

INTRODUCTION A LINUX

Par : Vanessa GIACOMONI

SUIVI DES VERSION		
version	date	auteur
1.0 (origine)	25/12/08	Vanessa GIACOMONI - Association ATLANTIC Bernard GIACOMONI - Association ATLANTIC
1.1	01/08/09	Vanessa GIACOMONI - Association ATLANTIC

Table des matières

I. GENERALITES	2
I.1.HISTORIQUE.....	2
I.2.CARACTERISTIQUES GENERALES:.....	2
I.3.LA LICENCE GPL:.....	2
I.4.NOYAU LINUX ET DISTRIBUTIONS:.....	3
II. ORGANISATION DES DONNEES	4
II.1.L'ARBORESCENCE DES FICHIERS:.....	4
II.2.MONTAGE D'UN SOUS-SYSTEME DE FICHER:.....	4
III. LA NOTION D'UTILISATEUR:	6
III.1. INTRODUCTION:.....	6
III.2. DEFINITION DES UTILISATEURS:.....	6
III.3. RELATIONS ENTRE FICHIERS ET UTILISATEURS:.....	6
III.3.1. PROPRIETAIRE D'UN FICHER (OU D'UN REPERTOIRE):.....	6
III.3.2. GROUPES D'UTILISATEURS:.....	6
III.3.3. NOTION DE DROITS D'ACCES:.....	6
III.3.4. LE SUPER UTILISATEUR:.....	7
IV. INSTALLATION DE LINUX:	9
IV.1. GENERALITES:.....	9
IV.2. DIFFERENTES PHASES DE L'INSTALLATION:.....	9
IV.2.1. PHASE N°1: PARTITIONNEMENT DU DISQUE DUR:.....	9
IV.2.1.1. POSITION DU PROBLEME:.....	9
IV.2.1.2. DESCRIPTION DU MASTER BOOT RECORD.....	10
IV.2.2. PHASE N° 2: FORMATAGE DES PARTITIONS:.....	10
IV.2.3. PHASE N° 3: CREATION DU NOYAU LINUX:.....	11
IV.2.3.1. PREPARATION DES SOURCES:.....	11
IV.2.3.2. CONFIGURATION DU NOYAU:.....	11
IV.2.3.3. CREATION DES DEPENDANCES:.....	11
IV.2.3.4. COMPILATION ET EDITION DE LIENS:.....	11
IV.2.3.5. AJOUT DE DEPENDANCES:.....	11
IV.2.4. INSTALLATION DU NOUVEAU NOYAU:.....	11
V. LANCEMENT DE LINUX:	13
V.1. SEQUENCE DE LANCEMENT DU NOYAU LINUX.....	13
V.1.1. GENERALITES:.....	13
V.1.2. LE PROCESSUS INIT:.....	13
V.1.3. LES NIVEAUX DE LANCEMENT:.....	13
V.1.4. LE FICHER /ETC/INITTAB:.....	13
V.1.5. LES SCRIPTS RC ET LE FICHER INIT.D.....	14
VI. COMMANDES FONDAMENTALES DE LINUX	15
VII. COMMANDES D'ADMINISTRATION (SOUS ROOT)	18
VIII. COMMANDES RESEAU:	19

INTRODUCTION A LINUX

Par : Vanessa GIACOMONI

I. GENERALITES

I.1. HISTORIQUE

L'«inventeur» de LINUX est Linus TORVALDS (1990-1991). C'est un logiciel libre distribué sous **licence GPL**.

I.2. CARACTERISTIQUES GENERALES:

Le système LINUX est un système d'exploitation bâti sur le modèle du système **UNIX**. En particulier:

- Le principe de l'organisation des systèmes de fichiers est pratiquement le même.
- Le langage de commande système est très semblable, surtout en ce qui concerne les commandes «de base».
- Les principes de gestion des processus et tâches logicielles sont pratiquement identiques.
- Etc...

L'administration de LINUX est calquée sur le modèle UNIX SYSTEME V de AT&T.

Comme UNIX, les systèmes Linux sont, à la base, un système en temps partagé. Ils sont conformes aux normes POSIX. Les différentes notions de base d'UNIX (processus, threads, événements, sémaphores, commandes de base, etc...) sont pratiquement identiques à celles d'un UNIX courant.

I.3. LA LICENCE GPL:

La licence GPL (General Public licence, appelée parfois «copyleft»):

- Permet la modification du programme original, et sa diffusion (sous licence GPL) ;
- Autorise la vente du logiciel libre sous sa forme originelle ou modifiée, à condition que le vendeur autorise la diffusion du logiciel ainsi modifié.
- Autorise l'utilisation du logiciel à des fins lucratives.
- Les logiciels sous licence GPL restent la propriété de leurs auteurs, personne ne peut donc s'approprier tout ou partie des droits d'auteur.
- La licence n'implique aucune forme de rémunération des auteurs.

Un logiciel libre n'est donc pas nécessairement gratuit. En revanche, les termes de la licence GPL implique que le **code source** soit libre d'accès. le code source du noyau de Linux doit donc demeurer accessible gratuitement: le système peut être compilé sur toute plate-forme ou c'est techniquement possible.

REMARQUE:

Le Projet GNU («GNU is Not Unix», développé par FSF (Free Software Foundation) vise à développer des logiciels soumis aux termes de la licence d'utilisation GPL.

INTRODUCTION A LINUX

Par : Vanessa GIACOMONI

I.4. NOYAU LINUX ET DISTRIBUTIONS:

On appelle **NOYAU** (en anglais, **KERNEL**) le «coeur» du système, c'est à dire la partie qui assure la gestion des processus et tâches et les mécanismes généraux de gestion des événements et de la périphérie.

Une **DISTRIBUTION** de Linux est composée:

- D'un noyau linux (à une version donnée)
- D'un ensemble de logiciels fonctionnant avec le noyau Linux et fournissant des fonctions de haut niveau ou améliorant l'ergonomie d'utilisation (couche graphique, bureau, outils d'installation, d'administration, etc...)

L'emploi d'une distribution n'est pas indispensable pour installer Linux, puisque toutes les sources sont accessibles gratuitement. Cependant, les distributions comprennent des procédures et outils d'installation qui facilitent énormément celle-ci.

Distributions les plus connues: RedHat, Mandriva, Debian, SuSe, Ubuntu, Fedora.....

INTRODUCTION A LINUX

Par : Vanessa GIACOMONI

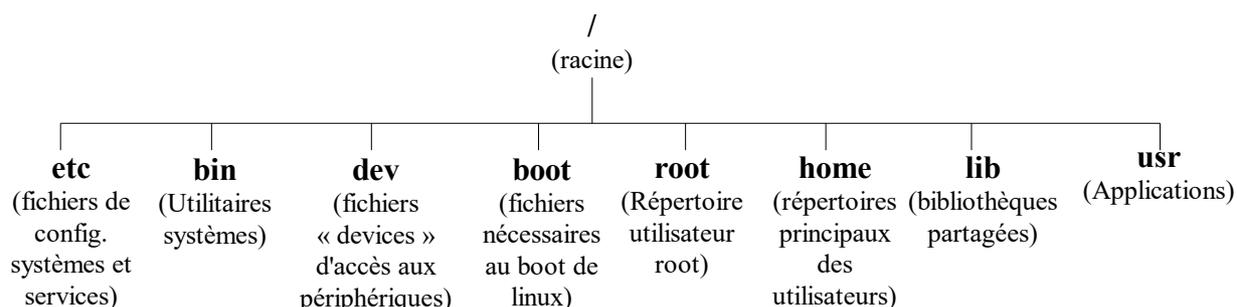
II. ORGANISATION DES DONNEES

II.1. L'ARBORESCENCE DES FICHIERS:

Dans Linux, comme dans UNIX, toutes les données sont organisées en une **arborescence unique** de fichiers commençant par une racine notée «/». A la différence de windows, pour lequel chaque support physique (ou chaque partition d'un support physique) est racine d'une arborescence de fichier (ce sont les «lecteurs» A, B, C...), sous LINUX les fichiers situés sur un support physique peuvent être attachés (on dit «**montés**») à n'importe quel noeud de l'arborescence unique.

Dans cette arborescence, tous les noeuds sont considérés comme des fichiers, même les répertoires: on trouvera ainsi des fichiers répertoires, des fichiers «sources», des fichiers exécutables (.exe), etc... certains fichiers représentent même des équipements périphériques (devices). On dit souvent «sous UNIX, tout est fichier».

L'arborescence d'un système UNIX n'est pas entièrement laissée à l'initiative des utilisateurs. Par exemple, pour LINUX, on trouve au premier niveau au moins les répertoires suivants:



Ces répertoires de premier niveau sont utilisés par le système d'exploitation. Ils ne doivent donc pas être détruits, sous peine de dysfonctionnement. Cependant, rien n'empêche d'ajouter d'autres répertoires à ce niveau ou des sous-répertoires ou fichiers dans les répertoires existants. Il est tout de même recommandé de respecter l'organisation définie.

II.2. MONTAGE D'UN SOUS-SYSTEME DE FICHIER:

Les systèmes de fichiers contenus par les périphériques de type «mémoire auxiliaire» (tels que les disques durs, les floppy, les cd-roms, les clefs USB...) peuvent être intégrés dans cette arborescence. Il suffit pour cela d'attacher ces systèmes de fichiers internes à un noeud de l'arborescence principale. Cette opération, appelée «**montage d'un système de fichier**», peut s'effectuer à tout moment par la directive système «mount», mais peut aussi être effectuée lors du lancement du système en paramétrant des fichiers de lancement.

REMARQUES:

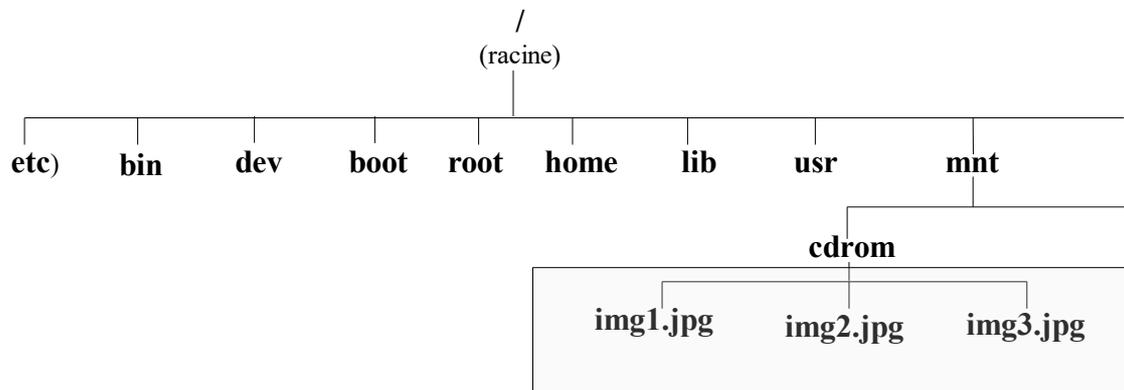
- En ce qui concerne les périphériques partitionnés, chaque partition constitue une sous-arborescence et peut être montée séparément.
- Le montage permet d'intégrer à l'arborescence des systèmes de fichiers de nature différente de celle de l'arborescence principale. Il est ainsi tout à fait possible de monter une partition windows sous linux.

INTRODUCTION A LINUX

Par : Vanessa GIACOMONI

EXEMPLE :

pour intégrer un CD-ROM, on pourra créer un répertoire mnt sous la racine et un sous répertoire cdrom, puis de monter sur le noeud cdrom la racine de l'arborescence contenue dans le CD-ROM. Imaginons que le CD-ROM contienne plusieurs fichiers images sous sa racine (img1.jpg, img2.jpg, img3.jpg). Après montage, l'arborescence générale deviendra :



INTRODUCTION A LINUX

Par : Vanessa GIACOMONI

III. LA NOTION D'UTILISATEUR:

III.1. INTRODUCTION:

Les systèmes basés sur UNIX intègrent la notion d'UTILISATEURS: a un instant donné, plusieurs utilisateurs peuvent être connectés à un système UNIX depuis différentes consoles d'exploitation et travailler en simultanéité apparente. LINUX est donc un système MULTI UTILISATEURS

III.2. DEFINITION DES UTILISATEURS:

Chaque utilisateur se voit attribuer un «compte» auquel sont attachés:

- Un nom (login)
- Un mot de passe (password).
- Un répertoire principal dans l'arborescence des fichiers (normalement, dans la sous-directorie home). Dans ce répertoire sont placés des fichiers qui s'exécutent à la connexion de l'utilisateur et permettent de définir son environnement personnel («bureau», alias, etc...)

Exemple: Dans une entreprise, le comptable pourra avoir pour login «compta» et le mot de passe «sqdqs». Logiquement, la directory principale de cet utilisateur sera: /home/compta.

Il existe un utilisateur privilégié: l'utilisateur «root», qui correspond à peu près au compte administrateur de windows.

Le fichier /etc/passwd contient la liste des utilisateurs couramment définis. Chaque entrée de ce fichier contient les paramètres liés à un utilisateur donnée: login, password (codé), nom complet d'utilisateur, chemin d'accès au répertoire principal.

Les systèmes UNIX peuvent être lancés en mode multi utilisateurs ou mono utilisateur. Ce dernier mode, (qui correspond à l'utilisateur root seul) est surtout utilisé pour la maintenance et le dépannage système.

III.3. RELATIONS ENTRE FICHIERS ET UTILISATEURS:

III.3.1. PROPRIETAIRE D'UN FICHIER (OU D'UN REPERTOIRE):

Tout noeud de l'arborescence (répertoire ou fichier) appartient à un utilisateur donné: cet utilisateur sera l'«owner» (le propriétaire) du fichier ou du répertoire.

III.3.2. GROUPES D'UTILISATEURS:

Les utilisateurs peuvent être regroupés en GROUPES D'UTILISATEURS, l'intérêt étant de donner à des utilisateurs travaillant sur les mêmes données des droits d'accès communs sur ces données (groupes de projets, groupes d'utilisateurs...).

III.3.3. NOTION DE DROITS D'ACCES:

Pour chaque ressource de la machine (fichier, répertoire), 3 types de droits d'accès sont définis:

- **Acces en écriture:** on peut créer et modifier la ressource.
- **Acces en lecture:** on ne peut que lire le contenu
- **Acces en exécution:** lorsque la ressource est un exécutable (programme), on

INTRODUCTION A LINUX

Par : Vanessa GIACOMONI

peut lancer l'exécution

Ces droits d'accès peuvent être définis pour 3 catégories d'utilisateurs:

- L'utilisateur qui possède la ressource (owner)
- Les utilisateurs du groupe auquel appartient l'owner.
- Les autres utilisateurs définis

Exemple:

Soit le fichier Fic et soient U1, U2, U3 et U4 les 4 utilisateurs définis pour la machine. Supposons que U1 soit l'owner de Fic, que U1, U3 et U4 appartiennent au groupe G1 et U2 au groupe G2. Supposons que l'administrateur système ait défini la répartition des droits suivante:

DEFINITION DES DROITS D'ACCES AU FICHER «Fic»									
	Droits du possesseur (owner)			Droits du groupe du possesseur			Droits des autres utilisateurs		
Types de droits	Lecture (R)	Ecriture (W)	Exécution (X)	Lecture (R)	Ecriture (W)	Exécution (X)	Lecture (R)	Ecriture (W)	Exécution (X)
Droits sur Fic	R	W	X	R	-	X	R	-	-
Users	Le possesseur (owner) de Fic est U1 on lui donne ici le droit de lire, modifier et exécuter Fic			U3 et U4 appartiennent au même groupe que le possesseur de Fic on leur donne ici le droit de lire et exécuter Fic (mais non de l'écrire)			U2 n'appartient pas au groupe du possesseur de Fic. Ici, on ne lui donne que le droit de lire Fic.		

Quel est l'intérêt de ce mécanisme ?

- Dans l'exemple ci-dessus, on peut supposer que U1, U3 et U4 ont été placés dans un même groupe (G1), car ils travaillent ensemble sur un projet commun. Le fichier Fic est possédé par U1, car c'est cet utilisateur qui a la charge de le développer (par exemple, Fic est une page HTML).
- U1 a donc besoin de tous les droits d'accès sur Fic.
- U3 et U4 travaillent avec U1 dans le groupe G1. Ils ont donc besoin d'accéder à ce qu'a fait U1, mais ils ne peuvent pas le modifier. On leur donne donc le droit de lire Fic et de l'exécuter, mais pas de l'écrire.
- U2 ne travaille pas sur le même projet. On peut cependant lui donner le droit de lire Fic, à titre de renseignement, mais sans pouvoir l'exécuter ni le modifier.

Le mécanisme permet donc de contrôler et de sécuriser les accès des utilisateurs aux données en répartissant les droits d'accès en fonction de l'activité de l'utilisateur.

III.3.4. LE SUPER UTILISATEUR:

Il existe cependant un utilisateur privilégié, l'utilisateur «**root**» à qui tous les droits d'accès sont attribués sur toutes les ressources: c'est l'équivalent du compte administrateur de windows.

L'utilisateur root est l'«owner» de tous les fichiers de configuration du système: il faut donc être «root» pour pouvoir modifier des fichiers comme

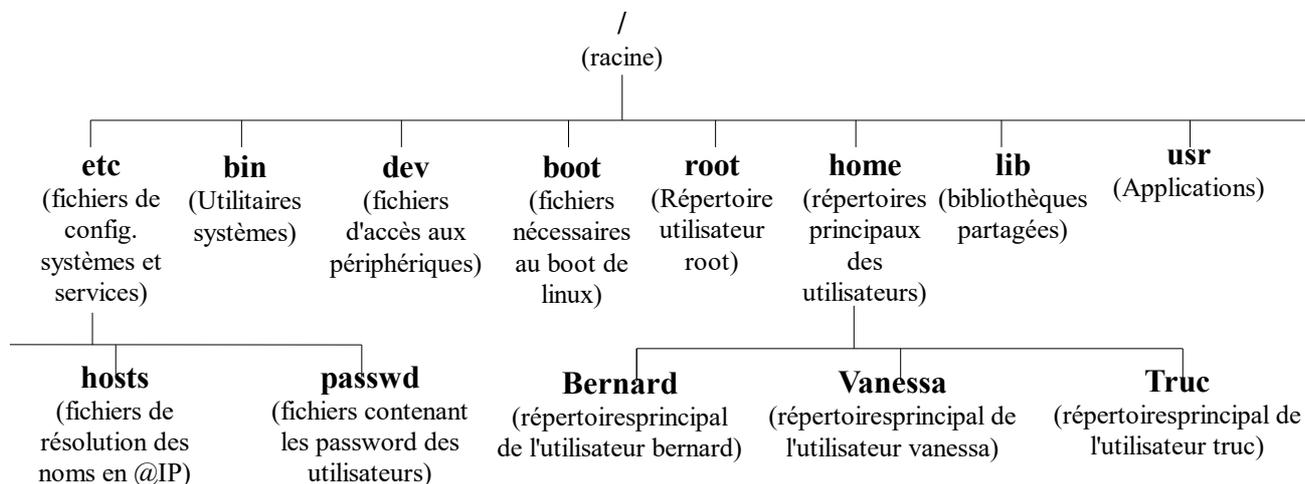
INTRODUCTION A LINUX

Par : Vanessa GIACOMONI

/etc/hosts (fichier de correspondance entre @IP et noms) ou /etc/passwd (fichier définissant les utilisateurs et leur mot de passe).

REMARQUE : La directory principale de l'utilisateur root n'est pas sous /home: c'est le répertoire /root.

SCHEMA DE L'ARBORESCENCE DES FICHIERS



Exemples :

- *Le fichier contenant les mots de passe des utilisateurs (fichier passwd) se trouvera dans le sous-répertoire etc. Le chemin pour y accéder sera /etc/passwd*
- *Ici, 3 utilisateurs sont définis.*

INTRODUCTION A LINUX

Par : Vanessa GIACOMONI

IV. INSTALLATION DE LINUX :

IV.1. GENERALITES :

En général, l'installation de LINUX s'effectue à partir d'une distribution (Red Hat, Mandriva, Ubuntu...). Dans ce cas, la procédure d'installation ne demande que peu de connaissances de l'utilisateur sur les différentes phases d'installation: après avoir booté sur le support d'installation de la distribution (en général, un CD-ROM fourni par un distributeur ou fabriqué à partir d'un téléchargement), il lui suffira de bien paramétrer les menus de configuration qui lui sont proposés (en général, des options par défaut sont proposés aux indécis...).

La procédure d'installation prend en charge toutes les phases, depuis le partitionnement du disque jusqu'à l'installation de modules additionnels. Le déroulement de cette procédure varie suivant la distribution choisie.

L'installation directe à partir des sources du noyau Linux (www.kernel.org) demande plus de connaissances techniques. Le paragraphe suivant en donne un bref aperçu.

IV.2. DIFFERENTES PHASES DE L'INSTALLATION :

Quel que soit le support d'installation choisi (sur CD, clef USB, par téléchargement ou à l'aide d'un CD de distribution), les premières étapes consistent à **partitionner** (ou repartitionner) le disque, puis à **formater** les partitions ainsi définies.

Il faudra ensuite **créer le noyau linux** et ses dépendances à partir des sources fournies (par une distribution ou par téléchargement à partir de www.kernel.org), puis l'**installer** à sa place dans le système de fichiers.

IV.2.1. PHASE N°1: PARTITIONNEMENT DU DISQUE DUR:

IV.2.1.1. POSITION DU PROBLEME :

En général, le problème est d'installer une distribution Linux sur un disque déjà partitionné. Il faut donc ajouter des partitions à celles qui existent déjà (ou à la place de celles-ci).

Dans ce cas, un (ou des) systèmes d'exploitation sont déjà installés sur la machine (dans les partitions du disque cible ou sur un autre disque). On pourra donc utiliser des outils fonctionnant avec un de ces systèmes d'exploitation:

- Outils LINUX:fdisk
- Outils Window: Partition Magic, fips.

Dans le cas contraire (aucun OS installé), il faudra recourir à une disquette de boot (window ou linux) pour disposer d'un système minimum permettant d'activer ces commandes.

En général, on créera au moins 2 partitions: une pour le nouveau système et une autre pour les données (qui pourront éventuellement être partagées avec un autre O.S.).

Le partitionnement revient à créer ou modifier le Master Boot Record (M.B.R) du disque dur. Un disque partitionné suivant le format INTEL présente la structure suivante:

INTRODUCTION A LINUX

Par : Vanessa GIACOMONI

IV.2.1.2. DESCRIPTION DU MASTER BOOT RECORD

MASTER BOOT RECORD (secteur 0)		PARTITION PRIMAIRE N° 1 (/dev/hda1)		PARTITION PRIMAIRE N°2 (dev/hda2)		PARTITION PRIMAIRE N°3 (dev/hda3)		PARTITION PRIMAIRE N°4 (dev/hda4)	
Routine d'amorçage	Table des partitions primaires (4 partitions)	Boot System		Boot System		Boot System			

Le MASTER BOOT RECORD, occupe les 512 octets du secteur 0 du disque. Il est composé principalement:

- D'une routine d'amorçage qui permet de lancer le «boot loader» d'une des partitions primaires
- D'une table des partitions primaires qui contient les données nécessaires pour localiser et identifier les 4 partitions primaires possibles.

4 partitions primaires peuvent être définies. Lorsqu'une partition primaire est une partition système (window, Linux,..), ses premiers secteurs contiennent un système de boot: pour linux, Lilo (Linux loader) ou Grub (Grand Unified Boot loader).

Une des partitions système est reconnue «partition active». C'est elle qui sera lancée au boot.

REMARQUES:

- 4 partitions primaires peuvent être créées sur un disque dur. Le M.B.R. Contient les informations nécessaires à leur identification.
- Sur ces 4 partition primaires, l'une d'elles peut être une partition «étendue». Une telle partition peut contenir une partition dite «logique» et une partition étendue, et ainsi de suite ... En définitive, un disque dur peut contenir jusqu'à 63 partitions.
- Une partition logique peut contenir un chargeur au même titre qu'un partition primaire.
- Le nom d'une partition obeit à la syntaxe suivante:

```
{ h|s }d{ a|b|c|...|z }{1|2|3|4|5|...|64}.
```

 - h désigne un disque IDE, s un disque scsi ou sata
 - a, b, c... désigne le disque dur physique
 - 1, 2, ,64 désigne la partition, sachant que les partitions 1 à 4 sont des partitions physiques, les autres étant des partitions logiques.
 Par exemple: sdc32 désigne la 32eme partition du disque sata (ou scsi)c.

IV.2.2.PHASE NO 2: FORMATAGE DES PARTITIONS:

Le formatage consiste à attribuer un type à une partition (FAT, VFAT, FAT32, NTFS, Ext2, Ext3, etc...), puis à la désigner (ou non) comme «partition d'amorçage»).

Le choix du type de partition n'est pas anodin: en effet:

- Si l'on désire conserver la possibilité d'utiliser une disquette de boot, cela implique de charger le système dans une partition de type FAT. Or, ces partitions (ainsi que VFAT) sont limitées à 2 GO.
- Si l'on désire partager des données entre deux systèmes d'exploitations coexistant sur la même machine, il faut créer une partition d'un type accepté par les deux O.S.
- Basiquement Ext2 et Ext3 sont des types uniquement reconnus par Linux.

CONSEIL:

Le Type NTFS est, à priori le plus performant en termes de temps d'accès et

INTRODUCTION A LINUX

Par : Vanessa GIACOMONI

de sécurité. Il est traité par windows et Linux. On pourra l'utiliser pour la partition de données, si on veut faire coexister ces deux systèmes. La partition LINUX pourra être une Ext2 ou Ext3.

- Outils de formatage windows: format
- Outils de formatage UNIX-LINUX: mkfs

IV.2.3.PHASE NO 3: CREATION DU NOYAU LINUX:

Le noyau Linux (comme, en général, les noyaux UNIX), est disponible sous la forme des sources d'un programme écrit en langage c. Il faut donc compiler ce programme (compilation+édition de liens) pour obtenir le noyau exécutable:

IV.2.3.1.PREPARATION DES SOURCES:

Le noyau Linux téléchargé à partir du site www.kernel.org se présente sous la forme d'une archive compressées, par exemple: linux-2.6.30.tar.bz2 ou bien linux-2.6.30.tar.gz. Les extensions bz2 ou gz dépendent de l'outil de compression (gzip ou bzip). Suivant l'extension, la décompression se fait par une commande du type:

tar xvjf linux-2.6.30.tar.bz2 ou **tar xzvf linux-2.6.30.tar.gz**

REMARQUE:

Les archives décompressées seront placées dans un répertoire correspondant au nom du fichier archive: ici, ce serait linux-2.6.30.

IV.2.3.2.CONFIGURATION DU NOYAU:

Cette opération permet d'adapter les fichiers de compilation au système hôte. Elle s'effectue par une commande qui peut être:

make menuconfig ou **make xconfig** (version graphique) ou **make config (en ligne)**

Cette commande va lancer un menu (en ligne ou graphique) qui permettra de personnaliser l'installation. En fait, elle crée un fichier de configuration qui sera utilisé pour la compilation.

IV.2.3.3.CREATION DES DEPENDANCES:

Cette action n'est nécessaire que pour les versions antérieures à la 2.6. C'est la commande: **make dep** qui est utilisée.

IV.2.3.4.COMPILE ET EDITION DE LIENS:

La commande:

make bzImage

crée l'exécutable correspondant au noyau sous forme d'une image compressée.

IV.2.3.5.AJOUT DE DEPENDANCES:

Cette phase permet d'intégrer au noyau les modules additionnels définis lors de la phase de configuration. Elle correspond aux commandes:

make modules et **make modules_install**

IV.2.4.INSTALLATION DU NOUVEAU NOYAU:

Il s'agit maintenant d'installer le nouveau noyau de façon à le rendre «bootable». La commande:

make install

Va réaliser les opérations suivantes:

- Installer le nouveau noyau comme système bootable dans le répertoire /boot (transfer de bzImage dans /boot/bzImage-2.6.9)

INTRODUCTION A LINUX

Par : Vanessa GIACOMONI

- Configurer le chargeur d'amorçage (Lilo ou Grub) pour qu'il prenne en compte le nouveau noyau (ajout d'une entrée dans: /etc/lilo.conf).
- Lancer le chargeur lilo pour qu'il prenne en compte son nouveau fichier de configuration (commande **lilo -v**).

Il suffit alors de rebooter pour se voir proposer le boot sur le nouveau noyau.

INTRODUCTION A LINUX

Par : Vanessa GIACOMONI

V.LANCEMENT DE LINUX:

V.1.SEQUENCE DE LANCEMENT DU NOYAU LINUX

V.1.1.GENERALITES:

- Le BIOS, après la phase de test (POST), lance la routine d'amorçage du MBR
- Cette routine lance le Boot Loader de la partition active.
- Le Boot Loader charge le noyau de LINUX et le «ramdisk» initrd.img en mémoire (Le ramdisk est un espace mémoire qui simule un disque pendant le chargement de linux), puis lance le noyau Linux.

V.1.2.LE PROCESSUS INIT:

Le noyau Linux lance le premier processus (processus «init»). Celui-ci est le «père» de tous les autres processus. Le processus init à son tour, lance les processus systèmes suivants en utilisant le contenu du fichier /etc/inittab:

V.1.3.LES NIVEAUX DE LANCEMENT:

LINUX possède 6 «niveaux de lancement différents»:

- Niveau 0: système à l'arrêt
- Niveau 1: mono-utilisateur
- Niveaux 2 à 5: multi-utilisateurs
- Niveau 6: mode arrêt du système.

A chaque niveau de lancement correspondra un fonctionnement différent de Linux, car les processus lancés ne seront pas les mêmes.

Le niveau de lancement peut être modifié:

- A l'installation.
- par la commande: `init <niveau>`
- En modifiant le fichier /etc/inittab

V.1.4.LE FICHIER /ETC/INITTAB:

Les entrées de ce fichier sont du type:

```
id:3:initdefault      <--Niveau de lancement par défaut (ici, = 3)
si::sysinit:/etc/rc.d/rc.sysinit
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      <--Suivant le niveau de lancement n, lance /etc/rc.d/rc n
14:4:wait:/etc/rc.d/rc 4
15:5:wait:/etc/rc.d/rc 5
16:6:wait:/etc/rc.d/rc 6
ud::once:/sbin/update
1:2345:respawn:/sbin/getty 38400 tty1
2:2345:respawn:/sbin/getty 38400 tty2
3:2345:respawn:/sbin/getty 38400 tty3
4:2345:respawn:/sbin/getty 38400 tty4
S0:2345:respawn:/sbin/mgetty -n 3 -s 115200 ttyS0 57600
```

INTRODUCTION A LINUX

Par : Vanessa GIACOMONI

```
S4:2345:respawn:/sbin/mgetty -r -s 19200 ttyS4 DT19200  
x:5:respawn:/usr/bin/X11/xdm -nodaemon
```

Suivant le niveau spécifié, on va lancer une des commandes **/etc/rc.d/rc n** ou n est le niveau.

V.1.5.LES SCRIPTS RC ET LE FICHIER INIT.D

Les scripts RC lancent des scripts situés dans `etc/rc.d/rc<n>.d` (ou `/etc/rc<n>.d`) ou n est le niveau de démarrage. Ces scripts permettent de démarrer les différents services systèmes (ftpd our ftp, httpd pour http, etc...)

INTRODUCTION A LINUX

Par : Vanessa GIACOMONI

VI. COMMANDES FONDAMENTALES DE LINUX

FONCTION	COMMANDE	UTILISATION
Créer un nouveau répertoire	<code>mkdir <chemin d'accès></code>	<code>mkdir /usr/people/Vanessa</code> : crée le répertoire Vanessa dans le répertoire <code>/usr/people</code> <code>mkdir ../Vanessa</code> : crée le répertoire Vanessa dans le répertoire parent du répertoire courant.
Déplacer ou renommer un fichier	<code>mv <chemin d'accès> <cible></code>	<code>mv /usr/people/F1 /usr/bin</code> : déplace le fichier F1 dans le répertoire <code>/usr/bin</code> sans le renommer <code>mv /usr/people/F1 /usr/bin/F2</code> déplace le fichier F1 dans le répertoire <code>/usr/bin</code> en le renommant F2
Editer un fichier	<code>vi, emacs, joe</code>	<code>vi</code> : éditeur de base (<code>vi <chemin d'accès></code>) <code>emacs</code> : très puissant <code>joe</code> : simple
faire une copie d'un fichier	<code>cp -i/l/r... <origine> <destination></code>	<code>cp /usr/people/F1 ./F2</code> : copie le fichier <code>/usr/people/F1</code> dans le répertoire courant, sous le nom F2. <code>cp -r /usr/people/D1 /usr/bin</code> copie le répertoire D1 et ses sous-répertoires dans le répertoire <code>/usr/bin</code>
Lister un répertoire donné	<code>ls -l/lu/m/F,t... <chemin d'accès></code>	<code>ls -l /usr/bin</code> affiche le contenu du répertoire <code>/usr/bin</code> ligne par ligne <code>ls -t /usr/bin</code> affiche le contenu du répertoire <code>/usr/bin</code> en classant les fichiers du plus récent au plus ancien
Naviguer dans les répertoires	<code>cd <chemin d'accès à un répertoire></code>	<code>cd ..</code> remonter vers répertoire parent <code>cd repl/rep2</code> aller vers répertoires enfants <code>cd /usr/include</code> chemin absolu
Rechercher un fichier	<code>Find <racine> -name <nom> -print</code> <code>locate <nom du fichier></code> <code>which <nom de fichier exécutable></code>	<code>find /bin -name toto -print</code> : recherche le fichier toto dans toute l'arborescence dont la racine est <code>/bin</code> , et affiche son chemin d'accès. <code>locate toto</code> : recherche toto dans la base de donnée des fichier manipulés récemment <code>which toto</code> : recherche un fichier exécutable
Savoir ou on se trouve dans l'arborescence des répertoires	<code>pwd</code>	<code>pwd</code> déclenche l'affichage du chemin d'accès (depuis la racine du répertoire courant).

INTRODUCTION A LINUX

Par : Vanessa GIACOMONI

FONCTION	COMMANDE	UTILISATION
Supprimer un fichier	rm -r <chemin d'accès>	rm /usr/people/toto: supprime le fichier toto rm -r /usr/people: supprime tous les fichiers du répertoire people et tous ses sous répertoires.Attention: rm / efface le disque !!!!!
Visualiser le contenu d'un fichi	cat, more	cat -n /usr/include/toto: visualise le contenu du fichier toto. -n permet de numérotter les lignes more /usr/people/toto: visualise le contenu du fichier toto page par page
Repérer une chaîne de caractères dans un fichier texte	Grep <cible> <chemin d'acces au fichier>	grep monNom /etc/passwd: recherche la chaîne «monNom» dans le fichier /etc/passwd
Créer des liens vers un fichier	ln <fichier source> <fichier lien> -s	ln /home/F1 /usr/people/F2: crée un lien «dur» vers le fichier /homeF1 à partir de /usr/people/F2. F1 et F2 pointent tous les deux sur F1 ln /home/F1 /usr/people/F2 -s: crée un lien «symbolique» vers le fichier /homeF1 à partir de /usr/people/F2. F1 et F2 pointent tous les deux sur F1 Lien symbolique: pointe sur un nom de fichier Lien dur: pointe sur un espace disque
Compresser ou décompresser des fichiers au format .gz	gzip <nom du fichier à compacter> gunzip <nom du fichier à décompacter>	gzip /usr/people/toto: compacte le fichier toto au format zip et le place dans le fichier /usr/people/toto.zip gunzip /usr/people/toto.zip: décompacte le fichier toto.zip et le place dans le fichier /usr/people/toto
la commande uncompress	uncompress <nom du fichier.Z>	uncompress /usr/people/toto.Z: correspond au compactage gzip -d toto
Archiver des fichiers ou des répertoires	tar cxvf z <nom du répertoire>	tar cvf /usr/people/vanessa: crée une archive du répertoire vanessa, sous forme d'un seul fichier vanessa.tar tar xvf /usr/people/vanessa.tar: recrée le répertoire vanessa dans le répertoire courant en utilisant l'archive vanessa.tar tar xvfz /usr/people/vanessa.tar.gz: recrée le répertoire vanessa à partir de l'archive compressés vanessa.tar.gz (équivalent de gunzip suivi de tar xvf).
Afficher l'espace libre sur le disque	df, du	df: affiche l'emplacement de montage des systèmes de fichiers. du: affiche utilisation du disque en kilo-octets

INTRODUCTION A LINUX

Par : Vanessa GIACOMONI

FONCTION	COMMANDE	UTILISATION
Gérer les processus	top ps -ax pstree kill <PID> (Processus IDentity) killall <nom de processus>	Top affiche des informations système en continu. (taper q pour arrêter) ps: affiche tous les processus liés à la console ps -ax affiche tous les processus en cours pstree: affichage des processus sous forme arborescente. kill 568: tue le processus de PID 568 killall toto: tue le processus toto (et tous ses fils)
Connecter plusieurs commandes entre elles	Pipes (en fait, un pipe, ou tube est représenté par une)	ls -l /usr/people more: liste le répertoire /usr/people suivant le formatage de la commande more (page par page)
Rediriger un flux	Par le signe > ou >>	cat toto > titi: cat toto provoque l'affichage de toto sur l'écran. Avec cat toto > tata, la sortie est redirigée dans le fichier titi. Si titi n'existe pas, il est créé. S'il existe, son contenu est remplacé par celui de toto. cat toto >> titi: redirige la sortie de toto dans le fichier titi. Si titi n'existe pas, il est créé. Si titi existe, les informations sont ajoutées en fin de titi.

NOTA: en général, l'option -help permet d'obtenir de l'aide sur les options.

INTRODUCTION A LINUX

Par : Vanessa GIACOMONI

VII.COMMANDES D'ADMINISTRATION (SOUS ROOT)

FONCTION	COMMANDE	UTILISATION
Changer les droits d'utilisation des fichiers	chmod	<p>chmod +w toto ajoute le droit d'écrire sur toto</p> <p>chmod -w toto supprime le droit d'écrire sur toto</p> <p>chmod +r toto ajoute le droit de lire toto</p> <p>chmod -r toto supprime le droit de lire toto</p> <p>chmod +x toto ajoute le droit d'exécuter toto</p> <p>chmod u : le changement ne concerne que le propriétaire du fichier</p> <p>chmod g : le changement ne concerne que le groupe du fichier</p> <p>chmod o : le changement ne concerne que les utilisateurs autres que le propriétaire et les membres du groupe.</p> <p>chmod a : le changement concerne tous les utilisateurs</p> <p>ex: chmod g+x toto ajoute le droit d'exécuter toto aux membres du groupe.</p>
Changer l'utilisateur et le groupe propriétaire des fichiers	<p>chown -R <nom groupe> <nom répertoire></p> <p>chown -R <nom groupe> <nom répertoire></p>	<p>chown -R MoiMeme /usr/toto: change le propriétaire de /usr/toto et des sous-répertoires. Le nouveau propriétaire est MoiMeme.</p> <p>chgrp -R projet /usr/toto: change le groupe de /usr/toto et des sous-répertoires. Le nouveau groupe est projet.</p>
Créer un nouvel utilisateur	adduser <nom utilisateur>	adduser Napoléon: crée un nouvel utilisateur appelé Napoléon
Gérer les mots de passe	passwd <nom utilisateur>	A la suite de la commande: passwd Napoléon, le système vous demande de saisir le mot de passe à attribuer à l'utilisateur Napoléon (avec confirmation)
Décrire un utilisateur	chfn	chfn va déclencher un dialogue de saisie d'infos sur l'utilisateur courant
Supprimer un utilisateur	userdel <nom d'utilisateur>	userdel Napoléon: supprime l'utilisateur Napoléon (on supprime non seulement l'utilisateur, mais aussi les répertoires et fichiers associés.
Les commandes tail et head	<p>tail <nom de fichier></p> <p>head <nom de fichier></p>	<p>Tail Toto: affiche les dernières lignes du fichier Toto par défaut, 10)</p> <p>head Toto: affiche les premières lignes du fichier Toto par défaut, 10)</p>

ASSOCIATION A.T.L.A.N.T.I.C - FICHE TECHNIQUE

INTRODUCTION A LINUX

Par : Vanessa GIACOMONI

INTRODUCTION A LINUX

Par : Vanessa GIACOMONI

FONCTION	COMMANDE	UTILISATION
Utiliser le cd-rom ou le lecteur de disquette	mount -t iso9660 <device> </mnt/<nom point de montage>	mount -t iso9660 /dev/hdb /mnt/cdrom: monte un CDROM (device /dev/hdb si il est sur ide 2, format iso9660) sur le point de montage /mnt/cdrom. Si ce répertoire n'existe pas, il faut le créer avant. umount /mnt/cdrom fait l'inverse
Mettre à jour le cache et les liens des bibliothèques	Idconfig -v	idconfig -v: met à jour les liens
Arrêter le système	shutdown	shutdown -r now: reboote le système shutdown -h now: arrête le système et la machine

VIII.COMMANDES RESEAU:

FONCTION	COMMANDE OU FICHIER	UTILISATION
Afficher les données d'interface	ifconfig....	ifconfig <@interface> netmask <masque> broadcast <@broadcast [up/down] configure un interface.
Ajouter une route à la table de routage d'un interface	route add	route add {-host net} <@ip> [netmask <masque>] [gw <@passerelle>] [metric <n>] Ajoute une route vers <u>l'@IP</u> par la passerelle <@passerelle>].
Résolution locale des noms	/etc/hosts	Entrées du type: @IP nom alias alias...
Résolution par DNS	/etc/resolv.conf	Exemple de fichier resolv.conf: domain <nom du domaine> search <nom domaine> .. <nom domaine> #on utilise d'abord /etc/hosts, puis #DNS) order host, bind #contiennent les @DNS dans l'ordre #d'utilisation (jusqu'à 6 adresses) nameserver <@ip 1> nameserver <@ip 2> nameserver <@ip 3> nameserver <@ip 4> nameserver <@ip 5> nameserver <@ip 6>