



Département ELECTRONIQUE

Polycopié pédagogique

Rédigé par: Dr . SAHRAOUI Leila
(Enseignante département Electronique)

Titre

SYSTEME LINUX

Cours destiné aux étudiants de Master

Master 1 Réseaux et Télécommunications

Année: 2022/2023

Table des matières

LISTE DES FIGURES	7
LISTE DES TABLEAUX	7
LISTE DES ACRONYMES	8
INTRODUCTION.....	9
Chapitre 1.....	11
INTRODUCTION AU SYSTEME GNU/LINUX	11
1. Historique	11
1.1. Qu'est-ce que Linux ?	11
1.2. D'où vient Linux ?	11
2. Linux est un logiciel libre (GNU/Linux)	13
2.1. Linux : un système d'exploitation	13
2.2. La licence GNU GPL	13
3. Les distributions Linux	14
3.1. REDHAT.....	14
3.2. Mandrake.....	14
3.2.1. MANDRIVE	15
3.3. SUSE.....	15
3.4. DEBIAN.....	15
3.5. Ubuntu.....	15
3.6. TURBOLINUX.....	16
4. Où chercher de l'aide ? :	16
4.1. Sur le disque dur	16
4.1.1. La commande man	16
4.1.2. La commande Info	17
4.1.3. Les HOWTO, FAQ et les documentations sur : /usr/doc (/usr/share/doc):	17
4.1.3.1. Les documentations.....	17
4.1.3.2. Les HOWTO, FAQ et Autres	18
4.2. Sur Internet	18
Chapitre II.....	19
DISQUES ET PARTITIONS	19

1. Structure d'un disque dur.....	19
1.1. Les Secteurs	19
1.2. Le cluster.....	19
2. Partitionnement du disque dur.....	19
2.1. Partitions.....	20
3. Conventions pour nommer les disques et partitions	20
3.1. Périphériques de stockage et montage	20
4. Systèmes de fichiers.....	24
4.1 Le système de fichiers FAT (File Allocation Table)	24
4.2. Le système de fichiers NTFS (New Technology File System)	24
4.3. Le système de fichiers de « Swap ».....	24
4.4. Le système de fichiers Ext2fs	24
4.4.1. Principales caractéristiques	25
4.5. Le système de fichiers Ext3fs	26
4.6. ReiserFS	27
4.7. JFS.....	27
5. Processus de démarrage et connexion au système	28
5.1. Démarrage du système.....	28
5.2. Structure du système GNU/linux	29
5.3. Système Multifenêtres:	30
6. Organisation des fichiers (l'arborescence du système).....	31
Chapitre III	33
ADMINISTRATION SOUS LINUX	33
1. Les utilisateurs sous Linux	33
1.1. Utilisateurs et groupes	34
1.1.1. Les utilisateurs.....	34
1.1.2. Les groupes	34
1.2. Gestion des utilisateurs	35
1.3. Droits d'accès sur les fichiers	36
Chapitre IV	38
INTRODUCTION A LA LIGNE DE COMMANDE	38
1. Interpréteur de commande sous Linux	38

2. Manipulation des fichiers et répertoires	38
2.1. Lister le contenu des répertoires :	38
2.2. Lister le contenu du fichier:	39
2.3. Création de répertoires et fichiers vides	39
2.3. 1. La commande mkdir	39
2.3.2. La commande touch	39
2.4. Supprimer des répertoires.....	40
Exemples d'application:	40
2.5. Déplacer ou renommer des fichiers	40
Exemples d'application:	41
2.6. Copier des répertoires et fichiers.....	41
Exemples d'application:	42
3. Droits d'accès sur les fichiers	42
3.1. Changer l'utilisateur d'un ou de plusieurs fichiers.....	42
Exemples d'application:	42
3.2. Changer le groupe propriétaire d'un ou de plusieurs fichiers :	43
Exemples d'application:	43
3.2. Changer droits d'accès sur les fichiers et les répertoires	43
3.2.1 En mode octal.....	43
Exemples d'application:	44
3.2.2. En mode expressions	44
Exemples d'application:	45
4. Accès à d'autres disques.....	45
4. 1. Monter un système de fichier	45
Exemples d'application:	46
4. 2. Monter un système de fichier	46
Exemples d'application:	46
Chapitre V	48
LE NOYAU LINUX	48
1. Introduction	48
2. Les modules	48
2.1. Afficher les modules chargés	48
2.2. Charger un module.....	49

Exemples d'application:	50
Exemples d'application:	51
2.4. Paramétrage des modules	52
3. Fichiers de dépendances	52
4. Outils nécessaires Pour compiler le noyau.....	53
4.1. Création du fichier de paramétrage.....	53
5. Compilation du noyau	54
6. Installation du noyau.....	54
Chapitre VI	55
PRÉSENTATION DE SERVICES RÉSEAUX SOUS GNU/LINUX	55
1. Configuration réseaux sous Gnu/Linux	55
1.1. Introduction	55
1.2. Protocole Description.....	55
1.3. Adresses IP et routage.....	55
Exemples d'application:	56
1.4. Rattachements des adresses IP avec le masque réseau	56
Exemples d'application:	56
Exemples d'application:	57
1.5. Fichiers réseaux.....	57
1.5.1. Le fichier hosts.....	57
1.5.2. Le fichier networks	57
1.6. Configuration et détection de cartes réseaux sous Linux	58
1.6.1. Différentes manières de voir les cartes réseaux	58
1.6.2. Attribution des noms d'interfaces par le noyau	58
1.6.3. Fichiers de configuration	59
Exemples d'application:	60
2. Réseaux sous GNU/LINUX: PARTEGE DE FICHERS.....	61
2.1. NIS (Network Information Service), NFS Système de fichier partagé (Network file system) et Système de partage avec Windows (Samba) :.....	61
2.1.1. NIS (Network Information Service).....	61
2.1.1.1. Services/démons utilisés par le service NIS:	61
2.1.1.2. Commandes utiles	61
2.1.2. NFS – Système de fichier partagé (Network file system).....	62

2.1.2.1. Services/démons utilisés par le service NFS:	62
2.1.2.2. Commandes utiles (voir man pour les détails).....	63
2.1.2.3. Configuration serveur	63
2.1.2.4. Le fichier /etc/exports :.....	64
2.1.2.5. Lancement de NFS sur un client	64
2.1.2.6. Application de partage avec NFS par les commandes côté serveur.....	66
2.1.2.7. Application de partage avec NFS par les commandes côté client.....	67
2.1.2.8. Exercices avec corrections sur le partage de fichiers NFS et le service NIS.....	67
2.1.3. Présentation des systèmes de fichiers de partage avec Windows le serveur Samba	69
2.1.3. 1. Les principales fonctionnalités du Samba	70
2.1.3.2. La configuration de Samba.....	70
2.1.3.3. Démarrage de Samba	70
Exemple : Avec le cas de ubuntu5 pour installation du serveur SAMBA	71
2.1.3.4. Vérification de l'installation	71
2.1.3.5. Les étapes pour créer des dossiers partagés	72
2.1.3.6. Paramétrage d'un client Linux (smbclient)	73
2.1.3.7. Exemples d'application:	73
CONCLUSION	75
RÉFÉRENCES BIBLIOGRAPHIQUES.....	76

LISTE DES FIGURES

<i>Figure 1.1: Affichage après exécution de la commande man ls</i>	17
<i>Figure 2-1 : Noms des partitions sous GNU/Linux</i>	22
<i>Figure 2-2: Noms des périphériques maître et esclave sur les bus IDE0 et IDE1</i>	22
<i>Figure 2-3: Noms des périphériques connecté sur les bus SATA et SCSI et la disquette</i>	23
<i>Figure 2-4: Partitionnement et noms des partitions sous GNU/Linux</i>	23
<i>Figure 2-5: Structure du système EXT2FS</i>	26
<i>Figure 2-6: Architecture d'un système GNU/Linux</i>	29
<i>Figure 2-7: Gestionnaire de fenêtre Gnome ou environnement Bureau pour (Ubuntu :Unity)</i>	30
<i>Figure 2-8: Gestionnaire de fenêtre ou environnement bureau pour KDE (Ubuntu : Kubuntu)</i>	31
<i>Figure 3-1: Droits d'accès affichés par la commande ls -l</i>	37
<i>Figure 5-1:Résultat de la commande ls mod</i>	47
<i>Figure 6-1: Le client accède à la ressource de façon transparente</i>	63
<i>Figure 6-2: Configuration du dossier de partage.</i>	70

LISTE DES TABLEAUX

Tableau 1: Les systèmes de fichiers/ Systèmes d'exploitation	28
--	----

LISTE DES ACRONYMES

BASH : BOURNE AGAIN SHELL
BIOS: BASIC INPUT OUTPUT SYSTEM
BL: BOOT LOADER
BS: BOOT SECTOR
CERN DE GENEVE : Conseil Européen pour la Recherche Nucléaire
CSH: C SHELL
EXT: EXTENDED FILE SYSTEM
LDP: LINUX DOCUMENTATION PROJECT
FAQ : FREQUENTLY ASKED QUESTION,
FAT: FILE ALLOCATION TABLE
FSF: FREE SOFTWARE FOUNDATION
GNOME: GNU NETWORK OBJECT MODEL ENVIRONMENT
GNU GPL : GNU GENERAL PUBLIC LICENSE
HFS: HIERARCHICAL FILE SYSTEM
HOWTO : COMMENT FAIRE
IBM AIX : CORPORATION ADVANCED INTERACTIVE EXECUTIVE
IBM: INTERNATIONAL BUSINESS MACHINES CORPORATION
JFS: JOURNALED FILE SYSTEM
KDE: K DESKTOP ENVIRONMENT
KERNEL: NOYAU
MACOS: MACINTOSH OPERATING SYSTEM
MBR: MASTER BOOT RECORDER
MFS: MACINTOSH FILE SYSTEM
MSDOS: MICRO SOFT OPERATING SYSTEM
NFS : NETWORK FILE SYSTEM
NIS : NETWORK INFORMATION SERVICE
OS: OPERATING SYSTEM
REISERFS: REISER FILE SYSTEM
SATA : SERIAL ADVANCED TECHNOLOGY ATTACHMENT
SCSI : SMALL COMPUTER SYSTEM INTERFACE
SH: SHELL
SMB: SERVER MESSAGE BLOCK
SWAP: SIZE WEIGHT AND POWER
TCSH: TEE-SEE-SHELL
UFS: UNIX FILE SYSTEM
UNIX : UNIPLEXED INFORMATION COMPUTING SYSTEM

INTRODUCTION

Ce document propose une structure organisée pour explorer les différents aspects du système d'exploitation Linux. Le système d'exploitation (en anglais Operating System : OS) est un logiciel ou un ensemble de logiciels qui permet la gestion des ressources de l'ordinateur et donc de permettre d'exécuter différentes applications.

Le Chapitre 1 commence par présenter l'historique du système Linux, en expliquant ce qu'est Linux et comment il a été conçu, son origine et aborde également la notion de logiciel libre qui adopte la licence GNU GPL. Ensuite, le chapitre explore les différentes distributions de Linux telles que REDHAT, Mandrake, SUSE, DEBIAN, Ubuntu et TURBOLINUX. Il fournit également des conseils sur la recherche d'aide, que ce soit sur le disque dur ou sur Internet.

Le Chapitre 2 présente la structure d'un disque dur, le partitionnement du disque et différents types des partitions, pour finalement mettre le point sur les conventions de nomination des disques et partitions sous GNU/LINUX, ainsi que les différents systèmes de fichiers tels que la FAT, NTFS, Swap, Ext2fs, Ext3fs, ReiserFS et JFS. Enfin, la dernière partie du chapitre expose le processus de démarrage du système et l'organisation des fichiers dans l'arborescence du système.

Le Chapitre 3 aborde l'administration sous Linux, en mettant l'accent sur les utilisateurs et les groupes. Il explique leur rôle, leur gestion et les droits d'accès sur les fichiers.

Le Chapitre 4 introduction à l'utilisation de la ligne de commande sous Linux. Il présente l'interpréteur de commande et explique comment manipuler les fichiers et les répertoires tels que, la création, suppression, le déplacement et la copie de fichiers et dossiers. Il couvre également des sujets tels que les permissions et la modification d'accès sur les fichiers et répertoires, ainsi que l'accès à d'autres disques.

Le Chapitre 5 est dédié au noyau Linux. Il présente la structure de base du noyau, la notion des modules, comment faire leur chargement et leur paramétrage. Le chapitre aborde également les fichiers de dépendances, les outils nécessaires pour compiler le noyau, ainsi que le processus et les étapes de compilation et d'installation du noyau.

Enfin, le Chapitre 6 présente les services réseau sous Linux. Il explique la configuration réseau, les adresses IP, le routage et les fichiers réseau sous Linux. La dernière partie du chapitre a été consacré aux services de partage de fichiers avec le NIS, le NFS et le Samba.

Linux est considéré réellement comme un logiciel open source multitâche, multiutilisateurs multiplateformes et multiprocesseur. Linux est présenté dans ce cours pour apporter un acquis pratique pour le parcours des étudiants.

Chapitre 1

INTRODUCTION AU SYSTEME GNU/LINUX

1. Historique

1.1. Qu'est-ce que Linux ?

Linux est un système d'exploitation créé par Linus Benedict Torvalds et enregistré en tant que marque déposée. Techniquement, Linux est un système d'exploitation, ou un "operating system" Au départ, Linux était basé sur le système d'exploitation Unix, mais il a été développé pour fonctionner sur des ordinateurs personnels plutôt que sur des stations de travail. Aujourd'hui, Linux est un système d'exploitation complet, puissant, rapide, stable et robuste qui peut rivaliser avec les plus grands systèmes d'exploitation existants.

1.2. D'où vient Linux ?

En 1991, les ordinateurs personnels étaient dominés par les compatibles PC fonctionnant sous les systèmes d'exploitation MS-DOS, Windows et OS/2. Cependant, aucun de ces systèmes d'exploitation n'exploitait correctement les capacités 32 bits et la gestion de la mémoire du microprocesseur Intel 80386, qui était alors abordable.

C'est à cette époque qu'un étudiant finlandais du nom de Linus Benedict Torvalds, qui étudiait les systèmes d'exploitation à l'université d'Helsinki. Linus Torvalds était un informaticien brillant et curieux qui avait récemment acquis un nouveau PC AT-386. Linus n'était intéressé que par des systèmes de type Unix, mais ceux-ci étaient hors de prix. Le DOS prédominait sur les architectures PC, mais ne pouvait satisfaire Linus Torvalds.

Dés lors, il a commencé à travailler sur un noyau (kernel) du système d'exploitation basé sur un système d'exploitation appelé le Minix. Le Minix était un système d'exploitation gratuit et son code source était disponible, mais avec une architecture différente et des fonctionnalités améliorées. C'est un système pédagogique dérivé et simplifié des systèmes Unix commerciaux conçu par Andrew Tannenbaum, et tournant sur les processeurs 8088. C'était le seul point de départ valable. Ce travail a abouti à la création du noyau Linux, qui est devenu le fondement de l'un des systèmes d'exploitation les plus populaires et les plus utilisés au monde.

C'est en 1991, que l'étudiant finlandais Linus Benedict Torvalds, entreprend d'écrire un noyau de système d'exploitation qu'on appellera plus tard « noyau Linux », en se basant sur le système Minix pour les plateformes i386.

La même année, en 1991, un certain Tim Berners Lee, au CERN de Genève, inventait le World Wide Web (WWW). Les deux événements ont convergé. Linus travaillait sur son système, qui allait devenir Linux. Il publiait l'entièreté de son travail (le code source de Linux) sur Internet. Plus fort encore, il invitait chacun à participer au développement de Linux ! Linux a donc été développé sur le net, par de nombreux informaticiens, donc Torvalds a collaboré avec d'autres développeurs pour ajouter des fonctionnalités et améliorer le noyau de Linux. Ce développement collaboratif a conduit à la création d'une communauté open-source active qui a continué à développer le système d'exploitation.

Le premier noyau Linux stable est sorti en 1991. Le 5 octobre 1991, il annonce sur le forum Usenet la disponibilité d'une ébauche version 0.02 de son noyau, la version 0.01 ayant eu une diffusion plus que confidentielle.

Le terme "Linux" a été utilisé pour la première fois par Ari Lemmke, un administrateur de système finlandais, lorsqu'il a téléchargé la première version du noyau Linux sur son ordinateur en 1991. Le nom Linux est un jeu de mots qui combine le nom de son créateur, Linus, et le mot Unix, le système d'exploitation sur lequel il était basé.

Au fil des ans, Linux est devenu un système d'exploitation populaire pour les serveurs, les ordinateurs personnels et les appareils mobiles. Il est utilisé par des entreprises telles que Google, Amazon et Facebook, ainsi que par des gouvernements, des établissements d'enseignement et des organisations à but non lucratif.

Aujourd'hui, Linux est un système d'exploitation open-source, ce qui signifie que le code source est disponible gratuitement et que toute personne peut le modifier et le distribuer librement. Cette nature open-source a conduit à une communauté de développement active et une grande variété de distributions Linux disponibles pour les utilisateurs, chacune avec ses propres caractéristiques et fonctionnalités.

2. Linux est un logiciel libre (GNU/Linux)

2.1. Linux : un système d'exploitation

Linux est un système d'exploitation. Parmi ses caractéristiques :

- multitâche : on peut faire tourner plusieurs programmes en même temps,
- multiplateformes : on le trouve sur tout type de machine,
- multiutilisateurs : plusieurs utilisateurs peuvent se connecter en même temps,

Linux est un système d'exploitation libre et open-source. Sa principale distinction par rapport à d'autres systèmes d'exploitation concurrents tels que Mac OS, Microsoft Windows et Solaris, est de proposer quatre libertés aux utilisateurs, telles que définies par la licence GNU GPL. Ces libertés permettent aux utilisateurs d'être indépendants de tout éditeur spécifique et encouragent la collaboration et le partage au sein de la communauté.

2.2. La licence GNU GPL

La création de la Free Software Foundation (FSF) et de la licence GPL (General Public License) pour les logiciels libres est attribuée à R. Stallman. Cette fondation a pour objectif de produire une licence adaptée aux logiciels libres, qui a été adoptée par un ensemble de logiciels libres sous le projet GNU (prononcé gnou : "GNU is Not Unix"). L'objectif était de créer une version libre de la plupart des outils que l'on trouve sous le système Unix, tels que le compilateur C/C++, appelé GCC (Gnu C Compiler), le débogueur GDB (Gnu Symbolic Debugger), et Emacs, un éditeur de fichiers

Linus Torvalds a adopté la licence GPL pour son noyau Linux et a utilisé les programmes développés par le projet GNU pour fournir une base d'application pour un système d'exploitation fiable, qui est plus précisément désigné par GNU/Linux. L'intégration des utilitaires GNU à Linux a eu pour conséquence d'incorporer la philosophie défendue par la FSF, qui a formalisé une approche de la notion de logiciels libres. Les idées sous-jacentes à cette approche sont les suivantes :

- liberté d'utiliser le logiciel
- accès au code source
- possibilité de modifier le code

-liberté de diffuser le programme, avec ou sans modification des variantes du concept de logiciel libre sont apparues, dot l'importante est le mouvement de open source.

Un logiciel libre n'est pas nécessairement un logiciel gratuit, et inversement tout logiciel non-commercial n'est pas forcément libre De plus, les logiciels libres ne sont pas exempts de droits d'auteur. C'est en vertu de ces droits que les contributeurs d'un logiciel libre accordent les quatre libertés essentielles : l'utilisation sans restriction, l'étude du code source, la modification du logiciel pour l'adapter à ses besoins et la redistribution sous certaines conditions précises.

Sur cette base, de nombreuses distributions de Linux ont vu le jour, basées sur le noyau Linux et incluant des logiciels libres utiles, ainsi que des outils de configuration et d'installation permettant d'installer et de faire fonctionner le système d'exploitation dans son ensemble.

3. Les distributions Linux

3.1. REDHAT



Société américaine fondée en 1994 (www.Redhat.com). Célèbre pour avoir introduit le système de gestion de paquets logiciel Redhat Package Manager (RPM),

3.1.1. FEDORA



Version grand publique gratuite de RedHat (fedora.redhat.com), utilise le système de gestion de paquets RPM.

3.2. Mandrake



Conçu par une société française, basé sur le RedHat en proposant des applications que la société mère refusait d'intégrer (tels que l'environnement de bureau KDE). Avec ces propres évolutions et succès, Mandrake a pu proposer une distribution originale avec de nombreux outils d'installation et d'administration. Cette société a pu s'affirmer et publier tous ses outils

et développements sous licences libres. Les programmes sont compilés pour les Pentium, ce qui empêche d'utiliser cette distribution sur des compatibles PC de type 486 ou 386.

3.2.1. MANDRIVE



Mandriva est le nom dérivé de la fusion entre la société **Mandrake** et Conectiva, du nom de la distribution brésilienne rachetée par Mandrakesoft en 2005. C'est la distribution française par excellence (www.mandriva.com). Très répandue dans le milieu éducatif, et en PME/PMI utilise le système de gestion de paquets RPM.

3.3. SUSE



Société allemande fondée à Nuremberg en 1993. Rachetée en 2003 par l'éditeur de logiciels américain Novell (www.novell.com/linux/suse), utilise le système de gestion de paquets RPM.

3.4. DEBIAN



Distribution issue d'un effort communautaire, le « projet Debian (www.debian.org), et non d'une entreprise distribution très soignée et ingénieuse austère à installer et à administrer (déconseillée aux débutants ?) Utilise le système de gestion de paquets Debian.

3.5. Ubuntu



Distribution populaire à base Débian lancée en 2004 (www.ubuntu-fr.org). Son nom provient d'un ancien mot bantou (langue d'Afrique), Ubuntu, signifiant « Humanité aux autres», ou encore « Je suis ce que je suis grâce à ce que nous sommes tous ».

3.6. *TURBOLINUX*



La distribution de Linux majeure pour les pays d'Asie particulièrement répandue en Chine (www.turbolinux.com).

4. Où chercher de l'aide ? :

4.1. Sur le disque dur

L'endroit le plus rapide, c'est sur le système lui-même, à savoir sur le disque dur de sa machine.

4.1.1. La commande man

Dans le monde Linux/Unix, c'est la commande **man** qui sert à la consulter des manuels d'aide. Cette commande est la plus souvent utilisée pour chercher de l'aide. Man est l'abréviation de "manual". Les manuels informatiques sur papier qui sont volumineux et coûteux, ils sont fournis sous forme électronique. La syntaxe pour l'utiliser de façons plus simple est la suivante :

– man xxx

Ceci affiche de l'aide à propos de la commande xxx, écran par écran.

Exemple :

Man ls

Le résultat de la commande est le suivant (Figure 1-1):

NOM

ls - Afficher le contenu de répertoires

SYNOPSIS

ls [OPTION]... [FICHIER]...

Afficher les informations des FICHIERS (du répertoire courant par défaut). Les entrées sont triées alphabétiquement si aucune des options `-cftuvSUX` ou `--sort` n'est utilisée.

Les paramètres obligatoires pour les options de forme longue le sont aussi pour les options de forme courte.

`-a, --all`
inclure les entrées débutant par « . »

`-A, --almost-all`
omettre les fichiers « . » et « .. »

`--author`
avec `-l`, afficher l'auteur de chaque fichier

`-b, --escape`
afficher la représentation octale des caractères non graphiques

Figure 1-1: Affichage après exécution de la commande man ls

– man man

Cette commande affiche de l'aide sur la commande man.

Pour sortir d'une page man, il suffit d'appuyer sur la lettre q.

4.1.2. La commande Info

Un autre format de documentation connu sous « **info** ». C'est le format utilisé pour enseigner la plupart des utilitaires de base du système. Souvent, la doc existe sous forme "man" et "info". La première peut être ancienne et non maintenue. Le format info est un format texte. Il autorise une navigation (via des hyperliens) dans le document d'aide (sous forme linéaire ou hiérarchique).

La commande info s'utilise comme la fonction man :

-info xxx

La doc sur la commande info est disponible au format info, il suffit de taper

info info

4.1.3. Les HOWTO, FAQ et les documentations sur : /usr/doc (/usr/share/doc):

4.1.3.1. Les documentations

A partir du répertoire /usr/doc ou /usr/share/doc, on trouve les documentations sur les paquetages installés. Le contenu dépend du concepteur du paquetage.

4.1.3.2. Les HOWTO, FAQ et Autres

Une autre source très intéressante et instructive se trouve dans l'arborescence `usr/doc` (`/usr/share/doc`). On y trouve pêle-mêle les HOWTO, FAQ, guides, documentations spécifiques à un logiciel.

-Les FAQ : sont les questions fréquemment posées (Frequently Asked Questions), accompagnées de leurs réponses, qui sont groupées par thèmes. Il est indispensable de lire les FAQ avant de poser une question sur le web :

-Les HOWTO : Les HOWTO Linux sont des documents complets qui expliquent en détail comment configurer ou installer certaines parties d'un système sous Linux, et qui peut recouvrir plusieurs commandes. Il existe par exemple un Sound-HOWTO consacré au système audio de Linux. Ils sont installés dans le répertoire `/usr/doc/HOWTO`. Ce sont les premiers documents à consulter lorsqu'on recherche une information pouvant s'exprimer par "Comment faire pour ?".

Les HOWTO sont rassemblés sur le site du LDP, le Linux Documentation Project (<http://www.linuxdoc.org>).

-Les guides : Il existe également des documentations qui abordent des thématiques plus larges telles que l'administration système sous Linux. Ces documents, appelés **guides** en anglais qui sont également disponibles sur le site du LDP. On les retrouve sur le disque local de certaines distributions Linux.

4.2. Sur Internet

-Les sites Web/ Certains sites web proposent de la documentation en ligne ou téléchargeable

- Linux Center : (<https://www.asustor.com/>)
- Unix Guru Universe : (<http://www.ugu.com/>)
- LDP : <https://tldp.org/>

-Les Forums

- <https://linuxfr.org/forums>
- fr.comp.os.linux.configuration
- comp.os.linux.security
- comp.os.linux.setup

Chapitre II

DISQUES ET PARTITIONS

1. Structure d'un disque dur

Un disque est physiquement divisé en petits secteurs. Une suite de secteurs peut former une partition. En fait, vous pouvez créer autant de partitions souhaitées, chacune d'entre elles étant considérées comme un disque dur séparé.

1.1. Les Secteurs

Un disque dur représente une suite de **secteurs**. Un secteur est la plus petite unité d'information sur un disque dur, et sa taille est de 512 octets. Les secteurs d'un disque dur de (n) secteurs sont numérotés de (0) à (n-1).

1.2. Le cluster

Il représente un regroupement de plusieurs secteurs, un cluster de 512 octets représente un secteur et un cluster de 64 ko est équivalent à 128 secteurs.

2. Partitionnement du disque dur

L'utilisation de plusieurs partitions vous permet de créer autant de disques durs virtuels sur votre disque dur réel. Ceci comporte plusieurs avantages:

- Des systèmes d'exploitation différents utilisent des structures de disque (systèmes de fichiers) différents; c'est notamment le cas pour Windows et GNU/Linux. Avoir plusieurs partitions sur un disque dur vous permet d'installer plusieurs systèmes d'exploitation sur le même disque matériel.
- GNU/Linux nécessite une partition appelée « swap » qui veut dire échange. Cette partition est désigné à accomplir des tâches complètement distinctes comme accueillir la mémoire virtuelle et compléter la mémoire physique.
- Il peut être très utile de séparer les différentes parties de votre OS en plusieurs partitions, même si elles utilisent le même système de fichiers. Vous pouvez répartir vos fichiers sur deux partitions, une pour vos données personnelles, une autre pour le système exploitation lui-même. Cela vous permet de mettre à jour ce dernier en effaçons toute la partition, toute en gardant vos fichiers personnels intacts puisque ils se retrouvent sur une autres partition.

- Les erreurs physiques sur un disque dur ne sont dispersées sur tout le disque ils se trouvent généralement situées sur des secteurs adjacents. Distribuer vos fichiers sur des partitions différentes limitera les pertes de données en cas de dommages physiques du disque dur.

2.1. Partitions

Les partitions procurent la possibilité d'installer plusieurs systèmes d'exploitation sur une même machine, la sauvegarde des données en cas de réinstallation et une séparation logique des répertoires (système, utilisateur et fichiers,...).

A l'installation, un disque dur n'est ni partitionné, ni formaté. Partitionner signifie définir des espaces réservés sur le disque, et formater signifie préparer la partition à recevoir des informations.

Une partition est définie par son type, son emplacement de début de partition et sa taille (ou son emplacement de fin de partition). Un partitionnement est réversible (non physique).

Une seule partition peut être activée sur un ordinateur : l'activation indique où le BIOS doit aller chercher le noyau du système d'exploitation pour le démarrage.

Il existe trois sortes de partitions :

- partition principale : on peut avoir au maximum quatre, pas de minimum et accepte tout type de système de fichiers
- partition étendue : ne peut contenir que des partitions logiques, ne peut pas recevoir de système de fichiers et ne peut exister que s'il existe une partition principale
- partition logique : contenue dans une partition étendue, pas de limite en nombre, accepte tout type de système de fichiers.

3. Conventions pour nommer les disques et partitions

3.1. Périphériques de stockage et montage

On accède aux partitions par des pointeurs stockés dans répertoire `/dev`. Ces pointeurs sont de la forme « dev/PPLN » :

Avec « PP » le type de bus (« hd » pour les disques IDE, « sc » pour les disques SCSI, « fd » pour les disquettes).

« L » la lettre du périphérique concerné

« N » le numéro de partition concernée (1 à 4 pour une partition principale ou étendue, > 4 pour les partitions logiques).

Exemple 1 :

- /dev/hda : périphérique maître du bus primaire IDE ;
- /dev/hdb : périphérique esclave du bus primaire IDE
- /dev/hdd : périphérique esclave du bus secondaire IDE;
- /dev/hda1 : première partition primaire du périphérique maître du bus primaire IDE ;
- /dev/hda7 : troisième partition logique du périphérique maître du bus primaire IDE.

- Les périphériques IDE (que ce soient des disques durs, lecteurs CD-ROM ou autres) primaires, maître et esclave, sont appelés respectivement /dev/hda et /dev/hdb
- Sur l'interface secondaire, ce sont /dev/hdc et /dev/hdd pour maître et esclave respectivement (Figure 0-2).
- Si votre ordinateur contient d'autres interfaces IDE (par exemple l'interface IDE présente sur certaines cartes Soundblaster), les disques s'appelleront alors /dev/hde, /dev/hdf, ..etc .
- les disques SCSI sont appelés /dev/sda, /dev/sdb , selon l'ordre de leur apparition sur la chaîne SCSI en fonction des ID croissants (Figure 0-3). Les lecteurs de CD-ROM SCSI sont appelés /dev/scd0, /dev/scd1, toujours dans l'ordre d'apparition sur la chaîne SCSI.

Ainsi, GNU/Linux nommera les partitions de la façon suivante :

- Les partitions primaires (ou étendues) sont nommées /dev/hda1 à /dev/hda4 (Figure 0-4) lorsqu'il y en a
- Les partitions logiques, s'il-y-en-a, sont nommées /dev/hda5, /dev/hda6, ...etc. selon l'ordre d'apparition dans la table des partitions logiques(Figure 0-4).

Exemple 1 (Figure 2-1) : GNU/Linux nommera les partitions de la façon suivante:

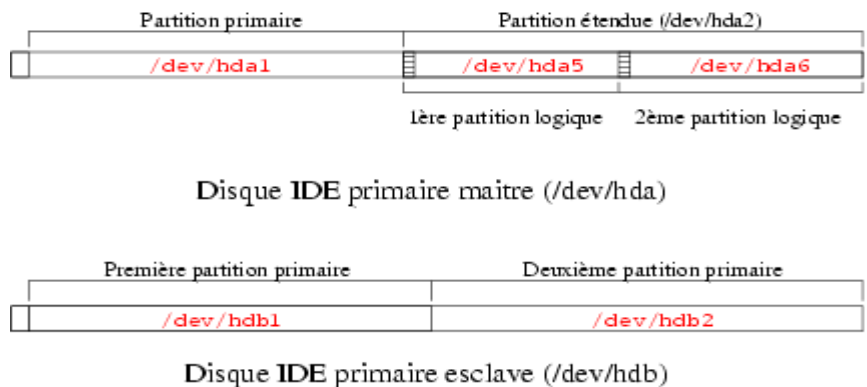


Figure 2-2: Noms des partitions sous GNU/Linux

Exemple 2 (Figure 0-2) : Nom des périphériques:

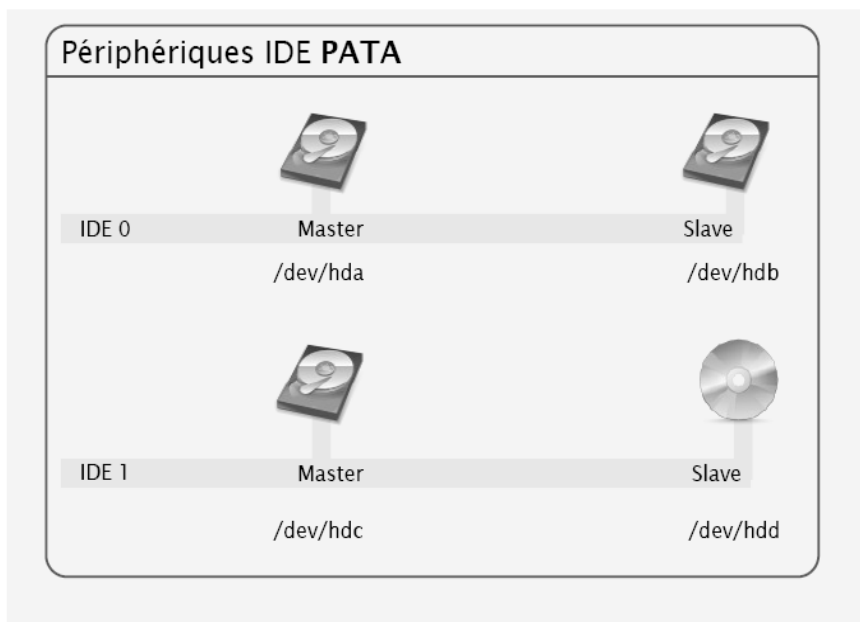


Figure 0-2: Noms des périphériques maître et esclave sur les bus IDE0 et IDE1

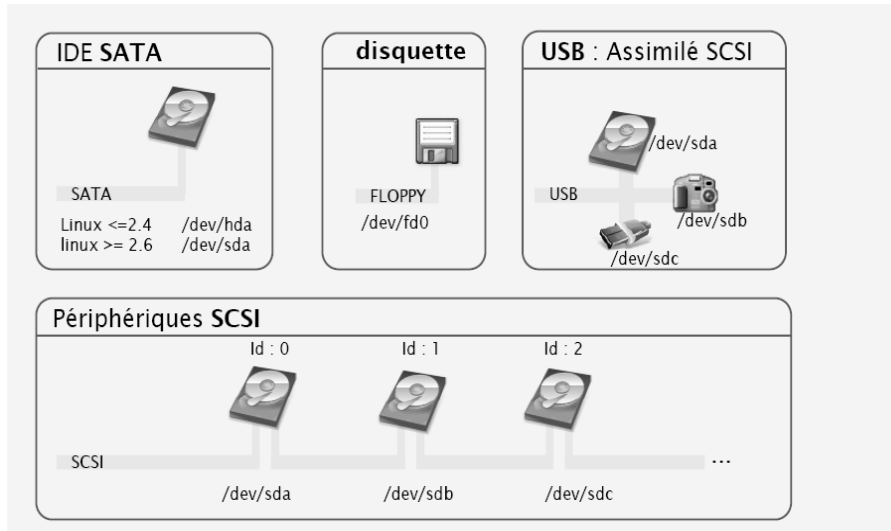


Figure 0-3: Noms des périphériques connecté sur les bus SATA et SCSI et la disquette

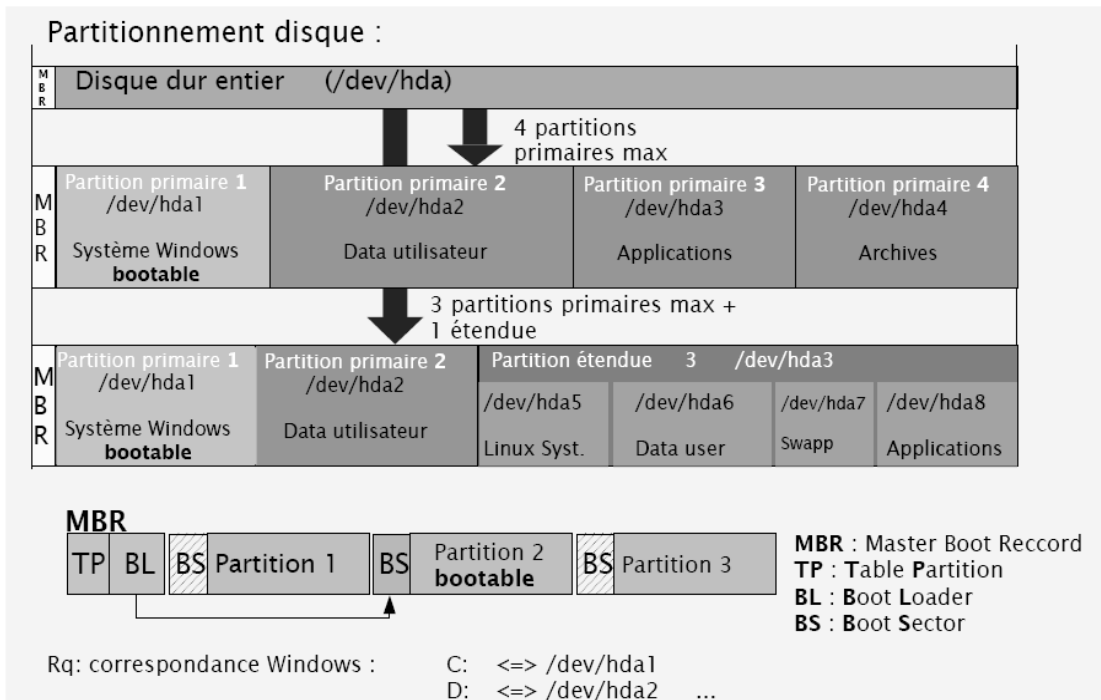


Figure 0-4: Partitionnement et noms des partitions sous GNU/Linux

4. Systèmes de fichiers

On appelle système de fichiers l'agencement logique et structuré des données sur un disque dur.

Linux supporte un grand nombre de systèmes de fichiers, et notamment ceux provenant du monde Windows (FAT 16, FAT32, NTFS en lecture seule actuellement pour certaines distributions).

4.1 Le système de fichiers FAT (File Allocation Table)

C'est l'un des plus simples que l'on puisse trouver. Il repose sur une table d'allocation des fichiers stockée au niveau supérieur de la partition. Il possède quatre limitations principales :

- le répertoire racine doit être stocké à un emplacement fixe sur la partition ;
- la mise à jour de la table d'allocation est très laborieuse pour le système ;
- il n'existe aucune organisation de la structure des répertoires, d'où une fragmentation importante
- la taille des clusters est variable en fonction de la taille des partitions

4.2. Le système de fichiers NTFS (New Technology File System)

Dans NTFS, l'organisation des fichiers est optimisée sur la partition, d'où une très faible fragmentation. NTFS respecte la norme POSIX.1 et permet une bonne gestion des disques de grande taille. Il prend également en charge le modèle de sécurité des systèmes Windows NT, 2000 et XP.

4.3. Le système de fichiers de « Swap »

Ce type de système de fichiers est un peu particulier : il est uniquement dédié à l'utilisation de la mémoire virtuelle et sert de mémoire complémentaire quand la RAM est saturée. Le Swap est utilisé intensivement par le système pour la décharge des logiciels en mémoire.

4.4. Le système de fichiers Ext2fs

Le Second Extended File System, en abrégé Ext2fs ou simplement ext2, est le système de fichier par défaut de GNU/Linux depuis de nombreuses années. Il est le successeur de Extended File System (d'où le nom « Second »), dont il corrige certains problèmes et certaines limitations.

Ext2FS respecte les standards usuels des systèmes de fichiers pour les systèmes de type Unix, tout en offrant une grande robustesse et de bonnes performances.

Ext2FS définit quatre catégories de fichiers :

- les fichiers normaux (textes, exécutables).
- les fichiers répertoires.
- les fichiers spéciaux, contenus dans le répertoire /dev, qui sont des points d'accès aux périphériques.
- les fichiers liens symboliques qui font référence à d'autres fichiers.

4.4.1. Principales caractéristiques

Ses principales caractéristiques sont les suivantes :

- répertoires séparés par des /
- fichiers cachés débutant par un « . ».
- espaces et noms longs acceptés.
- défragmentation quasi-inutile.

Une partition ext2fs comporte un secteur d'amorçage et un ensemble de groupes de secteurs. Le secteur d'amorçage est le premier sur la partition, est noté secteur 0 et a une taille de 1 Ko. Chacun des groupes de secteurs comprend six parties (Figure 02-5) :

- un super bloc, contenant les informations de structure de la partition.
- la liste des descripteurs de groupe permettant de localiser sur le disque les informations de chaque groupe.
- la table d'allocation des blocs du groupe.
- la table d'allocation des inodes.
- la table d'allocation des inodes du groupe.
- les blocs de données.

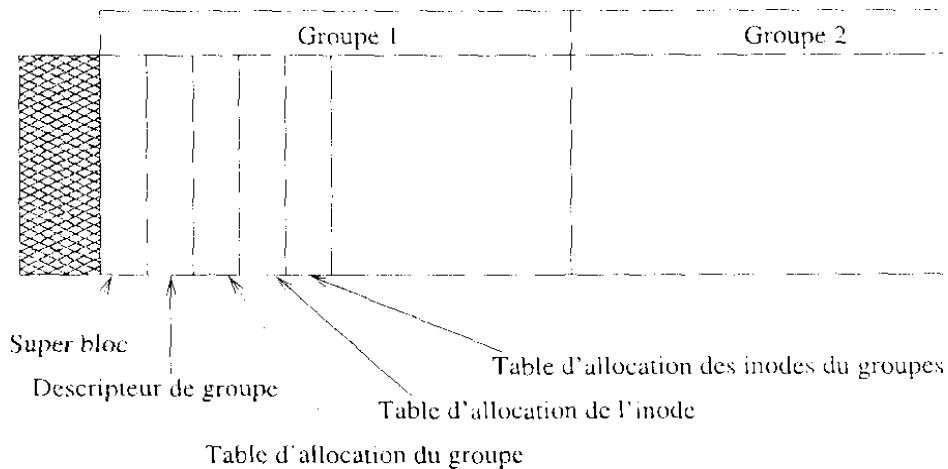


Figure 02-5: Structure du système EXT2FS

Le super bloc et les descripteurs de groupes sont répétés au début de chacun de chacun des groupes. Ces groupes ont pour but de regrouper les informations reliées entre elles et ont une taille de 8192 secteurs.

Pour chacun des fichiers créés sur un disque, un inode est associé. Il contient :

- l'identification du propriétaire.
- la taille en octets.
- les heures de modification, création ou accès.
- le nombre de liens associés.
- le type.
- les droits d'accès.

4.5. Le système de fichiers Ext3fs

Comme son nom l'indique : the **Third Extended File System** est le successeur d'Ext2FS. Il conserve une compatibilité avec celui-ci, mais ajoute une fonctionnalité très intéressante : la **journalisation**.

Un des problèmes majeurs avec les systèmes de fichiers « traditionnels » comme Ext2FS, est leur faible tolérance aux pannes, telles qu'un arrêt brutal du système (coupure de courant ou plantage logiciel). En général, de tels événements se soldent par un examen très long de la

structure du système de fichiers, des tentatives de corrections d'erreurs et une perte partielle ou totale des données enregistrées.

La journalisation est une réponse à ce problème. Pour simplifier, disons que le principe consiste à enregistrer les actions à effectuer dans un journal avant de les effectuer réellement. Le résultat est un système de fichiers qui reste toujours cohérent. En cas de problème, l'examen du système de fichiers consiste à regarder le journal et effectuer les actions qui n'ont pas eu le temps d'être effectuées avant la coupure. Le temps de vérification d'un système de fichiers n'est donc plus proportionnel à la taille de celui-ci, mais à son degré d'utilisation.

Ext3FS propose donc cette technologie, tout en conservant une structure interne basée sur Ext2FS, ce qui assure une excellente compatibilité.

4.6. ReiserFS

Au contraire d'Ext3FS, **ReiserFS** est un système de fichiers recréé en partant de zéro. Il est également journalisé comme Ext3FS, mais sa structure interne est radicalement différente. En particuliers, il utilise des concepts d'arbres binaires inspirés des logiciels de base de données.

4.7. JFS

JFS est le système de fichiers journalisé développé et utilisé par IBM. Initialement propriétaire IBM a récemment décidé d'ouvrir l'accès au monde du Logiciel Libre avec ce système de fichiers. Sa structure interne est proche de celle de Reiser FS.

Le tableau1 illustre les systèmes de fichiers usuellement utilisés correspondants aux systèmes d'exploitation.

Tableau 1: Les systèmes de fichiers/ Systèmes d'exploitation

Système d'exploitation	Types de système de fichiers supportés
Dos	FAT16
Windows 95	FAT16
Windows 95	FAT16, FAT32
Windows 98	FAT16, FAT32
Windows NT4	FAT, NTFS (version 4)
Windows 2000/XP	FAT, FAT16, FAT32, NTFS (versions 4 et 5)
Linux	Ext2, Ext3, ReiserFS, Linux Swap, (FAT, NTFS, ...)
MacOS	HFS, MFS
Sun Solaris	UFS (Unix File System)
IBM AIX	JFS (Journaled File System)

5. Processus de démarrage et connexion au système

5.1. Démarrage du système

Quand l'ordinateur est démarré, le premier élément à être lancé est le BIOS qui va choisir un disque de démarrage (disque dur ou disquette) et va lire le tout premier secteur

Dans le cas d'un disque dur, ce secteur est le Master Boot Record (MER). Ce secteur de démarrage contient un petit programme qui tient sur un seul secteur et dont la responsabilité est de lancer le système d'exploitation.

Pour un disque dur, le code contenu dans le secteur de démarrage va examiner la table des partitions, identifier la partition active, lire le secteur de démarrage depuis cette partition, et l'exécuter.

Il existe plusieurs systèmes de démarrage pour les systèmes Unix. Les plus connus pour Linux sont LILO (LInux LOader) et GRUB (GRand Unified Bootloader). Ces systèmes sont chargés de démarrer le noyau et d'initialiser le système en suivant les étapes suivantes :

- le noyau Linux est décompressé :

- le noyau Linux initialise les périphériques (disque dur, carte réseau,...) et configure automatiquement les pilotes.
- le noyau monte le système de fichiers racine qui est monté en lecture seule de manière à pouvoir vérifier les erreurs qu'il pourrait contenir.
- le noyau lance le programme init qui permet de lancer les processus de base
- le programme init passe alors en mode multiutilisateurs et lance le programme getty pour chaque console virtuelle, getty étant un programme permettant le login au système.
- finalement, le système de fichiers racine est remonté en lecture-écriture et les processus des services sont lancés.

5.2. Structure du système GNU/Linux

Un système GNU/Linux peut être vu selon trois couches complémentaires :

- Une couche « physique » comprenant les périphériques et le BIOS.
- Une couche « système » comprenant le noyau et les processus
- Une couche « interface » comprenant le shell et l'interface graphique basée sur X-Window.
- Les communications entre ces couches sont très réglementées par le noyau selon la Figure 2-6.

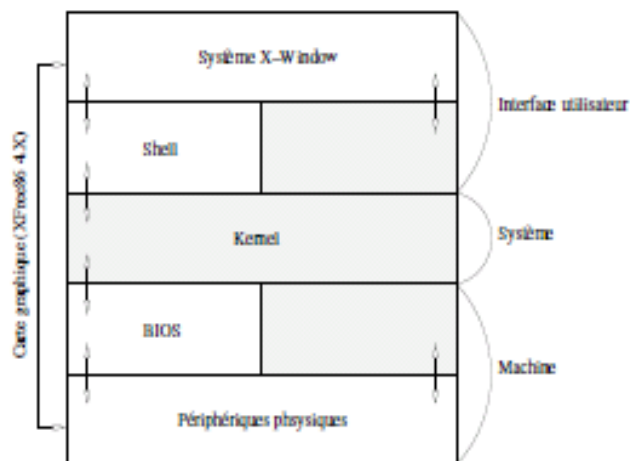


Figure 2-6: Architecture d'un système GNU/Linux

L'un des éléments essentiels à l'interaction avec les utilisateurs est le shell qui est un interpréteur de commandes qui lit et exécute les commandes de l'utilisateur. Il propose un contrôle des processus et permet de gérer les redirections en entrée et en sortie. Il propose en général un véritable langage de programmation et on retrouve en fait, plusieurs types de shell disponibles : bash, csh, ksh, tcsh,...

5.3. Système Multifenêtres:

Le système X-Window est l'interface graphique standard des systèmes Unix .il repose sur un processus particulier appelé "serveur X" et utilise un gestionnaire de fenêtre dont un grand nombre sont actuellement disponibles (KDE, Gnome, WindowMaker,...).

Les figures 2-7 et 2-8 illustre deux gestionnaire de fenêtre Gnome et KDE respectivement pour le système Ubuntu. Une des principales caractéristiques de ce système de fenêtres est la possibilité pour l'utilisateur de déporter des affichages graphiques à travers un réseau.

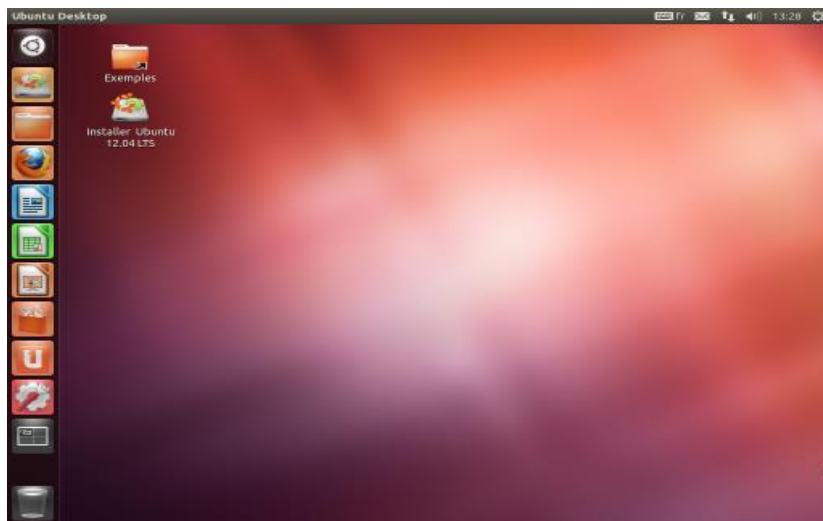


Figure2-7: Gestionnaire de fenêtre Gnome ou environnement Bureau pour (Ubuntu :Unity)

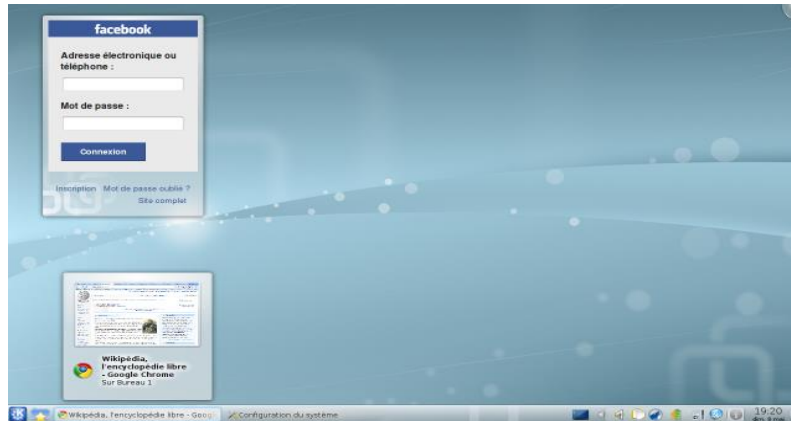


Figure 2-8: Gestionnaire de fenêtre ou environnement bureau pour KDE (Ubuntu : Kubuntu)

6. Organisation des fichiers (l'arborescence du système)

Il n'existe pas de norme d'organisation du système de fichiers, mais un standard est suivi par les différentes distributions de Linux.

Le répertoire de base / : s'appelle : répertoire racine (*root*) par analogie avec la racine d'un arbre représentant le système de fichiers.

L'organisation traditionnelle du répertoire racine est la suivante :

/bin : contient les fichiers exécutables nécessaires à l'initialisation

/boot : contient le noyau et les fichiers de démarrage

/dev : on retrouve les fichiers spéciaux appelés pointeurs

/etc : contient les fichiers de configuration du système et certains scripts

/home : la base des répertoires utilisateurs

/lib : les librairies système et les modules

/mnt : les points montage des systèmes extérieurs

/proc : un système de fichiers virtuels permettant l'accès aux variables du noyau

/root : le répertoire de base du super utilisateur (utilisateur système)

/sbin : les fichiers exécutables pour l'administration du système

/tmp : on retrouve les fichiers temporaires

/usr : les programmes, les bibliothèques et les fichiers d'aides accessibles pour l'utilisateur

/var : les données variables liées à la machine

Chapitre III

ADMINISTRATION SOUS LINUX

1. Les utilisateurs sous Linux

Linux distingue les utilisateurs A normaux « ordinaires A » du super-utilisateur.

Le super-utilisateur, appelé root, s'occupe de toutes les tâches d'administration et peut accéder à toutes les parties du système. Les utilisateurs normaux n'ont pas ces privilèges.

Tous les utilisateurs, y compris le super-utilisateur, ont leur répertoire personnel ou toutes leurs données privées (documents, fichiers, messages électroniques...) sont enregistrées. L'accès en écriture sur ces répertoires personnels est strictement limité à leur propriétaire. Il est même possible que des sous-répertoires d'un répertoire personnel contenant des données sensibles ne puissent pas être lus par d'autres utilisateurs.

Les répertoires système contenant la configuration globale ou les fichiers excusables ne sont accessibles et modifiables que par le super-utilisateur. Ce concept améliore la sécurité, un utilisateur sans privilèges root ne peut pas endommager le système.

-Tout dommage causé est strictement limité au compte et aux données personnelles de l'utilisateur.

-Toute opération exécutée avec les privilèges root peut potentiellement compromettre le système dans son ensemble.

-Une personne qui a l'intention d'endommager un système Linux doit d'abord obtenir les privilèges du root. C'est pour cette raison qu'il est bien plus difficile de créer des virus pour les systèmes Linux. Ils doivent d'abord passer la barrière du root.

-Si vous vous connectez régulièrement en tant que root, il est très facile de faire une erreur qui pourrait rendre votre système inutilisable. Une seule mauvaise manipulation peut suffire. En particulier, si vous n'avez pas mis de mot de passe à ce compte, n'importe qui peut altérer votre système.

En plus d'offrir des identités utilisateurs différentes pour les administrateurs et les utilisateurs normaux, plusieurs utilisateurs peuvent travailler sur un même système Linux en même temps. Ces utilisateurs peuvent se connecter au système par différents terminaux ou consoles.

1.1. Utilisateurs et groupes

1.1.1. Les utilisateurs

Linux est système multiutilisateurs, et pour pouvoir se connecter à un système GNU/Linux, il faut avoir un compte sur ce système. Quand vous avez créé des utilisateurs lors de l'installation, vous avez en fait ajouté des comptes utilisateur. Vous vous souvenez sans doute que la création d'un compte a exigé que vous entriez, les éléments suivants :

- le « nom » de l'utilisateur.
- un nom de connexion, considéré comme Username ou login
- un mot de passe.

Les deux paramètres importants ici sont le nom de connexion (très souvent appelé nom de login) et le mot de passe. Ce sont en effet eux que vous devrez utiliser pour vous connecter au système.

1.1.2. Les groupes

Les utilisateurs des systèmes Linux et UNIX sont assignés à un ou plusieurs groupes pour les raisons suivantes :

- Partager des fichiers ou d'autres ressources avec un petit nombre d'utilisateurs
- Faciliter la gestion des utilisateurs
- Facilité de surveillance des utilisateurs
- L'appartenance à un groupe est une solution parfaite pour les grandes installations Linux (UNIX)
- L'appartenance à un groupe vous donne, à vous ou à votre utilisateur, un accès spécial aux fichiers, répertoires ou périphériques qui sont autorisés pour ce groupe

-Un groupe peut contenir autant d'utilisateurs que vous le souhaitez, et il est très courant de voir une telle séparation dans les gros systèmes. L'inverse est également vrai : un utilisateur peut être membre d'un ou de plusieurs groupes, la limite étant de trente-deux groupes.

1.2. Gestion des utilisateurs

Sous Linux, un utilisateur est une personne physique ou virtuelle possédant des droits d'accès au système, un groupe d'utilisateur et éventuellement un répertoire personnel. L'utilisateur root est l'administrateur du système. Il possède tous les droits sur le système, les fichiers et les utilisateurs.

L'identification d'un utilisateur s'effectue par l'intermédiaire de son nom (« login ») et de son mot de passe. Tous les utilisateurs sont référencés dans les fichiers `/etc/passwd` ou `/etc/shadow`, tandis que `/etc/group` contient les références aux différents groupes. Le répertoire personnel d'un utilisateur est en général `/home/<login>`, et il peut changer son mot de passe par la commande `passwd`.

Les commandes `adduser`, `deluser` et `usermod` sont utilisables uniquement par root et permettent respectivement d'ajouter, de supprimer et de modifier un compte utilisateur.

Le fichier `/etc/passwd` contient toutes les informations sur les comptes utilisateurs du système

Seul le root a le droit d'écriture dessus et chacun des utilisateurs est référencé par une ligne

Exemple d'application :

`/etc/passwd` : Stocke la configuration des utilisateurs Linux

Voici un exemple de ce fichier:

```
root:x:0:0:root:/root:/bin/bash
```

```
bin:x:1:1:bin:/bin:/bin/bash
```

```
daemon:x:2:2:daemon:/sbin:/bin/bash
```

```
daemon:x:2:2:daemon:/sbin:/bin/bash
```

```
news:x:9:13:News system:/etc/news:/bin/bash
```

```
demo:x:1001:1001:,,,:/home/demo:/bin/bash
```

- **Nom d'utilisateur « demo »** : Il est utilisé lorsque l'utilisateur se connecte au système. Sa longueur doit être comprise entre 1 et 32 caractères.

- **Mot de passe** : Un caractère **x** indique que le mot de passe est chiffré est stocké dans le fichier `/etc/shadow`. Veuillez noter que vous devez utiliser la commande `passwd` pour mettre à jour le mot de passe dans le fichier `/etc/shadow`.

- **ID utilisateur (UID) : 1001** Chaque utilisateur doit se voir attribuer un ID utilisateur (UID). L'UID 0 (zéro) est réservé à l'utilisateur root et les UIDs 1-99 sont réservés aux autres comptes prédéfinis. Les UID 100-999 sont réservés par le système pour les comptes/groupes administratifs et système

- **ID de groupe (GID) : 1001** L'ID du groupe primaire (stocké dans le fichier `/etc/group`)

- **User ID Info** (commentaire) : ,, Le champ de commentaire, il vous permet d'ajouter des informations supplémentaires sur les utilisateurs, comme le nom complet de l'utilisateur, son numéro de téléphone, etc.

- **Répertoire d'accueil** : Le chemin absolu vers le répertoire de base de l'utilisateur avec le même nom de l'utilisateur que vous pouvez bien sur changer. Si ce répertoire n'existe pas, le répertoire de l'utilisateur devient /

-**Commande/shell** : Le chemin absolu d'une commande ou d'un shell (`/bin/bash`). Généralement, il s'agit du shell installé avec le système LINUX

1.3. Droits d'accès sur les fichiers

Il existe trois types droits d'accès ou de permissions sur les fichiers pour un utilisateur :

La lecture (Read), l'écriture (Write) et l'exécution (eXecute). À la création d'un nouveau fichier, des droits par défaut sont utilisés. Par ailleurs, trois catégories d'utilisateurs sont définies pour un même fichier : le propriétaire (User), les membres du groupe (Group) et les autres utilisateurs (Others). Chacun des fichiers est associé à un propriétaire et à un groupe.

La commande pour afficher les droits des fichiers est **ls -l** (Figure 3-1).

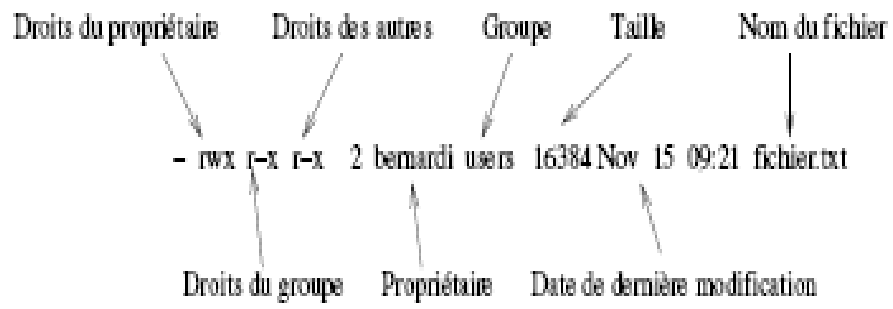


Figure 03-1: Droits d'accès affichés par la commande `ls -l`

Chapitre IV

INTRODUCTION A LA LIGNE DE COMMANDE

1. Interpréteur de commande sous Linux

Sous Linux, vous avez le choix entre une multitude d'interpréteur de commande (*sh, csh, ksh, tcsh, . . .*). Dans cette partie nous allons présenter le BASH (Bourne Again SHell) : l'interpréteur de commande sous Linux. L'invite de BASH sous Linux se présente généralement comme suit :

[<nom_utilisateur>@<ordinateur> <répertoire>] \$: Après ce signe les commandes sont utiliser par tous le monde (par tous l'utilisateur)

Ou

[root@<ordinateur> <répertoire>]# Après ce signe les commandes sont uniquement utiliser par le super-utilisateur.

2. Manipulation des fichiers et répertoires

2.1. Lister le contenu des répertoires :

ls [option] [chemin du répertoire]

Cette commande permet de lister le contenu d'un répertoire

Les options les plus utilisé :

-a ou --all : Liste tous les fichiers, y compris ceux cachés (avec un nom commençant par « . »).

-A ou --almost-all : Identique à l'option -a mais omet les entrées.

-t : Trie par date de dernière modification

-C : Liste par colonne

2.2. Lister le contenu du fichier:

cat [options] [chemin de fichier]

Exemple : **cat /etc/passwd** (Affiche le contenu du fichier passwd)

2.3. Création de répertoires et fichiers vides

2.3.1. La commande **mkdir**

mkdir (MaKe DIRectory) est utilisée pour créer des répertoires.

mkdir -[options] [répertoire ...]

-p : En fait, une seule option est vraiment intéressante : l'option **-p**. Si cette option est passée en argument, elle implique deux comportements :

1. **mkdir** créera les répertoires parents s'ils n'existaient pas avant. Sans cette option, **mkdir** échouerait, se plaignant que les répertoires parents n'existent pas.
2. **mkdir** terminera silencieusement si le répertoire que vous désirez créer existe déjà.

Exemples d'application:

mkdir TP2 : crée un répertoire du nom de TP2 dans le répertoire courant.

mkdir -p images/divers docs : crée un répertoire divers dans le répertoire images après avoir créé ce dernier s'il n'existait pas avec l'option **(-p)** et il va créer également un répertoire docs dans le répertoire courant.

2.3.2. La commande **touch**

touch pour créer des fichiers vides

Initialement, la commande **touch** n'a pas pour but de créer des fichiers mais de mettre à jour les dates d'accès, **touch** : créer les fichiers mentionnés comme des fichiers de taille 0 s'ils n'existaient pas déjà. La syntaxe est :

touch [options] fichier [fichier...]

Exemples d'application:

touch fichier1 images/fichier2 : Ce qui créera un fichier vide appelé fichier1 dans le répertoire courant et un fichier vide appelé fichier2 dans le répertoire images, si ces fichiers n'existaient pas avant.

2.4. Supprimer des répertoires

La commande **rm** : Pour supprimer des fichiers ou des répertoires

rm (ReMove) remplace les commandes del et deltree du DOS, et rajoute des options supplémentaires. Sa syntaxe est:

```
rm [options] [file|directory, file|directory...]
```

Parmi les options, on trouve :

-r ou **-R** : supprime récursivement. Cette option est obligatoire pour supprimer un répertoire, même vide. Toutefois, pour effacer des répertoires vides, vous pouvez également utiliser la commande rmdir.

-i : demande une confirmation avant chaque suppression.

-f: le contraire de **-i**, cette option force la suppression des fichiers ou répertoires, même si l'utilisateur n'a pas l'autorisation d'écriture sur les fichiers.

-v : mode **verbeux** qui rapporte tous les changements.

Exemples d'application:

rm -i images/*.jpg fichier1 : suppression de tous les fichiers dont le nom se termine par **jpg** dans le répertoire images, ainsi que le fichier fichier1 dans le répertoire courant. Une confirmation est demandée pour chacun des fichiers. Répondez **o** ou **y** pour confirmer, **n** pour annuler.

rm -Rf images/divers/ file* : suppression sans demande de confirmation de tous le répertoire divers dans le répertoire images. De plus, tous les fichiers du répertoire courant dont le nom commence par file sont également effacés.

2.5. Déplacer ou renommer des fichiers

La commande **mv** : pour déplacer ou renommer des fichiers

La syntaxe de la commande mv est la suivante :

```
mv [options] [file|directory] [file|directory ...] [destination]
```

Quelques options :

-r ou **-R** : déplacer ou renommer récursivement lorsque le fichier n'est pas vide

-f : force l'opération. Aucun avertissement en cas d'écrasement d'un fichier au cours de l'opération.

-i : le contraire -f. Demande une confirmation à l'utilisateur avant d'écraser un fichier existant.

Exemples d'application:

mv -i /tmp/pics/*.png . : déplace tous les fichiers du répertoire /tmp/pics/ dont le nom se termine par **png** vers le répertoire courant (.). Une confirmation est demandée avant d'écraser un fichier existant.

mv fichier1 /home/user/Documents : déplace le fichier1 dans le répertoire Documents de l'utilisateur user.

mv -rvf fichier* /images /trash : déplace récursivement, sans demander de confirmation, tous les fichiers du répertoire courant dont le nom commence par fichier, ainsi que tout le répertoire /images vers le répertoire /trash Tous les changements effectués sont mentionnés.(-**r** est obligatoire pour déplacer le répertoire images)

mv tp2 tp3 : renomme le fichier tp2 en tp3. Mais si un répertoire tp3 existait déjà, l'effet de cette commande serait de déplacer le fichier tp2 dans le répertoire tp3

2.6. Copier des répertoires et fichiers

- La commande **cp**: pour copier des fichiers et des répertoires

cp remplace les commandes copy et xcopy du DOS et contient d'autres options. Sa syntaxe est la suivante :

```
cp [options] [file|directory] [file|directory ...] [destination]
```

Voici les options les plus utilisées :

-R : copie récursivement, obligatoire pour copier un répertoire, même vide.

-i : demande une confirmation avant d'écraser des fichiers.

-f : le contraire de -i. Remplace tous les fichiers existants sans demander de confirmation.

-v : mode verbeux qui mentionne toutes les actions effectuées par cp.

Exemples d'application:

cp -i /tmp/images/* /images : copie tous les fichiers du répertoire /tmp/images dans le répertoire images/ du répertoire courant, en demandant une confirmation avant d'écraser un fichier.

cp -vR /docs /files/mp3/* /music : copie tout le répertoire docs du répertoire courant, en plus de tous les fichiers du répertoire mp3 qui se trouve sous le dossier files /files/mp3/ dans le répertoire music, lequel est situé dans le répertoire courant.

cp rép1 rép2: copie le fichier rép1 sous le nom de rép2 dans le répertoire courant.

3. Droits d'accès sur les fichiers

3.1. Changer l'utilisateur d'un ou de plusieurs fichiers

La commande **chown** : pour changer l'utilisateur d'un ou plusieurs fichiers.

La syntaxe de la commande chown est la suivante :

```
chown [options] [user.group] [file|directory, file|directory...]
```

ou plus simple encore

```
Chown [ new user] [ fichier] plus simple
```

Les options les plus utilisées :

-R : récursif : Change le propriétaire de tous les fichiers et sous-répertoires d'un répertoire donné.

-v : mode verbeux. Décrit toutes les actions effectuées par chown ; indique quels fichiers ont changé de propriétaire à la suite de la commande ainsi que ceux qui demeurent inchangés.

-c : comme -v, mais ne mentionne que les fichiers pour lesquels il y a eu un changement

Exemples d'application:

chown nobody /shared/book.tex : change le propriétaire du fichier /shared/book.tex pour nobody

chown -Rc reine *.mp3 /concerts : donne la propriété de tous les fichiers dans le répertoire courant se terminant par mp3 et de tous les fichiers et sous-répertoires du

répertoire concerts/ à l'utilisateur reine. Cette commande ne mentionne que les fichiers affectés par la commande.

chown -Rc reine.musique *.exe /concerts : donne la propriété de tous les fichiers dans le répertoire courant se terminant par exe du répertoire courant et de tous les fichiers et sous-répertoires du répertoire concerts/ à l'utilisateur reine et au groupe musique. Cette commande ne mentionne que les fichiers affectés par la commande.

3.2. Changer le groupe propriétaire d'un ou de plusieurs fichiers :

La commande **Chgrp** (CHange GRoup) : ne change que le groupe propriétaire d'un fichier ou d'un groupe de fichiers. Sa syntaxe est très semblable à celle de la commande chown :

chgrp [options] [group] [file|directory, file|directory...]

Les options de cette commande sont les mêmes que pour chown, et elle est utilisée de façon très similaire.

Chgrp [new group] [fichier]

Exemples d'application:

chgrp disk /dev/hd* : attribue au groupe disk tous les fichiers du répertoire /dev dont le nom commence par hd

3.2. Changer droits d'accès sur les fichiers et les répertoires

La commande **chmod** : changer les permissions ou les droits d'accès sur des fichiers et des répertoires

La commande chmod a une syntaxe bien particulière. Sa syntaxe générale est :

chmod [options] <change mode> [file|directory, file|directory...]

Mais ce sont les différentes formes que peut prendre le changement de mode qui la rendront très spécifique. Ceci peut se produire de deux façons :

3.2.1 En mode octal

Les droits d'accès de l'utilisateur propriétaire correspondent alors à des chiffres de la forme

x00, où « x » correspond au droit assigné à l'utilisateur : 4 pour lecture (Read) , 2 pour écriture (Write), 1 pour exécution (eXecute).

De même, les droits d'accès du groupe propriétaire sont de la forme **x0** et ceux des « autres » sont de la forme **x**.

- Pour obtenir le chiffre correct, il suffira d'additionner les droits d'accès assignés.

Exemples d'application:

rwxr-xr-- correspondent à :

rwx : droits d'accès de l'utilisateur propriétaire : 400+200+100

r-x : 40+10 (droits d'accès du groupe)

r-- : 4 (droits d'accès des autres, r--)

= 754 Les droits d'accès sont ainsi exprimés de manière absolue.

La commande sera alors **chmod 754 nom de fichier**

3.2.2. En mode expressions

Les droits d'accès sont ici indiqués par une suite d'expressions séparées par des virgules,

Une expression étant de la forme [catégorie] <+|-|=> [droits d'accès].

La catégorie peut être une combinaison de :

u (User, soit utilisateur, permission pour propriétaire) ;

g (Group, soit groupe, permission pour le groupe propriétaire) ou ;

o (Others, permission pour les autres utilisateurs qui n'appartiennent pas au même group).

Si aucune catégorie n'est spécifiée, le changement s'applique à toutes les catégories.

Un + applique un droit d'accès, un - le retire et un = établit la permission. Les droits d'accès sont définis par une ou plusieurs des lettres suivantes :

r (Read, soit lecture)

w (Write, soit écriture)

x (eXecute, soit exécution)

Remarque: Les options principales sont très similaires à celles de **chown** ou **chgrp**

Exemples d'application:

chmod -R o-w /dossier/docs : enlève de façon récursive le droit d'écriture aux autres utilisateurs (o) sur tous les fichiers et sous-répertoires du répertoire /dossier/docs/ ;

chmod -R og-w,o-x /prive/ : enlève de façon récursive le droit d'écriture pour le groupe et les autres sur tout le répertoire /prive, et retire le droit d'exécution pour les autres ;

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

4. Accès à d'autres disques

4. 1. Monter un système de fichier

-Dans tout système Unix, chaque périphérique (ou nom de partition) est identifié par un fichier. Les disques durs, les lecteurs de disquettes et CD-ROM ne sont pas identifiés par C, A ou D comme c'est le cas du windows.

-Pour pouvoir accéder à un périphérique ou une partition, on dit qu'il faut monter (ou encore attacher) son système de fichier sur un point de montage (un répertoire) situé dans l'arborescence des répertoires. Typiquement, chaque périphérique ou nom de partition est identifié comme suit (Revoir le chapitre 2) :

/dev/fd0 pour une disquette

/dev/cdrom qui est un lien symbolique vers un fichier de périphérique cdrom

/dev/hdb1 pour une première partition esclave du disque dur (hd)

/dev/sda pour un disque dur ou un autre périphérique de stockage SCSI.

(Revoir le chapitre 2)

La commande **mount** pour monter un système de fichier.

mount [options] <nom_de_partition> [point_de_montage]

L'option la plus couramment utilisée est l'option :

-t : permet de spécifier le type de système de fichier. Les systèmes de fichiers sont nombreux, les plus courants sont msdos, ext2, vfat.

Remarque : msdos correspond à un système de fichier FAT16 uniquement avec des noms courts alors que le vfat gère le FAT16 et le FAT32 avec des noms longs.

Exemples d'application:

Pour monter une disquette formatée msdos, la ligne de commande serait la suivante :

```
$ mount -t vfat /dev/fd0 /mnt/floppy
```

4. 2. Monter un système de fichier

La commande pour démonter un périphérique (ou une partition) est **umount**.

La syntaxe de cette commande est la suivante :

```
umount <nom_de_partition>
```

Exemples d'application:

1-Pour démonter une disquette quelque soit son format, la ligne de commande est la suivante :

```
$ umount /dev/fd0
```

2-Définir les expressions suivantes :

```
chmod 756 docs
```

```
mv -iR /home/master/image/ /docs
```

```
rm-Ri images/tp/ /docs/doc1
```

```
-rw-r-xr-x user users1024 Nov 12 11:15 files.
```

- En utilisant la dernière expression, écrire la commande qui vous permet de donner les droits d'accès en écriture pour les membres groupe.

Corrigé :

chmod 756 docs : Changement du droit d'accès du dossier bin en **rwxr-xrw**

mv -iR /home/master/image/ /docs : déplace le dossier **/image** de l'utilisateur master dans le dossier **/docs**

rm-Ri /images/tp/ /docs/doc1 : supprimer récursivement (R) tous le répertoire /tp qui se trouve sous le répertoire images, même si il n'est pas vide, avec demande de confirmation et supprimer aussi le fichier doc1 sous le répertoire docs avec demande de confirmation

-rw-r-xr-x user users1024 Nov 12 11:15 files : - le tiret indique que c'est un fichier, **rw-** : droits de lecture et écriture pour l'utilisateur, **r-x** : droits de lecture et exécution pour les membres du groupe, **r-x** : droits de lecture et exécution pour les autres utilisateurs (others), utilisateur : **user**, groupe : **users**, 1024 date et heure de la dernière modification Nov 12 11:15, **files** : nom du fichier.

-En utilisant la dernière expression, on peut écrire la commande qui nous permet de donner les droits d'accès en écriture pour les membres groupe :

Chmod 675 files ou **Chmod g+w files**

Chapitre V

LE NOYAU LINUX

1. Introduction

Beaucoup de personnes pensent que Linux est équivalent à l'ensemble des logiciels fournis par les distributions. Toutefois, techniquement parlant, Linux ne correspond qu'au noyau, et uniquement au noyau. Les logiciels indispensables tels que les interpréteurs de commandes ou les compilateurs sont en réalité indépendants du noyau.

2. Les modules

Les modules constituent une caractéristique importante du noyau Linux depuis sa version 2.0.x. Cette conception modulaire a permis de simplifier considérablement la configuration du matériel. Elle offre la possibilité de charger sur demande des pilotes pour certains matériels, ce qui est très pratique.

Les modules du noyau sont des fichiers objet produits par un compilateur C. La plupart d'entre eux sont déjà précompilés et disponibles dans la plupart des distributions Linux.

Ils sont généralement stockés dans le répertoire `/lib/modules/<version-du-noyau>`, où `<version-du-noyau>` est une chaîne de caractères décrivant la version du noyau.

Pour déterminer la version du noyau en cours d'exécution sur votre système, vous pouvez utiliser la commande `"uname -r"`.

Exemple : `"uname -r"` peut retourner `"2.4.7-10"` pour indiquer la version du noyau en cours d'exécution.

2.1. Afficher les modules chargés

La commande `"lsmod"` est utilisée pour afficher la liste des modules chargés par le système, ainsi que le nombre d'autres modules qui les utilisent. Voici un exemple d'utilisation de la commande `"lsmod"`:

Module	Size	Used by
nls_utf8	16384	1
vfat	20480	1
fat	81920	1 vfat
nls_iso8859_1	16384	2
nls_cp437	20480	2

Dans cet exemple, la commande "lsmod" affiche la liste des modules actuellement chargés dans le système, avec leur nom et leur taille. La colonne "Used by" indique le nombre d'autres modules qui utilisent chaque module. Par exemple, le module "fat" est utilisé par un autre module nommé "vfat".

Voici aussi un autre affichage de la commande ls mod (Figure 4-1)

Module	Size	Used by	Not tainted
iptable_filter	2412	0 (autoclean)	(unused)
ip_tables	15864	1 [iptable_filter]	
nfs	84632	1 (autoclean)	
lockd	59536	1 (autoclean)	[nfs]
sunrpc	87452	1 (autoclean)	[nfs lockd]
soundcore	7044	0 (autoclean)	
ide-cd	35836	0 (autoclean)	
cdrom	34144	0 (autoclean)	[ide-cd]
parport_pc	19204	1 (autoclean)	
lp	9188	0 (autoclean)	
parport	39072	1 (autoclean)	[parport_pc lp]
autofs	13692	0 (autoclean)	(unused)
e100	62148	1	
microcode	5184	0 (autoclean)	
keybdev	2976	0 (unused)	
mousedev	5656	1	
hid	22308	0 (unused)	
input	6208	0 [keybdev mousedev hid]	
usb-uhci	27468	0 (unused)	
usbcore	82752	1 [hid usb-uhci]	
ext3	91464	2	
jbd	56336	2 [ext3]	

Figure 5-1: Résultat de la commande ls mod

2.2. Charger un module

Pour charger un module dans un noyau en cours d'exécution, vous pouvez utiliser les commandes "insmod" ou "modprobe". Par défaut, ces commandes tentent de charger les modules à partir du répertoire /lib/modules/<version-du-noyau>/kernel/drivers/.

Toutefois, il est recommandé d'utiliser la commande "modprobe" plutôt que "insmod", car cette dernière ne vérifie pas les dépendances et peut causer des problèmes si des dépendances manquent ou sont chargées dans le mauvais ordre.

La commande "modprobe" effectue une vérification des dépendances avant de charger le module, ce qui assure que toutes les dépendances sont satisfaites et que les modules sont chargés dans le bon ordre.

Exemples d'application:

Voici quelques exemples pour charger des modules avec la commande "modprobe" :

Exemple 1 : Pour charger le module du système de fichier MS-DOS, vous pouvez utiliser la commande suivante :

```
$ sudo modprobe msdos
```

Cette commande charge le module "msdos" à partir du répertoire par défaut des modules du noyau.

Exemple 2 : Pour charger tous les modules réseau disponibles jusqu'au premier qui fonctionne, vous pouvez utiliser la commande suivante :

```
$ sudo modprobe -t net
```

Cette commande charge tous les modules réseau disponibles jusqu'au premier qui fonctionne.

Exemple 3 : Pour charger tous les modules réseau disponibles, vous pouvez utiliser la commande suivante :

```
$ sudo modprobe -a net
```

Cette commande charge tous les modules réseau disponibles. Si un module est déjà chargé, il est ignoré.

Exemple 4: Pour afficher la liste des modules réseau disponibles pour les cartes 3Com

```
# modprobe -lt net | grep 3c
/lib/modules/2.4.7-10/kernel/drivers/net/3c501.0
/lib/modules/2.4.7-10/kernel/drivers/net/3c503.0
/lib/modules/2.4.7-10/kernel/drivers/net/3c505.0
/lib/modules/2.4.7-10/kernel/drivers/net/3c507.0
/lib/modules/2.4.7-10/kernel/drivers/net/3e509.0
/lib/modules/2.4.7-10/kernel/drivers/net/3c515.0
/lib/modules/2.4.7-10/kernel/drivers/net/3c59x.o
/lib/modules/2.4.7-10/kernel/drivers/net/pcmcia/3c574_cs.o
/lib/modules/2.4.7-10/kernel/drivers/net/pcmcia/3c589_cs.o
```

vous pouvez utiliser la commande "modinfo" en utilisant le nom du module correspondant au matériel 3Com. Par exemple, si vous souhaitez afficher les informations pour le module "3c59x", vous pouvez utiliser la commande suivante :

```
$ modinfo 3c59x
```

Cela affichera des informations telles que la description du module, l'auteur, la version, les options de module, etc.

2.3. Retirer les modules

Les commandes "rmmod" et "modprobe" peuvent être utilisées pour retirer des modules d'un noyau en marche.

Exemples d'application:

Pour retirer les modules "fat" et "msdos" d'un noyau en marche, vous pouvez utiliser les commandes suivantes :

```
$ sudo modprobe -r fat msdos
```

Cette commande retire les modules "fat" et "msdos" du noyau.

```
$ sudo rmmod fat msdos
```

Cette commande retire également les modules "fat" et "msdos" du noyau.

Il est important de noter que le module ne peut être retiré que s'il n'est pas utilisé par d'autres modules ou processus en cours d'exécution. Si un module est utilisé par d'autres modules ou processus, vous devez d'abord retirer les modules ou les processus qui l'utilisent avant de pouvoir retirer le module.

2.4. Paramétrage des modules

Le fichier de configuration `/etc/modules.conf` (ou `/etc/modprobe.conf` selon les distributions) contient la liste de tous les modules qui doivent être chargés au démarrage du système. Les modules ainsi chargés peuvent être accessibles par le nom d'alias qui leur est attribué dans ce fichier.

Vous trouverez, ci-après, le listing du fichier `/etc/conf.modules` sur un ordinateur Linux Red Hat équipé d'une carte réseau Standard Microsystems et d'une carte réseau NE 2000 :

```
alias scsi_hostadapter aha1740
```

```
alias eth0 smc-ultra
```

```
alias eth1 ne
```

```
options ne io=0x280 irq=10
```

Ce fichier de configuration permet alors d'accéder à la carte réseau SMC par l'alias `eth0` et à la carte NE2000 par l'alias `eth1`.

Dans le cas de la carte réseau NE2000, la dernière ligne définit l'adresse de base d'entrées-sorties et le numéro d'interruption utilisé.

Les noms des cartes réseau pris en charge par le module du noyau se trouvent généralement dans le répertoire `/lib/modules/<version-du-noyau>/kernel/drivers/net/`.

3. Fichiers de dépendances

Le fichier `modules.dep` contient les dépendances entre les modules. Si un module utilise les fonctionnalités d'un autre module, ce fait est indiqué dans ce fichier. Le listing suivant représente un extrait de ce fichier :

```
/lib/modules/2.2.19-7.0.8-e-smith/net/ppp.o: \  
    lib/modules/2.2.19-7.0.8-e-smith/net/slhc.o  
/lib/modules/2.2.19-7.0.8-e-smith/net/ppp_mppe.o: \  
    lib/modules/2.2.19-7.0.8-e-smith/net/ppp.o
```

```
lib/modules/2.2.19-7.0.8-e-smith/net/ppp.o
/lib/modules/2.2.19-7.0.8/fs/nfs.o: \
    /lib/modules/2.2.19-7.0.8/misc/sunrpc.o \
    /lib/modules/2.2.19-7.0.8/fs/locked.o
/lib/modules/2.2.19-7.0.8/fs/nfsd.o: \
    /lib/modules/2.2.19-7.0.8/misc/sunrpc.o \
    /lib/modules/2.2.19-7.0.8/fs/locked.o
```

4. Outils nécessaires Pour compiler le noyau

Pour compiler le noyau, il est nécessaire d'avoir les outils de développement appropriés tels que :

Le compilateur GNU C,

L'utilitaire make,

Les fichiers sources du noyau;

Les fichiers d'en-têtes du noyau.

Le répertoire **/usr/src/linux** contient généralement les sources du noyau, Ce répertoire est un lien symbolique sur le répertoire **/usr/src/linux-<version-du-noyau>**.

4.1. Création du fichier de paramétrage

La première étape dans la création d'un nouveau noyau est la création du fichier de configuration **/usr/src/linux/.config** et la personnalisation du fichier **makefile**.

Il existe plusieurs façons de créer ce fichier :

`make config`

`make menuconfig`

`make xconfig`.

Il est nécessaire d'être dans le répertoire **/usr/src/linux** pour exécuter l'une de ces trois commandes.

5. Compilation du noyau

Une fois que les fichiers .config et makefile ont été personnalisés, le nouveau noyau peut être compilé en suivant ces étapes :

Étape 1 : `make dep` Les fichiers sources sont analysés afin d'identifier les dépendances entre les fichiers d'en-tête (.h). Un fichier (.dep) contenant les dépendances est créé dans chaque répertoire contenant des fichiers sources.

Étape 2 : `make clean` Les fichiers de sortie du noyau précédent sont supprimés pour créer un nouvel environnement de compilation propre. Il est parfois nécessaire d'utiliser la commande `make mrproper`.

Étape 3 : `make bzImage` Le nouveau noyau est créé à cette étape.

Étape 4 : `make modules` Les modules sont créés à cette étape.

Étape 5 : `make modules_install` Les modules compilés à l'étape précédente sont installés dans le répertoire `/lib/modules/<version-du-noyau>`.

Une fois ces étapes terminées, l'image du nouveau noyau se trouve dans le fichier `/usr/src/linux/arch/i386/boot/bzImage`.

6. Installation du noyau

La première étape de l'installation d'un noyau est de copier l'image au bon endroit :

```
cp -p /usr/src/linux/arch/<architecture>/boot/bzImage /boot/vmlinuz-<version-du-noyau>
```

-Il est recommandé de conserver une copie de sauvegarde de l'image `initrd` précédente :

```
mv /boot/initrd-<version-du-noyau>.img /boot/initrd-<version-du-noyau>.img.old
```

-Il est parfois nécessaire de créer une nouvelle image `initrd` à l'aide de la commande suivante :

```
mkinitrd -f /boot/initrd-<version-du-noyau>.img <version-du-noyau>
```

Chapitre VI

PRÉSENTATION DE SERVICES RÉSEAUX SOUS GNU/LINUX

1. Configuration réseaux sous Gnu/Linux

1.1. Introduction

Les systèmes d'exploitation Linux et d'autres systèmes Unix utilisent le protocole TCP/IP pour la communication en réseau. Ce chapitre explore les différents protocoles et concepts liés à la configuration réseau sous GNU/Linux.

1.2. Protocole Description

TCP (en anglais Transmission Control Protocol) : un protocole sécurisé, orienté connexion.

UDP (en anglais, User Datagram Protocol) : Un protocole sans connexion, non sécurisé.

ICMP (en anglais, Internet Control message Protocol) : C'est un protocole qui ne s'adresse en général pas à l'utilisateur, il s'agit plutôt d'un protocole de contrôle

IGMP (en anglais, Internet Group Management Protocol) :

Ce protocole contrôle le comportement des ordinateurs dans le cadre de la multidiffusion IP.

1.3. Adresses IP et routage

Adresses IP Chaque ordinateur de l'Internet possède une adresse 32 bits unique. Ces 32 bits (soit 4 octets)

Exemple : Structure d'une adresse IP

Adresse IP (binaire) : 11000000 10101000 00000000 00010100

Adresse IP (décimale) : 192. 168. 0. 20

Les quatre octets sont séparés par un point lorsqu'ils sont écrits dans le système décimal. L'adresse IP est associée à un ordinateur ou à une interface réseau.

La carte Ethernet aussi possède une adresse non équivoque, appelée adresse MAC (en anglais, Media Access Control). ...

Toutes les adresses d'un même réseau possèdent des chiffres en commun.

- La partie d'adresse commune à toutes les adresse s'appelle la « **partie réseau** » de l'adresse.
- les chiffres qui reste s'appellent « **partie hôte** ».

Le nombre partagé par toutes les adresse d'un même réseau est appelé « **masque réseau** »

Exemples d'application:

Adresse hôte (host address) :192.168.110.23

Masque de réseau (network mask) :255.255.255.0

Partie réseau (network portion) : 192.168.110. (net Id « identifiant réseau » l'adresse du réseau sur le quel l'ordinateur est connecté)

Partie hôte (host portion) : 23 (host Id « identifiant station adresse de l'ordinateur sur le réseau)

Adresse réseau (network address) 192.168.110.0

Adresse de diffusion (bordcast address) 192.168.110.255

1.4. Rattachements des adresses IP avec le masque réseau

Masques réseau et routage : Comme l'ordinateur avec l'adresse IP 192.168.0.0; ne peut tout simplement pas savoir où se trouve l'ordinateur avec l'adresse IP 192.168.0.20, il a fallu introduire les masques réseau.

Voilà ce qui se produit, avant l'envoi d'un paquet sur le réseau : on combine l'adresse cible et le masque réseau au moyen d'un **ET** binaire (en fait le complément à 2). Après cela, on combine aussi l'adresse source et le masque réseau au moyen d'un ET binaire (en fait le complément à 2).

Rattachements des adresses IP avec le masque réseau

Exemples d'application:

Adresse IP (192.168.0.20) : 11000000 10101000 00000000 00010100

Masque réseau (255.255.255.0) : 11111111 11111111 11111111 00000000

Après avoir fait le **ET** binaire le Résultat est le suivant :

Résultat (binaire) : 11000000 10101000 00000000 00000000

Résultat (décimal) : 192. 168. 0. 0 (Net ID)

Host portion: 20 (host Id)

Rattachements des adresses IP avec le masque réseau

Exemples d'application:

Adresse IP (213.95.15.200) : 11010101 10111111 00001111 11001000

Masque réseau (255.255.255.0) : 11111111 11111111 11111111 00000000

Après avoir fait le **ET** binaire le Résultat est le suivant :

Résultat (binaire) : 11010101 10111111 00001111 00000000

Résultat (décimal) : 213. 95. 15. 0

Host portion : 200 (host Id)

1.5. Fichiers réseaux

1.5.1. Le fichier hosts

/etc/hosts : Ce fichier associe les noms d'hôtes et les adresses IP correspondantes. Il faut énumérer ici tous les ordinateurs avec lesquels une connexion IP doit être établie.

On trouve dans ce fichier pour chaque ordinateur une ligne composée de:

L'adresse IP, du nom d'hôte et du nom de machine.

Exemple : */etc/hosts*

127.0.0.1 localhost

192.168.0.20 soleil.exemple.com soleil

192.168.0.0 terre.exemple.com; terre

1.5.2. Le fichier networks

/etc/networks : C'est ici que les noms de réseaux sont convertis en adresses réseau. Leur format ressemble à celui du fichier hosts, mais ce sont les noms de réseaux qui précèdent les adresses.

Exemple : */etc/networks*

loopback 127.0.0.0

localnet 192.168.0.0

1.6. Configuration et détection de cartes réseaux sous Linux

1.6.1. Différentes manières de voir les cartes réseaux

Une carte réseau est vue par le système de deux manières. Elle est d'une part considérée comme un périphérique (device) et d'autre part comme une interface. Le branchement ou la reconnaissance du périphérique entraîne un événement de branchement à chaud (hotplug event). Cet événement entraîne alors l'initialisation du périphérique par le biais du script **/sbin/hwup**. Lors de l'initialisation de la carte réseau en tant que nouvelle interface réseau, le noyau déclenche un événement supplémentaire. Cela entraîne l'installation de l'interface par le biais de **/sbin/ifup**.

1.6.2. Attribution des noms d'interfaces par le noyau

Le noyau associe les interfaces réseau suivant l'ordre de leur enregistrement. L'ordre d'initialisation est déterminant pour l'attribution des noms. Si, en présence de plusieurs cartes réseau, la première tombe en panne, la numérotation de toutes celles qui sont initialisées ensuite sera décalée.

Pour permettre une configuration flexible, la configuration des périphériques (matériel) et celle des interfaces sont séparées et l'association des configurations à leurs périphériques et interfaces respectifs ne dépend plus des noms des interfaces.

La configuration des périphériques se trouve dans **/etc/sysconfig/hardware/hwcfg-***

Tandis que la configuration des interfaces se trouve dans **/etc/sysconfig/network/ifcfg-***.

Les noms des configurations, c'est à dire tout ce qui suit **hwcfg-** ou **ifcfg** peuvent décrire le périphérique par l'endroit où il est monté, par un identifiant propre au périphérique ou par le nom de l'interface. Pour une carte PCI.

Il peut ainsi s'agir de **bus-pci-0000:02:01.0** (emplacement PCI)

Ou **devpid-0x8086-0x1014-0x0549** (identifiant du vendeur et du produit).

Pour l'interface correspondante, on peut aussi utiliser **bus-pci-0000:02:01.0** ou encore **wlan-id-00:05:4e:42:31:7a** (adresse MAC).

Si on ne veut pas associer une configuration à une carte donnée mais plutôt à une carte d'un type donné (une seule carte de ce type étant branchée à la fois), on choisit un nom de configuration moins particulier. On utilise alors par exemple **bus-pcmcia** pour toutes les cartes PCMCIA.

D'autre part, les noms peuvent aussi être limités par l'utilisation du type d'interface. Ainsi, **wlan-bus-usb** concernera toutes les cartes WLAN branchées avec USB.

La meilleure configuration sera recherchée dans le fichier **/sbin/getcfg**. Ce fichier fournit toutes les informations que l'on peut utiliser pour décrire un périphérique. La spécification exacte des noms de configuration se trouve dans la page de manuel du **getcfg**.

1.6.3. Fichiers de configuration

Cette section donne un aperçu des fichiers de configuration réseau et explique leur fonction ainsi que le format utilisé.

/etc/syconfig/hardware/hwcfg-*

Vous trouverez dans ces fichiers les réglages des cartes réseau et des autres périphériques. Ils contiennent les paramètres essentiels comme le module du noyau, le mode d'amorçage et l'ordre des scripts. Vous trouverez des détails à ce propos dans la page de manuel de **hwup**. Les fichiers de configuration **hwcfg-static-*** seront utilisées au démarrage de **Coldplug** indépendamment du matériel présent.

/etc/sysconfig/network/ifcfg-*

Ces fichiers contiennent les réglages des interfaces réseau. Ils contiennent entre autres le mode d'amorçage et l'adresse IP. Les paramètres possibles sont décrits dans la page de manuel de **ifup**.

Dans les fichiers **ifcfg-*** un réglage ne doit être utilisé que pour une seule interface.

/etc/sysconfig/network/config,dhcp,wireless

Le fichier config contient les réglages généraux concernant le comportement de **ifup**, **ifdown** et **ifstatus**. On trouve de même des commentaires dans **DHCP** et **wireless**, où sont regroupés

des réglages généraux concernant DHCP et les cartes réseau sans fil. Toutes les variables de ces fichiers peuvent également être utilisées dans **ifcfg-*** et y sont prioritaires.

/etc/sysconfig/network/routes,ifroute-* : Le routage statique pour TCP/IP est réglé ici.

Exemples d'application:

1. Calculez l'adresse réseau, l'adresse de diffusion et la partie hôte pour l'adresse IP 172.16.32.45 avec un masque réseau de 255.255.0.0.

Corrigé: Adresse réseau: 172.16.0.0 ici c'est **la classe B**

Adresse de diffusion: 172.16.255.255

Partie hôte: 32.45 ici c'est **la classe B**

2. Identifiez le protocole approprié à utiliser pour une application qui nécessite une communication sans connexion et rapide.

Corrigé: UDP (User Datagram Protocol)

3. Donnez un exemple de situation où le protocole ICMP est utilisé.

Corrigé: Le protocole ICMP est utilisé pour envoyer des messages de demande et de réponse pour vérifier la disponibilité d'une machine sur un réseau (par exemple, avec la commande "ping").

4. Quelle est la fonction principale du fichier **/etc/hosts**?

Corrigé: Le fichier **/etc/hosts** associe les noms d'hôtes aux adresses IP correspondantes, permettant la résolution de noms localement sans dépendre d'un serveur DNS.

5. Quel fichier convertit les noms de réseaux en adresses réseau dans le système GNU/Linux?

Corrigé: Le fichier **/etc/networks** convertit les noms de réseaux en adresses réseau.

2. Réseaux sous GNU/LINUX: PARTEGE DE FICHERS

2.1. NIS (Network Information Service), NFS Système de fichier partagé (Network file system) et Système de partage avec Windows (Samba) :

2.1.1. NIS (Network Information Service)

Permet de centraliser les fichiers de configuration sur un serveur NIS pour éviter de dupliquer tous ces fichiers en autant d'exemplaires que d'ordinateurs à gérer sur un réseau.

Évite les copies multiples, permet de n'avoir à maintenir les fichiers que sur une seule machine serveur.

NIS et NFS permettent de construire des systèmes informatiques répartis assurant un partage des fichiers et des données cohérent et transparent.

2.1.1.1. Services/démons utilisés par le service NIS:

portmap : fichier responsable de mise en correspondance numéro de ports TCP/IP et numéro de processus RPC (voir /etc/rpc pour les numéros réservés)

Côté serveur:

ypserv implémente le serveur NIS

yppasswd permet de changer un mot de passe sur le serveur NIS depuis un client NIS (démon rpc.yppasswd)

ypxfrd accélère les transferts entre serveur maître et esclave (démon rpc.ypxfrd)

Côté client:

ypbind implémente le client NIS

2.1.1.2. Commandes utiles

nisdomainname, domainname et ypdomainname : affiche le nom de domaine NIS

rpcinfo -P afficher les services portmap

Côté serveur

/usr/lib/yp/ypinit -m initialiser le serveur maître

cd /var/yp; make refaire les cartes du serveur maître, et les propager vers les serveurs esclaves

/usr/lib/yp/ypinit -s initialiser un serveur esclave

Côté client :

ypwhich afficher le nom du serveur NIS
yppasswd changer son mot de passe sur le serveur NIS
ypcat passwd afficher le fichier passwd du serveur
ypcat group afficher le fichier group du serveur
ypcat ... afficher le fichier... du serveur

2.1.2. NFS – Système de fichier partagé (Network file system).

NFS permet, comme NIS, de rendre un réseau transparent pour les utilisateurs. NFS permet de partager des systèmes de fichiers en réseau. NFS est un service asymétrique : Il se compose d'un **serveur NFS** et de **clients NFS**. Ces deux fonctions peuvent bien entendu coexister sur une même machine.

Celle-ci proposera dans le même temps des systèmes de fichiers au réseau "exportation" et montera des systèmes de fichiers appartenant à d'autres machines "importation".

2.1.2.1. Services/démons utilisés par le service NFS:

portmap fichier responsable de mise en correspondance numéro de ports TCP/IP <->
numéro de processus RPC (voir /etc/rpc pour les numéros réservés)

Côté serveur:

rpc.nfsd implémente la partie utilisateur du protocole
rpc.statd prise en compte du redémarrage des serveurs, pour une gestion correcte des
 verrous (implémente le protocole RPC NSM :Network Status Monitor) rpc.lockd gestion
 du verrouillage des fichiers (implémente le protocole NLM :Network Lock Manager)
rpc.mountd implémente la partie serveur du protocole, à l'écoute des demandes de montage

Côté client:

rpc.statd et rpc.lockd

2.1.2.2. Commandes utiles (voir man pour les détails)

Côté serveur:

showmount -a liste des clients/répertoire utilisé
-d liste des répertoires montés par des clients
-e liste des répertoires exportés

exportfs -a relit le fichier /etc/exportfs
-ra remet à jour la liste des répertoires exportés

rpcinfo -p afficher les services portmap

Côté client

rpcinfo -p [host] afficher les services portmap

showmount -e host liste des répertoires exportés par le serveur «host»

2.1.2.3. Configuration serveur

Créer un partage NFS côté serveur avec des options et des droits d'accès aux documents de partage qu'on retrouve dans le fichier exports

Principales options du fichier exports (serveur):

ro read-only (accès en lecture seule au répertoire exporté)

rw read-write: le client accède au répertoire en lecture/écriture

root_squash transforme les accès UID root en UID nobody

no_root_squash les accès root sur le client restent root sur le serveur

all_squash Convertit tous les UID/GID en utilisateurs anonymes. Utile pour exporter avec NFS des répertoires publics.

2.1.2.4. Le fichier */etc/exports* :

Le fichier **/etc/exports** du serveur NFS contient une liste des systèmes de fichiers locaux accessibles pour les clients NFS. Le contenu de ce fichier est maintenu par l'administrateur système. Chaque système de fichiers dans cette liste est suivi d'une liste d'options et d'une liste de contrôle d'accès. La liste est utilisée par le fichier `exportfs` pour renseigner le fichier `mountd`.

Le fichier `exports` peut ressembler au script suivant du fichier :

```
# /etc/exports

/home          soleil (rw) venus (rw)

/usr/X11       soleil (ro) venus (ro)

/usr/lib/texmf soleil (rw) venus (ro)

/              terre (ro,root_squash)

/home/ftp      (ro)

# End of exports
```

dans cette exemple le dossier `/home` est partager en lecture et écriture (rw) pour les deux machine soleil et venus, le dossier `X11` du répertoire `usr` est partager en lecture seul (ro) pour les deux machine soleil et venus, le dossier `/usr/lib/texmf` est mis à l'exportation en lecture et écriture (rw) pour la machine soleil et en lecture seul pour la machine venus, le dossier racine `/` est partagé en lecture seul `ro,root_squash` sans le privilège `root`, le dossier `/home/ftp` est partager en lecture seul (ro)

2.1.2.5. Lancement de NFS sur un client

Coté client, il suffit d'effectuer le montage de la ressource exportée par un serveur NFS : Le lancement des services se fait classiquement:

Il faut que le redirecteur de ports (`RPC portmapper`) soit en service

Exécuter la commande **`reportmap start`** en tant qu'utilisateur `root`

Utiliser la commande suivante pour le montage (importation) de fichier ou dossier :

`mount -t nfs serveur: pathdistant point demontage`

- pathdistant: est le nom du répertoire exporté par le serveur
- point de montage : est le nom du répertoire local à partir duquel seront utilisables les fichiers du serveur NFS

Pour automatiser le montage des ressources NFS sur un poste client, on utilise le fichier /etc/fstab.

consulter l'aide avec la commande **man mount** pour les options de montage supportées par NFS.

La figure 6-1 montre comment le client peut accéder à la ressource de l'exportation

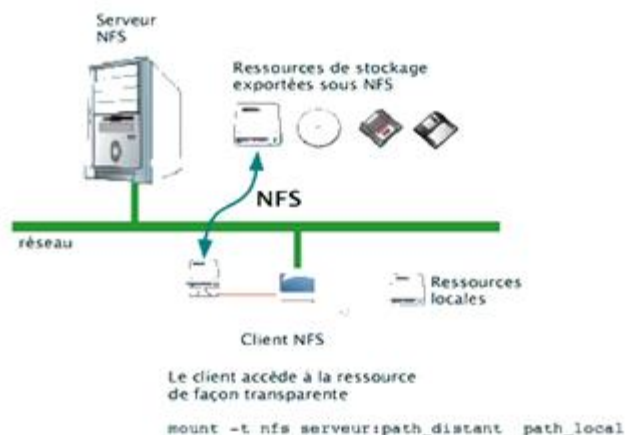


Figure 06-2: Le client accède à la ressource de façon transparente

Exemple de montage :

Mount -t nfs soleil:/home /home : le montage du répertoire /home de la machine soleil dans le répertoire /home point de montage avec le système nfs par l'option -t

Démons utilisés sont : Le fichier /etc/exports il est lu par **mountd** et **nfsd**.

Ainsi, lorsque ce fichier a été modifié, il est nécessaire de redémarrer **mountd** et **nfsd** afin de permettre à ces modifications d'être prises en compte. Pour ce faire, le plus simple est d'exécuter la commande :

rcnfsserver restart

2.1.2.6. Application de partage avec NFS par les commandes côté serveur

Créer un partage NFS côté serveur

Créer un partage NFS côté serveur avec les étapes suivantes :

1. Sur le poste serveur, qui peut être un simple ordinateur, ouvrir un **Terminal** en local ou une connexion distante **SSH**.

2. Installer le paquet requis :

```
sudo apt-get install nfs-kernel-server
```

3. Créer un répertoire local qui sera partagé :

```
sudo mkdir /srv/partage
```

4. Attribuer les droits d'accès et d'écriture, par exemple :

```
sudo chmod 755 /srv/partage/
```

```
sudo chgrp grplocal /srv/partage/
```

5. Ouvrir le fichier de configuration du partage NFS :

```
sudo vi /etc/exports
```

7. Déclarer le chemin du dossier à partager et les autorisations d'accès réseau :

```
/srv/partage          192.168.0.0/24(rw,async,no_subtree_check)
```

Ici, le dossier /srv/partage sera accessible depuis tout le sous-réseau 192.168.0.0 en lecture et écriture (rw ou read write).

8. Recharger le service serveur NFS pour prendre en compte les modifications.

```
sudo service nfs-kernel-server reload
```

9. Vérifier les montages offerts par le serveur NFS :

```
showmount -e
```

```
Export list for serveur:
```

```
/srv/partage 192.168.0.0/24
```

Si la commande ne renvoie pas l'export correct de la liste, faire un « sudo service nfs-kernel-server restart »

2.1.2.7. Application de partage avec NFS par les commandes côté client

Connecter un poste Linux client au partage NFS, avec les étapes suivantes :

1. Ouvrir un **Terminal** ou une connexion **SSH** sur le poste client.
2. Créer un dossier local qui sera le lien vers le partage réseau :

```
sudo mkdir /mnt/partage
```

3. Déclarer le partage réseau dans le fichier fstab :

```
sudo vi /etc/fstab
```

4. Ajouter une ligne :

```
serveur:/srv/partage /mnt/partage nfs defaults,user,auto
```

5. **Enregistrer et quitter** le fichier fstab.

6. Demander le montage :

```
sudo mount -a
```

7. Aller dans le nouveau partage, lister le contenu, écrire dedans.

```
cd /mnt/partage
```

```
ll /mnt/partage
```

```
touch test
```

2.1.2.8. Exercices avec corrections sur le partage de fichiers NFS et le service NIS

Exercice 1: En utilisant NFS, partagez le répertoire /data sur le serveur avec le client ayant l'adresse IP 192.168.1.10. Autorisez l'accès en lecture et en écriture au répertoire partagé.

Correction: Sur le serveur, modifiez le fichier /etc/exports pour ajouter la ligne suivante :

```
/data 192.168.1.10(rw)
```

Redémarrez le service NFS sur le serveur pour prendre en compte les modifications :

sudo systemctl restart nfs-server

Sur le client, montez le répertoire partagé :

sudo mount server:/data /mnt

Exercice 2 :

mount -t nfs post1/doc home/master/partage.

Corrigé :

mount -t nfs post1/bin/doc home/master/partage : monter le fichier doc qui se trouve dans le répertoire bin du post1 (avec le système de fichier nfs) dans le répertoire partage dans le home de l'utilisateur master.

Exercice 3: Configurez un serveur NIS pour centraliser les fichiers de configuration des utilisateurs sur le réseau. Créez un utilisateur "Master" avec le mot de passe "password123" et ajoutez-le au groupe "users".

Correction: Sur le serveur, installez les paquets nécessaires pour le service NIS :

sudo apt install ypserv yp-tools

Modifiez le fichier **/etc/ypserv.securenets** pour spécifier les sous-réseaux autorisés à accéder au service NIS. Créez l'utilisateur " Master" et ajoutez-le au groupe "users" :

```
sudo useradd Master -m -G users sudo passwd Master
```

Générez les fichiers NIS :

sudo /usr/lib/yp/ypinit -m sudo systemctl restart ypserv

Sur le client, configurez le fichier **/etc/yp.conf** pour spécifier l'adresse IP du serveur NIS. Redémarrez le service NIS sur le client :

sudo systemctl restart ypbind

Vous devriez maintenant pouvoir vous connecter en tant qu'utilisateur "Master " avec le mot de passe "password123".

Exercice 4: Changez le mot de passe de l'utilisateur NIS " Master " depuis un client NIS.

Correction: Sur le client, exécutez la commande suivante pour changer le mot de passe de l'utilisateur " Master " :

yppasswd Master

Suivez les instructions pour entrer le nouveau mot de passe.

Exercice 5: Affichez le nom de domaine NIS sur le client.

Correction: Sur le client, exécutez la commande suivante pour afficher le nom de domaine NIS :

`nisdomainname`

Exercice 6: Ajoutez le répertoire /shared au fichier exports du serveur NFS pour qu'il soit accessible en lecture seule par tous les clients.

Correction: Sur le serveur, modifiez le fichier /etc/exports pour ajouter la ligne suivante :

/shared (ro)

Redémarrez le service NFS sur le serveur pour prendre en compte les modifications :

sudo systemctl restart nfs-server

2.1.3. Présentation des systèmes de fichiers de partage avec Windows le serveur Samba

Samba a été initialement développée par l'Australien Andrew Tridgell dès 1991 et distribuée sous licence libre GNU GPL 3. Son nom provient du protocole de partage et d'impression de fichiers d'IBM et réutilisé par Microsoft, nommé SMB (Server message block), auquel ont été ajoutées les deux voyelles a : " SaMBa ".

Ce protocole est employé par Microsoft pour le partage de diverses ressources (fichiers, imprimantes, etc.) entre ordinateurs équipés de Windows. Samba permet aux systèmes Unix d'accéder aux ressources de ces systèmes et vice-versa.

Lorsque les deux systèmes de partage de fichiers (NFS, Samba) sont installés pour comparaison, Samba se révèle moins performant que NFS au niveau des taux de transferts.

Néanmoins, une étude a montré que Samba 3 était jusqu'à 2,5 fois plus rapide que l'implémentation SMB de Windows serveur 2003. Toutefois, samba n'est pas compatible

avec IPv6. Vous trouverez les informations concernant Samba sur le page d'accueil de ce site : <https://www.samba.org/>

2.1.3. 1. Les principales fonctionnalités de Samba:

- Partager un disque Linux pour des machines Windows
- Partager une imprimante Linux avec des machines Windows
- Partager une imprimante Windows à partir d'un hôte Linux
- Devenir un contrôleur de domaine (simuler le système Windows NT Server) et

permettre ainsi les authentification réseaux sur un domaine, le stockage centralisé des profils Windows et l'exécution de scripts de démarrage

- Gérer des listes de machines présentes sur le réseau et leur mise à disposition

Pour tous types de clients

2.1.3.2. La configuration du Samba

La configuration de Samba est effectuée par l'intermédiaire d'un fichier de configuration unique. La configuration de Samba sur une machine Linux (ou sur un autre Unix) est contrôlée par un seul fichier, `smb.conf`. Cependant, le nom par défaut de ce fichier de configuration est déterminé au moment de la compilation de l'outil `samba`. Dans beaucoup de cas, c'est `/etc/smb.conf`. Mais sur la Mandriva, avec le package `samba`, le fichier opérationnel est `/etc/samba/smb.conf`

`/etc/samba/smb.conf` : Ce fichier décrit les ressources que l'on désire partager la commande :

`/etc/init.d/samba restart` (afin de redémarrer le serveur `samba` et par la même occasion prendre en compte les modifications du fichier `smb.conf`).

2.1.3.3. Démarrage du Samba

L'architecture de Samba est constituée principalement d'un serveur et d'un client, ainsi que de quelques outils permettant de tester la configuration.

- Le serveur est constitué de deux démons (programmes chargés en mémoire) :

Le démon **smbd** est un noyau du serveur fournissant les services d'authentification et d'accès aux ressources. Le démon **nmbd** permet de montrer les services offerts par samba (affichage des serveurs samba dans le voisinage réseau,...).

- Le client smbclient est un client pour linux fournissant une interface permettant de transférer des fichiers, accéder à des imprimantes.

Exemple : Avec le cas de ubuntu5 pour installation du serveur SAMBA

- **Les paquets nécessaires** sont :

samba-common_3.4.0-3ubuntu5_all.deb

smbclient_3.4.0-3ubuntu5_i386.deb

samba_3.4.0-3ubuntu5_i386.deb

samba-doc_3.4.0-3ubuntu5_all.deb.

Et l'installation des paquets se fait avec les commandes suivantes :

```
sudo apt-get install samba-common
```

```
sudo apt-get install smbclient
```

```
sudo apt-get install samba
```

```
sudo apt-get install samba-doc
```

2.1.3.4. Vérification de l'installation

Après installation, le serveur de Samba devrait normalement être apte à démarrer (sans aucun partage de fichiers ou d'imprimante) en lançant la commande suivante:

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

Le serveur devrait alors renvoyer les lignes suivantes:

```
Starting SMB services
```

```
Starting NMB services
```

- La commande suivante permet de contrôler que les deux démons sont correctement lancés :

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

Le serveur devrait alors renvoyer les lignes suivantes:

smbd (pid 1054) is running...

nmbd (pid 1056) is running...

2.1.3.5. Les étapes pour créer des dossiers partagés

1. Depuis Poste de Travail, cliquez sur le dossier, le disque dur ou le lecteur de CD-ROM que vous voulez partager.
2. Sélectionnez le menu Fichier, puis Partager...
3. Cliquez sur Partager ce dossier, puis cliquez sur le bouton Autorisations. Par défaut, tout le monde peut accéder à votre dossier partagé. Vous pouvez modifier les droits de Tout le monde en cochant ou décochant les autorisations. (Voir figure)
4. Si vous voulez ajouter un utilisateur en particulier, cliquez sur le bouton Ajouter, puis définissez cet utilisateur. (Voir figure)
5. Cliquez sur le bouton OK. (figure)

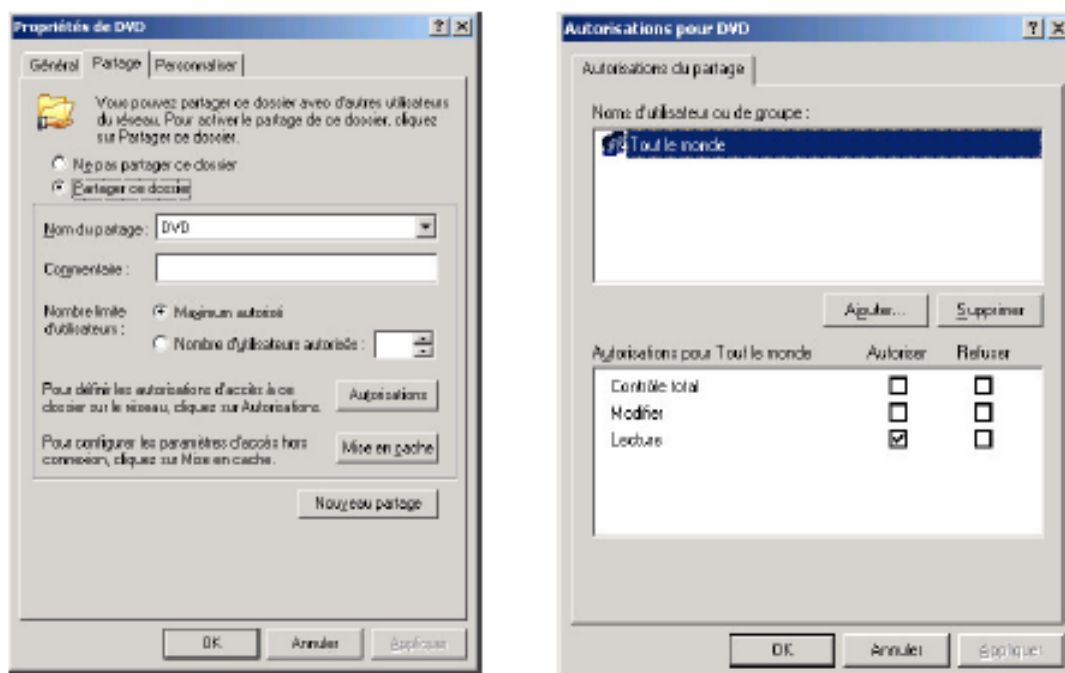


Figure 6-2: Configuration du dossier de partage

2.1.3.6. Paramétrage d'un client Linux (smbclient)

Un client SMB pour un hôte Unix est inclus dans la distribution de Samba. Vous pouvez utiliser cette utilitaire pour transférer des fichiers entre un « serveur » Windows et un client Linux.

Pour voir ce que partage une machine donnée, utilisez la commande :

`/usr/sbin/smbclient -non hôte`

Où « hôte » est le nom NetBIOS de la machine dont vous voulez voir les partages.

Vous obtiendrez une liste des « services » en partage, c'est-à-dire le nom des répertoires ou des machines qu'il partage pour vous. A moins que le serveur SMB n'ait aucune protection, vous allez devoir saisir un mot de passe. Utilisez le mot de passe pour le compte personnel sur cette machine

Les utilitaires smbmount et smbunmount permettent de monter et de démonter des ressources SMB

2.1.3.7. Exemples d'application:

Exemple 1 : Montez une ressource SMB sur votre client Linux en utilisant la commande mount. Assurez-vous de spécifier le nom d'utilisateur Windows approprié. Pour monter une ressource; la syntaxe est la suivante :

```
# mount -t smbfs /ordi1/Partage1 /mnt/test -o username=login_Windows
```

Avec "//ordi1/partage1" le chemin du partage SMB et "login_Windows" le nom d'utilisateur Windows approprié.

Exemple 2: Démontez une ressource SMB montée précédemment en utilisant la commande umount.

```
umount /mnt/test
```

Exemple 3: Accéder aux ressources Samba à partir d'une machine de type Unix.

Sous Linux un smbclient permet en premier lieu de vérifier l'existence d'un serveur Samba sur le réseau et de lister les ressources qu'il partage grâce à la commande:

smbclient nom_serveur_smb ou **smbclient 192.168.1.45 (@IP de serveur samba)** Ou
adresse machine

Une fois les ressources de partage identifiées (**ressource**, **nom_du_répertoire**) il est possible d'accéder à chacune d'entre elles par la commande avec nom de l'utilisateur:

smbclient \\nom_serveur_smb\ressource -U nom_utilisateur ou

smbclient //nom_du_serveur/nom_du_répertoire -U nom_utilisateur -I IP_du_server

CONCLUSION

En conclusion, ce parcours à travers les différents chapitres nous a permis d'explorer en profondeur le système d'exploitation Linux. Nous avons pu acquérir des connaissances générales sur son historique, sa structure de base étant un logiciel libre. Nous avons également évoqué certaines distributions les plus populaires (REDHAT, Mandrake, DEBIAN, Ubuntu ...) avec leurs caractéristiques distinctives.

Ensuite, nous avons exploré les différents systèmes de fichiers tolérés par le système d'exploitation Linux en passant par la nomination particulière des disques et partitions.

L'administration sous Linux a été un sujet clé, où nous avons étudié la gestion des utilisateurs, des groupes et des droits d'accès sur les fichiers et répertoires. Nous avons également appris à utiliser les commandes pour manipuler les fichiers, les répertoires et les droits d'accès dans le chapitre 4, ce qui constitue une compétence essentielle pour les administrateurs système. Cela nous a permis de mieux saisir comment monter des systèmes de fichiers et accéder à d'autres disques.

Le noyau Linux a été présenté comme l'élément central du système d'exploitation, avec des explications sur les modules, leur chargement et leur paramétrage. Nous avons parallèlement établi les étapes nécessaires pour compiler et installer le noyau.

Enfin, nous avons exploré les services réseaux sous Linux, en mettant l'accent sur la configuration réseau, les adresses IP, le routage et le partage de fichiers avec des services tels que NIS, NFS et Samba.

Ce cours nous a fourni les outils nécessaires pour tirer le meilleur parti de cette plateforme open-source qui est le système d'exploitation Linux. Que ce soit pour un usage personnel ou professionnel, les connaissances acquises dans ces différents chapitres nous permettront d'exploiter pleinement les capacités de Linux et d'accomplir une variété de tâches.

En conclusion, Linux offre une alternative puissante et flexible aux autres systèmes d'exploitation. Avec une communauté active et des ressources abondantes, Linux continue d'évoluer et de s'imposer comme un choix privilégié pour de nombreux utilisateurs à travers le monde.

RÉFÉRENCES BIBLIOGRAPHIQUES

Bovet, D. P., & Cesati, M. (2005). Understanding the Linux Kernel (3ème édition). O'Reilly. ISBN: 978-0596005658

Love, R. (2010). Linux Kernel Development (3ème édition). Addison-Wesley Professional. ISBN: 978-0672329463

Love, R. (2013). Linux System Programming (2ème édition). O'Reilly Media, Inc. ISBN: 978-1449339531

Michael Wielsch, Jens Prahm, H.G. Esser (2000). La Bible Linux (2ème édition). Collection PC 100% pratique. ISBN: ISBN non disponible.

Nicolas, Pons (2022). Linux, Principes de base de l'utilisation du système (8ème édition). Ressources informatiques. ISBN: 978-2-409-03776-4

Philippe, Banquet (2023). Linux, Administration système et exploitation des services réseau (5ème édition). Expert. ISBN-13 : 978-2746092174

Webographie

<https://docplayer.fr/5130641-Le-centre-de-controle-de-mandriva.html>

<http://livre21.com/LIVREF/F6/F006034.pdf>

<https://jplu.developpez.com/tutoriels/systeme/presentation-linux>

<https://linux.goffinet.org/administration/processus-et-demarrage/noyau-linux/>

https://www.academia.edu/Documents/in/Administration_Reseau_Sous_Linux_Tri

https://doc.ubuntu-fr.org/nfs#avoir_les_droits_d_administration_sur_les_deux_machines

<https://linux.goffinet.org/administration/configuration-du-reseau/synthese-des-commandes-reseau/>

<https://www.doc-developpement-durable.org/file/Projets-informatiques/cours-%26-manuels-informatiques/Linux/Administration-Reseau-Linux.pdf>

Unité d'enseignement: UED 1.1**Matière 1 : Système Linux****VHS : 22h30 (Cours : 1h30)****Crédits : 1****Coefficient : 1****Objectifs de l'enseignement :**

Cette matière permettra à l'étudiant d'avoir une idée sur l'utilisation et la personnalisation d'un système GNU/Linux.

Connaissances préalables recommandées :

Des connaissances sur le système d'exploitation Windows et Programmation.

Contenu de la matière :**Chapitre 1. Installation du système GNU/Linux (3 Semaines)**

- Historique du système linux.
- Linux est un logiciel libre (GNU/Linux)
- Les distributions linux
- Où chercher de l'aide sur linux ?

Chapitre 2. Gestion des disques et autres périphériques (2 Semaines)

- Structure d'un disque dur
- Partitionnement du disque dur
- Conventions pour nommer les disques et partitions sous GNU/linux
- Systèmes de fichiers/Le système de fichiers Ext2fs (Second Extended File System)
- Processus de démarrage et connexion au système linux

Chapitre 3. Administration de base (2 Semaines)

- Gestion des utilisateurs sous Linux
- Utilisateurs et groupe sous linux
- Droits d'accès sur les fichiers

Chapitre 4. Introduction à la ligne de commande (2 Semaines)

- Interpréteur de commande
- Commandes de manipulation des répertoires et des fichiers.
- Commandes de manipulation des utilisateurs et des groupes.
- Accès à d'autres disques (Monter un système de fichier).

Chapitre 5. Noyau, Performances et Tuning (3 Semaines)

- Les modules du noyau
- Manipulation des modules du noyau
- Paramétrage des modules
- Compilation du noyau et installation du noyau

Chapitre 6. Présentation de services réseaux sous GNU/Linux (3 Semaines)

- Configuration réseaux sous Gnu/Linux
- Les réseaux sous GNU/Linux: partage des fichiers
- Présentation des systèmes de fichiers (NFS, NIS)
- Présentation des systèmes de fichiers de partage avec Windows (Samba)

Mode d'évaluation :

Examen : 100%.

Références bibliographiques :

1. [Richard Stallman et la révolution du logiciel libre.](#)
2. [Linus](#) Torvalds et David Diamond, *“Il était une fois Linux : L’extraordinaire histoire d’une révolution accidentelle”*, Osman Eyrolles Multimédia, 2001.
3. Chris DiBona (dir.), *“Tribune libre : Ténors de l’informatique libre”*, O’Reilly, 1999.