



Linux à l'EPFL

Premiers pas

Michèle Coulmance, DIT-SB EPFL

Avril 2008

Préambule

Ce support de cours doit être considéré comme un document de travail, à commenter et à annoter au fur et à mesure. Il est destiné à fournir références et éléments, certains d'entre eux n'auront pas une durée de vie très longue (tout change si vite en informatique !), d'autres sont très spécifiques à l'utilisation de linux à l'EPFL. Il ne s'agit donc pas d'un manuel « de référence ».

Quelques liens qui pourront vous aider à en trouver d'autres !

Le portail linux de l'epfl

<http://linux.epfl.ch>

Le guide de survie du débutant sous linux

<http://www.delafond.org/survielinux>

Le portail linux francophone

<http://linuxfr.org>

lea Linux Electronique Aide, ou Linux Entre Amis

<http://lea-linux.org>

Aides diverses

[http:](http://www.ac-creteil.fr/reseaux/systemes/linux/Welcome.html)

[//www.ac-creteil.fr/reseaux/systemes/linux/Welcome.html](http://www.ac-creteil.fr/reseaux/systemes/linux/Welcome.html)

Table des matières

Préambule	iii
1 Historique et distributions, rappel	1
1.1 Petit historique	1
1.2 Distributions :	2
2 Structure du système d'exploitation	3
2.1 Le noyau	3
2.2 Le système de fichier	3
2.3 L'interface shell	3
2.4 Le serveur X Window	4
3 Un tour dans l'environnement graphique	7
3.1 Pré-requis : notion d'utilisateur	7
3.2 Super-utilisateur :	7
3.3 Démarrer une session	8
4 Les premières commandes	9
4.1 Avant tout	9
4.2 Liées aux utilisateurs	10
4.3 Fichiers et chemins	10
4.4 Manipulation de texte	10
4.5 Les redirections des commandes	11
4.6 Les tubes (pipe)	11
4.7 La commande script :	11
5 Objectif installation	13
5.1 L'exploration du système	13
5.1.1 Le partitionnement	13
5.1.2 Les paramètres réseau :	15
5.1.3 Les paramètres pour X :	15
5.1.4 Le matériel pour booter	16
5.2 Une première installation	16

6	L'utilisateur	19
6.1	Informations et authentifications, bases de données locales et centralisées	19
6.1.1	Locales :	19
6.1.2	Centralisées :	20
6.2	Création de compte	20
6.2.1	local	20
6.2.2	centralisé	21
6.3	Environnement	22
6.4	Fichiers de configuration	24
6.4.1	Configuration par défaut, générale :	24
6.4.2	Fichiers de configuration de l'utilisateur :	24
6.4.3	Quelques essais de modifications	24
6.5	Environnement X	26
6.6	En prime : les scripts, un aperçu	26
7	Première approche du système de fichier	31
7.1	Disques physiques, partitions et systèmes de fichiers	31
7.1.1	Disques	31
7.1.2	Partitions :	31
7.1.3	RAID et LVM	31
7.1.4	Système de fichier	32
7.2	Montage	33
7.2.1	Montage d'une ressource distante :	33
7.2.2	Automontage :	33
7.3	Gestion de périphériques :	34
7.3.1	Gestion dynamique des périphériques, udev et HAL	34
7.4	Structure des répertoires	36
7.5	Quelques commandes relatives aux fichiers	37
7.5.1	Quelques exemples d'utilisation	38
7.5.2	Les liens	38
7.6	Permissions de base :	38
7.6.1	Modification des permissions :	39
7.6.2	Appartenance	39
7.6.3	Droits spéciaux sur un fichier : sticky bit, setuid, setgid	39
7.7	ACL (Access Control List)	40
7.8	Archivage	40
7.8.1	Options :	41
7.8.2	Utilisation	41
8	Niveaux de fonctionnement, processus	43
8.1	Démarrage	43
8.2	Arrêt du système	44
8.3	Niveaux de fonctionnement	44

8.4	démarrer en mode single user :	45
8.5	Scripts de démarrage	46
8.5.1	examen du fichier /etc/inittab	46
8.5.2	Examen des scripts du répertoire /etc/init.d	46
8.6	En cas de problèmes	47
8.7	Processus	47
8.7.1	Examen de l'état du système	47
8.7.2	Tuer un processus	47
8.7.3	Modes d'exécution d'un processus	48
9	Gestion des packages	49
9.1	RPM	49
9.2	yum	50
9.2.1	Configuration	50
9.2.2	Utilisation en ligne de commande :	51
9.2.3	Nouveaux outils graphiques	51
10	NFS	53
10.1	Côté serveur	53
10.2	Côté client	54
11	Sécurité	55
11.1	Introduction à SELinux :	55
11.2	Firewall	56
11.2.1	Paramétrage plus fin, exemple d'un serveur NFS3	56
12	SAMBA	59
12.1	Serveur	59
12.2	Client, les outils	60
12.2.1	Graphiques	60
12.2.2	Lignes de commande	61
13	Connexions à distance	63
13.1	SSH, Secure SHell	63
13.1.1	ssh et scp	63
13.1.2	ssh et les clés	64
13.2	Autres	64
13.3	Terminal distant	64
13.4	VPN	64
	Annexe	65

A	Organisation linuxline.epfl.ch	65
A.1	Répertoires de base	65
A.1.1	fedora	65
A.1.2	RedHatEnterprise	65
A.2	Mises à jour Fedora	65
A.3	Accès	66
B	Editeur VI	67
B.1	Fonctionnement	67
B.1.1	Mode commande	67
B.1.2	Mode insertion	67
B.1.3	Mode ligne	68
B.1.4	Quelques indications sommaires	68
B.2	Mode d'emploi	68
B.3	Autres éditeurs	69
C	kickstart	71
D	grub	75
D.1	Spécifier une partition et des fichiers dans le GRUB :	75
D.1.1	Désignation des périphériques de stockage	76
D.1.2	Désignation des partitions	76
D.1.3	Désignation des fichiers par leur nom	76
D.1.4	Désigner un périphérique par défaut	76
D.2	Quelques exemples de commandes	77
D.2.1	Commandes de base pour amorcer un noyau	77
D.2.2	Retrouver sur quelle partition sont les fichiers d'amorçage	77
D.2.3	Restaurer un secteur de boot sur un disque dur	77
D.3	Exemple de fichier <code>/etc/grub.conf</code>	77
E	vncviewer et X-win32	79
E.1	La base, vnc, Virtual Network Computer :	79
E.1.1	Côté linux :	79
E.1.2	Côté windows :	80
E.2	X-win32 :	80
E.2.1	Côté linux :	80
E.2.2	Côté windows :	80
E.3	Outils graphiques sous linux, reposant sur VNC	80
E.3.1	Serveurs	80
E.3.2	Clients	81

F	CD-DVD commandes en lignes	83
F.1	Obtenir l'adresse d'un graveur :	83
F.2	commande cdrecord	83
F.3	Créer une image iso :	84
F.4	Autres utilitaires	84
F.4.1	paquetage dvd+rwtools	84
F.4.2	growisofs	84
G	Quelques raccourcis clavier	85
	Index	89

Chapitre 1

Historique et distributions, rappel

1.1 Petit historique

1969 : Ken Thompson (laboratoires de Bell) écrit la première version de ce qui devait par la suite s'appeler Unix. Le nom "Unix" était un calembour sur le système d'exploitation Multics (Multiplexed Information and Computing Service) et a été écrit "Unics" en premier lieu (UNiplexed Information and Computing System).

1971 : Ritchie et Thompson écrivent le noyau d'Unix en C.

1974 à 1977 : le code source d'Unix est distribué librement aux universités. UNIX gagne la faveur de la communauté scientifique et universitaire. Il a été ainsi à la base des systèmes d'exploitation des principales universités.

1978 : UNIX, 7ème édition. Cette version a été développée expressément pour être portée sur diverses architectures matérielles. En outre, avec la version 7, ATT annonce qu'ils font payer des licences pour accéder aux sources du système. Ce qui incite l'université de Californie à Berkeley à créer sa propre variante : BSD UNIX.

1984 : Richard Stallman, au MIT (Massachusetts Institute of Technology) lance un mouvement pour le développement des logiciels libres ; ce qui débouche sur la création de la FSF : Free Software Foundation, autour du projet GNU (GNU's Not Unix) : <http://www.gnu.org>

1987 : X Window, une interface client/serveur graphique développée au MIT.

1991 : Linus Torvald décide de développer un système d'exploitation utilisant au mieux les fonctionnalités du processeur 386 d'Intel : naissance de Linux.

Chaque composant du noyau (système de fichiers, gestion de mémoire etc.) est sous la responsabilité d'une personne qui centralise les développements de centaines de développeurs communiquant via Internet. Code source, et code binaire sont librement et gratuitement distribués. Tout acquéreur peut librement

utiliser, modifier, et même commercialiser un produit, mais il est tenu de mettre gratuitement à la disposition de la communauté les modifications.

1.2 Distributions :

Une distribution comprend le noyau, des pilotes de périphériques, des bibliothèques, des utilitaires d'installation et de post-installation, ainsi que des logiciels.

Quelques distributions :

- Slackware, la première des distributions qui a popularisé Linux.
- RedHat : a développé le paquetage, un paquetage comprenant un logiciel, sa documentation, et des utilitaires pour simplifier l'installation, la désinstallation et la mise à jour. Ce concept a été repris ensuite par d'autres distributions. La distribution de Redhat étant devenue commerciale, c'est Fedora qui a repris la suite pour la distribution non commerciale.
- Debian : le développement de Debian est réalisé via Internet de la même manière que le développement du noyau Linux. Utilise son propre format de paquetage. C'est la distribution des utilisateurs confirmés et des puristes.
- Ubuntu : c'est **LA** distribution à la mode ! C'est une distribution basée sur Debian, et adaptée par la société Canonical afin d'être plus facilement "prise en main". (pour faire un peu de mauvais esprit : debian.org alors que ubuntu.com)
- SuSE : distribution d'origine allemande, possède ses propres outils d'installation, utilise les paquetages RedHat. SuSE sponsorisait le groupe Xfree86 pour le développement de serveurs X de cartes graphiques récentes.
- Mandriva (ex Mandrake) : une bonne distribution pour les débutants

Chapitre 2

Structure du système d'exploitation

2.1 Le noyau

- Contrôle les terminaux, les imprimantes, les disques, et autres périphériques
- Gère et alloue les ressources pour les utilisateurs.
- Gère le système de fichier

2.2 Le système de fichier

- Range les données de façon hiérarchique

2.3 L'interface shell

Un processus exécuté en mode utilisateur ne peut pas accéder directement aux ressources de la machine. Le shell, interpréteur de commande, représente l'interface utilisateur.

Les utilisateurs vont interagir sur le Noyau du Système d'Exploitation grâce à un **interpréteur de commande**. Il existe plusieurs interpréteurs de commande : C-Shell, Korn-shell, Bourne-shell. Les fonctionnalités sont les mêmes, les syntaxes sont différentes.

Ils vont jouer le rôle d'une interface entre l'utilisateur et le noyau. Un shell lit ce qui provient du terminal, et traduit les requêtes des utilisateurs en actions transmises au noyau. Une requête d'un utilisateur peut être simple, lister le contenu d'un répertoire par ex., ou beaucoup plus complexe. Une fois la requête formulée, le shell a la responsabilité d'interpréter la commande et de la transmettre au noyau pour que les ressources nécessaires puissent être allouées, le noyau étant responsable de l'ordonnancement et de la synchronisation de la requête.

2.4 Le serveur X Window

Cf. historique, c'est un système graphique fonctionnant sur le mode client/serveur, le serveur désignant ici un terminal X, les clients des applications X, clients et serveurs dialoguant selon un protocole X.

L'application serveur gère non seulement l'écran, mais également la souris et le clavier (=> on peut avoir des différences de fonctionnement de clavier en mode console et en X-Window).

Même si la célèbre ligne de commande demeure l'outil favori des utilisateurs chevronnés, il y a donc déjà bien longtemps que les utilisateurs peuvent bénéficier aussi d'un environnement graphique, et d'une souris à cliquer ! Linux offre une panoplie d'interfaces graphiques qui peuvent être installées sur une distribution.

Mais : l'interface graphique n'est rien d'autre qu'un programme qui tourne sur le système. Elle ne fait pas partie du noyau Linux ni du système de base. Elle est simplement un outil puissant qui permet d'utiliser les capacités graphiques de votre machine.

Dans le souci du respect des standards, un standard avait été créé pour définir comment dessiner et déplacer des fenêtres sur l'écran, comment permettre à l'utilisateur d'agir avec le clavier et la souris ainsi que d'autres aspects importants d'un système graphique. Ce standard a été nommé X Window System, souvent abrégé en X11 ou simplement X. Il est utilisé sur la plupart des systèmes Unix, Linux et apparentés.

Au départ : projet Xfree86.

Mais ce dernier ayant opté pour une licence incompatible avec la GPL (GNU General Public License), l'utilisation de Xorg est désormais recommandée. Les différences entre Xorg et XFree86 sont actuellement minimales. Si l'on connaît un des deux systèmes, on connaît à peu près l'autre.

Sur RedHat Enterprise jusqu'à la version 3 : Xfree86-4.3, depuis la version 4
xorg-x11

Sur Fedora : xorg-x11

A la base : des applications graphiques (/usr/X11R6/bin), et un gestionnaire de fenêtres graphiques .

Puis : développement d'ensembles cohérents de clients X : c'est ce qu'on appelle bureau (desktop). On dispose alors de beaucoup d'outils, on peut accéder aux applications graphiques sans même connaître leur nom.

Le choix du **bureau** reste un choix avant tout personnel : KDE, ou Gnome ?

Les deux ont des avantages ... et des inconvénients. Cependant : même si vous préférez et utilisez Gnome, veillez à installer konqueror (paquet kdebase) un explorateur de KDE qui présente des fonctionnalités intéressantes.

Pour mémoire, des gestionnaires anciens mais très légers :

- *twm* : Le Tab Window Manager minimaliste, proposant la boîte à outils la plus basique.
- *fvwm2* : Dérivé du gestionnaire de fenêtres twm incorporant une apparence 3D
- un site parmi d'autres : <http://www.plig.org/xwinman/intro.html>

Chapitre 3

Un tour dans l'environnement graphique

3.1 Pré-requis : notion d'utilisateur

Chaque utilisateur déclaré peut démarrer une session de travail, et accéder à un certain nombre de ressources.

Chaque utilisateur dispose :

- D'un nom d'utilisateur, et d'un numéro d'utilisateur.
- D'un groupe, et numéro de groupe.
- D'un mot de passe.
- D'un répertoire initial de travail.
- D'un interpréteur shell.

3.2 Super-utilisateur :

Compte **root** : compte de l'administrateur.

*ATTENTION : **root** a tous les droits. Evitez de travailler systématiquement sous le compte root !*

*On verra plus tard la commande **sudo** [6.2.1](#) à la page [20](#)*

3.3 Démarrer une session

Un petit tour dans l'environnement graphique

- Les 4 bureaux
- le menu principal
- les préférences
- etc.

*Un paramétrage fin de **Gnome** est aussi possible avec **gconf-editor***

C'est un peu comparable à la base de registres Windows (en mieux, forcément ...). Toutes les options de configurations sont accessibles d'un seul coup, et on peut également intervenir sur des paramètres « cachés »

*On peut visualiser les configurations en entrant dans un browser l'url **about:config***

On peut aussi démarrer par défaut en mode console, c'est même parfois indispensable ... on reviendra sur ce point plus tard

Chapitre 4

Les premières commandes

Pour ouvrir un terminal :

Menu principal, system tools

ou bien bouton de droite dans un bureau

et ... à vos lignes de commandes !

- *Le rappel des commandes : en appuyant sur les flèches de défilement haut/bas on peut obtenir un rappel des précédentes lignes de commandes entrées.*
- *La complétion : la touche **tab** permet de compléter automatiquement une ligne de commande (recherche automatique de chemins par exemple)*

4.1 Avant tout

man : « Quand tout a échoué, lisez le manuel ! »

*Avec **konqueror** , un des explorateurs possibles faisant partie du paquetage **kdebase** , on peut pointer sur l'URL **#une_commande** pour obtenir l'affichage de la page du manuel de **une_commande***

whatis, **apropos** : recherche par mots-clé ou par fragments de mots-clé

*Pour créer les bases de données de recherche **whatis** et **apropos** : lancer la commande **/usr/sbin/makewhatis***

locate *la_commande* : emplacement de la commande

Pour créer les bases de données : **locate -u**

4.2 Liées aux utilisateurs

who : quels sont les utilisateurs connectés

whoami : afficher l'U-ID effectif

id : caractéristiques (U-ID et G-ID) d'un utilisateur

4.3 Fichiers et chemins

cd : se déplacer dans l'arborescence

pwd : où est-on dans l'arborescence

rm : effacer un fichier

ls : afficher les caractéristiques des fichiers et répertoires

cp : copier un fichier

cat ou **more** : examiner le contenu d'un fichier

Notations particulières dans les arborescences :

. représente le répertoire courant

.. représente le répertoire père, cad le répertoire de niveau immédiatement supérieur au répertoire courant.

~ représente le répertoire initial (répertoire home) d'un utilisateur

4.4 Manipulation de texte

grep : très puissant, recherche une chaîne ou une expression régulière dans un fichier et affiche les lignes contenant cette chaîne

diff : compare 2 fichiers ASCII et affiche les lignes différentes.

cat : affichage et regroupement de fichiers

head -n/tail -n : affichage des n premières/dernières lignes d'un fichier

tail -f : lecture d'un fichier en continu, visualisation du flux d'entrée dans ce fichier au fur et à mesure. Pratique pour examiner les fichiers de log, et voir "en direct" ce qui se passe

4.5 Les redirections des commandes

Le shell travaille avec 3 « fichiers » :

- **stdin** : entrée standard associée au clavier
- **stdout** : sortie standard associée au clavier
- **stderr** : sortie standard d'erreur associée au clavier

Il est possible de rediriger les entrées ou sorties standard vers des fichiers, ou vers d'autres commandes.

Redirection de la sortie standard : **commande > nom-de-fichier**

```
ls -al > mon_listing
```

le résultat de la commande sera écrit dans un fichier

Redirection de l'entrée standard : **commande < nom-de-fichier**

Redirection de l'erreur standard : **(commande) >& nom-de-fichier**

4.6 Les tubes (pipe)

La barre verticale | (**pipe**) est utilisée pour permettre la communication entre 2 processus.

```
commande_1 | commande_2
```

La sortie d'une commande est utilisée comme l'entrée de la commande suivante. On peut dire que c'est équivalent à rediriger la sortie standard d'une commande vers un fichier, puis utiliser ce fichier comme entrée standard pour la commande suivante.

Plusieurs commandes peuvent être ainsi combinées.

```
ls /usr/bin | grep lp
ls /etc/init.d/p* | xargs cat
ls | xargs du -sh
```

4.7 La commande script :

Permet de sauver dans un fichier les commandes et leurs résultats

```
script -f essai.txt
...
exit (pour terminer la commande script et fermer le fichier)
```


Chapitre 5

Objectif installation

5.1 L'exploration du système

Quelques outils graphiques :

hardware browser

system monitor

*on peut retrouver toutes les informations dans les fichiers du répertoire **/proc***

Pour une nouvelle machine :on peut vérifier les listes de compatibilité hardware sur le site <http://bugzilla.redhat.com/hwcert/>

le site <http://bugzilla.redhat.com> permet aussi de rechercher les bugs répertoriés, ou d'en signaler.

5.1.1 Le partitionnement

swap

Presque toutes les distributions vous obligent à utiliser une partition de swap (=fichier d'échange). Cette partition servira à écrire lorsque la mémoire vive déborde. Dans le temps, on disait que le swap devait faire le double de la taille de la RAM. Maintenant que le kernel a changé et que les RAM sont énormes ainsi que les disques durs, une fois la RAM est généralement suffisant..

racine, notée /

Ici se trouve la partition racine du système. Obligatoire. Plus ou moins grande selon ce que vous mettrez dessus. Sachez que la saturation de cette partition est

dangereuse pour le système. Tous les répertoires ci-dessous se placeront ici si vous ne leur spécifiez pas de partition spécifique.

/boot

Il peut être bon que le système de démarrage soit sur une partition à part. Donc un /boot en ext2, de très petite taille (150 Mo).

/home

Ici seront vos données. L'intérêt de les séparer physiquement est de pouvoir par exemple réinstaller Linux sans toucher à vos données. Donnez la taille que vous voulez. Le format peut être ext2 ou ext3.

Attention : il peut être intéressant de prévoir un automontage home. Dans ce cas là il faudra donner un autre nom à cette partition, et les répertoires des utilisateurs seront montés sous le nom symbolique de home.

/usr

Ici seront installés les programmes. Cette partition doit être de grande taille (minimum 1,5 Go) si vous voulez beaucoup de programmes. Vous pouvez ne pas créer de /usr. Simplement, votre / devra alors être plus grande. Et attention à ne pas installer trop de programmes qui satureraient alors /. Format ext2 ou ext3. (Cette partition est facultative)

/var

C'est la partition dans laquelle le système écrit régulièrement des données. Comme la partition racine (/) ne doit jamais saturer, il est prudent que les données changeantes soient sur une partition à part. Une centaine de Mo minimum peut convenir. Beaucoup plus si vous voulez être un serveur web. Format ext2 ou ext3 (Cette partition est facultative).

En résumé

Tous les répertoires qui n'ont pas une partition spécifique seront automatiquement installés dans la partition /, dite partition racine, ou partition root, mais à ne pas confondre avec le répertoire /root qui est le répertoire de l'utilisateur administrateur root !

Au minimum : **swap, /boot, / (racine, root) , /ma_partition**

Et toujours disposer d'au moins UNE partition qu'on ne touchera pas si l'on doit réinstaller le système.

*lorsqu'on installe une nouvelle version du système, il est bon de sauvegarder le répertoire **/etc** , le répertoire **/var** , le répertoire **/home***

*Si le répertoire **/home** est sur une partition, on pourra réinstaller sans toucher à cette partition.*

Examinez le partitionnement existant :

fdisk ou **parted**

df

cat /etc/fstab

ou bien outils graphiques

Regardez si on le modifiera ou non, et comment.

Vérifiez la taille mémoire.

free

ou bien outils graphiques

5.1.2 Les paramètres réseau :

Examinez votre configuration réseau :

- Nom de machine et routeur :

fichier /etc/sysconfig/network

commande /sbin/arp

- adresse IP :

fichier /etc/sysconfig/network-scripts/ifcfg.eth0

commande ifconfig -a

- Serveur de nom : **fichier /etc/resolv.conf**

*Si on peut bénéficier d'un service **DHCP** , **D**ynamic **H**ost **C**onfiguration **P**rotocol, c'est le serveur **DHCP** qui fournit toutes ces informations*

5.1.3 Les paramètres pour X :

Dans le fichier **/etc/X11/xorg.conf** ou **/etc/X11/XF86Config-4** (pour Xfree86), on peut noter les spécifications pour

keyboard, keyboard_layout, mouse, monitor, device

Il peut être utile de sauvegarder ce fichier.

5.1.4 Le matériel pour booter

un CD boot.iso :

L'image iso pour démarrer une nouvelle installation se trouve dans le sous répertoire **images** des distributions.

*Attention à la manière de graver le CD **boot.iso** , ce CD DOIT être « bootable ».*

*La commande **dd** permettait la fabrication de disquettes d'amorçage, elle est plutôt utilisée maintenant pour fabriquer un support d'amorçage USB :*

dd if=bootdisk.img of=/dev/sda

(à condition que le bios supporte le démarrage à partir d'un périphérique USB)

L'image qui va permettre le démarrage de l'installation est spécifique à la distribution que l'on veut installer : si on a gravé un CD de boot à partir du répertoire fedora/fc5, on ne pourra pas démarrer une installation de fedora fc7 !

*Il est donc très important de bien repérer le répertoire à partir duquel on a fabriqué le matériel de boot pour pouvoir ensuite indiquer le bon répertoire **NFS** (Network File System) lors de l'installation.*

5.2 Une première installation

- Redémarrez l'ordinateur à partir de la disquette ou du CD
- Choisissez mode **nfs**
- Entrez les paramètres réseaux

Attention, en cas d'erreur à ce niveau, les corrections ne sont pas toujours prises en compte, il faut alors impérativement redémarrer une installation.

– Serveur **linuxline.epfl.ch**

– Répertoire

Fedora : /export/mirror/fedora/f8/i386/os

RedHat Enterprise : /export/RHE/ws-5/update1/x86_32/iso

- Install custom

- Partitionnement : choisir Manually partition DiskDruid, examiner le partitionnement proposé, s'exercer à le modifier
- Use grub : ne PAS cocher use a GRUB password
- Firewall : no firewall (juste pour la première installation)
- Entrez le mot de passe root
- Choix des groupes de paquetage : Select individual packages

NE PAS INSTALLER TOUS LES PAQUETAGES, réfléchir à ce qu'on veut !

*les paramètres utilisés pour l'installation sont stockés automatiquement dans un fichier **/root/anaconda-ks.cfg**
On peut donc ensuite sauvegarder, éventuellement modifier ce fichier, et l'utiliser pour une autre installation en mode kickstart, voir [C](#) à la page 71.*

RHE : ATTENTION

Lors du premier démarrage : SAUTEZ l'étape d'enregistrement proposée par défaut ! Cliquez sur « Tell me why i need ... », et sur l'écran suivant sélectionnez « I can no complete registration ... ». Vous pourrez ensuite accréditer votre machine en utilisant un script qui vous est fourni lorsque vous recevez votre accréditation, ou en utilisant une ligne de commande.

Chapitre 6

L'utilisateur

6.1 Informations et authentications, bases de données locales et centralisées

6.1.1 Locales :

Examen des fichiers `/etc/passwd`, `/etc/shadow`, `/etc/group` à l'aide des commandes `cat` ou `more` ou en « glissant » le fichier dans un éditeur

- `passwd` : différents champs
 - nom
 - emplacement pour le mot de passe crypté (ou renvoi au fichier `/etc/shadow`)
 - numéro d'utilisateur
 - numéro de groupe
 - répertoire initial de travail
 - interpréteur shell
- `group` : différents champs
 - Nom
 - emplacement pour un éventuel mot de passe
 - numéro de groupe
 - éventuelle énumération des membres du groupe

*Un même utilisateur peut appartenir à plusieurs groupes différents, la commande **newgrp** permet de s'identifier comme appartenant à tel ou tel groupe, et de bénéficier des permissions allouées à ces groupes. La commande **groups** permet de voir à quels groupes on appartient*

Les **ACL**, *Access Control Lists*, sont reconnues et utilisables, elles permettent de positionner des permissions avec un grain beaucoup plus fin que les simples niveaux utilisateur/groupe. Voir la section 7.6.3.3 à la page 40.

6.1.2 Centralisées :

Il est possible de chercher des données dans d'autres bases de données que les fichiers locaux.

La plus importante pour l'EPFL : **LDAP**, Lightweight Directory Access Protocol, annuaire pour toute la communauté de l'EPFL

Server : ldap.epfl.ch Base : o=epfl, c=ch

Accès aux données publiques contenues dans LDAP :

<http://bottin.epfl.ch>

Recherche dans la base LDAP en ligne de commande (si le paquetage **openldap-clients** a été installé !)

```
ldapsearch -x -v -h ldap -b 'o=epfl,c=ch' cn='coulmance'
```

```
ldapsearch -x -v -h ldap -b 'o=epfl,c=ch' cn='WinAd'
```

Accès à une personnalisation :

<http://dinfo.epfl.ch/cgi-bin/accountprefs>

vous permet de modifier les paramètres répertoire home et interpréteur de commande de votre compte, après authentification.

L'ordre de consultation pour les principales bases de données est fixé dans un fichier

```
#/etc/nsswitch.conf
```

```
...
```

```
passwd : files ldap
```

```
shadow : files ldap
```

```
group : files ldap
```

```
hosts : files dns
```

6.2 Création de compte

6.2.1 local

utilisation d'un outil graphique au clic souris

Systems settings, Users and Groups

ou appelé en ligne de commande

system-config-users

Accès aux commandes en tant que "root"

On peut, en tant qu'utilisateur, lancer une commande qui nécessite les droits de "root" en faisant précéder cette commande de **sudo**. Il faut cependant que l'utilisateur fasse partie d'un groupe possédant les droits "sudoers".

man sudo

Ajouter l'appartenance au groupe **wheel** d'un utilisateur local
Examiner et modifier avec la commande **visudo**

6.2.2 centralisé

Paramétrage et validation de l'accès à une base LDAP :

utilisation d'un outil graphique au clic souris

Systems settings, Authentication

ou appelé en ligne de commande

system-config-authentication

(au besoin relancer ssh **/etc/init.d/sshd**, et le gdm manager **gdm-restart**)

Création d'un répertoire de travail pour un utilisateur existant dans LDAP

cp -r /etc/skel /repertoire-home/nom-utilisateur

Attribution de ce répertoire à l'utilisateur :

chown -r nom-utilisateur /repertoire-home/nom-utilisateur

On DOIT filtrer les autorisations de connexions en modifiant dans le fichier

/etc/ldap.conf

*la valeur du paramètre **pam_filter**.*

Par exemple pour autoriser les utilisateurs des groupes 10072 et 10073

pam_filter |(gidNumber=10072)(gidNumber=10073)

*Il est possible d'utiliser un mode encrypté pour communiquer avec le serveur LDAP, cf. la case **Use_TLS**. Si on utilise cette option, il faut installer un certificat.*

<http://certauth.epfl.ch/CA/cacert.html>

*Ce certificat est normalement installé dans le répertoire **/etc/openldap/cacerts**, mais on peut configurer le nom d'accès au certificat dans le fichier **/etc/ldap.conf***

Il peut être nécessaire d'ajouter dans le fichier `/etc/ldap.conf` une ligne `nss_initgroups backlink` pour diminuer le temps de login des utilisateurs déclarés dans le serveur LDAP de l'EPFL

Il existe aussi des groupes gérés au niveau du serveur LDAP de l'EPFL, voir <http://dinfo.epfl.ch/groupes>

ATTENTION :si on utilise l'authentification LDAP, la configuration par défaut peut rendre impossible l'accès aux comptes locaux, y compris le compte `root` si l'accès au réseau est défectueux

On peut pallier à ce problème en modifiant le fichier `/etc/pam.d/system-auth` et en ajoutant des options :

```
default=bad authinfo_unavail=ignore success=ok
user_unknown=ignore
```

sur la ligne

```
account [default=bad...] /lib/security/pam_ldap.so
```

Il est **indispensable** d'avoir une **copie de sauvegarde du répertoire `pam.d`** avant toute intervention sur les fichiers contenus dans ce répertoire !

6.3 Environnement

Un utilisateur va pouvoir personnaliser son environnement en paramétrant les variables d'environnement dans des fichiers dits fichiers cachés (dont le nom commence par un point, et qui ne sont « vus » que si on ajoute l'option `-a` à la commande `ls`).

Si on a choisi l'interpréteur de commande `bash` les deux fichiers qui vont permettre de personnaliser l'environnement de l'utilisateur sont :

- `.bash_profile` exécuté à la connexion, personnalisation de la session de travail
- `.bashrc` lu au moment de l'exécution de `.bash_profile` : définition des alias, définitions de fonctions etc.

Les variables d'environnement

Elles véhiculent de l'information entre plusieurs application. Chaque application ou programme peut lire ou modifier la valeur d'une de ces variables, il suffit qu'ils connaissent le nom de la variable pour y accéder.

Tous les systèmes d'exploitation proposent un certain nombre de ces variables qui permettent aux applications d'obtenir des informations sur la configuration du système, par ex. la variable **TMP** fournit le chemin du dossier temporaire. La valeur d'une variable d'environnement peut être propre à chaque utilisateur ; par exemple, la variable **HOME** destinée à fournir le chemin d'accès au répertoire personnel de l'utilisateur contiendra la valeur **/disqueA/titi** pour l'utilisateur titi, et **/disqueC/toto** pour toto. Pour les deux utilisateurs, un programme utilisant la variable **HOME** (le nom de la variable est précédé du \$ si l'on veut indiquer la valeur de la variable) fonctionnera de la même manière, quelque soit l'utilisateur, et quelque soit l'emplacement physique des répertoires personnels. Un utilisateur peut définir et utiliser ses propres variables d'environnement. Il suffit de modifier le fichier **.bashrc** pour définir, par exemple, une variable donnant le chemin d'accès à un répertoire de données :

```
REPertoire_DONNEES=/partage/donnees_a_traiter
```

Afin que cette variable soit utilisable par les programmes lancés par l'utilisateur, il faut exporter cette variable, la rendre publique, en ajoutant

```
export REPertoire_DONNEES
```

Les programmes et autres processus lancés ensuite hériteront de cette variable, et pourront l'utiliser.

Quelques variables d'environnement :

HOME répertoire de connexion de l'utilisateur
USER nom de login de l'utilisateur
PATH chemin de recherche pour les commandes externes
MANPATH chemin de recherche pour les sections de manuel
SHELL type de shell
PRINTER imprimante par défaut.

Pour examiner et positionner les variables d'environnement : **set** et **unset**

Les alias

C'est le mécanisme qui permet de désigner une commande, ou une suite de commandes, par un nouveau nom, le nom d'alias, et d'ajouter des options qui seront automatiquement associées à une commande.

La commande **alias** permet de voir quels sont les alias, et d'en définir de nouveaux. Pour supprimer un alias : **unalias**

Exemple d'alias :

```
alias l.='ls -d .* --color=tty'  
alias ll='ls -l --color=tty'  
alias ls='ls --color=tty'
```

6.4 Fichiers de configuration

6.4.1 Configuration par défaut, générale :

On trouve les fichiers de configuration générale dans le répertoire `/etc`
`/etc/profile`, `/etc/bashrc`, répertoire `/etc/profile.d`

Seul l'administrateur peut les modifier.

Le répertoire `/etc/skel` contient un ensemble pré-défini de fichiers de configuration pour un utilisateur type. C'est pourquoi on le recopie au moment de la création d'un compte utilisateur, voir la section 6.2.2 à la page 21.

6.4.2 Fichiers de configuration de l'utilisateur :

Rappelons que plupart du temps il s'agit de fichiers dits silencieux, leur nom commence par un point, et par défaut ils n'apparaissent pas avec la commande `ls` lancée sans option, voir la section 6.3 à la page 22. Ils sont pris en compte APRES les fichiers de configuration par défaut, et vont permettre à l'utilisateur de personnaliser son environnement :

`~/.bash_profile`, `~/.bashrc`, mais aussi `~/.emacs` si on travaille avec l'éditeur `emacs`, les répertoires `~/.gnome` ou `~/.kde`

Rappel : ~ représente le répertoire home de l'utilisateur)

6.4.3 Quelques essais de modifications

Variable PATH

- Examiner cette variable d'environnement :
`set | grep PATH`
- Modifier la variable d'environnement `PATH` en éditant et en modifiant le fichier `bash_profile` pour ajouter un nouveau chemin, par ex. `/usr/local`
- Faire prendre en compte les nouveaux paramètres :
`source .bash_profile`
- Réexaminer la nouvelle valeur de la variable :
`set | grep PATH`

Personnaliser les couleurs d'affichage de la commande **ls**

- Examiner le contenu du fichier par défaut
/etc/DIR_COLORS
- Recopier ce fichier en fichier de configuration de l'utilisateur
cp /etc/DIR_COLORS ~/.dir_colors
- Examiner la variable d'environnement **LS_COLORS**
set | grep LS_COLORS
- Faire prendre en compte l'évaluation du fichier **.dir_colors**
en ajoutant dans le fichier **~/.bashrc**
eval '/usr/bin/dircolors -b \$HOME/.dir_colors '

- Eventuellement modifier la variable **LS_OPTIONS**
export LS_OPTIONS='--color=auto'
- Modifier le fichier **.dir_colors** comme on veut
- Faire prendre en compte les modifications
source .bashrc
- Tester :
set | grep LS_COLORS
ls

Essayer de créer ou de modifier un alias, et faire prendre en compte cette modification

PATH** est initialisé avec une liste minimale de chemin dans le fichier **/etc/profile** . L'utilisateur pourra compléter cette liste dans son fichier **.bash_profile

***MAIS ATTENTION** le répertoire courant, noté **.**, ne doit pas toujours être systématiquement exploré afin de diminuer le risque d'utilisation de commandes non standard, et non fiables (éventuels leurres).*

Si le répertoire courant est exploré, pour ces mêmes raisons, il est préférable qu'il le soit en dernier.

S'il n'est pas exploré, il ne faudra pas oublier de donner la référence de la commande à exécuter dans le répertoire courant :

***./ma-commande** , et non pas **ma-commande** .*

6.5 Environnement X

Le bureau chargé par défaut pour l'ensemble des utilisateurs est défini dans le fichier `/etc/sysconfig/desktop`.

Un utilisateur peut choisir le bureau qu'il souhaite utiliser :

Commande **switchdesk** (paquetages **switchdesk** et **switchdesk-gui**);
ou bien trouver l'entrée **Desktop switching tool** dans les menus de configuration.

Ou bien, tout à la main : Créer un fichier **fichier** `~/.Xclients` puis définir le chemin complet du gestionnaire que l'on veut appeler dans ce fichier :

`/usr/bin/startkde`, ou bien `/usr/bin/gdm`

On rend ce fichier « exécutable », **chmod +x .Xclients**, et cette nouvelle configuration individuelle sera prise en compte lorsque l'utilisateur lancera la commande **startx**

Ressources X :

Fichiers de configuration des ressources

`/usr/lib/X11/app-defaults` entre autres

6.6 En prime : les scripts, un aperçu

INITIATION

```
#!/bin/bash
# exemple2:test des parametres passes en ligne de commande
#
echo "mon nom est $0"
echo " J'ai $# parametres"
echo " Les voici : $* "
echo " Le premier est : $1"
#
```

APPEL DE FONCTIONS

```
#!/bin/bash
# tstfic, appel de fonction
# entrer sur la ligne de commande des noms de fichiers et de repertoires
#
# Fonction
    ifile() {
        if test -f $f
        then
            return 0
        else
            return 1
        fi
    }
# Corps du script
for i
do
f=$1
echo "$1"
shift
if ifile
then
    echo "$f est un fichier"
else
    echo "$f n'est pas un fichier"
fi
done
#
```

CHAINES DE CARACTERES

```
#!/bin/bash
# Substitution d'un pattern à un autre dans un fichier texte
# i.e., "subst Smith Jones letter.txt".
ARGS=3
E_BADARGS=65 # Wrong number of arguments passed to script.
# Test number of arguments to script (always a good idea).
if [ $# -ne "$ARGS" ]
then
    echo "Usage: 'basename $0' old-pattern new-pattern filename"
    exit $E_BADARGS
fi
#
old_pattern=$1
new_pattern=$2
#
if [ -f "$3" ]
then
    file_name=$3
else
    echo "File \"$3\" does not exist."
    exit $E_BADARGS
fi
#
# Substitution pour TOUTES les occurrences, cf. man sed
sed -e "s/$old_pattern/$new_pattern/g" $file_name
# 's' is, of course, the substitute command in sed,
exit 0 # Succes : returns 0.
```

VARIABLES

```
#!/bin/bash
# Variables: assignment and substitution
a=375
hello=$a
echo hello      # Not a variable reference, just the string "hello".
echo $hello
echo ${hello} # Identical to above.
#
hello="A B C D"
echo $hello      # A B C D
echo "$hello"    # A B C D
# Quoting a variable preserves whitespace.
echo '$hello'    #$hello
# Notice the effect of different types of quoting
exit 0
```

LECTURE D'UN FICHER

```
#!/bin/bash
# Lecture et affichage du fichier /etc/fstab.
File=/etc/fstab
{
read line1
read line2
} < $File
#
echo "Premiere ligne de $File:"
echo "$line1"
echo
echo "Seconde ligne de $File:"
echo "$line2"
exit 0
```

Quelques exemples sur
/net/linuxline/export/mirror/divers/cours/exo_rep

Chapitre 7

Première approche du système de fichier

7.1 Disques physiques, partitions et systèmes de fichiers

7.1.1 Disques

Descripteur de disque : `/dev/hda` ou `/dev/sda`

Sous linux, tout est fichier, toute ressource est vue comme un fichier, exemple du répertoire `/dev`

7.1.2 Partitions :

Un disque physique est en général découpé en partitions, ce sont des divisions fonctionnelles du disque. Des informations sur ces partitions sont obtenues grâce à la commande `df` Examen (et/ou création) des partitions d'un disque :

`fdisk /dev/hda`

`parted`

7.1.3 RAID et LVM

RAID Redundant Array of Independent Disk : RAID 1 « mirroring », RAID 5 « disk array with block-interleaved distributed parity »

LVM Logical Volume Management : permet de s'affranchir des contraintes physiques, et de créer des groupes de volumes pouvant contenir plusieurs disques physiques, ou fragments de disques physiques. Un groupe de volume sera ensuite découpé en **volumes logiques**.

Quelques commandes pour examiner :

- `pvdisk`

- **vgdisplay**
- **lvdisplay**

Une commande utile : **lvmdump**

7.1.4 Système de fichier

Dans chaque partition (ou volume logique), l'administrateur installe un système de fichiers.

Chaque partition peut contenir un et un seul système de fichier, et un système de fichier ne peut être sur plusieurs partitions.

Systèmes de fichiers sous linux : **ext2** , le classique ; **ext3** : systèmes de fichiers « journalisés », ce qui permet de maintenir la cohérence des metadonnées (inodes etc.)

Mais linux connaît également **vfat** , et d'autres !

*Tous les détails sur le système de fichier Linux par défaut, **ext2** à l'adresse <http://e2fsprogs.sourceforge.net/ext2intro.html>*

Les structures de contrôle du système de fichier sont : la VTOC (= label), le superbloc, les blocs de groupe de cylindres, et la table des inodes.

Le système de fichiers est décrit par le superbloc. Etant donné qu'il contient des données critiques, le superbloc est dupliqué. Les copies sont utilisées lorsqu'une erreur disque altère le premier superbloc.

Lors du montage d'un système de fichiers, on copie le superbloc en mémoire, et toutes les opérations de création et d'effacement de fichiers modifient ce superbloc en mémoire. D'où les problèmes d'incohérence lorsque le système n'est pas arrêté « proprement ». La commande **fsck** permet de vérifier l'intégrité d'un système de fichier et de corriger, dans la mesure du possible, les incohérences. Par défaut, elle est lancée au démarrage du système.

Paramétrage possible : cf la 6ème colonne de **/etc/fstab** , cf. également le positionnement de « Maximum mount count »

```
dumpe2fs /dev/sda1 | grep 'ount count'
```

*Le répertoire **lost+found** est un répertoire particulier d'un système de fichier Unix dans lequel l'OS place les fichiers ou les morceaux de fichiers retrouvés ou endommagés lors d'une vérification de l'intégrité d'un système de fichier (**fsck**) A examiner dans le cas où il y a eu des problèmes !*

7.2 Montage

Lorsque le système démarre, le système de fichier root (cad /) est monté. Puis les autres systèmes de fichiers seront montés selon les instructions contenues dans le fichier `/etc/fstab`

```

LABEL=/          /          ext3    defaults    1 1
LABEL=/boot1    /boot      ext2    defaults    1 2
LABEL=/travail  /travail  ext3    defaults    1 2
/dev/hda        /media/cdrecorder auto    pamconsole,exec,
                                   noauto,managed 0 0

```

7.2.1 Montage d'une ressource distante :

On peut également « monter » une partie d'un système de fichier se trouvant physiquement sur le disque dur d'un ordinateur distant : c'est ce qu'on appelle un montage **NFS** (Network File System) :

```
mount linuxline:/export/mirror/redhat /monredhat
```

Ce type de montage est possible

- Si le super-utilisateur de l'ordinateur distant a autorisé cette « exportation »
Examen des fichiers exportés par un ordinateur : **showmount -e nom_machine**
Paramétrage des autorisations d'exportation : fichier `/etc/exports`
- Si linux est capable de reconnaître le type du système de fichier à monter (cf. option `-t` de la commande **mount**)

7.2.2 Automontage :

Lorsqu'on a besoin d'accéder à une ressource, un démon **automount** peut lancer la commande **mount** de façon transparente pour l'utilisateur.

Exemple : **ls /net/linuxline/export/mirror/redhat**

est équivalent à

```
mount linuxline :/export/mirror/redhat \
/net/linuxline/export/mirror/redhat
```

```
ls /net/linuxline/export/mirror/redhat
```

Fichiers de configuration :

```
/etc/auto.master
# Sample auto.master file
# This is an automounter map and it has the following format
# key [ -mount-options-separated-by-comma ] location
# For details of the format look at autofs(5).
#/misc /etc/auto.misc --timeout=60
/net /etc/auto.net
/home /etc/auto.home
```

```
/etc/auto.home
#
coulmance localhost:/maison/coulmance
fabbri localhost:/groupe_dit/fabbri
michele ditsbpc16:/travail/michele
#
```

Pour créer un point de montage : **mkdir point_de_montage**
 Un point de montage n'est rien d'autre qu'un répertoire **VIDE**
 Un montage n'est possible que si le point de montage est **VIDE**

7.3 Gestion de périphériques :

7.3.0.1 Préambule :

On peut également utiliser la commande **mount** pour monter les périphériques amovibles disquettes, cdrom, zip etc. Le point de montage prédéfini est actuellement **/media** (jadis **/mnt**)

Exemple :

```
mount /dev/fd0 /media/floppy
```

ou simplement

```
mount /media/floppy s'il existe une ligne décrivant ce point de montage dans le fichier /etc/fstab.
```

```
umount /media/floppy pour libérer la ressource Attention : il faut que le point de montage ne soit plus utilisé pour pouvoir « démonter » !
```

7.3.1 Gestion dynamique des périphériques, udev et HAL

Ou : les mystères de « l'apparition de l'icône sur le bureau » enfin dévoilés.

A lire à tête reposée ...

Le kernel (à partir de la version 2.6) détecte l'ajout d'un nouveau périphérique. Il charge les modules nécessaires au bon fonctionnement du nouveau matériel. Toutes les caractéristiques techniques que le kernel obtient en « interrogeant » le matériel sont disponibles dans `/sys`. Tous les fichiers trouvés dans cette partie de l'arborescence sont générés en direct par le kernel, c'est une gestion totalement dynamique des périphériques qu'on ajoute/supprime à chaud.

```
ls /sys
```

Pour que les programmes extérieurs au kernel puissent accéder à un périphérique, il faut « nommer » le périphérique en mettant un fichier dans `/dev` (sous linux TOUT est fichier, y compris le matériel).

C'est **udev** qui se charge de créer dynamiquement les fichiers dans `/dev`. **udev** est un programme qui va être informé par le kernel qu'un nouveau matériel est connecté et qu'il faut lui donner un nom (=fichier) dans `/dev`, à travers des règles définies dans `/etc/udev`

```
ls -lr /dev/disk/bypath
udevinfo -a -p /sys/block/sda
udevtest /sys/block/hda
```

Avec fedora6 ou RHEL5 :

on peut examiner ce qui se passe lorsqu'on connecte une clé USB

```
udevmonitor -env
```

Une fois que le nouveau périphérique est devenu utilisable à travers un fichier, il reste à avertir les utilisateurs que ce matériel est disponible. C'est **HAL**, **H**ardware **A**bstraction **L**ayer, qui va s'en charger en collectant et en rendant accessibles les informations

```
lshal
```

Ensuite il faut faire parvenir les informations collectées par **HAL** à l'utilisateur. **HAL** va « poster » un message, message qui sera pris en compte par d'autres programmes utilisateurs grâce à **D-bus** (qui permet aux applications de communiquer entre elles)

Finalement des logiciels de l'environnement graphique

gnome-volume-manager par exemple, vont être capable grâce à **DBUS** de savoir quand un CDROM est introduit, de quel type de CD il s'agit, et quelle est l'application à lancer : lire un cd audio par exemple, et jouer la musique, ou bien monter le périphérique et ouvrir nautilus pour montrer les fichiers.

```
ls -l /etc/hal/device.d
```

HAL est également capable de lancer le programme **fstab-sync**, qui va consulter **/etc/fstab** et voir si il y a une entrée pour le nouveau périphérique. Si il n'en voit pas alors il va en ajouter une à la fin du fichier **fstab**. Les modifications sont conservées en mémoire, et l'entrée ajoutée dans **fstab** est supprimée si le périphérique est débranché. On a donc un **fstab** dynamique.

7.4 Structure des répertoires

Les fichiers sont organisés en structure arborescente et hiérarchique, qu'on peut représenter sous la forme d'un arbre renversé.

tout est fichier, toute ressource est vue comme un fichier, voir la section 7.1.1 à la page 31

Les répertoires sont des fichiers permettant d'organiser la structure en arborescence.

Le point de départ de l'arborescence est représenté par **/**, c'est ce qu'on appelle le répertoire racine, ou root.

Répertoires à la racine :

- **/bin** commandes essentielles, utilisables en mode monutilisateur, entre autres **bash**, **csch** ...
- **/boot** tout ce qui est nécessaire au boot,
- **/dev** fichiers spéciaux = points d'entrée vers les périphériques
- **/etc** fichiers de configuration
- **/home** répertoire ou point de montage pour les répertoires home des utilisateurs
- **/lib** bibliothèques nécessaires aux programmes de **/bin** et **/sbin**
- **/media** point de montage pour les périphériques amovibles
- **/mnt** ancien point de montage pour ces périphériques amovibles
- **/opt** répertoire prévu pour des commandes optionnelles
- **/proc** toutes les infos. sur l'état du système
- **/root** répertoire home du compte administrateur (doit être accessible même en fonctionnement niveau 1)
- **/sbin** commandes essentielles utilisées plutôt par le compte administrateur.
- **/sys** voir la section 7.3.1 à la page 35
- **/tmp** fichiers temporaires générés par le système et les processus
- **/usr** point de montage d'une arborescence contenant des programmes et bibliothèques partagées ; **/usr/bin**, **/usr/sbin**, **/usr/lib**, **/usr/share**
...
- **/var** fichiers susceptibles de changer, journaux des activités du système(logs), files d'attente pour les impressions, applications journalisées(cron) etc.

/dev/null et /dev/zero : les données écrites sur ces fichiers virtuels spéciaux sont éliminées. On peut aussi se servir de ces pseudo-fichiers pour effacer le contenu d'un fichier :

cat /dev/null > fichier_à_effacer

Mais ATTENTION ces fichiers ne DOIVENT JAMAIS être effacés.

Commandes pour les recréer « au cas où » :

mknod -m 666 /dev/null c 1 3

mknod -m 666 /dev/zero c 1 5

chown root :root /dev/null /dev/zero

Chaque fichier ou répertoire peut être caractérisé par un nom absolu :

ls /etc/init.d

ou par un nom relatif, c'est à dire exprimé par rapport au répertoire courant.

Par ex., à partir de **/home/user1** :

ls ../../etc/init.d

Rappel : ~ représente le répertoire home de l'utilisateur

7.5 Quelques commandes relatives aux fichiers

- **cd** pour se déplacer dans l'arborescence
- **pwd** pour savoir quel est le répertoire courant
- **mkdir** pour créer un répertoire
- **rmdir** pour effacer un répertoire
- **cp** copie d'un fichier ou répertoire
- **mv** déplacement d'un fichier ou répertoire
- **rm** effacement d'un fichier
- **touch** création d'un fichier vide, ou modification des dates de création d'un fichier existant
- **cat/more** examiner le contenu d'un fichier
- **find** recherche d'un fichier
- **du** affichage de l'espace occupé par un fichier ou un répertoire
- **ls** affichage des caractéristiques de fichiers et répertoires

Nombreuses options communes, dont

-R, -r : récursif

-i : interactif

7.5.1 Quelques exemples d'utilisation

7.5.1.1 ls

Nombreuses options, parmi les principales :

- a affiche tous les fichiers, y compris les fichiers dits silencieux
- l donne les principales caractéristiques des fichiers, le type (fichier ou répertoire), les permissions, le propriétaire, la taille, la date de la dernière modification, etc

Obtenir le listing des répertoires d'un utilisateur :

```
ls -l | grep ^d
```

Obtenir le listing des fichiers dits silencieux, dont le nom commence par un point :

```
ls -a | grep "^\."
```

7.5.1.2 find

Rechercher (et pouvoir éliminer automatiquement) certains fichiers :

```
find . \( -name core \) -size +1c -print
```

```
find . -type f -mtime +7 -print0 | xargs -0 rm -f
```

7.5.1.3 du

Afficher l'espace occupé par des utilisateurs :

```
cd /home
```

```
ls | xargs du -sk
```

7.5.2 Les liens

Ils permettent d'associer plusieurs noms à un même fichier, d'accéder à un fichier par différents chemins d'accès.

Création d'un lien : **ln -s ce-qui-existe nom-symbolique**

Visualisation des liens : **ls -l**

7.6 Permissions de base :

Un répertoire/fichier appartient à un utilisateur, et à un groupe.

Les utilisateurs permettent ou non l'accès aux objets leur appartenant.

Seul le super utilisateur peut passer outre les permissions.

- 3 niveaux d'identifiants :

1. Propriétaire **User**, utilisateur qui crée l'objet
2. Groupe **Group**, groupe auquel appartient le propriétaire du fichier
3. Autres **Others**, utilisateurs autres que le propriétaire et les membres du groupe

- 3 types de permissions :
 1. Lecture **read**, permet d'afficher le contenu d'un répertoire ou d'un fichier
 2. Ecriture **write**, permet de modifier le contenu d'un fichier ou d'un répertoire, de créer, de copier, ou de supprimer un fichier.
 3. Exécution **execute**, permet d'exécuter un programme, permet de pouvoir entrer dans un répertoire.

7.6.1 Modification des permissions :

chmod

Utilisation de la notation symbolique :

Classes :

- **u**ser utilisateur
- **g**roup groupe
- **o**ther autres
- **a**ll tous

Permissions :

- **r**ead
- **w**rite
- **x**execute

Opérateurs :

- **=** affectation
- **+** ajout
- **-** suppression

Exemple : **chmod g+rw, o-w afac**

7.6.2 Appartenance

chown : changer le propriétaire d'un fichier **uniquement si on est root**.

Exemple :

chown coulmance:dit-sb un_fichier

*ATTENTION : vérifier où l'on est dans l'arborescence avant de lancer une commande **chmod** ou **chown** en mode récursif !*

7.6.3 Droits spéciaux sur un fichier : sticky bit, setuid, setgid

7.6.3.1 sticky bit :

Un utilisateur ayant le droit d'écrire dans un répertoire peut aussi effacer un fichier dans ce répertoire. Lorsqu'on positionne le « sticky bit » pour ce répertoire, il faut, pour pouvoir effacer un fichier dans ce répertoire, avoir le droit d'écrire, et

être propriétaire du fichier qu'on veut effacer, ou propriétaire du répertoire, ou administrateur.

Exemple : le répertoire **/tmp** est « protégé » par un « sticky bit »

Appliqué à un fichier, le sticky bit prend un sens totalement différent : un fichier exécutable ayant le sticky bit positionné reste en mémoire même après la fin de son exécution, pour pouvoir être relancé plus rapidement. Seul l'administrateur a le droit de positionner le sticky bit sur un fichier exécutable.

7.6.3.2 setuid et setgid :

Si un fichier exécutable est « setuid », il va être exécuté avec les droits du propriétaire, et non avec les droits de l'utilisateur qui le lance.

Exemple : la commande **/usr/bin/passwd**, qui permet à l'utilisateur de modifier son mot de passe est « setuid root », c'est à dire qu'elle permettra à un utilisateur de modifier le fichier **/etc/passwd**, fichier dans lequel seul root a un accès en écriture.

De même un programme exécutable « setgid » s'exécutera avec les droits du groupe auquel il appartient.

Exemple : **/usr/sbin/sendmail**

7.6.3.3 Positionnement de ces droits spéciaux :

- sticky bit : **chmod o+t nom_de_repertoire**
- setuid : **chmod u+s nom_de_fichier**
- setgid : **chmod g+s nom_de_fichier**

7.7 ACL (Access Control List)

Les ACL permettent de positionner les permissions avec une granularité plus fine que la granularité de base. Elles permettent d'étendre le nombre d'utilisateurs et de groupes ayant des droits sur un même fichier.

Le système de fichier ext3 supporte les ACL, à condition qu'il soit « monté » avec l'option **acl** (voir la section 7.1.4 à la page 32), ce qui est actuellement une option par défaut pour RedHat et Fedora.

Commandes : **getfacl**, **setfacl**

Exemple : **setfacl -m user:titi:rwX un_fichier**

smbcacls si on travaille sur des ressources « montées » par Samba, voir la section 11.2.1 à la page 58

7.8 Archivage

la commande **star** (ancienne commande = **tar**) permet de créer une archive en regroupant une partie d'une arborescence de fichiers dans un fichier unique. Ce

fichier peut être créé sur le disque dur, sur une bande de sauvegarde etc.

7.8.1 Options :

- **c** générer une archive
- **t** visualiser le contenu d'une archive
- **x** extraire une archive
- **v** verbose (afficher des informations sur les opérations)
- **z** compresser ou décompresser
- **f** nom du fichier

7.8.2 Utilisation

```
créer en mode "verbose" une archive qui sera stockée (option -f
) dans un fichier /export/etc.tar de toute l'arbores-
cence /etc
star -cvf ./etc.tar /etc

lister le contenu du fichier archive etc.tar
star -tvf ./etc.tar

extraire le contenu de l'archive etc.tar
star xvf etc.tar
```

La taille du fichier etc.tar est identique à la taille de l'arborescence originale, mais on peut compresser ce fichier (outils gzip).

Chapitre 8

Niveaux de fonctionnement, processus

8.1 Démarrage

Le disque d'un PC peut être partitionné. On peut faire coexister plusieurs systèmes d'exploitation sur un même disque dur, un par partition.

Un et un seul système d'exploitation est actif à la fois, c'est celui qui est chargé en mémoire au moment du démarrage de l'ordinateur.

Le premier secteur d'un disque contient ce qu'on appelle le Master Boot Record (MBR) qui décrit le partitionnement du disque. Il est aussi à l'origine du chargement, on dit aussi amorce ou boot du système actif.

Le processus de démarrage d'un PC commence par l'exécution automatique d'un programme de chargement situé dans le BIOS (Basic Input Output System), qui réside dans la mémoire morte (ROM).

C'est ce programme qui va charger le programme de chargement du MBR..

Le chargeur peut offrir la possibilité de choisir le système d'exploitation que l'on veut effectivement démarrer.

Le programme de chargement, à son tour, chargera le secteur de boot du système d'exploitation choisi.

A l'heure actuelle : tant sous Windows que sous linux les chargeurs sont multiboot, donc on peut toujours avoir le choix de démarrer Windows ou linux

De préférence : installer d'abord Windows, puis linux, qui prendra en compte l'existence de Windows, et permettra un multi-boot.

Chargeur linux multiboot : actuellement **GRUB** (voir la section [D](#) à la page [75](#)) a remplacé **LILO**

Examen des messages de démarrage :

- **dmesg**
- fichiers **/var/log**
- outil graphique **system logs**

8.2 Arrêt du système

commande **shutdown** :

Options

-h halt

-r reboot

shutdown -h now

shutdown -r now

Il existe des risques d'incohérence au niveau du système de fichiers si le système n'est pas arrêté proprement, si la copie en mémoire du super-bloc n'a pas été correctement réécrite.

*Update du super bloc : Commande **sync***

*Lorsqu'il y a des problèmes, le système tente de réparer les incohérences au reboot suivant en lançant la commande **fsck** (ou **e2fsck**).*

*On peut ensuite aller regarder dans le répertoire **lost+found** pour essayer de récupérer ce qui n'a pas été reconstitué (répertoire toujours créé lorsqu'on crée un système de fichier).*

8.3 Niveaux de fonctionnement

Le chargement du noyau linux va permettre l'initialisation des structures de données internes, le repérage des périphériques présents, le chargement des modules nécessaires.

Le système linux offre plusieurs niveaux de fonctionnement. A chacun d'eux va correspondre un certain nombre de services pour les utilisateurs. Le rôle de la commande **init**, qui a comme argument le niveau de fonctionnement choisi, est d'activer les processus nécessaires à ce niveau.

Niveaux de fonctionnement :

0 Hors service : l'alimentation peut alors être coupée sans danger

1 Mode mono-utilisateur (pour administration système)

2 Mode multi-utilisateurs mais sans les fonctionnalités réseau.

3 Mode multi-utilisateurs avec réseau

4 Inutilisé

5 X11

6 Mise hors service et redémarrage : sert durant le redémarrage du système à partir d'un niveau de fonctionnement (2, 3, 4, 5). Le système passera ensuite au niveau 0.

Lors de la phase de démarrage, **init** doit savoir à quel niveau placer le système. Pour cela, il recherche dans le fichier **/etc/inittab** la ligne qui configure ce niveau par défaut :

id :5 :initdefault :

Pour connaître le niveau de fonctionnement : **runlevel**

Pour changer de niveau de fonctionnement : **telinit**, ou **init**

telinit 3 pour entrer en mode « réseau » si on est en niveau 1

telinit 5 pour entrer en mode graphique

***telinit** est surtout utile lorsqu'on a du démarrer en mode « single user » et que l'on teste au fur et à mesure le démarrages des différents services*

*à noter : **Ctrl-Alt-F1 ... Ctrl-Alt-F6** : passage en consoles texte*

***Ctrl-Alt-Backspace** pour « tuer » X*

8.4 démarrer en mode single user :

A partir de la fenêtre de démarrage
e pour entrer en mode édition
e pour éditer la ligne **kernel /boot/vmlinuz ...**
ajouter **single** à la fin de cette ligne
b pour booter

=> Mise en évidence des problèmes de sécurité. . .

8.5 Scripts de démarrage

8.5.1 examen du fichier `/etc/inittab`

Le processus **init** commence d'abord par exécuter le script `/etc/rc.d/rc.sysinit`, qui contient des commandes de contrôle et d'initialisations pour les différents processus :

Initialisation du **PATH**, initialisation du nom de la machine (hostname), du nom de domaine éventuellement, initialisation du **swap**, montage du pseudo-système de fichier `/proc`, etc.

Puis il exécute les scripts spécifiques au niveau d'exécution demandé, scripts dont l'énumération se trouve dans les répertoires

`/etc/rc.d/rc0.d, rc1.d, ..., rc6.d` (correspondant aux niveaux de fonctionnement). On remarquera qu'en fait ne se trouvent dans ces répertoires que des liens symboliques vers les scripts à lancer ; tous les scripts concernant le démarrage et l'arrêt d'un service sont regroupés dans le répertoire `/etc/init.d`. On peut décider des scripts qu'on veut lancer au démarrage de linux, et à quel niveau de fonctionnement on veut démarrer ce service

8.5.2 Examen des scripts du répertoire `/etc/init.d`

```
more /etc/init.d/lpd
```

A tout moment, pour stopper, démarrer, relancer, examiner l'état d'un service :

```
/etc/rc.d/init.d/lpd status
```

```
/etc/rc.d/init.d/lpd stop
```

```
/etc/rc.d/init.d/lpd start
```

```
/etc/rc.d/init.d/lpd restart
```

Equivalent à la commande : **service** nom_du_service

Il n'est donc pas nécessaire d'arrêter et relancer le système lorsqu'on modifie un paramètre de configuration d'un service, il suffit de relancer ce service.

menu System settings, Server settings, Services : outil graphique

chkconfig offre un outil simple en ligne de commande pour maintenir la hiérarchie du répertoire `/etc/rc.d` tout en évitant aux administrateurs de manipuler les différents liens symboliques de ce répertoire.

Pour afficher les paramètres des services pour chaque niveau d'exécution :

```
chkconfig -list
```

Pour modifier les démarrages de services au prochain reboot :

```
chkconfig -level n service on
```

```
chkconfig -level n service off
```

Il est important de bien contrôler les services lancés au démarrage, en particulier tous les services pouvant mettre en cause la sécurité de la machine

8.6 En cas de problèmes

Pas d'affolement !

Essayer de démarrer en mode **single user** , voir la section 8.3 page 45

Essayer de démarrer avec un cd de boot : Après une installation, ou après une mise à jour du kernel, il est sage de fabriquer un CD pour redémarrer au cas où il y a un problème de bootloader.

mkbootdisk :pour fabriquer une image iso qui permettra de redémarrer un noyau 2.6.9-55.EL

mkbootdisk -device rescue-2.6.9-55.iso -iso 2.6.9-55.EL

Pour savoir quel est le noyau actif : **uname -a**

Essayer d'utiliser un « Linux Live CD » : Mini-linux sur un CD pour dépannages, démonstration ,etc. (DSL, Knoppix, Basilisk etc.)

http://fr.wikipedia.org/wiki/Damn_Small_Linux :

<http://distrowatch.com/dwres.php?resource=cd>

A partir du CD 1 de la distribution, ou de la disquette de boot au prompt, **boot>** entrez **linux rescue**

Vous aurez ensuite la possibilité de monter par défaut toutes les partitions détectées, ou bien de n'en monter aucune.

A partir d'une disquette de démarrage **Grub** (voir la section D à la page 75) retrouver ce qu'il faut pour démarrer le noyau existant sur le disque, restaurer le MBR

8.7 Processus

8.7.1 Examen de l'état du système

Menu system tool, **system monitor** outil graphique

uptime, **free**, **ps**, **top** commandes en ligne

8.7.2 Tuer un processus

kill -signal PID

- **PID** numéro du processus qu'on veut tuer et qu'on a repéré par ex. grâce à **ps -el**
- **signal** signal qu'on envoie au processus, les plus utilisés étant **SIGHUP** et **SIGKILL**

SIGHUP ou 1 : relire les fichiers de configuration **SIGKILL** ou 9 : « Crève charogne ! »,

kill -9 à éviter autant que possible, mais fort utile dans les cas désespérés ! On ne peut tuer que les processus qui vous appartiennent, heureusement, attention lorsqu'on lance cette commande comme administrateur !

8.7.3 Modes d'exécution d'un processus

- avant-plan, ou foreground, ou **fg**
- arrière-plan, ou background, ou **bg**

Exécution séquentielle en avant-plan

Le shell crée un processus fils qui va exécuter la commande, puis se met en attente de la terminaison du fils avant d'afficher à nouveau l'invite de commande, ou de passer à la commande suivante : le processus s'exécute en avant-plan.

Exécution parallèle en arrière-plan

Le shell crée un processus fils qui va exécuter la commande, puis sans attendre la terminaison du fils, affiche le numéro du **job** ainsi que le numéro d'identification du premier processus fils, et affiche son invite de commande, ou passe à la commande suivante.

Syntaxe d'exécution en arrière-plan

commande &

Pour passer de foreground à background : **Ctrl-z**, puis **bg**

Chapitre 9

Gestion des packages

9.1 RPM

RedHat PackageManager : outil puissant, très utile

Cf une documentation <http://linuxline.epfl.ch/Doc/>

- installation de nouveaux packages
rpm -ivh nom_package
Option intéressante -aid permet d'ajouter les paquets nécessaires à la résolution de dépendances, si c'est possible.
- mise à jour :
rpm -Fvh nom_package
important pour appliquer les patches de sécurité
- désinstallation :
rpm -e nom
- interrogations :
 - **rpm -qa** pour voir les packages installés
 - **rpm -ql nom** pour voir la liste des fichiers dépendant d'un package installé
 - **rpm -qlp nom_package** pour voir la liste des fichiers contenus dans un package
 - **rpm -q -scripts nom** pour voir les scripts de pré et post installation.
 - **rpm -qc nom** pour voir les fichiers de configuration liés à un package
 - **rpm -qf nom_de_fichier** pour savoir à quel package appartient un fichier
 - **rpm -qip nom_package** pour obtenir des informations sur un package
- Importation de clés : **rpm -import *GPG-KEY***

*Un cas particulier : la mise à jour du noyau A l'heure actuelle l'utilisation de la commande **rpm** permet de mettre à jour le noyau et de modifier automatiquement le fichier **grub.conf**. Par défaut l'ancien noyau est conservé, on pourra donc toujours redémarrer avec l'ancienne version en cas de problème.*

9.2 yum

Yellowdog Updater Modified (un équivalent de **apt-get** pour Debian) Une couche au dessus de RPM, permet d'installer et de mettre à jour plus facilement, et en paramétrant les sites à partir desquels on veut rechercher les mises à jour ou les nouveaux paquetages.

9.2.1 Configuration

fichiers **/etc/yum.conf**

*la ligne **gpgcheck=1** signifie que par défaut les paquets doivent être « signés » et qu'on doit avoir la clé permettant de vérifier l'authenticité de ces paquets.*

Répertoire **/etc/yum.repos.d** : il contient les fichiers qui vont permettre de s'adresser à certains serveurs.

/etc/yum.repos.d/fedora.repo

[base]

name=Fedora Core \$releasever - \$basearch - Base

#baseurl=ftp://linuxline/fedora/fc6/os

mirrorlist=ftp://linuxline/fedora/yum-fc6/local-base

enabled=1

gpgcheck=1

/etc/yum.repos.d/fedora-updates.repo

[updates-released]

name=Fedora Core \$releasever - \$basearch - Released Updates

#baseurl=ftp://linuxline/fedora-updates/5/i386

mirrorlist=ftp://linuxline/fedora/yum-fc6/local-updates

enabled=1

gpgcheck=1

*Ne pas oublier d'importer les clés pour la vérification des paquets, surtout avec l'option **gpgcheck=1***

On peut préciser dans le fichier de configuration **/etc/yum.conf** que certains paquetages ne DOIVENT PAS être mis à jour.

exclude=mysql*

Sous Fedora :
Aller sur le site <http://rpm.livna.org>
Installer un paquetage livna-release
Examiner les changements apportés dans le répertoire **yum.repos.d**

9.2.2 Utilisation en ligne de commande :

- Examiner :
 - **yum list installed**
 - **yum list updates**
 - **yum list gcc***
 - **yum info**
- Mettre à jour :
 - **yum update nom_paquetage**
 - **yum update**
- Installer :
 - **yum install nom_paquetage**
- Désinstaller :
 - **yum remove nom_paquetage**

9.2.3 Nouveaux outils graphiques

A partir de Fedora 5 et de RHEL5 on dispose de boîtes à outils graphiques pour utiliser **yum**

pirut (add/remove software) pour gérer l'installation et la désinstallation de paquetages

pup (Package Updater) pour gérer les mises à jour

Comparaison des outils **rpm** et **yum** : Recopier un paquetage, par exemple php-5 à partir de linuxline
Essayer de l'installer avec la commande **rpm -ivh php-5***
Puis utiliser la commande **yum install php***

*Les distributions Debian (et donc ubuntu's) utilisent un autre format pour gérer les paquetages (.deb) et donc un autre outil : **Advanced Package Tool** pour gérer les paquetages. Cependant actuellement il existe des outils **apt** dans les distributions RedHat, et des outils **rpm** dans les distributions Debian.*

Chapitre 10

NFS

Network File System

10.1 Côté serveur

autoriser l'exportation

Fichier `/etc/exports` :

```
#  
#Exemple d'exportation  
#  
#Rep_local_exporté      à_qui(comment)  
#  
/export/sic-ii          suntest1.epfl.ch(rw)  
/export/mirror          *.epfl.ch(ro,insecure)  
#
```

relancer les services NFS

```
/etc/init.d/nfs restart
```

vérifier

```
showmount -e localhost
```

Attention si on a activé un firewall, voir la section [11.2](#) à la page [56](#)

10.2 Côté client

vérifier les ressources exportées

```
showmount -e serveur_distant
```

Essayer un montage

```
mount machine_distante_serveur:/rep_exp/mnt/rep_test
```

automontage

Voir la section 7.2.1 à la page 33 Si c'est bon, on peut essayer un auto-montage :

Fichier **/etc/auto.master**

```
/home auto.home --timeout 100
```

```
/net auto.net
```

/home et **/net** : répertoires sous lesquels seront monté les ressources d'après les règles définies dans **auto.home** et **auto.net**

Ces points de montage doivent être VIDES

#exemple de fichier /etc/auto.home:

#

```
user1 host1:/export/&
```

```
user2 localhost:/rep_local/&
```

Chapitre 11

Sécurité

11.1 Introduction à SELinux :

<file:///net/linuxline/export/RHE/doc/rhe/RH-DOCS/rhel-rg-fr-4/ch-selinux.html>

<http://www.redhat.com/docs/manuals/enterprise/RHEL-4-Manual/selinux-guide/>

« SELinux fournit un système de contrôle d'accès obligatoire (ou MAC, de l'anglais **Mandatory Access Control**) intégré au noyau Linux. Sous un système standard de contrôle d'accès discrétionnaire Linux (ou DAC, de l'anglais **Discretionary Access Control**), une application ou un processus exécuté en tant qu'utilisateur (UID ou SUID) reçoit la permission de l'utilisateur sur des objets tels que des fichiers, des sockets et d'autres processus. L'exécution d'un noyau MAC SELinux permet de protéger le système contre des applications malveillantes ou défectueuses qui peuvent endommager ou détruire le système. SELinux définit les droits d'accès et de transition de chaque utilisateur, application, processus et fichier du système. SELinux gouverne alors les interactions de ces sujets et objets à l'aide d'une politique de sécurité qui spécifie le degré de rigueur ou de souplesse d'une installation donnée de Red Hat Enterprise Linux. »

Les applications sont « confinées » : le comportement d'une application peut être étudié, quels sont les ports utilisés, quels sont les fichiers utilisés, en écriture, en lecture, etc ; des règles de fonctionnement « normal » de cette application sont ainsi générées, et toute déviation à ces règles sera soit empêchée si le mode est **enforcing**, soit signalée si le mode est **permissive**.

Pour vérifier le statut de SELinux : **sestatus -v**

Pour examiner les droits d'accès SELinux des fichiers : **ls -Z (user :role :type)**

11.2 Firewall

Activation d'un firewall : Boîte à outil graphique :

Menu System Settings, Security level

system-config-securitylevel

Fichier : **/etc/sysconfig/iptables**

Examiner le fichier **/etc/sysconfig/iptables**
Activer le firewall en autorisant **ssh**
Examiner à nouveau le fichier

A tout moment on peut stopper l'ensemble des contrôles du :

/etc/init.d/iptables stop Voir la section 8.5.1 à la page 46

11.2.1 Paramétrage plus fin, exemple d'un serveur NFS3

Avec le firewall actif : toutes les demandes NFS sont rejetées. Que faire pour autoriser les accès sur le port NFS ?

La commande **rpcinfo -p localhost** permet de visualiser les associations port/services. Les informations se trouvent dans le fichier **/etc/services**.

MAIS un certain nombre de services peuvent utiliser des ports **dynamiquement** alloués par le **portmapper**. Si l'on veut autoriser l'accès à ces services au travers d'un système protégé par un firewall, il est indispensable d'empêcher l'allocation dynamique et de contraindre le **portmapper** à allouer des ports fixes pour ce service.

On va paramétrer cette allocation statique dans un fichier **/etc/sysconfig/nfs** qui, s'il existe, est lu par défaut au moment du démarrage du service **NFS**, on peut vérifier en examinant le script de démarrage **/etc/init.d/nfs**.

Création du fichier

/etc/sysconfig/nfs

#fichier /etc/sysconfig/nfs:

MOUNTD_PORT=4002

LOCKD_TCPPOINT=4001

LOCKD_UDPPOINT=4001

STATD_PORT=4000

#

redémarrer le service

/etc/init.d/nfs restart

configurer des nouvelles règles

: Il faut ouvrir les ports :

- **portmapper 111**
- **nfs 2049**
- **mountd 4000 à 4003**

D'où les règles suivantes :

```
-A RH-Firewall-1-INPUT -p tcp -m tcp --dport 111 -j ACCEPT
-A RH-Firewall-1-INPUT -p tcp -m tcp --dport 2049 -j ACCEPT
-A RH-Firewall-1-INPUT -p tcp -m tcp --dport 4000:4003 -j ACCEPT
-A RH-Firewall-1-INPUT -p udp -m udp --dport 111 -j ACCEPT
-A RH-Firewall-1-INPUT -p udp -m udp --dport 2049 -j ACCEPT
-A RH-Firewall-1-INPUT -p udp -m udp --dport 4000:4003 -j ACCEPT
```

à introduire avec un éditeur de texte dans le fichier **/etc/sysconfig/iptables**
Et miracle, les ressources **NFS** seront alors disponibles même si l'on active le **firewall**

*ATTENTION toujours faire une copie du fichier
/etc/sysconfig/iptables avant et après les modifica-
tions*

Chapitre 12

SAMBA

C'est un utilitaire qui va permettre d'intégrer une machine OS/linux dans un réseau Wins.

12.1 Serveur

Depuis la version 3, Samba va également permettre l'intégration d'une machine linux dans Active Directory, et l'interrogation du serveur kerberos d'Active Directory pour authentifier les utilisateurs, ce qui devrait permettre une unification de la gestion des utilisateurs : authentification avec LDAP côté linux, avec Kerberos côté Active Directory, l'identité, le numéro d'utilisateur, le mot de passe étant alors identiques.

Pour permettre à des PC sous Windows d'accéder à des ressources disponibles sur un système linux, il faut décrire les ressources mises à disposition : où, à qui etc.

Cf. des exemples de configurations dans

`/net/linuxline/export/mirror/divers/samba3`

Fichier `/etc/smb.conf`

```
[homes]
comment = Home {%}u
browseable = no
writable = yes
```

```
[reputil]
comment = util
path = /export/util
writable = yes
browseable = yes
```

Positionnement du type d'authentification dans /etc/samba/smb.conf :
security=ADS si on veut utiliser Active Directory
 Ne pas oublier de configurer également l'authentification **kerberos**

Configuration **kerberos** :

Outil graphique **system-config-authentication**

Ou édition et modification du fichier **/etc/krb5.conf**)

Exemple :

Realm SCX.INTRANET.EPFL.CH

KDCs SCX.INTRANET.EPFL.CH :88

12.1.0.1 Intégration d'une serveur samba dans Active Directory

- Positionner **default_realm** (fichier **/etc/krb5.conf**) sur le domaine dans lequel on veut faire entrer le serveur, par exemple :
default_realm = SCX.INTRANET.EPFL.CH
- Positionner également la valeur de realm dans le fichier **/etc/samba/smb.conf**
realm = SCX.INTRANET.EPFL.CH
- Obtenir un ticket kerberos :
kinit coulmance@SCX.INTRANET.EPFL.CH
- Inscrire le serveur en précisant (option **-U**) le nom d'un utilisateur ayant des droits d'administration dans Active Directory
net ads join -U coulmance@SCX.INTRANET.EPFL.CH
- vérification : **net ads testjoin**

*La commande **net ads join** génère beaucoup de messages d'erreurs, pas toujours cohérents ni explicites, et parfois malgré les messages d'erreurs le serveur a bien été intégré dans Active Directory, ce dont on s'aperçoit d'autant moins facilement que le temps de rafraichissement d'Active Directory est . . .parfois un peu long*

12.2 Client, les outils

12.2.1 Graphiques

Dans **konqueror**, browser de **kde**, on peut indiquer le protocole **smb** dans la demande d' **url** (Uniform Resource Locator)

```
smb://SERVERS
smb://linuxline
```

Pour accéder à une ressource d'un PC dans Active Directory avec **konqueror** on pourra entrer dans l' URL

```
smb://SCX.INTRANET.EPFL.CH\coulmance@siipc20/coulmance
```

C'est à dire

```
smb://domaine\user_name@host_name/share_name
```

Konqueror 3 permet également un accès en protocole **webdav** (Web-based Distributed Authoring and Versioning), et on pourra ensuite copier/coller etc.

```
webdav://linux.epfl.ch/site/linuxline/
```

12.2.2 Lignes de commande

12.2.2.1 Examiner les ressources d'une serveur Samba

```
[coulmance@siipc22 util] smbclient -L linuxline
```

```
Domain=[SERVERS] OS=[Unix] Server=[Samba 3.0.7-1.3E.1]
```

```
Sharename Type Comment
```

```
-----
fedora-doc Disk fedora doc
fedora-6 Disk fedora fc6
fedora-7 Disk fedora fc7
RHEL-WS5 Disk RHEL Workstation
RHEL-DOC Disk RHEL Documentation
```

```
Domain=[SERVERS] OS=[Unix] Server=[Samba 3.0.9-1.3E.13.2]
```

```
...
```

12.2.2.2 Montages de ressources d'un serveur Windows

Commande en ligne en tant que **root** :

ATTENTION aux différences de syntaxe selon les versions !

- à l'ancienne :

```
mount -t smbfs -o username=coulmance/SCX.intranet.epfl.ch \
//scxdata/coulmance$ /home/coulmance/scxdata
```

- A la moderne : (Common Internet File System)
`mount -t cifs -o user=SCX.intranet.epfl.ch/coulmance \
//scxdata/coulmance$ /home/coulmance/scxdata`

mount.cifs est une application relativement « jeune » ; il y a encore pas mal de problèmes, pas mal d'instabilité, des difficultés de « négociations », et des bugs qui ne se manifestent pas de la même manière selon le serveur qui fournit la ressource.

*Truc : **modinfo** commande pour connaître la version du module **cifs***

Trouvez le numéro de version du module **cifs.ko**

Le module pam_mount Il permet le montage automatique et transparent de ressources Windows au moment de la connexion des utilisateurs, avec demande d'un seul mot de passe qui sera ensuite transmis aux différents modules de **PAM** et permettra une succession d'authentification, unix, puis ldap, puis kerberos ... Exemple de configuration de `/etc/security/pam_mount.conf` pour activer un montage :

```
volume * smb scxdata &$ /home/&/scxdata uid=&,gid=10072 - -
```

Chapitre 13

Connexions à distance

*AUCUN accès sur une machine de l'EPFL n'est possible depuis l'extérieur; voir **DIODE**
<http://dit.epfl.ch/page51041.html>
Certaines machines pouvant être « ouvertes » dans certaines conditions et sur demande, la forteresse n'est cependant pas inexpugnable!*

13.1 SSH, Secure SHell

13.1.1 ssh et scp

Permet les connexions à distance avec encryption :

Commande

ssh -l login_name nom_machine_distante

exit pour quitter

La redirection X n'est pas autorisée par défaut. C'est à dire qu'on n'a pas la possibilité de lancer en ligne de commande à partir de la fenêtre terminal de la machine distante, lorsqu'on est connecté, une commande permettant d'ouvrir une application X, par ex. **xclock** sur l'écran de la machine sur laquelle on travaille. L'autorisation de la redirection X se fait :

- côté client : fichier **/etc/ssh/ssh_config ForwardX11 yes**
- côté serveur : fichier **/etc/ssh/sshd_config X11Forwarding yes**

Même principe de fonctionnement pour la commande **scp** qui va permettre de copier d'une machine à l'autre :

scp un_fichier utilisateur@machine_distante:/chemin_acces/

Ou dans l'autre sens, bien sûr, d'une machine distante à un compte local. L'option **-r** permet la copie récursive d'un répertoire avec l'ensemble des sous-répertoires.

13.1.2 ssh et les clés

La commande **ssh-keygen** permet de générer des clés privées et publiques. Ces clés sont stockées dans un sous répertoire `~/.ssh`. On peut ensuite recopier la clé publique **id_dsa.pub** (pour la version 1 du protocole ssh) ou **id_rsa.pub** (pour la version 2 du protocole SSH) dans un fichier **authorized_keys** dans le sous répertoire `~/.ssh/` sur la machine distante.

- Générer des clés
- Examiner les variables d'environnement liées à SSH
- Tester la commande **ssh-add**

On peut ensuite ajouter la commande **ssh-add** dans les commandes à lancer au démarrage d'une session Gnome (ou KDE), ce qui évitera l'entrée d'un mot de passe ou d'une « passphrase » lorsqu'on se connectera avec **ssh**

Examiner **Preferences, More preferences, session**

13.2 Autres

putty, winscp : pour se connecter à une machine linux depuis Windows

13.3 Terminal distant

Voir [E](#) à la page 79

Côté serveur : **vncserver**

Côté client : **vnc**, ou **tsclient**

Ou bien **X-Win32** disponible sur DistriLog.

13.4 VPN

Virtual PrivateNetwork

<http://network.epfl.ch/vpn/>

Annexe A

Organisation linuxline.epfl.ch

A.1 Répertoires de base

A.1.1 fedora

`/export/mirror/fedora/f(c)n` (n : numéro de version)

`/iso` : images des CD

`/os` : distribution

*Ces deux répertoires peuvent se trouver dans des sous-répertoires **i386** et **x86_64***

A.1.2 RedHatEnterprise

`/export/RHE/ws-n` ou `/as-n` (n : numéro de version)

`/iso` : images des CD

`/i386` : distribution

`/update n` : mises à jour

*les répertoires **iso** et **os** peuvent se trouver dans des sous-répertoires **x86_32** et **x86_64***

`/export/RHE/ws-3/epfl` : quelques fichiers exemples pour **kickstart**

A.2 Mises à jour Fedora

Configuration de **yum**, voir la section [9.1](#) à la page [50](#)

`/export/mirror/fedora-updates`

F-Errata :

/master : tous les derniers patches

/patches : TOUS les patches

A.3 Accès

- montage NFS **/net/linuxline**
- Samba, à partir de l'explorateur réseau sous Windows, group **Servers** ,
linuxline
- <http://linuxline> pour RHEL à partir de la version 4, pour fedora à partir de la version 7
- <ftp://linuxline> pour les distributions fedora

Annexe B

Editeur VI

Editeur VI : pas très convivial, mais si utile qu'il est indispensable d'en connaître les bases.

C'est un éditeur

- Interactif
- Visuel
- Utilisant un fichier temporaire (le document original est conservé jusqu'à ce qu'une demande explicite d'écriture sur disque soit faite)

B.1 Fonctionnement

- Mode commande
- Mode insertion
- Mode ligne

B.1.1 Mode commande

Il est invoqué par défaut, et on retourne en mode commande grâce à ESC à partir du mode insertion. Il permet les déplacements, les suppressions, les déplacements et copies de parties de fichiers.

B.1.2 Mode insertion

Il permet l'ajout, l'insertion de texte.

Il est invoqué à partir du mode commande par la frappe d'un caractère.

- **a** append : ajout **après** le curseur
- **A** Append : ajout **à la fin** de la ligne
- **i** insert : insère **avant** le curseur
- **I** Insert : insère **au début** de la ligne
- **o** open : ajoute une ligne **après** la ligne courante
- **O** Open : ajoute une ligne **avant** la ligne courante

B.1.3 Mode ligne

Il est invoqué à partir du mode commande par la frappe du caractère **:**

Ce caractère **:** apparaît sur la dernière ligne de l'écran.

Il permet entre autres les écritures sur le disque, l'exécution d'une commande, et la sortie !

- **:q!** annulation des modifications apportées au texte, et sortie de vi.
- **:wq** écriture sur le disque du fichier modifié et sortie de vi.
- **:r nom_fichier** insertion d'un fichier

Il est possible de configurer certaines variables qui vont modifier le fonctionnement de VI :

- soit en les positionnant lorsqu'on est en mode ligne :
 - **:set nu** permet la numérotation des lignes
 - **:set nonu** supprime la numérotation des lignes
- soit en les déclarant dans un fichier `~/ .exrc`

Exemple de paramétrage des variables dans `.exrc` :

#numérotation des lignes

set nu

#affichage des modes

set showmode

B.1.4 Quelques indications sommaires

- Effacement :
 - **x** effacement du caractère sur lequel est positionné le curseur.
 - **dw** effacement du mot sur lequel est positionné le curseur (curseur sur le premier caractère)
 - **dd** effacement de la ligne sur laquelle est positionnée le curseur.
- Remplacement :
 - **r** permet le remplacement du caractère sur lequel se trouve le curseur, puis retour au mode commande.
 - **R** remplacement de plusieurs caractères. ESC pour revenir en mode commande lorsqu'on a terminé les remplacements.
- Copies et déplacements :
 - **yy** copie de la ligne sur laquelle se trouve le curseur.
 - **p** placer la ligne copiée au-dessous de la ligne sur laquelle est le curseur.
 - **P** placer la ligne copiée au-dessus de la ligne sur laquelle est le curseur.

Pour déplacer : **dd** (`ndd` place n ligne dans le buffer), puis **p** ou **P**.

B.2 Mode d'emploi

Merci à Jean-Claude De Giorgi !

Voir le document `vi_command_reference.sxw`
sur `/net/linuxline/export/mirror/divers/cours`

B.3 Autres éditeurs

- nano, ou pico.
- emacs particulièrement adapté aux besoins des programmeurs

Annexe C

kickstart

Exemple de fichier Kickstart

```
# Dans un premier temps on peut ajouter la directive interactive ce qui
# permettra une lecture des infos. et une possibilite de modifier
#
# INSTALLATION EN MODE INTERACTIF
interactive

#System language
lang en{\_}US
#Language modules to install
langsupport en{\_}US fr{\_}FR --default en{\_}US

# CLAVIER
keyboard fr{\_}CH-latin1
# autre clavier courant :keyboard us
# SOURIS
mouse generic3ps/2

#System Timezone
timezone Europe/Zurich

# A MODIFIER OBLIGATOIREMENT LORS DE LA PREMIERE CONNEXION
# OU EN INTERACTIF DURANT INSTALLATION
# Root password
rootpw rootepfl

#Reboot after installation
#Ici inactif CAR IL FAUT avoir le temps d' ENLEVER LA DISQUETTE
# reboot
```

```
#Install Red Hat Linux instead of upgrade
install

# INSTALLATION VIA NFS
# CE QUI IMPLIQUE QUE LE RESEAU FONCTIONNE
# ET QUE LA MACHINE POSSEDE UNE ADRESSE IP
#
nfs --server 128.178.50.24 --dir /export/RHE/ws/update/i386

#System bootloader configuration
bootloader --location=mbr
#Clear the Master Boot Record
zerombr yes

#!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
# ATTENTION ATTENTION ATTENTION PARTITIONNEMENT
# PREMIER EXEMPLE : ON PART DE RIEN
# ON VA CREER SUR LE PREMIER DISQUE IDE LES 3 PARTITIONS MINIMALES
# ON CREE SUR CE DISQUE 3 PARTITIONS INDISPENSABLES
# A PEU PRES STANDARD POUR LA TAILLE

# DECOMMENTER CE QUI SUIIT POUR CE PREMIER CAS
#clearpart --drives hda
#part /boot --ondisk=hda --fstype ext2 --size 100
#part swap --ondisk=hda --size 512
#part / --ondisk=hda --fstype ext3 --size 3000

# SECOND EXEMPLE : IL EXISTE DEJA DES PARTITIONS
# ON LES CONNAIT, ON VA LES UTILISER
# FORMATTER CELLES INDISPENSABLES
# NE PAS TOUCHER AUX AUTRES
# DECOMMENTER ET ADAPTER CE QUI SUIIT
#part /boot --onpart hda1 --fstype ext2
#part swap --onpart hda5 --size 512
#part / --onpart hda2 --fstype ext3
#part /export --onpart hda3 --noformat
#
#!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
# ATTENTION ATTENTION ATTENTION CONFIG RESEAU
# A MODIFIER OBLIGATOIREMENT
#
# SI ON UTILISE UNE CONFIG. STATIQUE :
```

```
# REMPLACER LES xxx et yyy PAR LES BONNES VALEURS
#network --bootproto static --ip 128.178.xxx.yyy --netmask 255.255.255.0
--gateway 128.178.xxx.1 --nameserver 128.178.15.7
#
# SI on peut utiliser un serveur DHCP :
# DECOMMENTER LA LIGNE network --bootproto dhcp
# COMMENTER LA LIGNE network --bootproto static
#network --bootproto dhcp

#!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
# UTILISATION DU SERVICE LDAP
# TRES PRATIQUE
#System authorization information
auth --useshadow --enablemd5 --enableldap --enableldapauth --ldapservers
ldap.epfl.ch --ldapbasedn o=epfl,c=ch

# SINON : COMMENTER LA LIGNE AU DESSUS,DECOMMENTER LA SUIVANTE
#auth --useshadow --enablemd5

# Firewall configuration
firewall --enable

#Video
xconfig --depth 16 --resolution 640x480

# Choix des paquetages :
# Moins crucial, on peut toujours enlever/ajouter ensuite

%packages --resolvedeps

@GNOME Desktop Environment
@KDE Desktop Environment
@Editors
@Engineering and Scientific
@Graphical Internet
@Office/Productivity
@Sound and Video
@Graphics
@Authoring and Publishing
@Server Configuration Tools
@Windows File Server
@Development Tools
@Kernel Development
@Administration Tools
```

```
@System Tools
@Printing Support
```

```
# SCRIPT EXECUTE APRES INSTALLATION
%post
# MODIFIER CERTAINS FICHIERS
#
# HOSTS.ALLOW ET HOSTS:DENY
cp /etc/hosts.allow /etc/hosts.allow.orig
cat <<EOF>> /etc/hosts.allow
ALL : localhost.localdomain
ALL : .epfl.ch
portmap: 128.178.0.0/255.255.0.0
EOF
#
cp /etc/hosts.deny /etc/hosts.deny.orig
cat <<EOF>> /etc/hosts.deny
ALL : ALL
EOF
#
#
rpm --import /usr/share/doc/rpm-*/RPM-GPG-KEY
rpm --import /usr/share/doc/rpm-*/RPM-PGP-KEY

# RHN
# INDIQUER votre cl'e d'activation puis d'ecommentez
#rhnreg{\_}ks --activationkey=nnnnnn
# configuration de la cliente RHN pour site proxy EPFL
#wget -O- http://rh-proxy/pub/conf{\_}proxy | bash
```

Annexe D

grub

<http://doc.polynum.org/howto/grub/grub-howto.html>

<file:///net/linuxline/export/mirror/Docs/cours/grub>

Le GRUB, **G**rant **U**nified **B**ootloader, possède de nombreux avantages par rapport à LILO, ne serait-ce que la possibilité d'utiliser un shell : le GRUB est un mini-système d'exploitation. Le programme, par exemple installé sur une disquette, permet, sans noyau disponible sur celle-ci, d'aller en chercher un, et de lui passer la main.

Le GRUB a été défini, dès l'origine, comme un outil dont l'objectif est d'être un gestionnaire d'amorçage universel, puissant et flexible. Il dispose d'un mini-shell autorisant l'exploitation de toutes les commandes internes. A l'aide d'un CD contenant le GRUB, vous pouvez démarrer n'importe quel poste dès lors que vous parvenez à retrouver, sur l'un des disques, un noyau. Et si le poste n'a pas de disque ? C'est qu'il doit démarrer par le réseau, et cela, le GRUB sait le faire aussi.

D.1 Spécifier une partition et des fichiers dans le GRUB :

Le gestionnaire d'amorçage doit charger en mémoire des données, et doit donc pouvoir accéder à des champs de caractères placés quelque part : localement sur des périphériques, ou téléchargeables. En gros, ces données sont réparties sur des champs de données (les disques par exemple), dans ces champs de données éventuellement dans des subdivisions (les partitions), auxquelles on doit pouvoir accéder soit en indiquant la liste des blocs à lire, soit, si le champ de données est équipé d'informations (un système de fichiers), en fournissant le nom de cet ensemble de blocs dans le système de fichiers.

La désignation d'un ensemble de données sera donc composée de la désignation du périphérique de stockage (le disque par exemple), de la partition dans le disque, et du nom de fichier dans le système de fichiers si ce système existe, ou de la liste des blocs.

D.1.1 Désignation des périphériques de stockage

Pour le GRUB, il n'existe pas de différence entre des disques IDE ou SCSI, ils seront tous appelés de la même façon : **hdN** où **N** est un entier. Le premier disque trouvé est désigné par **hd0**, le deuxième par **hd1**, etc... La numérotation démarre de **0**.

Pour désigner une unité de données on place son nom entre parenthèses : **(hd0)** désigne le premier disque dur.

Une disquette sera désignée par le préfixe **fd**, avec le même système de numérotation : **(fd0)** désigne donc la première unité de disquette.

Enfin, un périphérique spécial, **(nd)**, existe lorsque le réseau a été initialisé (il s'agit du network device).

D.1.2 Désignation des partitions

Les unités physiques (disques) sont partitionnées. Pour désigner une partition sur une unité physique, on fait suivre la désignation de l'unité physique par une virgule et l'identifiant de la partition.

Ainsi **(hd0,0)** désigne la première partition sur le premier disque dur. A noter que l'écriture **(hd0)** est tout à fait légitime : dans ce cas on désigne le périphérique (disque) dans son intégralité. Cette notation est utile pour accéder, par exemple, au MBR : le MBR n'appartient à aucune partition. De même, dans le cas d'une disquette qui n'est pas partitionnable (en fait l'intégralité de la disquette se confond avec une partition), on utilisera également la syntaxe **(fd0)**.

D.1.3 Désignation des fichiers par leur nom

Dès lors que le périphérique est désigné, si le système de fichiers est supporté, vous y avez accès en appelant les fichiers par leur nom.

Ainsi :

(hd2,4)/vmlinuz

va désigner le fichier **vmlinuz** situé à la racine du système de fichiers existant sur la 5ème partition (première partition logique) du troisième disque dur.

ATTENTION ! Les systèmes de fichiers ne sont pas montés. Donc, si dans la `fstab` cette partition était d'habitude montée sous `boot`, le nom n'est pas `/boot/vmlinuz`, la racine pour ce système de fichiers est... la racine, pas le point de montage hypothétique

D.1.4 Désigner un périphérique par défaut

Plutôt que d'avoir à re-désigner, à chaque commande, le périphérique, on peut signifier au GRUB le périphérique par défaut.

root (hd0,0)

stipule donc que le système de fichiers par défaut est celui existant sur la première partition du premier disque dur.

D.2 Quelques exemples de commandes

D.2.1 Commandes de base pour amorcer un noyau

- Dire au GRUB qu'il doit charger un noyau situé à tel emplacement :
kernel (hd0,0)/tartempion-linux root=/dev/hda2

ATTENTION! Les éventuels paramètres passés au noyau, ici root=/dev/hda2 sont destinés à celui-ci ; la syntaxe pour les périphériques est donc celle compréhensible par ce noyau.

- Puis lui dire de passer la main à cet OS :
boot

D.2.2 Retrouver sur quelle partition sont les fichiers d'amorçage

```
grub> find (hd0,0) /grub/grub.conf
```

D.2.3 Restaurer un secteur de boot sur un disque dur

- Spécifier sur quelle partition de quel disque sont les fichiers boot
grub> root (hd0,1)
- Installer GRUB sur le MBR du disque 0
grub> setup (hd0)
- Quitter **grub> quit**

D.3 Exemple de fichier /etc/grub.conf

```
# grub.conf generated by anaconda
#
# Note that you do not have to rerun grub after making changes to this file
# NOTICE: You have a /boot partition. This means that
#           all kernel and initrd paths are relative to /boot/, eg.
#           root (hd0,0)
#           kernel /vmlinuz-version ro root=/dev/sda2
#           initrd /initrd-version.img
#boot=/dev/sda
default=0
timeout=15
splashimage=(hd0,0)/grub/splash.xpm.gz
#hiddenmenu
```

```
title Red Hat Enterprise Linux AS (2.6.9-42.EL)
    root (hd0,0)
    kernel /vmlinuz-2.6.9-42.EL ro root=LABEL=/ rhgb quiet
    initrd /initrd-2.6.9-42.EL.img
title Red Hat Enterprise Linux AS (2.6.9-55.0.2.ELsmp)
    root (hd0,0)
    kernel /vmlinuz-2.6.9-55.0.2.ELsmp ro root=LABEL=/ rhgb quiet
    initrd /initrd-2.6.9-55.0.2.ELsmp.img
```

grub.conf generated by anaconda

Note that you do not have to rerun grub after making changes to this file

NOTICE: You do not have a /boot partition. This means that all kernel and initrd paths are relative to /, eg.

A ajouter pour un Dual boot, si Windows est sur le second disque :

```
title Windows 2000
map (hd0) (hd1)
map (hd1) (hd0)
rootnoverify (hd1,0)
chainloader +1
```

Annexe E

vncviewer et X-win32

E.1 La base, vnc, Virtual Network Computer :

E.1.1 Côté linux :

Les fichiers de configuration sont dans un répertoire `$HOME/.vnc`

E.1.1.1 La commande `vncpasswd` :

Elle permet à un utilisateur de fixer le mot de passe qui sera demandé lors d'une connexion. Ce mot de passe est stocké dans le répertoire `~/.vnc`.

***ATTENTION**, ce mot de passe est parfois stocké en clair, non encrypté, il faut veiller tout particulièrement aux droits associés au répertoire `.vnc`.*

E.1.1.2 Un fichier `xstartup`

permet de personnaliser la session X qui sera exécutée.

Exemple :

```
#!/bin/sh
# Uncomment the following two lines for normal desktop:
unset SESSION{\\_}MANAGER
exec /etc/X11/xinit/xinitrc
[ -x /etc/vnc/xstartup ] && exec /etc/vnc/xstartup
[ -r $HOME/.Xresources ] && xrdp $HOME/.Xresources
xsetroot -solid grey
vncconfig -iconic &
xterm -geometry 80x24+10+10 -ls -title "$VNCDESKTOP Desktop" &
startx &
```

E.1.1.3 La commande `vncserver`

permet de démarrer le service.

E.1.1.4 La commande `vncserver -kill :n`

permet de stopper le service ; **n** représente le numéro de l'écran. **ATTENTION** à l'espace obligatoire entre `-kill` et `:`.

E.1.2 Côté windows :

Logiciels possibles : RealVNC , ou WinVNC ; commande `vncviewer`

E.2 X-win32 :

E.2.1 Côté linux :

Autoriser le protocole XDMCP en modifiant le fichier
`/etc/X11/gdm/gdm.conf`
`XDMCP enable=true`

Il faut évidemment être en « runlevel »5.

E.2.2 Côté windows :

Ouvrir le port 16001 sur le firewall Windows (sinon le logout ne se fait pas.)

E.3 Outils graphiques sous linux, reposant sur VNC

E.3.1 Serveurs

E.3.1.1 Serveur vino pour Gnome

Vino est un serveur vnc pour Gnome.

Configuration :

- Outil graphique : **Applications, preferences, remote desktop**
- Ligne de commande :
`gconftool-2 -s -t bool`
`/desktop/gnome/remote_access/enabled true`
- Edition du fichier :
`/home/.gconf/desktop/gnome/remote_access/%gconf.xml`

L'administrateur peut modifier les configurations par défaut de l'environnement Gnome

```
gconftool-2 --direct --config-source \  
xml :readwrite :/etc/gconf/gconf.xml.defaults \  
--type bool \  
--set /desktop/gnome/remote_access/enabled true
```

E.3.1.2 Serveur krfb pour KDE

Pour KDE : KDE Desktop Sharing, mêmes principes que **vino**

E.3.2 Clients

Un outil client pour KDE, qu'on peut aussi utiliser sous Gnome, il suffit d'avoir installer le paquetage **kdenetwork**.

Accessible à partir de :

Applications, Internet, Remote Desktop Connection

Ou bien appeler la fenêtre graphique avec la commande **krdc**

Rappel : avec VNC, les communications ne sont pas chiffrées. On peut sécuriser en utilisant un « tunnel »SSH

Annexe F

CD-DVD commandes en lignes

F.1 Obtenir l'adresse d'un graveur :

F.1.0.1 Interne :

```
cdrecord dev=ATA -scanbus
...
>scsibus0:
    0,0,0  0) 'HL-DT-ST' 'RW/DVD GCC-4481B' 'E106' Removable CD-ROM
    0,1,0  1) *
.....
```

Le graveur sera alors identifié par **dev=ATA :0,0,0**

F.1.0.2 Port USB :

```
cdrecord -scanbus
...
>scsibus1:
    1,0,0 100) 'PLEXTOR ' 'DVDR PX-716A ' '1.02' Removable CD-ROM
    1,1,0 101) *
...
```

Le graveur sera alors identifié par **dev=1,0,0**

F.2 commande **cdrecord**

- Graver une image iso : **cdrecord -v dev=n,n,n /chemin/image.iso**
De nombreuses options permettent de fixer de nombreux paramètres : la vitesse, **speed=nnn** , **-dao** (Disk at once) ou **-tao** (track at once), etc.
Mode multi session : option **-multi**
Très utile pour tester avant de graver : option **-dummy**
- Effacer un CD réinscriptible : **cdrecord dev=n,n,n blank=fast**

F.3 Créer une image iso :

```
mkisofs -R -o mon_image.iso /chemin/repertoire_a_stocker
```

-R : Rock Ridge protocole pour les systèmes de fichiers iso9660
-o : nom de l'image iso qui va être créée
/chemin/repertoire_a_stocker : chemin du répertoire qui fournit le contenu de l'image iso.

F.4 Autres utilitaires

F.4.1 paquetage dvd+rwtools

Il fournit des outils supplémentaires, entre autres :

```
dvd+rw-mediainfo /dev/scd0
```

fournit des indications sur le type de media présent dans un périphérique
/dev/scd0

F.4.2 growisofs

Pour un support DVD, on peut aussi utiliser la commande **growisofs** pour fabriquer l'image et la graver.

Il est préférable de **TESTER** avant de graver, en utilisant l'option **-dry-run**

Exemple :

```
growisofs -dry-run -dvd-compat -Z /dev/scd0 -J -R  
/chemin/repertoire
```

- option **-Z** pour graver une **première** session
- option **-dvd-compat** : **ATTENTION** cette option ferme le disque, on ne pourra pas faire ensuite de multi-session (option **-M**, cf. ci-dessous), mais elle offre une grande compatibilité ensuite avec les lecteurs DVD.
- option **-M** : multisession, permet d'ajouter une nouvelle session aux précédentes, tant que l'on n'a pas utilisé l'option **-dvd-compat**

Exemple pour des multisessions :

```
growisofs -Z /dev/scd0 -J -R /chemin/premier_repertoire  
growisofs -M /dev/scd0 -J -R /chemin/repertoire_suivant
```

Annexe G

Quelques raccourcis clavier

Quelques combinaisons pour la console Linux :

Alt-F1 à F6 : Passer à d'autres pseudo-terminaux

Ctrl-Alt-F1 à F6 : Passer à d'autres pseudo-terminaux (X-window)

Alt-F7 : Retourner à X-window

Ctrl-Alt-moins : Changer la résolution de l'écran dans X-window

Ctrl-Alt-plus : Changer la résolution de l'écran dans l'autre sens dans X-window

Ctrl-Alt-Backspace : Arrêter X-window

Alt-X, Alt-C, Alt-V : Combinaisons de touches usuelles Windows/Mac pour Couper, Copier, Coller avec la touche Ctrl- qui est remplacée par ces Alt- dans certain programmes.

Quelques liens :

<http://www.novell.com/cool solutions/tip/2289.html>

Une référence pour d'autres jeux de clavier :

Linux essential keyboard shortcuts and sanity commands

<http://linux-newbie.sunsite.dk/html#6.Linux%20Shortcuts%20and%20Commands|outline>

Quelques extraits, sans traduction !

<Ctrl><Alt><F1> Switch to the first text terminals. Under Linux you can have several (6 in standard setup) terminals opened at the same time. This is a keyboard shortcut, which means : "press the control key and the alt key, hold them. Now press <F1>. Release all keys."

<Ctrl><Alt><Fn> (n=1..6) Switch to the nth text terminal. (The same could be accomplished with the rarely used command chvt n. "chvt" stands for "change virtual terminal"). In text terminal (outside X), you can also use <Alt><Fn> (the key <Ctrl> is not needed).

- <Ctrl><Alt><Fn> (n=7..12)** Switch to the nth GUI terminal (if a GUI terminal is running on screen n-1). On default, the first X server is running on terminal 7. On default, nothing is running on terminals 8 to 12—you can start subsequent X server there.
- <Tab>** (In a text or X terminal) Autocomplete the command if there is only one option, or else show all the available options. On newer systems you may need to press <Tab><Tab>. THIS SHORTCUT IS GREAT, it can truly save you lots of time.
- <ArrowUp>** (In a text or X terminal) Scroll and edit the command history. Press <Enter> to execute a historical command (to save on typing). <ArrowDown> scrolls back.
- <Shift><PgUp>** Scroll terminal output up. This works also at the login prompt, so you can scroll through your bootup messages. The amount/usage of your video memory determines how far back you can scroll the display. <Shift><PgDown> scrolls the terminal output down.
- <Ctrl><Alt><+>** (in X-windows) Change to the next X-server resolution (if you set up the X-server to more than one resolution). For multiple resolutions on my standard SVGA card/monitor, I have the following line in the file /etc/X11/XF86Config (the first resolution starts on default, the largest resolution determines the size of the "virtual screen") :
- ```
Modes "1024x768" "800x600" "640x480" "512x384" "480x300"
 "400x300" "1152x864"Z
```
- Of course, first I had to configure the X server, either by using Xconfigurator, xf86config, or manually by editing the file /etc/X11/XF86Config, so that it supports the above resolutions (mostly the matter of uncommenting the line that defines my video chipset, and specifying the synchronization frequencies my monitor supports). XFdrake (Mandrake configuration utility) can do it from GUI. See also the commands xvidtune and xvidgen.
- <Ctrl><Alt><->** (in X-windows) Change to the previous X-server resolution.
- <Ctrl><Alt><Esc>** (in X-windows, KDE) Kill the window I am going to click with my mouse pointer (the pointer changes to something like a death symbol). Similar result can be obtained with the command xkill (typed in X-terminal). Useful when an X-window program does not want to close (hangs ?).
- <Ctrl><Alt><BkSpc>** (in X-windows) Kill the current X-windows server. Use if the X-windows server cannot be exited normally.
- <Ctrl><Alt><Del>** (in text terminal) Shut down the system and reboot. This is the normal shutdown command for a user at the text-mode console. Don't just press the "reset" button for shutdown !
- <Ctrl>c** Kill the current process (works mostly with small text-mode applications).

- 
- <Ctrl>d** (pressed at the beginning of an empty line) Log out from the current terminal. See also the next command.
- <Ctrl>d** Send [End-of-File] to the current process. Don't press it twice else you also log out (see the previous command).
- <Ctrl>s** Stop the transfer to the terminal.
- <Ctrl>q** Resume the transfer to the terminal. Try if your terminal mysteriously stops responding. See the previous command.
- <Ctrl>z** Send the current process to the background.
- exit** Logout. I can also use `logout` for the same effect. (If you have started a second shell, e.g., using `bash`, this command will make you exit the second shell, and you will be back in the first shell, not logged out. Then use another `exit` to `logout`.)
- reset** Restore a screwed-up terminal (a terminal showing funny characters) to default setting. Use if you tried to "cat" a binary file. You may not be able to see the command as you type it, but it still will work.
- <MiddleMouseButton>** Paste the text which is currently highlighted somewhere else. This is the normal "copy-paste" operation in Linux. It a fast and powerful supplement to the widely-known GUI "copy-paste" menu-based operation. (It doesn't work inside older versions of Netscape which use the Mac/MS Windows-style "copy-paste" exclusively. It does work in the text terminal if you enabled "gpm" service using "setup". It also works inside any dialog boxes, etc.–really convenient !) It is best used with a Linux-ready 3-button mouse (Logitech or similar) or else set "3-mouse button emulation". The `<MiddleMouseButton>` is normally emulated on a 2-button mouse by pressing both mouse buttons simultaneously.
- `~` (tilde character) My home directory (normally the directory `/home/my_login_name`). For example, the command `cd ~/my_dir` will change my working directory to the subdirectory "my\_dir" under my home directory. Typing just "cd" alone is an equivalent of the command "cd ~". I keep all my files in my home directory.
- (dot )** Current directory. For example, `./my_program` will attempt to execute the file "my\_program" located in your current working directory.
- (two dots )** Directory parent to the current one. For example, the command `cd ..` will change my current working directory one level up.
- Some additional KDE keyboard shortcuts (useful, but non-essential)
- <Alt><Tab>** Walk through windows. To walk backwards : `<Alt><Shift><Tab>`
- <Ctrl><Tab>** Walk through desktops. To walks backwards :  
`<Ctrl><Shift><Tab>`
- <Ctrl><Esc>** Show the table of processes running on my system. Allow me to kill any of the processes I started (or send other signals to them)

- <Alt><F1> Access the K-menu ("Equivalent to MS Windows "Start" menu)
- <Alt><F12> Emulate the mouse using the arrow keys on the keyboard.
- <Alt><LeftMouseButton> Drag a window to move it. Normally, I move a window by dragging its top title bar, but occasionally I manage to get it off the screen. With this shortcut, I can drag by any part of the window.
- <Alt><PrintScreen> Take a snapshot of the current window into the clipboard.
- <Ctrl><Alt><PrintScreen> Take a snapshot of the entire desktop into the clipboard.
- <Ctrl><Alt><l> Lock the desktop.
- <Ctrl><Alt><d> Toggle hide/show the desktop (great to hide the Solitaire game when your boss walks in).

# Index

- [/proc](#), [13](#)
- [/swap](#), [13](#)
- [/var](#), [14](#)
  
- [ACL](#), [20](#), [40](#)
- [Active Directory](#), [59](#)
- [alias](#), [23](#)
- [apt](#), [51](#)
- [automount](#), [33](#)
  
- [background](#), [48](#)
- [boot](#), [45](#)
- [bureau](#), [4](#)
  - [Gnome](#), [8](#)
  - [gnome](#), [8](#)
  
- [cifs](#), [61](#)
- [commandes](#)
  - [alias unalias](#), [23](#)
  - [apropos](#), [9](#)
  - [arp](#), [15](#)
  - [cat](#), [10](#), [19](#), [37](#)
  - [cd](#), [10](#), [37](#)
  - [cdrecord](#), [83](#)
  - [chkconfig](#), [46](#)
  - [chmod](#), [39](#), [40](#)
  - [chown](#), [21](#)
  - [cp](#), [10](#), [37](#)
  - [dd](#), [16](#)
  - [df](#), [15](#)
  - [diff](#), [10](#)
  - [dmesg](#), [44](#)
  - [du](#), [37](#)
  - [fdisk](#), [15](#)
  - [find](#), [37](#)
  - [free](#), [15](#), [47](#)
  - [fsck](#), [32](#)
  - [gdm-restart](#), [21](#)
  - [getfacl,setfacl](#), [40](#)
  - [grep](#), [10](#)
  - [groups](#), [19](#)
  - [head](#), [10](#)
  - [id](#), [10](#)
  - [ifconfig](#), [15](#)
  - [kill](#), [47](#)
  - [konqueror](#), [9](#)
  - [locate](#), [10](#)
  - [ls](#), [10](#), [22](#), [37](#)
  - [lvmddump](#), [32](#)
  - [makewhatis](#), [9](#)
  - [man](#), [9](#)
  - [mkdir](#), [37](#)
  - [modinfo](#), [62](#)
  - [more](#), [10](#), [19](#), [37](#)
  - [mount](#), [33](#), [54](#)
  - [mv](#), [37](#)
  - [newgrp](#), [19](#)
  - [parted](#), [15](#)
  - [pipe](#), [11](#)
  - [ps](#), [47](#)
  - [pvdisplay,vgdisplay,lvdisplay](#), [31](#)
  - [pwd](#), [10](#), [37](#)
  - [redirection](#), [11](#)
  - [rm](#), [10](#), [37](#)
  - [rmdir](#), [37](#)
  - [rpcinfo](#), [56](#)
  - [scp](#), [63](#)
  - [script](#), [11](#)
  - [service](#), [46](#)
  - [sestatus](#), [55](#)
  - [set unset](#), [23](#)
  - [showmount](#), [33](#), [53](#)
  - [shutdown](#), [44](#)

- smbacls, 40
  - smbclient, 61
  - ssh, 63
  - ssh-add, 64
  - ssh-keygen, 64
  - sshd, 21
  - sudo, 7, 21
  - sync, 44
  - system-config-authentication, 21
  - system-config-securitylevel, 56
  - tail, 10
  - tar, 40
  - telinit, 45
  - top, 47
  - touch, 37
  - umount, 34
  - uname, 47
  - uptime, 47
  - visudo, 21
  - whatis, 9
  - who, 10
  - whoami, 10
- DHCP, 15
- distribution, 2
- environnement, 22, 23
- fichiers de configuration, 24
- .ssh, 64
  - auto.home, 54
  - auto.master, 34, 54
  - auto.net, 54
  - cachés, 22, 24
  - DNS resolv.conf, 15
  - exports, 33, 53
  - fstab, 15, 32
  - group, 19
  - ifcfg, 15
  - inittab, 45
  - iptables, 56
  - ldap.conf, 21
  - network, 15
  - nfs, 56
  - nsswitch.conf, 20
  - passwd, 19
  - services, 56
  - shadow, 19
  - skel, 21
  - smb, 59
  - ssh, 63
  - system-auth, 22
  - X11, 15
  - yum, 50
- firewall, 17, 55
- foreground, 48
- gateway, 15
- grub, 17, 43, 47, 75
- HAL, 35
- kerberos, 59
- kickstart, 17, 71
- knoppix, 47
- LDAP, 20, 20, 21
- liveCD, 47
- LVM, 31
- MBR, 43
- mount, 33
- multiboot, 43
- NFS, 16, 53, 56
- outils graphiques
- authentication, 21
  - hardware browser, 13
  - konqueror, 60
  - messages, 44
  - pirut, 51
  - pup, 51
  - Security level, 56
  - system monitor, 47
  - users, 20
- pam\_mount, 62
- pare-feu, voir firewall
- partitionnement, 13, 31
- permissions, 38

portmapper, 56

RAID, 31

routeur, voir gateway

RPM, 49

runlevel, 44, 45

samba, 59

scripts, 26

SELinux, 55

services, 46

setgid, 40

setuid, 40

shell, 3, 7, 22

single user, 45

SSH, 63

sticky bit, 39

system monitor, 13

udev, 34

variables d'environnement, voir  
environnement

vi, 67

vim, voir vi

vnc, 64, 79

VPN, 64

yum, 50