedule a shutdown for 2 minutes
# from now. This does, of course, assume you have powerd installed and your UPS connected and working correctly .
pf::powerfail:/sbin/shutdown -f -h +2 "Power Failure; System Shutting Down"

# If power was restored before the shutdown kicked in, cancel it.
pr:12345:powerokwait:/sbin/shutdown -c "Power Restored; Shutdown Cancelled"

# Run gettys in standard runlevels
1:12345:respawn:/sbin/mingetty tty1
2:2345:respawn:/sbin/mingetty tty2
3:2345:respawn:/sbin/mingetty tty3
4:2345:respawn:/sbin/mingetty tty4
5:2345:respawn:/sbin/mingetty tty5
6:2345:respawn:/sbin/mingetty tty6

# Run kdm in runlevel 5
x:5:respawn:/opt/kde/bin/kdm -nodaemon
```

Résultats des Commandes

ls

ls -l

L'affichage sera de forme suivante :

permissions	ln*	owner	group	size ⁺	date	name
-------------	-----	-------	-------	-------------------	------	------

* : le nombre de liens sur le fichier

⁺ : exprimé en bytes

ex. de ls -l

total 3						
drwx-----	5	root	root	1024	Jun 19 20:33	Desktop
-rw-r--r--	1	root	root	1171	Jul 12 16:19	cmd_la
-rw-r--r--	1	root	root	157	Jul 12 16:19	cmd_ls

ex. de ls -al dans la directory /root

total 17							
drwxr-xr-x	4	root	root	1024	Jul 12 16:19	.	la directory courante
drwxr-xr-x	18	root	root	1024	Jun 19 17:34	..	la directory parente
-rw-r--r--	1	root	root	1126	Aug 23 1995	.defaults	
-rw-r--r--	1	root	root	1334	Jul 11 16:11	.bash_history	historique des commandes
-rw-r--r--	1	root	root	24	Jul 14 1994	.bash_logout	shell de sortie
-rw-r--r--	1	root	root	238	Aug 23 1995	.bash_profile	configuration du shell bash
-rw-r--r--	1	root	root	269	Jul 1 18:27	.bashrc	alias
-rw-r--r--	1	root	root	180	Mar 4 1996	.cshrc	configuration du cshell
-rw-r--r--	1	root	root	108	Jun 19 17:33	.emacs	configuration d'emacs
-rw-r--r--	1	root	root	123	Jun 19 17:33	.inputrc	
drwxr-xr-x	3	root	root	1024	Jun 19 20:33	.kde	configuration de kde
-rwxr-xr-x	1	root	root	435	Jun 26 20:00	.kderc	"
-rw-r--r--	1	root	root	166	Mar 4 1996	.tcshrc	configuration du shell tcsh
drwx-----	5	root	root	1024	Jun 19 20:33	Desktop	
-rw-r--r--	1	root	root	1171	Jul 12 16:19	cmd_la	
-rw-r--r--	1	root	root	157	Jul 12 16:19	cmd_ls	

nota : les fichiers commençants par un . ne sont pas affichés par la commande *ls*. Il faut spécifier *ls -a* pour les voir apparaitre.

Table des matières

DEBUTER SUR UNIX (INSTALLATION / ARRET / DEMARRAGE)	1
INSTALLATION D'UN SYSTEME UNIX	1
<i>Lequel choisir</i>	1
<i>Où se le procurer</i>	1
<i>Installation</i>	3
<i>Partitions</i>	3
DEMARRAGE	6
<i>Démarrage d'un système Unix (ex. Linux)</i>	6
<i>Arrêt d'un système Unix (ex. Linux)</i>	8
PRESENTATION DU SYSTEME UNIX	9
CARACTERISTIQUES DU SYSTEME	9
<i>Composantes d'UNIX</i>	9
L'ADMINISTRATEUR SYSTEME	10
L'EDITEUR DE TEXTE VI	12
COMMANDES	13
LIGNE OU EXPRESSIONS	15
<i>Les commandes les plus utiles</i>	17
LE SYSTEME DE FICHIERS UNIX	1
ORGANISATION ARBORESCENTE	18
<i>Notion de systèmes de fichiers (File System=FS)</i>	18
<i>Arborescence pré-SVR4</i>	18
<i>Arborescence SVR4</i>	18
<i>Arborescence de Linux</i>	19
LES TYPES DE FICHIERS	19
<i>Types de fichiers Unix et création</i>	19
<i>Détermination du type</i>	20
<i>Extensions courantes</i>	21
STRUCTURE D'UN FS	22
VERIFICATION DE L'INTEGRITE D'UN FS	24
GESTION DE L'ESPACE DISQUE	24
<i>Savoir dans quel répertoire on est</i>	24
COMMANDES RELATIVES A UN SYSTEME DE FICHIERS	24
<i>Gestion des fichiers</i>	25
<i>Gestion des répertoires</i>	25
LES PERIPHERIQUES, LE NOYAU	26
ARCHITECTURE MATERIEL	26
LES PERIPHERIQUES	27
L'AJOUT DE PERIPHERIQUES	28
<i>Recompilation du noyau</i>	28
AJOUT D'UNE IMPRIMANTE	29
<i>Lancer une impression</i>	29
<i>Etat de l'impression</i>	29
<i>Arrêter une demande d'impression</i>	30
<i>Pilotage de l'imprimante</i>	30
MONTAGE/DEMONTAGE D'UN FS	31
<i>Démonter un device</i>	31
<i>Le fichier fstab</i>	32
FICHIERS DE DEMARRAGE	35
<i>inittab</i>	35
RESULTATS DES COMMANDES	36
<i>ls</i>	36