



Licence Informatique
IUT Fotso Victor de Bandjoun

ENVIRONNEMENT LINUX

Support de cours

par

Narcisse TALLA

Université de Dschang
Narcisse.talla@univ-dschang.org

Mars 2016

SOMMAIRE

SOMMAIRE	i
Chapitre 1. Fondamentaux sur l'environnement Linux	1
1.1. Introduction.....	1
1.1.1. Définition de Linux.....	1
1.1.2. La communauté Linux.....	1
1.1.3. Logiciel libre et le GPL.....	1
1.1.4. Les différentes licences d'utilisation des logiciels	2
1.2. Historique des systèmes Unix	2
1.3. Caractéristiques des systèmes Unix.....	3
1.3.1. Caractéristiques de Linux.....	3
1.3.2. Les distributions Linux.....	4
Chapitre 2. Environnement GNU/Linux.....	7
2.1. Installation de Linux.....	7
2.1.1. Préparation à l'installation de Linux	7
2.1.2. Partitionnement des disques.....	7
2.1.3. Systèmes de fichiers journalisés	9
2.1.4. Présentation des environnements graphiques	9
2.1.5. Les disques durs et les partitions	12
2.2. Configuration réseau et Linux.....	14
2.2.1. Informations indispensables.....	14
2.2.2. Configuration initiale lors de l'installation	15
2.2.3. Interface Ethernet.....	15
2.2.4. Fichiers de configuration.....	15
2.3. Installer un chargeur ou booter Linux	16
2.4. Gestion des packages et applications	17
2.4.1. Les commandes usuelles.....	18
2.5. Outils communs d'utilisation d'un réseau sous Unix.	19
2.5.1. Quelques fichiers importants.....	19
2.6. Quelques commandes utiles.....	20
2.7. Prise en main du shell.....	22
2.7.1. Commandes usuelles d'Unix.....	22
2.7.2. Métacaractères du shell.....	24
2.7.3. Création de liens.....	24
2.8. Système de fichiers	24
2.8.1. Structure du système de fichiers	25
2.8.2. La gestion des droits d'accès aux fichiers.....	25
Chapitre 3. Gestion des droits et Initiation au Shell.....	27
3.1. Rôle de l'administrateur réseau sous Unix.....	27
3.2. Gestion des utilisateurs.....	27
3.2.1. Notion d'utilisateur	27
3.2.2. Le système d'authentification et ses composants	27
3.3. Initiation au Shell.....	32
3.3.1. Jokers.....	32
3.3.2. Groupement de commandes.....	32
3.3.3. Les redirections	32
3.3.4. Editeurs de textes et utilitaires Unix	33
Chapitre 4. Utilitaires.....	35

4.1.	Outils multimédia sous Linux	35
4.1.1.	Outils de musique	35
4.1.2.	Outils de graphisme	35
4.1.3.	Le logiciel de retouche d'image : GIMP.....	36
4.2.	Impression	36
4.2.1.	Partager une imprimante Linux pour Windows.....	36
4.2.2.	Partager une imprimante Windows pour Linux.....	37
4.3.	Archivage, compression.....	37
4.3.1.	Contenu du dossier.....	37
4.3.2.	Compression/décompression de fichiers	38
4.3.3.	Commandes gzip/gunzip.....	38
4.3.4.	Commandes bzip2 et bunzip2.....	39
4.4.	Sauvegarde et archivage de fichiers.....	39
4.4.1.	Sauvegarde de fichiers.....	39
4.4.2.	Archivage de fichiers : Commande tar.....	39
4.5.	Configuration du service d'impression.....	40
4.6.	Outil graphique printtool	40
Chapitre 5.	Linux UBUNTU administration	42
5.1.	The user root.....	42
5.2.	Administrator group users	42
5.3.	Useful Comands for Linux Administration	42
5.3.1.	Useful Comands for Linux newbies.....	42
5.3.2.	Advanced Commands for Middle Level Linux Users.....	50
5.3.3.	Advanced Commands for Linux Experts.....	61
Bibliographie.....		73

Chapitre 1. Fondamentaux sur l'environnement Linux

1.1. Introduction

Cet enseignement a pour objectif de donner à l'étudiant des connaissances de base sur l'environnement Linux.

1.1.1.Définition de Linux

Linux est un système d'exploitation multiutilisateur et multitâche qui peut être installé sur plusieurs plateformes différentes (Intel, Apple, DEC, Sparc, etc.). Il est conforme à la norme POSIX et peut fonctionner sans problème avec d'autres systèmes d'exploitation comme Apple, Microsoft et Novell. Le système d'exploitation Linux est libre. Il peut être copié et redistribué gratuitement. Le code source est également disponible pour tous de la même manière.

1.1.2.La communauté Linux

La communauté Linux comprend l'ensemble des développeurs, pas seulement du noyau d'ailleurs, et les utilisateurs. Ils mettent à profit l'Internet pour tisser des liens et promouvoir une entraide à tous les niveaux. Aussi de nombreuses associations se sont créées pour regrouper les enthousiastes et faire connaître Linux au plus grand nombre en organisant diverses manifestations. La communauté des développeurs de Linux, répartie à travers le monde, développe de nombreux logiciels au sein de projets tout aussi nombreux. Les principaux, ou du moins les plus médiatiques, sont le noyau lui-même, bien sûr, mais aussi deux projets de « bureaux », KDE et GNOME, et de nombreux projets de développement de solutions de clusters, de haute disponibilité, ...

Cette communauté Linux présente deux caractéristiques qu'il faut souligner :

- ❖ Plutôt que de se plaindre que « ça ne marche pas », le linuxien tente généralement de découvrir l'origine du problème, et dans la mesure du possible, de le résoudre par lui-même.
- ❖ Sur Internet, on peut trouver aisément la solution à la plupart des problèmes courants rencontrés sous Linux. En contrepartie, on attend la même démarche des nouveaux utilisateurs. Une fois un problème résolu, il peut être utile de faire profiter la communauté en publiant la solution sur une page Web, par exemple.

1.1.3.Logiciel libre et le GPL

On qualifie de libre un logiciel qui répond aux principaux critères suivants :

- ❖ il est livré avec ses sources, qu'il est possible de modifier ;
- ❖ il est possible de le redistribuer sans contrainte ;
- ❖ il est possible de distribuer les modifications que l'on a faites. Ce type de diffusion n'interdisant pas la commercialisation.

Au début des années 80, Richard Stallman, alors professeur au M.I.T, le célèbre Massachusetts Institut of Technologie, s'est vu refuser l'accès au code d'un programme. Il prit conscience du risque de la propriétérisation des logiciels. Il affirme qu' « il n'est pas normal que lorsqu'on utilise un logiciel, on ne puisse pas savoir ce qu'il y a dedans ». En effet, les logiciels commerciaux sont alors livrés uniquement en format exécutable ; ils sont donc illisibles par le commun des mortels, ou du moins des programmeurs.

L'idée de Stallman l'amènera rapidement à la création d'une fondation appelée la *Free Software Foundation (FSF)*, comprendre Fondation pour le logiciel libre. Le premier travail de la FSF a été de produire une licence parfaitement adaptée au logiciel Libre : il s'agit de la GPL (General Public Licence, ou Licence Publique Générale). En 1984 il initialise le projet GNU (prononcer gnou). Ce terme est un acronyme récursif qui signifie *GNU is Not Unix*.

La GPL, la licence des produits GNU est une licence libre. Elle ajoute une contrainte particulière : elle autorise l'utilisation d'un programme GPL à l'intérieur d'un autre programme, à la condition que celui-ci soit distribué sous licence GPL. Elle garantit ainsi la non-propriétérisation des logiciels. Le hasard et la nécessité se rencontrent : Linus Torvalds a adopté la GPL pour son noyau Linux, de même qu'il a récupéré l'ensemble des programmes développés par le projet GNU pour fournir une base d'applications pour le système d'exploitation fiable que nous connaissons et qu'il est plus exact de désigner par GNU/Linux.

1.1.4. Les différentes licences d'utilisation des logiciels

Tout usage d'un logiciel doit s'accompagner de l'acquisition d'une licence d'utilisation. Il s'agit d'un contrat précisant la manière dont vous aurez le droit d'utiliser ce logiciel. On peut classer les logiciels en plusieurs catégories en fonctions du degré de liberté que procure la licence d'utilisation. C'est ainsi que l'on distingue :

- ❖ Les logiciels propriétaires. Ils s'accompagnent de licences qui n'autorisent que l'utilisation, généralement assortie de restrictions (limitation d'usage à un nombre donné d'utilisateurs, installation restreinte à un ordinateur, etc.).
- ❖ Les partagiels (*sharewares*) et les gratuiels (*freewares*). Il s'agit de logiciels propriétaires mais les partagiels sont distribués gratuitement, charge à l'utilisateur de s'acquitter de son prix s'il continue à l'utiliser après une période définie. Les gratuiels sont des logiciels dont le prix est nul.

1.2. Historique des systèmes Unix

En 1989, un étudiant américain du nom de Andrew S. Tannenbaum, trouve que le système proposé sur son ordinateur muni d'un processeur de la famille i386 ne lui convient pas tout à fait. Il s'attèle à une tâche ardue puisqu'il se met à écrire son propre système. Un an plus tard, il sort la version 1.1 de son système qu'il appela Minix. C'est un système à vocation éducative, accessible pour un étudiant. Mais très rapidement Andrew tannenbaum s'essouffle ; il n'a pas que ça à faire et puis il est tout seul.

En 1991, le jeune finlandais Linus Torvalds est étudiant en informatique à l'université d'Helsinki. Il utilise Minix sur son ordinateur tout neuf. Il est convaincu que l'informatique

doit être pour l'utilisateur un moyen de gagner du temps, un outil et non une contrainte. En dehors des systèmes qu'il peut utiliser sur les serveurs de son université, il trouve que tous les systèmes (y compris Minix) auxquels il a accès pour sa machine personnelle sont soit trop coûteux, soit trop peu efficaces. Le système Minix qu'il a installé sur son ordinateur doté d'un processeur 386 s'avère trop limité pour lui. En conséquence, il décide d'écrire lui-même un système d'exploitation qui le satisfasse. Pour cela, il se fonde sur le système d'exploitation Minix.

Au mois d'août 1991 est né Linux (linus'Unix), L'Unix de Linus dans sa version 0.0.1. Cette version n'était même pas exécutable. L'archive ne contenait que les rudiments des sources du noyau et il fallait compiler Linux sous Minix pour l'utiliser un tant soit peu.

Le 5 octobre 1991, Linux devient autonome lors de la sortie officielle de la version 0.0.2. Celle-ci permet de faire tourner quelques applications GNU (logiciels libres) essentielles comme le compilateur gcc ou le shell bash. Mais Linus Torvalds réalise très bien que réitérer l'expérience d'Andrew Tannenbaum sur Minix n'est pas viable. Il se convainc qu'il faut répartir la tâche de développement, trouver des collaborateurs et étendre ce projet à un cercle le plus large possible. Pourquoi pas la planète ? A cette époque où la guerre froide vient de se terminer, l'Internet commence à envahir les universités et amorce l'explosion que nous connaissons. Linus Torvalds en profite pour lancer un appel qui est resté célèbre, sur le forum comp.os.minix et décide de mettre le code source sous licence GPL de telle sorte que tout le monde peut alors participer au développement de Linux.

1.3. Caractéristiques des systèmes Unix

1.3.1. Caractéristiques de Linux

1.3.1.1. La sécurité

La sécurité est nécessaire pour protéger les données des utilisateurs mais aussi pour éviter de servir de point de départ à une attaque sur Internet. L'expérience a montré par exemple que le pirate précautionneux n'attaque jamais directement sa cible depuis sa machine ; cela permettrait de remonter jusqu'à lui sans trop de difficultés. En piratant au préalable une machine peu défendue, celle d'un particulier en général, le pirate peut brouiller les pistes, causant en fait des ennuis à sa victime. Autre phénomène qui prend de l'ampleur : la propagation rapide des virus qui apparaissent à l'air de l'Internet.

Linux offre de bonnes garanties de sécurité. D'abord il est conçu dès le départ pour fonctionner en réseau. Ensuite il profite de son mode de développement. Tous les développeurs intéressés peuvent examiner le logiciel. Par conséquent, il y a plus de chances qu'un oeil averti découvre un bogue potentiellement exploitable. Une fois un problème diagnostiqué, il n'est pas nécessaire d'attendre la publication d'un correctif par l'éditeur original du logiciel. Les chances sont très grandes qu'un développeur membre de la communauté le corrige rapidement si ce n'est le responsable du projet ou de son équipe.

1.3.1.2. Fiabilité et stabilité

Un effort particulier a été fait lors du développement de linux pour s'assurer que le système soit fiable et stable :

- ❖ Les applications sont cloisonnées et ne peuvent faire planter le système entier ;
- ❖ Linux est conçu pour des serveurs devant fonctionner des mois durant ;
- ❖ les bogues découverts sont corrigés très rapidement.

1.3.1.3. Simplicité

De grands progrès ont été réalisés ces dernières années pour rendre le système accessible à tous. L'abondance de la documentation permet à tous de résoudre la plupart des problèmes les plus courants ; La grande similarité de principes entre la plupart des composants du système facilite la tâche de compréhension des bases du système. Il suffit pour tirer pleinement profit du système, de passer un peu de temps pour comprendre ces bases.

Avec une bonne configuration, Linux se révèle plus simple que la plupart des autres SE dans le cadre de la maintenance quotidienne. D'où un gain de temps appréciable.

Sa conception multi-utilisateur rend possible et même facilite la mise en place d'un environnement distinct pour chacune des personnes amenées à utiliser la machine. Cet environnement peut être personnalisé, empêchant un utilisateur d'accéder aux données des autres. C'est d'ailleurs par défaut le cas.

1.3.2. Les distributions Linux

1.3.2.1. Pourquoi les distributions

L'abondance des applications disponibles et la multiplicité des programmes ayant les mêmes objectifs, c'est-à-dire disponibles pour répondre à un même besoin, pose un problème de choix pour un utilisateur ; Compte tenu du nombre impressionnant d'applications que l'on peut être amené à installer sur une machine, il faut absolument disposer d'un outil d'installation et de gestion de ces applications ; Enfin, faciliter la prise en main pour les débutants doit être une préoccupation des éditeurs et de la communauté. D'où l'importance des manuels d'installation, et autres outils de configuration du système.

1.3.2.2. Notion de distribution

Les distributions sont des ensembles logiciels contenant le système linux, une interface graphique (contrairement à Ms-Windows, l'interface graphique est dissociée du reste du système) et un ensemble de programmes. En général elles disposent aussi d'un outil de gestion des applications installées (comme InstallShield sous Ms-Windows, mais en plus riche) et d'autres outils de configuration du système, plus orientés débutant que les méthodes de configuration traditionnelle qui s'adressent principalement aux administrateurs.

1.3.2.3. Les distributions majeures Linux

La distribution Slackware

C'est l'ancêtre des distributions Linux. Pionnier dans ce domaine, la société Slackware commercialisa une distribution Linux contenue sur une centaine de disquettes (le CD-

Rom n'étant pas aussi répandu comme de nos jours). Compte tenu de son âge, cette distribution ne disposait pas d'interfaces conviviales de configuration du système et de ce fait s'adresse à des utilisateurs expérimentés.

La distribution Redhat

Considérée par certains comme la référence professionnelle, l'apparition de cette distribution a constitué un événement marquant dans le monde Linux. Elle intègre un système complexe de gestion d'applications qui permet la mise à jour d'une application tout en conservant les fichiers de configuration et qui gère les dépendances. Ce système (Redhat Package Manager : rpm) peut donc installer automatique tous les éléments logiciels requis pour faire fonctionner une application.

La facilité de trouver des paquets *rpm* milite également en faveur de Red Hat et de Mandriva, mais on trouve également tout en *.tar.gz*, ce qui permet d'installer tout logiciel sur toute distribution ; il suffit de le compiler si le package ne contient pas une version binaire.

La distribution Suse

Diffusée par la société allemande SuSE GmbH créée en 1992, la première version de cette distribution sortira en 1993 sur disquettes et était basée sur la distribution Slackware. Depuis elle a beaucoup évolué. Techniquement très proche de la Redhat, elle est basée sur le même type de packages : RPM. Mais elle se différencie essentiellement par le programme d'installation basé sur l'outil YAST et son utilitaire SaX de configuration de l'interface graphique.

- ❖ YAST - Yet Another Setu Tool - est un puissant outil d'installation et de configuration disposant des caractéristiques suivants :
 - peut être lancé après l'installation ;
 - prend en charge une bonne partie de la configuration de la machine dont l'installation du réseau (config de base, NFS, ISDN, Samba, ...) grâce à une détection hardware et un partitionnement automatiques ;
 - permet d'effectuer des mises à jour et des sauvegardes ;
 - dispose d'une fonction intégrée de gestion des utilisateurs et des groupes ;
 - permet une compilation et une personnalisation du noyau ;
- ❖ SaX - SuSE Advanced X-Configuration - est l'utilitaire d'installation et de configuration du serveur Xwindow (l'interface graphique).

La distribution Debian

La distribution Debian est la seule distribution non commerciale et totalement libre et gratuite. Gérée par une association de plus de 400 volontaires bénévoles dont la majorité sont des développeurs, elle est aussi la plus complète : elle propose un très grand nombre d'applications/d'outils regroupés en plus de 2500 packages.

Son objectif est triple :

- ❖ Fournir une distribution entièrement libre ;

- ❖ Développer un outil de gestion d'application puissant ;
- ❖ N'intégrer que des applications stables.

Points faibles de cette distribution :

- ❖ le mode graphique n'est pas privilégié : de ce fait, son installation et sa configuration sont relativement plus difficile ;
- ❖ les nouvelles fonctionnalités sont tradivement diffusées.

En conclusion, on peut retenir que la distribution Debian est une bonne distribution très complète et 100% libre mais pour les connaisseurs, et essentiellement pour les serveurs où la fiabilité et la sécurité sont primordiales. Autrement on peut utiliser Ubuntu qui à l'origine est basée sur Debian mais qui à su s'imposer comme une distribution conviviale et facile à prendre en main.

La distribution Corel Linux

Cette distribution est l'oeuvre de l'éditeur canadien Corel (qui édite aussi Corel Wordperfect).

Elle est basée sur une Debian, pour laquelle un programme d'installation graphique et une amélioration de KDE ont été développés, de même qu'un nouveau gestionnaire de fichiers (Corel File Manager) qui permet par exemple d'accéder au réseau Windows par Samba sans passer par les outils complexes de configuration. La gestion de l'impression a aussi été facilitée.

La distribution Mandriva

La distribution Linux-Mandriva anciennement Mandrake est une distribution récente, à l'origine basée sur La RedHat. Elle a été la première (avec sa version 5.x) à intégrer l'interface graphique KDE, conviviale et simple d'emploi. Un effort tout particulier a été fait au niveau de la traduction, pour la rendre utilisable par le plus grand nombre. A partir de la version 6, Linux-Mandriva a pris son indépendance, et n'est plus basée sur la Redhat.

La distribution Linux-Mandriva offre de nombreux packages supplémentaires par rapport à la Redhat ainsi qu'une installation et une utilisation faciles. Pour cela, elle est dotée d'outils graphiques pour toutes les étapes de l'installation (programme DrakX), pour la gestion des différents programmes (programme RpmDake), et pour l'installation de nouveaux périphériques (DrakConf). Tous ces outils sont intégrés dans le gestionnaire de fenêtres KDE, une interface graphique intuitive, avec une traduction en français et une personnalisation du bureau KDE.

Chapitre 2. Environnement GNU/Linux

2.1. Installation de Linux

2.1.1. Préparation à l'installation de Linux

Linux ne supporte pas tous les matériels qui existent dans le commerce ou ailleurs.

Prérequis :

Microprocesseur : Intel x86 à base de microprocesseur 386 minimum (ou compatible comme AMD, CYRIX, etc.).

Disque dur : prévoir pour le système lui-même de l'ordre de 10 Go pour une distribution assez récente.

Mémoire : prévoir environ 2 Go avec un système X. 512 Mo ou même moins peut aussi aller, mais avec les nouveaux environnements sous X, la rapidité n'est pas garantie.

2.1.2. Partitionnement des disques

2.1.2.1. Préparation du disque dur

Avant de commencer l'installation, il faut libérer de la place pour les nouvelles partitions Linux. Si l'on désire installer Linux sur un disque vierge, pas de problème. Par contre, si l'on veut faire cohabiter Windows et Linux sur le même disque, et que Windows est déjà installé sur la totalité du disque, il va falloir des aménagements. Alors soit on vire tout et on réinstalle Windows, c'est le plus simple évidemment, soit on redimensionne la partition Windows. Pour ce faire, on peut utiliser un logiciel approprié, comme FIPS ou *Partition Manager* (un freeware sous DOS) ou encore *Partition Magic* (produit commercial).

Remarque 1 : *Partition Manager* a l'air intéressant, il est *freeware* et fonctionne sous DOS. Il permet de gérer les partitions, de sauvegarder le MBR (Master Boot Record, c'est là que vient s'enregistrer LILO dans un fichier), de le restaurer, de modifier la taille d'une partition, de la formater, etc.

Remarque 2 : la distribution Ubuntu, tout comme plusieurs autres distributions Linux, est livrée avec un utilitaire de repartitionnement très performant, sachant même redimensionner les partitions au format NTFS.

2.1.2.2. Partitionnement des disques avant installation

Après s'être logué en root, il faut taper "fdisk" AVANT de taper "setup").

Pour partitionner un disque, on doit appeler la commande fdisk suivie du nom du disque sur lequel sera installé Linux, par exemple "fdisk /dev/hda". Des logiciels plus conviviaux existent comme *DiskDruid* (disponible lors de l'installation d'une RedHat par exemple) ou *DiskDrake* de Mandriva fonctionnant en mode graphique.

Les principales commandes de fdisk sont :

- ❖ **n** Crée une partition, de type linux native (ext2fs) par défaut.
- ❖ **t** Change le type d'une partition (83 linux native, 82 linux swap).

Dans fdisk, tapez "m", afin d'accéder à la liste de toutes les commandes possibles. Pour toutes les autres commandes de fdisk, on peut se référer à l'installation-HOWTO

Attention : le nombre de partitions dites "primaires" est limité à 4. Ainsi, si vous souhaitez définir plus de 4 partitions sur un même disque, l'une de ces 4 partitions primaires doit être définie comme "étendue", elle contiendra alors toutes les autres partitions. Les partitions primaires sont numérotées de 1 à 4, les partitions créées à l'intérieur de la partition étendue sont numérotées à partir de 5.

Pour ce qui est du nombre de partitions, il faut tout d'abord une partition pour le swap. Le swap c'est la mémoire virtuelle de Linux. Sa taille dépend de la mémoire physique. Il est conseillé de lui attribuer le minimum entre le double de la RAM et 1024 Mo.

Pour le reste, on a le choix : La chose la plus simple est une seule partition (en plus du swap), qui contiendra toutes les données (système & utilisateur) ; En fait, il est intéressant de créer au moins deux partitions : une qui contiendra le système et une autre pour les données. On peut ainsi réinstaller Linux et même formater la partition système sans perdre les données sauvegardées. Pour la même raison, on peut pousser le raffinement jusqu'à avoir une troisième partition afin de contenir les logiciels installés (OpenOffice par exemple).

2.1.2.3. Si nous installons un serveur :

L'une des méthodes les plus utilisées, c'est de mettre le / sur un petit disque ou une partition, puis /usr sur un autre disque en point de montage. Si notre système va être un serveur avec beaucoup d'utilisateurs qui vont avoir beaucoup de données personnelles à stocker, on peut créer aussi un /home en point de montage sur un autre disque. On va aussi généralement réserver un disque ou une partition pour le répertoire temporaire que l'on montera dans /tmp, et encore une autre pour le /var. En fait, on ne va laisser dans le disque contenant la racine que les informations qui bougent peu et dont le volume augmente peu. Ainsi on ne charge pas le disque de point de montage / des répertoires /usr, /home, etc., ce qui permet de facilement maintenir le système (si /home est plein, on prend un disque plus gros que l'on monte en /home et on recopie tout dessus, etc.) De plus si un disque crashe, si ce n'est pas le disque contenant la racine /, ça permet de rebooter quand même la machine, de travailler un minimum pour réinstaller un disque sans à avoir à réinstaller un système de base.

2.1.2.4. Si vous installez sur une machine perso :

Sur une machine mono-utilisateur, voire mono-disque, on peut faire une partition pour la racine / (mini 1Go, recommandé 2Go au moins), une pour /home et une pour le swap (1024 Mo par exemple). On peut aussi éventuellement faire une partition pour /opt qui contiendra les logiciels auxiliaires (qui ne proviennent pas de la distribution) installés. Cela permet de mettre à jour, voire de réinstaller la distribution sans trop de peine.

Lorsqu'on dispose d'assez de place sur le disque (par exemple 500Go), pour une utilisation personnelle, 5 à 10Go suffisent pour la racine ; 5 à 10Go suffisent pour le /usr,

on garde le swap entre 512 et 1024 Mo, et enfin, le reste pour **/home**. On peut aussi créer une partition **/mnt/data** de 200Go pour stocker les données.

2.1.3. Systèmes de fichiers journalisés

Le système de fichiers (File System ou FS), c'est la façon dont le système d'exploitation structure les données sur le disque dur. Comme tout système d'exploitation, Linux dispose de ses propres systèmes de fichiers :

- ✿ **EXT2FS (ext2)** : c'est le système de fichiers historique de Linux, il n'est pas forcément le meilleur choix car, n'étant pas journalisé, si vous éteignez mal votre ordinateur, linux prendra beaucoup de temps pour tester les disques durs (de la même manière que le *scandisk* de Microsoft).
- ✿ **EXT3FS (ext3)** : c'est l'évolution logique du système de fichiers *ext2*. Il présente l'avantage d'être compatible avec *l'ext2* tout en étant journalisé. Il n'implémente toutefois pas les toutes dernières innovations en matière de journalisation. Ce système de fichiers est indiqué lorsque l'on veut pouvoir relire des informations depuis windows.
- ✿ **REISERFS** : c'est un système de fichiers journalisé. C'est un bon choix. Il est réimplémenté à partir de zéro et bénéficie de beaucoup d'innovations.

Windows utilise quant à lui un système VFAT (partitions limitées à 2 Go) ou FAT32, voire FAT (l'ancien système de fichier du DOS). Il n'est pas possible d'installer Linux sur une partition de ce type, car elles ne gèrent pas les permissions. Certaines version récentes de Windows utilisent aussi le système de fichiers NTFS, mais Microsoft n'a pas fourni les spécifications techniques de ce format et Linux (ainsi que la plupart des autres systèmes d'exploitation) est incapable de le gérer correctement (il faut donc prévoir une partition en VFAT-FAT32 pour partager des informations facilement entre Linux et Windows) : Linux sait lire les informations sur une partitions NTFS mais pas en écrire (cette fonctionnalité est en cours de développement pour certaines distributions).

2.1.4. Présentation des environnements graphiques

2.1.4.1. Objectif de l'interface graphique

L'objectif de l'interface graphique est de fournir un environnement de travail convivial pour les utilisateurs. Elle n'est qu'une application parmi d'autres et donc est considérée comme étrangère au système d'exploitation qui au sens strict ne doit que servir d'interface entre les périphériques (clavier, carte vidéo, souris, disques, etc.) et les différentes applications.

2.1.4.2. Eléments constitutifs de l'interface graphique

L'interface graphique comporte trois types d'éléments distincts : le serveur d'affichage X, le gestionnaire de fenêtres et les applications. La modularité de l'interface graphique permet de changer un aspect sans modifier les autres et de donner à chaque utilisateur son environnement favori.

Le serveur d'affichage X : Encore appelé X Window System ou X11, le serveur d'affichage X est le programme chargé d'afficher les programmes sur le ou les écrans disponibles et d'autoriser ou non les applications à afficher à l'écran. Dans un environnement multi-utilisateurs, la nécessité de X11 s'explique par le fait que plusieurs utilisateurs connectés en même temps sur la même machine lancent des applications différentes et ont besoin de les rediriger vers leur écran.

Le gestionnaire de fenêtres : Le gestionnaire de fenêtres, seconde couche de l'interface graphique, est un programme chargé de définir l'ergonomie de l'affichage. Il définit la façon dont les fenêtres s'affichent, les boutons présents sur leur barre de titre, les icônes présentes à l'écran, etc. Sous linux il y en a quelques dizaines parmi lesquelles KDE et GNOME.

Les applications : La troisième couche se compose des applications (ou programmes). Tous les programmes sont conçus pour fonctionner indépendamment des couches inférieures. Ils peuvent éventuellement communiquer entre eux en utilisant une interface commune. KDE offre par exemple une telle interface aux différentes applications.

2.1.4.3. Les interfaces courantes sous Linux

Il existe plus d'une cinquantaine d'interfaces graphiques à l'heure actuelle dont on peut citer :

GNOME : Bien que moins répandu et moins utilisé que KDE, GNOME est l'interface qui monte. Elle est soutenue par le projet GNU et a été officiellement adopté par plusieurs fabricants d'Unix pour devenir leur interface principale.

WindowMaker : C'est un gestionnaire de fenêtres (window manager) pour XWindow libre et populaire. WindowMaker est un gestionnaire de fenêtres assez rapide, facile à prendre en main et à configurer. Sa popularité grandissante s'explique aussi par sa finition irréprochable et par son esthétique travaillée.

KDE : C'est l'acronyme de K Desktop Environment, est très utilisé sous Linux. C'est l'interface retenue par défaut par Linux-Mandriva et d'autres distributions. Le premier objectif de KDE fut de fournir une interface graphique offrant un moyen simple permettant aux applications graphiques, membres de la couche la plus haute, de communiquer entre elles et de devenir un environnement de bureau complet.

2.1.4.4. L'environnement KDE

Sur le plan technique, KDE vise l'intégration des caractéristiques suivantes qui manquent dans les environnements graphiques de X :

- ❖ un protocole Glisser-Lacher commun pour les applications et le bureau ;
- ❖ une configuration du bureau simple, fondée sur des dialogues, pour toutes les fonctionnalités et les applications de bureau ;
- ❖ un système d'aide uniformisé pouvant être accessible de toutes les applications ;
- ❖ un atelier de développement logiciel commun ;
- ❖ un atelier de préparation de document semblable à CORBA ou à ActiveX ;

- ❖ une transparence réseau améliorée au niveau application ;
- ❖ un type de développement simple d'emploi (SDK) pour faciliter la création d'applications compatibles.

Du point de vue de l'utilisateur, les objectifs les plus importants sont :

- ❖ la présentation d'un bureau attrayant et moderne ;
- ❖ le barrage aux problèmes que les applications rencontrent avec les connexions réseau (rendant le réseau transparent pour le bureau et ses applications) ;
- ❖ une aide en ligne intégrée fournissant des points d'accès et une interface utilisateur uniforme pour toutes les applications ;
- ❖ une apparence uniforme pour toutes les applications y compris les menus, les raccourcis clavier, les palettes de couleurs, etc. ;
- ❖ la création d'un produit international, avec un support clavier, menu, aide en ligne et application, pour de nombreuses langues ;
- ❖ la fourniture de centaines d'applications intégrées.

La version 2.0 de KDE intègre une suite bureautique complète, appelée Koffice. A terme, KDE pourrait devenir l'environnement standard de Linux, voire d'Unix s'il parvient à intéresser les éditeurs d'Unix propriétaires.

L'interface est constituée de trois principaux composants : le lanceur d'applications, le menu de sélection du bureau et les icônes d'applications.

Le lanceur d'applications : Ce menu déroulant encore appelé menu K permet de lancer les applications faisant partie de l'environnement KDE. Celles-ci sont classées par thème à l'intérieur du menu. Son contenu précis dépend des logiciels choisis à l'installation.

Le menu de sélection du bureau : KDE peut afficher les fenêtres sur plusieurs écrans virtuels (quatre par défaut). Chacun de ces écrans se comporte comme un bureau totalement indépendant, sur lequel on peut choisir un fond, des icônes, et des styles de fenêtres différents.

Les icônes d'applications : Sur la barre d'outils se trouvent des icônes représentant des applications que l'on souhaite avoir sous la main. Il suffit de cliquer sur une icône pour démarrer l'application correspondante. Mais il est toujours possible de lancer une application non accessible via une icône en tapant **Alt-F2**.

2.1.4.5. Personnalisation du bureau

Lancement automatique des applications

Il est possible de lancer les applications automatiquement à la connexion de l'utilisateur grâce au dossier Démarrage, représenté par la première icône présente sur le bureau. Voici les étapes de la procédure :

- ❖ Ouvrir le dossier Démarrage ;
- ❖ Cliquer droit dans l'espace de travail ;
- ❖ Sélectionner Nouveau/Application ;

- ❖ Taper le nom de l'application dans la zone Exécuter. Si l'application n'est pas graphique, il faut plutôt taper *rxvt -e « commande »* ou cocher la case « Lancer dans un terminal ».

Ajout d'une icône sur le bureau ou d'une entrée dans le lanceur

Un clic droit n'importe où sur le bureau affiche un menu qui permet de rajouter une ou plusieurs icônes sur le bureau. Il suffit de sélectionner Nouveau/Application. Le dialogue est identique à celui du dossier Démarrage.

L'ajout d'une entrée dans le lanceur se fait à l'aide de l'éditeur de menu, accessible par Lanceur d'application/Utilitaires/Editeur de menus. Des fenêtres représentant les menus s'affichent. Pour en rajouter une, il suffit de cliquer avec le bouton droit sur la représentation du lanceur, choisir Nouveau et insérer la nouvelle entrée.

Dans la fenêtre qui s'affiche, il est possible de choisir le nom de l'élément à insérer, le programme à exécuter en mode graphique, puis cliquer sur OK pour confirmer.

2.1.5. Les disques durs et les partitions

2.1.5.1. Les disques durs

Pour les disques *IDE*, la numérotation se fait avec un préfixe "*hd*", suivi par une lettre, "*a*" pour le premier IDE maître, "*b*" pour le premier esclave, etc.

En résumé on a :

- ❖ ***hda*** Disque maître sur le 1er contrôleur IDE
- ❖ ***hdb*** Disque esclave sur le 1er contrôleur IDE
- ❖ ***hdc*** Disque maître sur le 2eme contrôleur IDE
- ❖ ***hdd*** Disque esclave sur le 2eme contrôleur IDE

Les lecteurs de CD-ROM IDE apparaissent comme un disque dur. Si votre CD est maître sur la deuxième nappe par exemple, il sera nommé *hdc*.

Pour les disques *SCSI*, cette fois c'est l'ID SCSI qui va donner la lettre, le préfixe étant "*sd*". Prenons par exemple 3 disques SCSI d'id 0, 1 et 3 on va avoir :

- ❖ ***sda*** Disque d'ID 0
- ❖ ***sdb*** Disque d'ID 1
- ❖ ***sdd*** Disque d'ID 3

Les lecteurs de CD-ROM SCSI, les ZIP, JAZ, etc. SCSI ainsi que les périphériques de stockage USB (appareils photo, clefs USB...) apparaissent comme un disque dur SCSI ; ils se nomment aussi *sdx* suivant leur ID. Astuce : pour connaître la liste de vos périphériques de ce type, tapez la commande : *cdrecord -scanbus*.

2.1.5.2. Partitions et organisation des fichiers sous Linux

Les *partitions* sont quant à elles représentées par le nom du disque dur, suivi d'un chiffre représentant la partition. Par exemple, la première partition du premier disque IDE se nomme *hda1*, de même, la troisième partition du disque SCSI d'ID 1 se nommera *sdb3*.

L'arborescence

Sous Linux, ou tout autre Unix d'ailleurs, il n'y a pas de lettre représentant le disque dur comme le C: de Windows/DOS. Il y a une seule arborescence, qui débute à la "racine", notée '/', et tous les disques durs/partitions apparaissent dans cette arborescence avec leur contenu de façon transparente comme un répertoire.

Les partitions / les points de montage

Les partitions que l'on va créer sur les disques durs seront attachées à des répertoires de l'arborescence, ou "*points de montage*". C'est ce qu'on appelle "*monter une partition*". Le point de montage est un simple répertoire, vide avant le montage, et qui après le montage, représente le contenu de la partition montée. Par exemple, supposons que nous ayons créé une partition `/dev/hda2` (premier disque IDE, 2ème partition) pour contenir le système. On *montera* cette partition dans le répertoire racine `/`. Si la partition `/dev/hda3` est destinée à contenir les données utilisateur, on la montera dans le *point de montage* `/home`. On verra le contenu de cette partition dans le répertoire `/home` comme si c'était n'importe quel autre répertoire, alors que physiquement les données sont sur une autre partition. De même, pour accéder à la disquette, on montera le périphérique `/dev/fd0` dans le point de montage `/mnt/floppy` (ou `/floppy` selon la distribution) et pour accéder au CD-ROM, on montera le périphérique `/dev/cdrom` dans le point de montage `/mnt/cdrom` ou `/cdrom`.

Les répertoires standards

Linux possède des répertoires "spéciaux" à la racine (un peu comme le `C:\windows`), qui sont classiques dans les systèmes Unix et peuvent ou non représenter un point de montage pour une partition.

Tableau 1. Les répertoires standards de Linux sont :

<code>/</code>	La racine du système = la base.
<code>/boot</code>	Fichiers utilisés pour booter le noyau
<code>/usr</code>	Programmes, données, accessibles par les utilisateurs et non nécessaires lors du boot
<code>/home</code>	Tous les répertoires de base des utilisateurs et de certains services comme le FTP, le serveur Web, Samba, etc.
<code>/dev</code>	Tous les périphériques
<code>/bin</code>	Programmes nécessaires au boot, et disponibles ensuite à tous les utilisateurs
<code>/sbin</code>	Programmes nécessaires au boot, disponibles au root uniquement
<code>/var</code>	Fichiers de log, spool d'impression, de mail, etc. En général, fichiers qui changent tout le temps
<code>/etc</code>	Fichiers de configuration du système
<code>/root</code>	Répertoire home de l'administrateur
<code>/lib</code>	Bibliothèques et modules (comme les DLL de Windows) nécessaires au boot
<code>/tmp</code>	Fichiers temporaires
<code>/mnt</code>	Répertoire proposé pour monter des périphériques (ou <code>/mnt/cdrom</code> , <code>/mnt/floppy</code> , <code>/mnt/zip</code> , etc.)
<code>/proc</code>	Fichiers d'information sur la configuration système (les fichiers sont générés

	dynamiquement, pas de modifications possibles). Les fichiers dans ce répertoire n'existent pas sur le disque, c'est Linux qui les crée en temps réel lorsque vous en avez besoin.
/opt	Dans ce répertoire (optionnel), on peut ajouter des logiciels afin qu'ils ne s'installent pas dans les répertoires système.

Les périphériques

Tout périphérique est identifié à un fichier qui se trouve dans le répertoire */dev* (comme *device*). Par exemple, le disque maître du deuxième port IDE est identifié par */dev/hdc*, sa deuxième partition par */dev/hdc2*. Le lecteur de disquette est représenté par */dev/fd0*, etc.

Pour "monter" un périphérique ou une partition, il faut utiliser la commande *mount*.

Syntaxe : `mount -t type-du-support fichier-périphérique point-de-montage`

Traditionnellement, les périphériques tels que les disques et partitions sont montés dans le répertoire */mnt* : par exemple */mnt/cdrom* pour un cdrom.

Exemples : Montage d'un lecteur de CD-ROM : `$ mount -t iso9660 /dev/hdb /mnt/cdrom`

Montage d'une partition Windows : `$ mount -t vfat /dev/hda1 /mnt/win98`

Pour démonter un périphérique, il faut utiliser la commande *umount*.

2.2. Configuration réseau et Linux

Il s'agit de paramétrer un système LINUX pour qu'il soit connecté à un réseau local, et puisse éventuellement accéder au réseau Internet via un accès distant par routeur ou par modem. Un certain nombre d'éléments matériels et logiciels (daemons lancés,...) doivent être installés, configurés et activés au démarrage de la machine.

2.2.1. Informations indispensables

- ❖ **Adresse IP** de l'adaptateur (on dit aussi interface) réseau : c'est une machine de type passerelle ayant 2 cartes. Par exemple : 192.168.1.5 se composant de l'adresse du réseau (supposé de classe C) **192.168.1.0** et du numéro de la machine, ici 5 (de 1 à 254)
- ❖ **Adresse de "boucle"** : une machine isolée a toujours l'adresse **127.0.0.1**, ce qui lui permet de se connecter à elle-même
- ❖ **Masque de sous-réseau (netmask)** : ce qui détermine les adresses qu'il est possible d'attribuer aux machines de ce sous-réseau, ici **255.255.255.0**
- ❖ **Adresse IP générale du sous-réseau** : elle se déduit du masque et d'une adresse d'une machine quelconque du réseau; ici 192.168.1.0.
- ❖ **Nom complet** : nom station + nom du domaine

2.2.2. Configuration initiale lors de l'installation

Prenons l'exemple d'une machine Linux, installé avec la distribution Mandriva

La configuration réseau comprend 2 parties :

- 1) Détection de l'adaptateur réseau Ethernet et intégration du pilote dans le noyau.
- 2) Paramétrage réseau **TCP/IP**: il peut s'effectuer à l'aide de fenêtres de dialogue successives à renseigner.

Par exemple :

- ❖ adresse ip : 192.168.2.100 à 192.168.2.119
- ❖ masque de sous-réseau : 255.255.255.0
- ❖ passerelle par défaut : 192.168.2.245
- ❖ serveur de nom primaire : 195.98.246.50
- ❖ nom de domaine : iutfv.univ-dschang.org
- ❖ nom de machine : talla.iutfv.univ-dschang.org
- ❖ 2ème serveur de nom : 195.98.246.10

2.2.3. Interface Ethernet

Pour vérifier que l'adaptateur réseau est bien lié à la couche réseau du noyau Linux et activé, passer la commande **ifconfig**.

- ❖ `$ ifconfig nom-interface` renseigne sur l'interface, son paramétrage et son activité
- ❖ `$ ifconfig lo` renseigne sur l'interface loopback (bouclage local)
- ❖ `$ ifconfig eth0` renseigne sur la première interface Ethernet
- ❖ `$ ifconfig nom-interface adresse-IP` assigne cette adresse à l'interface et l'active

2.2.4. Fichiers de configuration

/etc/HOSTNAME nom de la machine dans le domaine (obtenu aussi par la commande `$ hostname`)

/etc/hosts table de correspondance des adresses IP des machines du sous-réseau et de leur nom d'hôtes

/etc/host.conf

order hosts, bind

multi on

fin du fichier.

indique que chaque machine cherche d'abord l'adresse de toute machine dans le fichier **/etc/hosts** et si la correspondance n'est pas trouvée, chercher par requête DNS.

/etc/sysconfig/network-scripts/ifcfg-eth0 fichier d'activation au démarrage de l'interface eth0

DEVICE = eth0

IPADDR = 192.168.2.101

NETMASK = 255.255.255.0

NETWORK = 192.168.2.0

BROADCAST = 192.168..2.255

ONBOOT = yes

Plus généralement, les interfaces à activer automatiquement au boot sont définies dans les fichiers du répertoire **/etc/sysconfig/network-scripts/**

/etc/sysconfig/network

NETWORKING = yes

FORWARD_IPV4 = no(empêche le transfert automatique des paquets ?)

HOSTNAME= pc01

DOMAINNAME = misfu.com

GATEWAY= 192.168.2.245

#passerelle par défaut, par où chercher si l'adresse IP n'est pas dans le sous-réseau

GATEWAYDEV = eth0

/etc/resolv.conf

#nom de domaine local de l'ordinateur

domain misfu.com

#adresse du serveur primaire DNS

nameserver 195.98.246.50

#liste de domaines à essayer, si le nom d'hôte ne précise pas son domaine. On peut mettre jusqu'à 3 serveurs de noms

domainsearch wanadoo.fr

/etc/networks

décrit les noms et adresses IP des différents sous-réseaux routables à partir de l'hôte

/etc/conf.modules

#modules installés

alias eth0 wd

options wd io=0x240

2.3. Installer un chargeur ou booter Linux

LILLO (Linux LOader) est le petit utilitaire qui permet de faire démarrer Linux. D'où son importance.

C'est, en fait, deux programmes :

- ❖ un chargeur de linux
- ❖ le programme qui installe le chargeur sur le disque.

Désinstaller LILO

- ❖ Depuis Linux : taper *lilo -u*.
- ❖ Depuis DOS : tapez *fdisk /mbr*.

GRUB : Le GRUB est le bootloader du projet GNU.

LOADLIN : C'est un utilitaire fonctionnant sous DOS que l'on peut utiliser pour démarrer Linux, si l'on ne peut/souhaite pas utiliser LILO, et que l'on dispose d'une partition DOS/Windows bootable. Grâce à lui, on peut choisir le système que l'on souhaite utiliser : Linux, FreeBSD, Windows, OS/2,..., lors du lancement de la machine. Pour cela, lilo est placé dans le secteur de boot du disque dur. Il est disponible en général sur les CD des distributions. Pour le faire fonctionner, il est conseillé de créer un répertoire **C:LINUX**, d'y placer LOADLIN, de copier le noyau **/boot/vmlinuz-2.x.x** sous Linux dans ce répertoire sous le nom **VMLINUZ**, et d'y créer un fichier **linux.bat** contenant la ligne de commande :
LOADLIN C:LINUXVMLINUZ root=/dev/hdb1 ro %1 %2

-> Il faut ensuite remplacer */dev/hdb1* par la partition "racine" / ou encore, si on utilise initrd, il faudra copier */boot/initrd* dans le même répertoire, et placez cette ligne dans *linux.bat* : *LOADLIN C:LINUXVMLINUZ initrd=C:LINUXINITRD %1 %2*.

Dès lors, on peut créer une entrée dans le *c:config.sys* pour avoir un choix "Linux" au boot qui en fait lance *linux.bat*. Ou bien on boote en "ligne de commande seulement" et on démarre Linux en tapant *linux* au prompt *C:>*.

En cas d'erreur de manipulation, vous risquez de rendre votre configuration inutilisable. Pensez à avoir toujours une disquette de démarrage sous la main au cas où un problème surviendrait. Vous pouvez également faire une copie de votre secteur de démarrage sur une disquette avec la commande:

```
dd if=/dev/hda of=/fd/MBR/ bs=512 count=1
```

Pour modifier la configuration de lilo, il suffit de modifier son fichier de configuration, lequel est généralement */etc/lilo.conf*.

Problèmes avec le CD-ROM : Essayer de copier le contenu du CD sur le disque dur et de lancer l'installation depuis le disque dur, cela règle bien des problèmes si le lecteur de CD est mal reconnu.

2.4. Gestion des packages et applications

Le format RPM (RedHat Package Manager), a été, créé à l'origine pour la distribution RedHat. Depuis, de nombreuses distributions l'ont adopté et l'utilisent. D'autres formats de packages dans le monde Linux sont DEB (pour Debian) et TGZ (pour Slackware).

Il existe de nombreux utilitaires tels que Kpackage, GnoRPM, Midnight Commander etc. qui permettent d'installer et de gérer les packages RPM. L'installation depuis la ligne de commande reste toutefois la solution la plus rapide et la plus efficace pour ce travail.

2.4.1. Les commandes usuelles

Pour installer un nouveau paquetage :

```
[talla@misfu talla]$ rpm -ivh mon_package.rpm
```

Les options **h** et **v** ne sont pas obligatoires, **-i** est suffisant pour faire une installation. L'option **h** provoque l'affichage d'une barre de progression et **v** rend l'installation verbeuse, ce qui fournit des messages plus explicites en cas d'erreur.

Pour procéder à la mise à jour d'un paquetage :

```
[talla@iutfv.univ-dschang.org]$ rpm -Uvh mon_package.rpm
```

Le "U" correspond à "Update".

Pour supprimer un paquetage :

```
[talla@iutfv.univ-dschang.org]# rpm -e mon_package [talla@iutfv.univ-dschang.org]#  
rpm -e mon_package-version
```

Le "e" correspond à "erase".

Pour faire des requêtes sur les paquetages :

```
[talla@iutfv.univ-dschang.org]# rpm -qa
```

affiche la liste de tous les paquetages déjà installés.

L'option « q » signifie « query » et l'option « a » signifie "all".

```
[talla@iutfv.univ-dschang.org]# rpm -qa | grep mon_package
```

Ainsi par exemple `rpm -qa | grep 'util'` donne la liste de tous les RPM installés tels que `util-linux`, `nautilus`, `bind_utils` etc.

Trouver la liste des fichiers contenus dans un paquetage `mon_package` qui est déjà installé.

```
[talla@iutfv.univ-dschang.org]# rpm -ql mon_package
```

L'option "l" signifie "list".

Obtenir des informations relatives à un paquetage

```
[talla@iutfv.univ-dschang.org]# rpm -qi mon_package
```

L'option "i" signifie "information".

Cette commande donne les informations relatives à ce paquetage, une brève description de ce que fait le paquetage, des dépendances qui doivent être satisfaites, etc.

```
[talla@iutfv.univ-dschang.org]# rpm -qpi mon_package
```

L'option "p" signifie "package".

Retrouver le paquetage d'origine d'un fichier

```
[talla@iutfv.univ-dschang.org]$ rpm -qf chemin_absolu/mon_fichier
```

Ici le "f" signifie "file".

[talla@iutfv.univ-dschang.org]\$ rpm -i --test mon_package effectue simplement une vérification sans installer le paquetage mon_package. Permet essentiellement de vérifier si l'installation pourrait se dérouler sans encombre ou alors si des dépendances non satisfaites risquent de faire avorter l'installation.

Si rpm ne veut pas désinstaller le paquetage mon_package, faire :

[talla@iutfv.univ-dschang.org]\$ rpm -qa | grep mon_package affiche la liste des paquetages dont le nom contient mon_package, par exemple :

mon_package-1.1 mon_package-1.2

[talla@iutfv.univ-dschang.org]# rpm -e mon_package-1.1

Désinstalle le paquetage désigné par son nom et son numéro de version.

2.5. Outils communs d'utilisation d'un réseau sous Unix.

Sous Linux, tout est fichier. Que ce soit un répertoire, une imprimante, un disque ou une unité périphérique, le système le perçoit comme un fichier. Il existe deux modes de lecture ou d'ouverture de fichier sous Linux : le **mode lecture seul** et le **mode édition**.

L'ouverture du fichier en mode lecture seule peut se faire grâce à la commande cat qui offre soit la possibilité d'afficher le contenu du fichier à l'écran, soit de le rediriger vers un fichier.

L'ouverture du fichier en mode édition peut se faire avec des outils tels que **gedit** ou **vi**. Le contenu est alors affiché à l'écran et l'utilisateur peut procéder aux mises-à-jours.

La synthèse de tous ces outils et commandes est la même : Nom de la commande suivi du chemin d'accès au fichier.

2.5.1. Quelques fichiers importants

Certains fichiers sont fréquemment utilisés pour l'administration des systèmes Linux. Parmi ceux-ci, les plus en vue sont les suivants :

2.5.1.1. Fichier de noms d'hôtes : /etc/hosts

Ce fichier est structuré sous la forme :

adresse IP	nom de machine	liste d'alias <i>commentaire</i>
127.0.0.1	localhost.localdomain	localhost
192.168.11.5	directeur.iutfv-bandjoun	Directeur
192.168.11.1	dptinfo.iutfv-bandjoun	ChefDepartement
192.168.11.2	salletp.iutfv-bandjoun	SalleTP

Il assure la correspondance (nom, adresse IP). S'il existe un serveur de noms sur cette machine, il contient très peu de lignes et ne sert qu'au démarrage de la machine avant que le serveur de noms ne soit lancé. Si le serveur de noms est sur une machine distante il ne contient que les machines du réseau local.

2.5.1.2. Résolveurs de nom : /etc/resolv.conf

```
search iutfv-bandjoun info.univ-dschang.org univ-dschang.org
nameserver 172.20.41.2
```

Il précise les noms de domaines avec lesquels compléter le nom d'une machine pour connaître son nom complet, ainsi que les adresses de serveurs de noms à interroger. On peut aussi y trouver le domaine d'appartenance de la machine sous la forme domain iutfv-bandjoun par exemple.

2.5.1.3. Liste des protocoles : /etc/protocols

Il contient la liste des protocoles connus et utilisés dans Internet sous la forme :

```
nom du protocole      numéro du protocole      liste d'alias commentaire
#/etc/protocols: ip 0 IP    #internet protocol,      pseudo protocol number
```

2.5.1.4. Liste des services : /etc/services

Il contient la liste des services Internet connus sous la forme :

nom du service	numéro de port/protocole	liste d'alias	commentaire
• ftp-data	20/tcp		
• fsp	21/udp	fspd	
• ssh	22/tcp		# SSH Remote Login Protocol

2.5.1.5. Fichier des liens : /etc/inetd.conf

Il contient les liens entre nom de services et « exécutables » réalisant ce service.

• ftp	stream	tcp	nowait	root	/usr/sbin/tcpd	in.proftpd
-------	--------	-----	--------	------	----------------	------------

2.6. Quelques commandes utiles

Les commandes sont illustrées avec leurs résultats après les avoir lancées. Il est conseillé d'utiliser la commande man du système pour obtenir de plus amples informations sur celles-ci.

hostname retourne le nom de la machine.

finger renvoie des informations sur un utilisateur, en utilisant s'ils existent les fichiers .plan et .project.

ping permet de tester l'accessibilité d'une machine. Elle envoie une requête ICMP (echo) à destination d'une machine cible, spécifiée par son nom ou son adresse IP, qui lui retourne une réponse ICMP (echo). Si une machine ne répond pas au ping, elle est inutilisable pour toute autre application.

- ❖ ping talla
- ❖ PING talla.iutfv-bandjoun (192.168.1.7): 56 data bytes
- ❖ 64 bytes from 172.20.41.7: icmp_seq=0 ttl=64 time=0.2 ms

Ping renvoie également le numéro de la séquence ICMP exécutée, la valeur du champ TTL et le temps d'aller-retour entre les deux machines. Elle peut le faire car dans le paquet expédié pour la requête elle place son heure d'émission et lorsqu'elle reçoit la réponse elle la soustrait de l'heure système. Puis, en fin d'exécution (provoquée par l'utilisateur ou suite à l'envoi du nombre de paquets spécifiés), quelques données statistiques sont affichées permettant d'évaluer la qualité de la liaison.

traceroute renvoie la route prise par des paquets pour atteindre une destination. Elle utilise le champ TTL (Time to Live) des paquets IP transmis selon le protocole UDP.

```
❖ |dptinfo|/etc>/usr/sbin/traceroute www.iutfv.univ-dschang.org
❖ traceroute to www.iutfv.univ-dschang.org (195.46.203.182), 30 hops max, 38 byte packets
❖ 1 fokam (172.20.41.1) 0.185 ms 0.133 ms 0.121 ms
```

arp permet de visualiser et modifier (si on a le droit) la table de translation adresses Ethernet/adresses Internet. Cette table est en fait un cache qui évolue au fur et à mesure des sollicitations des machines du réseau.

```
❖ |dptinfo|/etc>/sbin/arp -a
❖ fokam.iutfv-bandjoun (172.20.41.1) at 00:A0:C9:EE:EA:6A [ether] on eth0
❖ g104-6.iutfv-bandjoun (172.20.41.146) at 00:02:B3:21:05:BA [ether] on eth0
```

nslookup permet d'interroger les serveurs de noms d'Internet. Elle fonctionne en mode interactif quand on l'appelle sans argument ou avec les arguments - nom ou adresse IP de machine, à ce moment-là on obtient :

```
❖ |dptinfo|/etc>nslookup
❖ Default Server: nes.iutfv-bandjoun
❖ Address: 172.20.41.2>
```

où l'on a le nom du serveur de noms par défaut. Dans le mode interactif on obtient de l'aide sur les différentes commandes à l'aide de la commande ?

```
> ?
$Id: nslookup.help,v 8.5 2000/03/30 23:25:35 vixie Exp $
Commands:  (identifiers are shown in uppercase, [] means optional)
  domain=NAME - set default domain name to NAME
  srchlist=N1[/N2/.../N6] - set domain to N1 and search list to N1,N2, etc.
```

whois permet d'interroger une base de données sur les réseaux et leurs administrateurs. Par défaut un serveur global est interrogé, mais on peut spécifier quel serveur whois on interroge, comme par exemple dans la commande suivante : **whois univ-dschang.org@whois.nic.fr**

On peut aussi utiliser **whois nom_de_personne@nom_de_serveur**.

netstat permet d'obtenir des statistiques sur le nombre de paquets, les erreurs, les collisions etc... sur une interface en donnant la liste de toutes les sockets ouvertes.

On obtient la table de routage par l'option : **netstat -rn**

Kernel IP routing table						
Destination	Gateway	Genmask	Flags	MSS Window	irrt	Iface
172.20.41.8	0.0.0.0	255.255.255.255	UH	40	0	0 eth0

127.0.0.0	0.0.0.0	255.0.0.0	U	40	0	0	lo
0.0.0.0	172.20.41.1	0.0.0.0	UG	40	0	0	eth0

U route utilisable, G route indirecte (passerelle), H hôte

tcpdump visualise différentes informations (selon les options de la commande) sur les paquets qui passent par l'interface réseau de la machine. C'est une commande réservée à root.

- |dptinfo|~>tcpdump -c 20
- tcpdump: listening on eth0

tcpdump tcp port 21 ne renverra que les paquets liés au protocole tcp sur le port 21, à savoir ftp.

2.7. Prise en main du shell

2.7.1. Commandes usuelles d'Unix

Le tableau suivant présente les principales commandes reconnues par l'interpréteur de commande Linux (Shell).

Tableau 2. Commandes usuelles

Commande	Action
who -s	Utilisateurs connectés et leur station
man	Obtenir de l'aide sur une commande Unix
man -k	Chercher de l'aide sur un mot clé
exit	Se délogger
pwd	Obtenir le chemin actif
ps -A	Obtenir la liste des jobs actifs
pg ou more	Paginer une commande (arrêt fin écran)
kill -9	Arrêter un job
ls ("LS" minuscule)	Liste des fichiers du répertoire courant
shutdown	Arrêt serveur
su nom_utilisateur	Changer de profil utilisateur
cp fichier_source fichier_destination	Copier un fichier
LS -L (en minuscules, ça suffit)	Commande équivalent au DIR du DOS
CAT nom_fich	Commande équivalente au TYPE du DOS. Utiliser MORE pour paginer
> nom_fichier	Redirection des sorties vers un fichier

2.7.1.1. Autres commandes utiles

hostname	Nom de la station sur laquelle on est connecté
uname	Nom de l'utilisateur courant
date	renvoie la date

cal	calendrier
logname	nom de login courant
id	renvoie les UID et GID de l'utilisateur courant
more	permet de paginer une sortie écran
clear	efface l'écran
echo	visualise une chaine ou variable
man	aide d'UNIX
env	visualise les variables d'environnement
whereis	localise une commande sur les disques
apropos	recherche des rubriques connexes de l'aide
grep	recherche une chaine
tty	donne le nom du périphérique de sortie standard
mkdir	créer un répertoire
mv	déplace un fichier/répertoire
cp	copie un fichier/répertoire
touch	crée un fichier vide ou met à jour les flags d'un fichier
df	donne l'utilisation des disques
cd ~ ou cd	replaces dans le \$HOME de l'utilisateur courant
whoami	donne le nom de login de l'utilisateur courant
mkdir -p	crée une arborescence
rm -r	supprime une arborescence
ls -R	équivalent au tree
su - loginname	cette commande permet de changer de login et en plus de lire l'environnement complet de l'utilisateur
cat fich1 fich2 >fich3	concaténation des fichiers fich1 et fich2 et envoi du résultat dans fich3
tail -f	visualise la fin d'un fichier

2.7.1.2. Commandes de tris et d'actions d'extraction

sort	Tri standard
uniq	élimine les lignes identiques consécutives
head	extraie le début d'un fichier
tail	extraie la fin d'un fichier (tail +2 pour sauter une entête par exemple)
split	découpage en paquets
cut	extraction d'une partie d'un fichier, colonnes, caractères, etc...
grep	recherche une chaine et extraie les lignes correspondantes
paste	colle ligne à ligne deux fichiers

join	réalise une jointure entre deux fichiers de type base de données
wc	compte le nombre de lignes ou de mots

2.7.1.3. Commandes d'entrée/sortie basiques

read var	attend une saisie au clavier et positionne var
echo	affiche à l'écran
echo -e	affiche avec codes de contrôle étendus
tput	locate + inversion vidéo

2.7.2. Métacaractères du shell

Le tableau suivant présente les caractères étendus compris par le Shell

Tableau 3. Caractères étendus compris par le Shell

Caractère	Action
' '	permet de spécifier une chaîne dont le contenu ne sera PAS interprété par le shell
" "	permet de spécifier une chaîne dont le contenu sera interprété par le shell (peut contenir des variables \$ qui seront interprétées)
*	tous les fichiers
?	remplace un caractère dans le nom d'un fichier
[]	spécifie une liste
;	permet d'enchaîner des commandes (attention, différent du pipe)
&	lancer un processus en arrière-plan
#	commentaire
` `	récupère le résultat d'une commande
\$	= contenu de

2.7.3. Création de liens

La notion de lien est tout aussi importante dans Linux. Ils permettent à plusieurs utilisateurs de travailler sur un même fichier. On distingue deux sortes de liens, les liens physiques et les liens logiques. Pour le lien logique, le fichier est physiquement unique, mais le lien crée une vue logique dessus, une sorte de duplication du nom du fichier tel que les deux noms indexent le même fichier physique. Le lien physique quant à lui duplique physiquement le fichier.

ln fichier permet créer un lien physique.

ln -s répertoire permet de créer un lien logique.

2.8. Système de fichiers

2.8.1. Structure du système de fichiers

2.8.1.1. Organisation

Comme tous les systèmes Unix, Linux organise ses fichiers selon un modèle en arborescence. Ceci est devenu pratiquement une règle pour la plupart des systèmes d'exploitation modernes. Mais par rapport au DOS qui est issu d'Unix, il y a des différences notables qu'il faut toujours avoir à l'esprit :

Sous DOS, chaque lecteur (disquette, CD-Rom, disque, partage, USB,...) dispose d'une arborescence indépendante qui se déploie sous une racine identifiée par une lettre (A:, B:, C:, etc.). A l'inverse, sous Linux, tous les lecteurs sont intégrés sous une arborescence unique dont la racine est désignée par le caractère /. Cette organisation bénéficie d'une souplesse et d'une efficacité qui feraient pâlir le système DOS.

Contrairement à DOS/Windows, Linux respecte les casses des caractères dans les noms de fichiers. Il est fréquent d'avoir des noms de fichiers avec plusieurs points comme par exemple `bash.tar.gz`.

Dans le nom d'un fichier on peut choisir n'importe quel caractère, à l'exception de la barre oblique (on dit aussi slash) qui sert de séparateurs entre les répertoires. Il est possible d'avoir des noms de fichier relativement longs.

Le système de fichiers par Défaut ext2FS peut contenir des fichiers d'une taille maximale de 2 Go.

2.8.1.2. Quelques règles et conventions de nommage

Les fichiers dont le nom commence par le caractère point sont des fichiers cachés qui ne sont pas affichés par défaut. Il est conseillé :

- ❖ d'utiliser les minuscules pour le nommage des fichiers. Toutefois, la tradition fait choisir une lettre majuscule pour la première lettre des répertoires personnels.
- ❖ d'éviter les caractères spéciaux et les blancs dans les noms de fichiers.
- ❖ de ne pas dépasser plus de 1024 caractères pour un nom de fichiers.

2.8.2. La gestion des droits d'accès aux fichiers

Les droits (ou autorisations) d'accès constituent un moyen pour protéger les données (fichiers UNIX) appartenant à chacun contre une destruction ou altération accidentelle (ou intentionnelle) par soi-même ou les autres utilisateurs.

Principe : Chaque fichier ou répertoire peut avoir trois types d'utilisateurs et quatre types d'autorisation ou modes (ou encore droits) d'accès.

2.8.2.1. Les droits (ou autorisations ou modes) d'accès

Ils définissent les types d'actions que l'on peut effectuer sur le fichier. On en distingue 4 conformément au tableau suivant :

Tableau 4. Spécification des droits d'utilisateurs

Code d'autorisation.	Signification
r (readable)	Accès en lecture. Permet de lire le contenu ou de copier le fichier Les éléments du répertoire sont accessibles en lecture, <i>ls</i> est possible.
w (writable)	Accès en écriture. Modification du contenu du fichier possible Création et suppression de fichiers dans le répertoire possible (mais indépendant des autorisations. d'accès aux fichiers).
x (executable)	Accès en exécution. Le fichier contient un programme (binaire ou script) et peut être exécuté. Le nom du répertoire peut apparaître dans un chemin d'accès. Cd possible.
- (pas d'accès)	Mode protégé. Aucun accès n'est possible

2.8.2.2. Les types d'utilisateurs

Pour chaque fichier ou répertoire, les autorisations d'accès en lecture, en écriture ou en exécution peuvent être positionnées séparément pour trois types d'utilisateurs : le propriétaire, le groupe et les autres.

Le tableau suivant présente une spécification

Tableau 5. Spécification des codes d'utilisation des fichiers

Code d'utilisation.	Signification
u (user)	Propriétaire. Créateur du fichier ou du répertoire. Les trois premiers caractères de la liste d'autorisations.
g (groupe)	Chaque utilisateur est membre d'un groupe défini par l'administrateur du système ; les membres d'un groupe partagent l'accès aux fichiers des uns et des autres. 4 ^e , 5 ^e et 6 ^e caractères de la liste d'autorisation.
o (Others)	Tous les autres utilisateurs autorisés à se servir du système. 7 ^e , 8 ^e et 9 ^e caractères.

2.8.2.3. Affichage des autorisations d'accès des fichiers

On utilise la commande *ls* (de l'anglais **list**) avec l'option *-l* combinable avec l'option *-a* Types de fichiers et répertoires (le premier caractère de la liste élargie d'autorisation)

Tableau 6. Options de la commande *ls*

Code	Signification
-	Le type le plus courant, type standard
d	Répertoire (de l'anglais directory)
b	Fichier de périphérique bloc (programme de gestion de périphérique)
c	Fichier de périphérique caractère.
s	Fichier "socket" utilisé pour la communication entre des programmes fonctionnant simultanément
l	Indique que le fichier est un lien, utilisé pour relier logiquement deux fichiers.

Chapitre 3. Gestion des droits et Initiation au Shell

3.1. Rôle de l'administrateur réseau sous Unix

Unix étant un système d'exploitation Multi-Utilisateurs, la gestion du système et des utilisateurs est confiée à un super utilisateur nommé root ou racine.

Le rôle de l'administrateur ou root est de :

1. configurer le noyau du système d'exploitation ;
2. sauvegarder les données et réparer les systèmes de fichier ;
3. gérer les utilisateurs ;
4. installer de nouveaux logiciels ;
5. intégrer de nouveaux disques durs et de nouvelles partitions ;
6. configurer le processus de démarrage de Linux ;
7. configurer le réseau

3.2. Gestion des utilisateurs

3.2.1. Notion d'utilisateur

Même si vous êtes le seul utilisateur, il est indispensable de créer des utilisateurs ne serait-ce que pour des raisons de sécurité. L'utilisateur root ayant tous les droits, il est recommandé de se connecter avec un autre compte utilisateur afin d'éviter d'effectuer de fausses manipulations qui pourraient avoir de lourdes conséquences sur la stabilité du système. Lorsque vous avez besoin de faire de l'administration système, vous avez toujours la possibilité de changer d'utilisateur et de devenir root à partir de la commande `su : $ su root ;`

Chaque utilisateur dispose d'un répertoire personnel sous `/home`, par exemple `/home/talla` pour l'utilisateur *talla*. Outre les fichiers personnels de l'utilisateur, son compte comprend des fichiers cachés de configuration du shell ainsi que les préférences de l'interface graphique X-Window.

3.2.2. Le système d'authentification et ses composants

3.2.2.1. Session utilisateur

Le système d'authentification comprend, pour chaque utilisateur déclaré dans le système, deux informations capitales qui sont demandées à l'utilisateur, au début de la session : son nom d'utilisateur et son mot de passe. C'est grâce à ces informations que le système arrive à conserver, pour chaque utilisateur, ses préférences d'une session sur l'autre ainsi que ses droits d'accès et lui permet de retrouver son fond d'écran, ses icônes, la configuration de ses programmes, ... ainsi que ses données privées. C'est donc un système de reconnaissance de l'utilisateur.

3.2.2.2. Le nom d'utilisateur

On distingue deux grand types de comptes utilisateur sur un système linux : les comptes administratifs (sys, mail, etc...) qui ne sont utilisés que par le système pour des tâches spécifiques et les comptes d'utilisateurs réels utilisés par les linuxiens.

3.2.2.3. Le mot de passe

Le mot de passe permet à l'utilisateur d'apporter au système la preuve de son identité. On dit que l'authentification se fait au moyen d'un mot de passe. Et si quelqu'un connaît le mot de passe d'un utilisateur, il peut accéder à toutes ses données, ou usurper son identité pour déclencher des actions qui pourraient s'avérer nuisibles. C'est pourquoi il faut prendre les précautions suivantes concernant son mot de passe :

- ❖ Ne pas le communiquer à une autre personne quelle qu'elle soit ;
- ❖ Ne pas l'écrire ou le noter de quelque manière que ce soit ;
- ❖ Ne pas choisir un mot de passe trop simple ou que les autres peuvent deviner sans difficultés.

Un bon mot de passe doit :

- ❖ comporter un mélange de lettres, de chiffres, de ponctuations, etc...
- ❖ être suffisamment long (8 caractères) ;
- ❖ être dénué de sens (par exemple, ne pas choisir une date d'anniversaire, ou un mot du dictionnaire, son nom d'utilisateur, etc.)

3.2.2.4. Gestion des comptes et groupes d'utilisateurs sur Linux

Les comptes utilisateur servent à beaucoup de choses sur les systèmes UNIX et Linux :

- ❖ Ils permettent de distinguer les différents utilisateurs qui ont accès au système, pour des raisons de sécurité et d'organisation.
- ❖ Ces utilisateurs peuvent définir des permissions d'accès à leurs données, afin d'en autoriser ou d'en interdire l'exploitation par les autres.
- ❖ Les comptes permettent l'authentification de chaque utilisateur accédant au système, ce qui permet bien d'autres actes de gestion tels que gérer les courriers électroniques des utilisateurs par exemple ou savoir qui a fait quoi, en examinant les fichiers de trace du système.

3.2.2.5. Gestion des comptes d'utilisateurs

Création d'un compte d'utilisateur

La création d'un compte demande plusieurs opérations : l'ajout d'une entrée dans le fichier */etc/passwd*, la création du répertoire personnel de l'utilisateur, et la mise en place de la configuration par défaut dans le répertoire de cet utilisateur.

L'ensemble de ces opérations est pris en charge par la commande **adduser**. (ou **useradd** selon la distribution) dont la syntaxe se présente comme suit :

adduser [-c commentaires] [-d rep_personnel] [-e date_expiration] [-f tps_inactivité] [-g groupe_initial] [-G groupe] [...] [-m [-k squelette_rep | -M] [-p motdepasse] [-s shell] [-u uid [-o]] [-n] [-r] utilisateur

Utilisée sans aucun argument, cette commande demande de manière interactive toutes les informations nécessaires pour identifier l'utilisateur en cours de création. Utilisée sans option (donc uniquement avec l'argument nom d'utilisateur), cette commande applique un ensemble de valeurs par défaut qui sont consignées dans le fichier : */etc/default/useradd*

Voici un exemple de ce fichier, auquel de brefs commentaires sont rajoutés.

GROUP=100 identifiant du groupe primaire

HOME=/home racine des rép. Personnels

INACTIVE=-1 (nb de jours avant destruction du compte)

EXPIRE= nb de jours avant expiration du mot de passe

SHELL=/bin/bash shell de connexion attribué au compte

SKEL=/etc/skel fichiers recopiés par défaut dans chaque répertoire personnel

Tableau 7. Options les plus utilisées

Option	Résumé
-c commentaires	Nom complet de l'utilisateur et des commentaires divers
-d rep_personnel	Par défaut dans le répertoire /home
-e date_expiration	Fixe la date d'expiration du compte (format MM/JJ/AA)
-f tps_inactivité	Nombre de jours entre l'expiration et la désactivation permanente du compte
-g groupe_initial	Groupe d'affectation du compte. Doit exister avant la création du compte.
-G liste	Fixe l'appartenance de l'utilisateur à une liste de groupes secondaires (séparateur, sans espace)
-m	Le répertoire personnel sera créé s'il n'existe pas. L'option -k n'est valide qu'en conjonction avec l'option -m
-k squelette_rep	Recopie le contenu du répertoire squelette_rep dans le rép. Personnel ; par défaut /etc/skel
-p motdepasse	Mot de passe crypté. Par défaut le compte est désactivé
-s shell	Par défaut, attribution du shell par défaut bash
-u uid	Pour fixer l'identifiant uid à l'utilisateur
-n	Un groupe ayant le même nom que le compte en cours de création sera créé sur le système par défaut. Cette option désactivera ce comportement spécifique à Linux Mandriva

Suppression d'un compte d'utilisateur :

Pour supprimer le compte d'un utilisateur (non connecté), on utilise la commande **userdel**

userdel [-r] utilisateur

L'option **-r** supprime aussi le répertoire personnel et les fichiers de l'utilisateur.

La commande supprime toute trace de l'utilisateur dans le fichier de configuration : */etc/passwd* y compris dans les groupes d'utilisateurs.

Modification d'un compte d'utilisateur :

Pour modifier le compte d'un utilisateur, on utilise la commande **usermod** comme suit :

usermod [options] utilisateur

Les options sont les mêmes que **adduser**

usermod -G stagiaire, prof stagex ajoute l'utilisateur stagex dans les 2 groupes stagiaire et prof (qui doivent exister au préalable)

Pour lui attribuer le nouveau mot de passe stgx : passwd stagex. La saisie du nouveau mot de passe sera demandée deux fois.

Remarques :

- ❖ Si root souhaite redéfinir son propre mot de passe, il doit passer la commande **passwd** ;
- ❖ Un utilisateur quelconque ne peut pas créer de compte, même s'il a le privilège de faire partie du groupe root. ! Par contre, il peut modifier lui-même son mot de passe.
- ❖ Pour pouvoir se connecter au réseau SAMBA, à partir d'une station distante Windows, il faut créer un compte Samba avec l'utilitaire smbpasswd

Gestion des groupes d'utilisateurs

Un groupe sous Linux est un ensemble d'utilisateurs qui partagent les mêmes fichiers et répertoires. Nous verrons que les fichiers accordent des droits d'accès réglables à ces groupes.

Chaque utilisateur doit faire partie au moins d'un groupe, son *groupe initial ou primaire*. Celui-ci est défini au moment de la création du compte, et *par défaut sous Linux Ubuntu*, l'utilisateur appartient à un nouveau groupe créé, portant son nom.

Ainsi, dans */etc/passwd* chaque utilisateur possède un groupe par défaut, précisé par son identifiant *gid* dans ce fichier.

Pour lister tous les groupes (primaire et secondaires) d'un utilisateur : **groups utilisateur**. Pour créer un nouveau groupe : **groupadd groupe**

Supprimer le groupe .stagiaires : **groupdel stagiaires** (Le groupe est alors supprimé du fichier */etc/group*).

Pour ajouter un utilisateur à un groupe, le plus simple est d'éditer le fichier */etc/group* et d'ajouter une liste d'utilisateurs (séparés par des virgules) sur la ligne du groupe (ou utiliser **Linuxconf**).

Tableau 8. Récapitulatif des principales commandes

Commande	Action
useradd, usermod, userdel	gestion des comptes utilisateur
groupadd, groupmod, groupdel	gestion des groupes
passwd	changer le mot de passe d'un utilisateur
id, groups, finger	utilitaires divers

3.2.2.6. Outils graphiques de gestion des comptes et groupes

Les outils que nous présentons brièvement nécessitent tous une connexion au système en tant que root.

Webmin

Rubrique "Système" puis "Utilisateurs et groupes"

- ❖ Cliquer sur un utilisateur ou un groupe pour modifier leurs propriétés.
- ❖ Cliquer sur "Créer un nouvel utilisateur" pour ajouter un utilisateur.
- ❖ Cliquer sur "Créer un nouveau groupe" pour ajouter un groupe.

Linuxconf : En général il se lance en ligne de commande (on peut utiliser linuxconf sous X-KDE sur les systèmes Redhat Section Comptes utilisateurs)

Kuser : c'est une petite application graphique intégrée à l'interface graphique KDE. Pour le lancer, on peut utiliser la commande Kuser dans une fenêtre console.

Userdrake : Pour le lancer on peut utiliser la commande **userdrake** dans une fenêtre console.

Le *superutilisateur* ne se nomme pas obligatoirement **root**. Pour le changer, il suffit de remplacer le nom du compte root par celui désiré. Un compte privilégié est un compte dont l'identifiant (UID, User ID) vaut zéro.

L'administrateur configure un prompt identique pour tous les utilisateurs du système en modifiant la variable PS1 dans le fichier **/etc/profile**. S'il veut un prompt du genre **"Bonjour#"**, il lui faut d'éditer le fichier. Toutes les lignes concernant la variable PS1 doivent alors être précédées d'un dièse: #. Il suffit alors d'ajouter la ligne **PS1='Bonjour#'**. Il est également possible d'utiliser des variables dans le prompt (pour afficher par exemple l'heure ou le nom de la machine...) :

- ❖ **d** pour ajouter la date (format anglais)
- ❖ **t** pour ajouter l'heure (HH:MM:SS)
- ❖ **u** pour ajouter le nom de l'utilisateur
- ❖ **w** pour ajouter le chemin complet du répertoire courant
- ❖ **W** pour ajouter le répertoire courant
- ❖ **h** pour ajouter le nom de la machine

3.3. Initiation au Shell

3.3.1. Jokers

Commande	Action
?	remplace un seul caractère
*	remplace 0 ou plusieurs caractères
[]	remplace par n'importe quel caractère situé à l'intérieur des crochets
[-]	remplace par un caractère qui se situe dans la suite (0-9 de 0 à 9, a-z de a à z)
[!]	le contraire de []

3.3.2. Groupement de commandes

Opération	Action
commande1 ; commande2	Lance commande2 après commande 1
commande1 commande2	Renvoie le résultat de commande1 à commande2
commande &	Lance commande en arrière-plan
commande1 commande2	SINON : ne lance commande2 que si commande1 échoue
Commande1 && commande2	ET : Lance commande2 si commande1 réussi
(liste)	Groupe de commandes
{ liste; }	Groupe de commandes
`Commande` \${commande}	Substitution de commandes :

3.3.3. Les redirections

Entrée/Sortie standard (E/S)	
0	Entrée Standard (clavier)
1	Sortie Standard (écran)
2	Erreur Standard (écran)
Entrée / Sortie standard	
> fichier	Redirection en sortie vers un fichier, écrase le fichier si le fichier existe
< fichier	L'entrée standard est un fichier
>> fichier	Redirection en sortie vers un fichier, sans écrasement de fichiers
<< mot	Lecture au clavier jusqu'à "mot"
<> fichier	L'entrée et la sortie standard est un fichier

3.3.4. Editeurs de textes et utilitaires Unix

3.3.4.1. L'éditeur de texte Vi

Commande	Signification	Action
i	insert	insertion avant le curseur
I	insert	insertion au début de la ligne
a	append	insertion après le curseur
A	append	insertion à la fin de la ligne
o	open	ouvre une ligne blanche en dessous de la ligne courante
O	open	ouvre une ligne blanche au-dessus de la ligne courante
J	concatenate	concatène la ligne suivante à la fin de la ligne courante
<=	curseur	déplacement d'un caractère vers la gauche
=>	curseur	déplacement d'un caractère vers la droite
w	word	avance au début du mot suivant
F	forward	page suivante
B	backward	page précédente
^ ou 0		début de la ligne courante
\$		fin de la ligne courante
/toutou		recherche la chaîne "toutou" à partir de la position actuelle du curseur vers le bas du fichier
?text		recherche la chaîne "text" à partir de la position actuelle du curseur vers le haut du fichier
n	new	recherche la prochaine occurrence de la chaîne "toutou" <i>recherche vers le bas du fichier</i>
N	new	recherche l'occurrence précédente de la chaîne "toutou" <i>recherche vers le haut du fichier</i>
x		suppression du caractère sous le curseur
X		suppression du caractère précédent le curseur
dd	delete	suppression de la ligne courante
yy	yank	copie la ligne courante dans le buffer
p	paste	colle la ligne contenue dans le buffer après la ligne courante
P	paste	colle la ligne contenue dans le buffer avant la ligne courante
u	undo	annule la dernière commande et seulement la dernière
r	replace	remplace le caractère sous le curseur par un nouveau caractère
R	replace	remplace tous les caractères par de nouveaux caractères.
C	change	remplace la fin de la ligne par de nouveaux caractères

Par défaut, l'éditeur s'ouvre en mode commande. Pour activer le mode insertion, il faut appuyer sur la touche « i » au clavier. Pour abandonner le mode insertion et retourner au mode commande, appuyer sur la touche ESCAP.

3.3.4.2. Expressions régulières et mode commande globale

Les expressions régulières servent à manipuler le fichier texte dans son ensemble.

Tableau 9. Spécification des expressions régulières

.	représente un caractère quelconque
*	multiplicateur du caractère précédent
^	début de ligne
\$	fin de ligne ou fin de fichier
	permet de représenter les caractères « . », « * », « ^ » et « \$ » en le mettant devant. Exemple : * ou \$
1	permet la réécriture de l'expression régulière lors d'une substitution
s	commande de substitution
d	commande de destruction de lignes
:	passage en mode commande globale
g	parcours global du fichier ou de la ligne

3.3.4.3. Exemples de commandes avec expressions régulières

Tableau 10. Quelques commandes avec expressions régulières

:1,\$s/neant/bof/g	de la ligne 1 à la dernière ligne, substitution de la chaîne neant par la chaîne bof
:1,\$s/neant/bof/	idem mais seulement pour la première occurrence de la chaîne neant par ligne
:1,3s/^.*=/BRAVO/	de la ligne 1 à la ligne 3, substitution du début de la ligne (^) jusqu'au (.*) caractère = par la chaîne BRAVO
1,\$s/B.*O/1 Veinard/	rajoute aux mots commençant par B et terminant par O, le mot "Veinard"
:1,\$s/.\$//	suppression du dernier (\$) caractère quel qu'il soit (.)
:10,20d	suppression des lignes 10 à 20
:g/^#/d	suppression des lignes commençant par un #
:g/^M/s//v=""^M/g	suppression des ^M en milieu de ligne et substitution par un vrai retour à la ligne. Taper sur "Enter" pour obtenir le ^M.

3.3.4.4. Autres commandes utiles

Tableau 11. Quelques commandes diverses

Tableau 11. Quelques commandes diverses		
:set showmode		indique le mode (insertion ou commande) en bas de la fenêtre
:set number		affiche le numéro de chaque ligne
:155		positionne sur la ligne 155
:set filetype=unix		transforme le type DOS d'un fichier en type UNIX
:w	write	sauvegarde le fichier en cours d'édition
:q	quit	abandonne l'édition
:q!	quit	abandonne l'édition sans sauvegarder le fichier
:x	exit	sauvegarde le fichier en cours d'édition et abandonne l'édition (idem :wq)

Chapitre 4. Utilitaires

4.1. Outils multimédia sous Linux

4.1.1. Outils de musique

4.1.1.1. Lecteurs de CD audio et multimédia XMMS

Le lecteur de CD audio de KDE porte le nom de `kscd`. Il offre toutes les fonctionnalités d'un platine classique. Il fait partie des nombreuses applications intégrées dans KDE. Il est important d'avoir toujours à l'esprit que lorsqu'il est lancé, il prend le contrôle du lecteur et qu'il devient impossible d'éjecter le CD en utilisant les contrôles habituels, que ce soit le gestionnaire de fichiers ou le bouton de la façade du lecteur.

X MultiMedia System ou XMMS, c'est le nom d'un logiciel centralisé capable de jouer des fichiers sons de tous types et qui supporte aussi la diffusion de fichiers par Internet, via son option **Lire à un emplacement**, en tapant l'URL correspondante. Il est multifonction : permet de gérer une liste d'attente des fichiers à jouer, d'utiliser un equalizer, etc.

4.1.1.2. Outils d'enregistrement et de mixage

Sous Linux, il est possible de transcrire des parties d'un CD audio -en indiquant un numéro de plage ou une durée - sous forme de fichiers, ce qui permet ensuite de les traiter à loisir dans une table de mixage. Il existe de nombreux programmes sous Linux permettant de procéder à ce genre de transcription. Citons pour exemple **cdda2wav**. Afin de réduire l'espace ainsi occupé sur le disque, il est possible de convertir ces fichiers au format mp3, avec l'utilitaire **bladeenc**.

`kmix` est le nom de l'application intégrée de KDE, permettant de mixer les différentes entrées de son. On y accède via le lanceur d'applications, dans Multimédia/Console de mixage.

4.1.2. Outils de graphisme

4.1.2.1. `kview`

`kview` est une application intégrée de KDE qui permet de visualiser de nombreux formats de fichiers graphiques contenant des images statiques. Elle permet également de manipuler des images simples et de procéder à des conversions de format.

4.1.2.2. `Xv` et `kpaint`

`xv` est un petit outil d'affichage d'image. Il reconnaît la plupart des formats courants : gif, jpg, png, bmp, tiff, etc. Il permet aussi de procéder à des captures d'écrans. Son défaut est qu'il n'est pas libre.

A contrario `kpaint` est une application intégrée à KDE destinée à la manipulation d'images bitmap ; il permet aussi de procéder à de légères modifications d'images.

4.1.2.3. Aktion! et ksnapshot

Aktion! Est ce que l'on appelle un player vidéo (un visualiseur d'animation) de KDE utilisant l'application xanim. Il supporte un grand nombre de formats : AVI, Quick-Time, gif, FLI, etc. Il supporte pour la plupart des formats la gestion du son. Ksnapshot est quant à lui une application de capture d'écrans intégrée à KDE.

4.1.3. Le logiciel de retouche d'image : GIMP

Gimp est l'acronyme de GNU image Manipulation Program. Il s'agit d'un logiciel d'édition et de retouche d'images numériques issu du projet GNOME. Gimp est dans son ensemble un logiciel extrêmement complet et complexe. Il permet de faire du dessin mais aussi de la retouche photo ou du photo-montage. Il n'est pas prévu pour réaliser un dessin de A à Z ; pour cela, d'autres outils de dessin sont plus adaptés, comme kpaint ou killustrator pour le dessin vectoriel. En fait, il a pour but de reprendre en mieux toutes les fonctionnalités offertes par un logiciel comme Photoshop.

4.2. Impression

4.2.1. Partager une imprimante Linux pour Windows

Pour partager une imprimante Linux pour des stations Windows, il faut s'assurer que l'imprimante est configurée pour fonctionner sous Linux. La mise en place d'un partage de l'imprimante par SMB est quasi-automatique.

Ajouter la configuration d'impression à votre smb.conf comme dans l'exemple ci-dessous:

```
[global]
printing = bsd
printcap name = /etc/printcap
load printers = yes
log file = /var/log/samba-log.%m
lock directory = /var/lock/samba
[printers]
comment = Toutes les imprimantes
security = server
path = /var/spool/lpd/lp
browseable = no
printable = yes
public = yes
writable = no
create mode = 0700
```

```
[ljet]
security = server
path = /var/spool/lpd/lp
printer name = lp
writable = yes
public = yes
printable = yes
print command = lpr -r -h -P %p %s
```

Il faut s'assurer que le « path » corresponde au répertoire de spool défini dans **/etc/printcap**.

4.2.2. Partager une imprimante Windows pour Linux

Pour partager une imprimante sur une machine Windows, il faut suivre les points suivants:

1. Il faut avoir les entrées correspondantes à l'imprimante dans **/etc/printcap** et elles doivent correspondre à la structure locale des répertoires (pour le répertoire de **spool**, etc...).
2. Il faut avoir le script **/usr/bin/smbprint**. Il est fourni avec les sources de Samba, mais pas avec toutes les distributions binaires.
3. Si on veut convertir des fichiers ASCII en Postscript, il faut avoir **nenscript**, ou équivalent. **nenscript** est un convertisseur Postscript et se trouve généralement dans **/usr/bin**.

4.3. Archivage, compression

4.3.1. Contenu du dossier

La commande **find** permet de localiser facilement un fichier. Elle recherche les fichiers remplissant certaines conditions, en commençant par le répertoire indiqué ou le répertoire courant. Sa syntaxe est la suivante :

find Répertoire_de_départ [Critères] [Options de commande]

Les critères spécifient les restrictions de recherche, notamment par date, par taille,... Les options quant à elles spécifient les traitements à appliquer, par exemple que le résultat s'affiche à l'écran, qu'une commande s'applique aux fichiers trouvés,...

Exemple : \$ find -mtime -3 -print

Dans cet exemple, on cherche à afficher tous les fichiers dont les dernières modifications remontent à moins de 3 jours à partir du répertoire courant.

4.3.2. Compression/décompression de fichiers

Commandes **compress/uncompress** et **zcat** :

La commande **compress** permet de réduire la taille des fichiers spécifiés en utilisant le codage adaptatif de Lempel-Ziv. Si la réduction est possible, chaque fichier est remplacé par un nouveau fichier de même nom avec une extension **.Z**.

La commande **uncompress** restitue les fichiers compressés dans leur état original. Les fichiers résultats ont les noms, propriétaire et droits d'accès originaux et l'extension **.Z** est enlevée.

La commande **zcat** restitue les fichiers compressés dans leur état original et envoie le résultat sur la sortie standard.

Syntaxes :

`compress [-d] [-f|-z] [-z] [-v] [-c] [-V] [Fichier ...]`

`uncompress [-f] [-v] [-c] [-V] [Fichier ...]`

`zcat [-V] [Fichier ...]`

`compressdir [options][Répertoire ...]`

`uncompressdir [options][Répertoire ...]`

Exemples :

`$ compress -v Fessai`

Fessai : Compression : 23.55% -- replaced with Fessai.Z

`$ uncompress Fessai.Z`

`$ compress -f Fessai`

`$ compress -d Fessai.Z`

`$ uncompress -c arch.tar.Z | tar -tvf -`

4.3.3. Commandes **gzip/gunzip**

La fonction de base de la commande **gzip** consiste à prendre un fichier, le compresser, sauvegarder la version compactée sous la forme **.gz**, puis à supprimer l'original. Elle stocke le nom du fichier original dans la version compressée. Si aucun fichier n'est spécifié, **gzip** tentera de compresser les données arrivant sur l'entrée standard. De même l'option **-c** de **gunzip** permet d'écrire les données décompactées sur la sortie standard.

zcat est souvent identique à **gunzip -c**.

Syntaxe :

`gzip [-l] [Fichier]`

`gunzip [-c] [-N] [Fichier]`

4.3.4. Commandes **bzip2** et **bunzip2**

bzip2 compacte des fichiers en utilisant l'algorithme de réduction de texte par tri de blocs de Burrows-Wheeler, et le codage d'Huffman. Les options de ligne de commandes sont délibérément très similaires à celles de GNU **gzip**, mais elles ne sont pas identiques.

bzip2 attend une liste de noms de fichiers pour accompagner les options de ligne de commandes. Chaque fichier est remplacé par une version compactée de lui-même, avec le nom « nom_original.bz2 ».

bzip2 et **bunzip2**, n'écraseront par défaut pas les fichiers existants. Par contre, l'adjonction de l'option **-f** permet de le faire.

Syntaxe :

```
bzip2 [-cdztkqv] [noms_fichiers]
```

```
bunzip2 [-fkvs] [noms_fichiers]
```

4.4. Sauvegarde et archivage de fichiers

4.4.1. Sauvegarde de fichiers

La commande **cpio** copie une entrée sur une sortie en archivant tout au passage. Par tube, on peut diriger la sortie des commandes **ls** ou **find** vers **cpio**. Une archive au format **cpio** est composée de la concaténation du contenu des fichiers qui ont servi à la créer.

Syntaxe :

```
cpio -o [v]
```

```
cpio -i [dfmrtuv]] [motifs]
```

```
cpio -p [dmmruv] catalogue
```

Exemples :

```
$ ls | cpio -o > /dev/floppy
```

```
$ find . -depth -print | cpio -o > /dev/rmt0
```

Dans cet exemple, on sauvegarde le contenu du répertoire courant avec les sous-répertoires.

```
$ mkdir /tmp/test; cd /tmp/test
```

```
$ cpio -id < /tmp/f.cpio
```

Ici on restaure le répertoire courant.

4.4.2. Archivage de fichiers : Commande **tar**

La commande **tar** (**t**ape **a**rchive), permet d'assembler de nombreux fichiers en une seule archive, tout en conservant tous les attributs des fichiers (date, permissions, propriétaires)

Syntaxe : *tar ...*

Exemples : \$ tar -cf. mt.tar mt

Dans cet exemple, on archive le répertoire mt. Attention à la position de l'option **f**. Elle doit, si elle est utilisée, être toujours la dernière et le nom de fichier qui la suit doit être le nom de l'archive.

4.5. Configuration du service d'impression

La configuration lors de l'installation est recommandée.

Le fichier de config **/etc/printcap** contient des informations sur les imprimantes rattachées localement ou lointaine avec lesquelles le daemon doit communiquer. On conseille de ne pas directement le modifier (très risqué à cause de la difficulté de respecter les formats d'écriture). Pour tous les détails voir le manuel : **man printcap**

Exemples

```
# nom court de l'imprimante locale lp:\
# nom du répertoire de la file d'attente (sd=spool directory)
:sd=/var/spool/lpd/lp:\
# la taille maximum du fichier est illimitée (car 0)
:mx#0:\
# pas de page de séparation
:sh:\
# nom du fichier spécial pour printer locale
:lp=/dev/lp0:\
# nom du fichier de traitement du fichier
:if=/var/spool/lpd/lp/filter:
# nom court de l'imprimante REMOTE
lp:\
# nom du répertoire de la file d'attente
:sd=/var/spool/lpd/lp:\
# nom du serveur d'impression distant (rm=remote machine)
:rm=pc1.cfipen.fr:\
# nom de l'imprimante distante (rp=remote printer)
&nbModifications, ajout d'une imprimantesp; :rp=lp:\
```

4.6. Outil graphique printtool

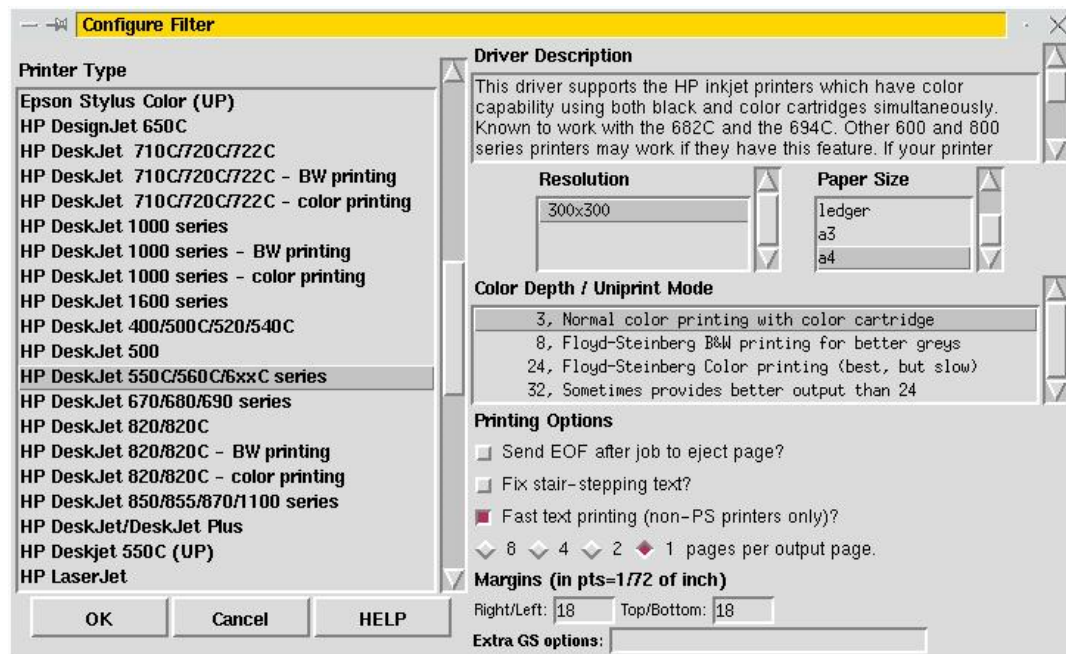
On peut complètement modifier le paramétrage du service d'impression, supprimer ajouter des imprimantes grâce à l'utilitaire graphique **printtool**

Le lancer dans un terminal **X** ou par : **K/appli non KDE/Système/Printtool**, puis **Add**

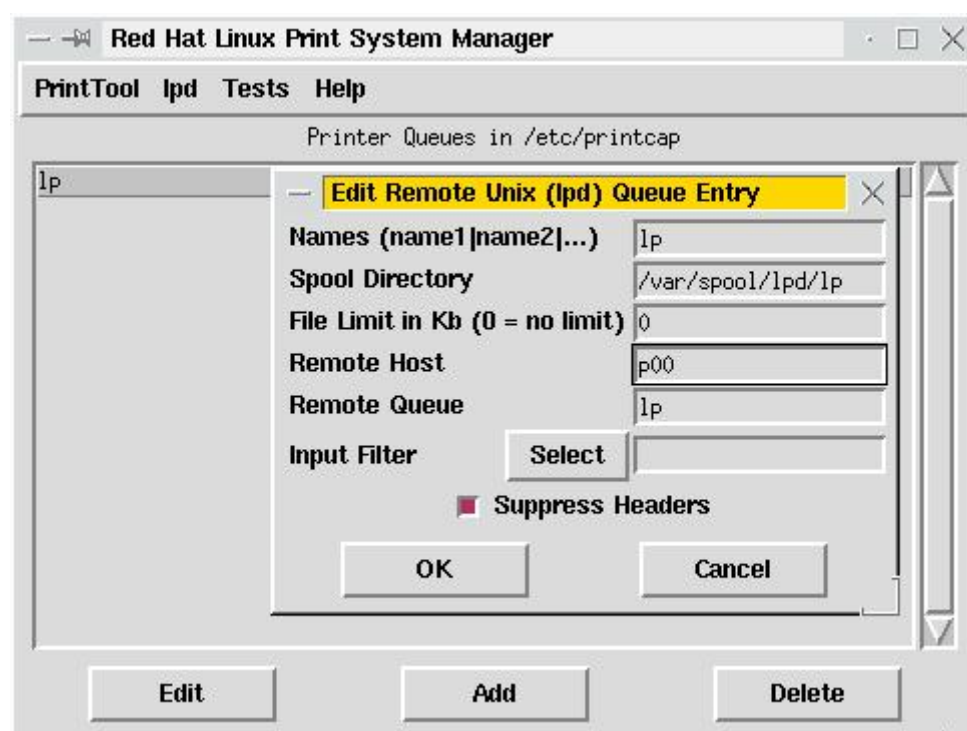
Si l'imprimante est connectée à un port parallèle, elle est auto-détectée, en général sur **/dev/lp0**, l'équivalent de LPT1.

Normalement, l'installation va créer un rép. de spool dans **/var/spool/lpd/** qui appartient à root avec les droits 755

Exemple d'installation d'une imprimante locale



Exemple d'installation d'une imprimante lointaine



Chapitre 5. Linux UBUNTU administration

5.1. The user root

By default, the password for the user “**root**” (the system administrator) is locked. This means that it is not possible to login as root or to use “**su**” command. Instead, the installer will set up “**sudo**” to allow the user that is created during install to run all administrative commands.

This means that in the terminal, one can use **sudo** for command that require **root privileges**. All programs in the menu will use a graphical sudo to prompt for a password. When sudo ask for a password, it needs the password of the user account created during installation process. This means that root password is not needed.

To run a comman which requires root privileges in a terminal, simply prepend sudo in front of it. To get an interactive root shell, use “**sudo -i**”.

5.2. Administrator group users

By default, only the user who installed the system is permitted to run sudo. To add more administrators, one just have to assign the group “admin” to the concerned users through the following command:

From a shell, do the following

Sudo adduser username admin

From the GUI, do the following

1. Access the system setting;
2. Select “Users and groups” program;
3. Add the new user to the admin group.

To change the rights of files and directories

```
find /srv - type f -exec chmod 640 {} \;
```

```
find /srv - type d -exec chmod 750 {} \;
```

5.3. Useful Comands for Linux Administration

5.3.1. Useful Comands for Linux newbies

5.3.1.1. Command: ls

The command “**ls**” stands for (**List Directory Contents**), List the contents of the specified location. If no location is specified, it gives the content of the current folder.

```
root@TALLA:~# ls
```

```
Android-Games    Music
Pictures         Public
Documents        TALLA-Sync
Downloads        Templates
```

The command “**ls -l**” lists the content of current (or specified) folder, in **long listing** fashion.

```
root@TALLA:~# ls -l
total 40588
drwxrwxr-x 2 talla talla 4096 May 8 01:06 Android Games
drwxr-xr-x 2 talla talla 4096 May 15 10:50 Desktop
drwxr-xr-x 2 talla talla 4096 May 16 16:45 Documents
drwxr-xr-x 6 talla talla 4096 May 16 14:34 Downloads
```

Command “**ls -a**”, lists the content of folder, including **hidden** files starting with ‘.’.

```
root@TALLA:~# ls -a
.          .gnupg      .dbus       .goutputstream-PI5VWV      .mission-control      .adobe      deja-
dup        .grsync     .mozilla    .themes      .Xauthority          .gconf      .local      .Xauthority .cache
```

Note: In **Linux**, a file name starting with ‘.’ is hidden. Every file/folder/device/command is a file. The output of **ls -l** is:

1. **d** (stands for directory).
2. **rwxr-xr-x** is the file permission of the file/folder for owner, group and world.
3. The 1st **talla** in the above example means that file is owned by user talla.
4. The 2nd **talla** in the above example means that file belongs to user group talla.
5. **4096** means file size is 4096 Bytes.
6. **May 8 01:06** is the date and time of last modification.
7. And at the end is the name of the **File/Folder**.

5.3.1.2. Command: **lsblk**

The “**lsblk**” stands for (**List Block Devices**), print block devices by their assigned name (but not **RAM**) on the standard output in a tree-like fashion.

```
root@TALLA:~# lsblk
NAME    MAJ:    MIN RM  SIZE RO  TYPE MOUNTPOINT
sda      8:0      0      232.9G 0 disk
├─sda1   8:1      0      46.6G 0 part /
├─sda2   8:2      0       1K 0 part
├─sda5   8:5      0     190M 0 part /boot
├─sda6   8:6      0     3.7G 0 part [SWAP]
├─sda7   8:7      0    93.1G 0 part /data
└─sda8   8:8      0    89.2G 0 part /personal
sr0     11:0     1     1024M 0 rom
```

The “**lsblk -l**” command lists block devices in ‘**list**’ structure (not tree like fashion).

```
root@TALLA:~# lsblk -l
NAME MAJ: MIN RM  SIZE RO TYPE MOUNTPOINT
sda  8:0  0 232.9G 0 disk
sda1 8:1  0 46.6G 0 part /
sda2 8:2  0  1K 0 part
sda5 8:5  0 190M 0 part /boot
```

```
sda6 8:6 0 3.7G 0 part [SWAP]
sda7 8:7 0 93.1G 0 part /data
sda8 8:8 0 89.2G 0 part /personal
sr0 11:0 1 1024M 0 rom
```

Note: **lsblk** is very useful and easiest way to know the name of **New Usb Device** you just plugged in, especially when you have to deal with disk/blocks in terminal.

5.3.1.3. Command: md5sum

The “**md5sum**” stands for (**Compute and Check MD5 Message Digest**), md5 checksum (commonly called **hash**) is used to match or verify integrity of files that may have changed as a result of a faulty file transfer, a disk error or non-malicious interference.

```
root@TALLA:~# md5sum teamviewer_linux.deb
47790ed345a7b7970fc1f2ac50c97002 teamviewer_linux.deb
```

Note: The user can match the generated **md5sum** with the one provided officially. Md5sum is considered less secure than **sha1sum** for example.

5.3.1.4. Command: dd

Command “**dd**” stands for (**Convert and Copy a file**), Can be used to convert and copy a file and most of the times is used to copy an iso file (or any other file) to a usb device (or any other location), thus can be used to make a ‘**Bootable**’ Usb Stick.

```
root@TALLA:~# dd if=/home/user/Downloads/debian.iso of=/dev/sdb1 bs=512M; sync
```

Note: In the above example the usb device is supposed to be sdb1 (one can verify it by using the command **lsblk**, otherwise the disk and OS could be overwritten), use name of disk very cautiously!

dd command takes some time ranging from a few seconds to several minutes in execution, depending on the size and type of file and read and write speed of Usb stick.

5.3.1.5. Command: uname

The “**uname**” command stands for (**Unix Name**), print detailed information about the machine name, Operating System and Kernel.

```
root@TALLA:~# uname -a
Linux TALLA 3.8.0-19-generic #30-Ubuntu SMP Wed May 1 16:36:13 UTC 2015 i686 i686 i686 GNU/Linux
```

Note: **uname** shows type of kernel. **uname -a** output detailed information. Elaborating the above output of **uname -a**.

1. “**Linux**”: The machine’s kernel name.

2. **"TALLA"**: The machine's node name.
3. **"3.8.0-19-generic"**: The kernel release.
4. **"#30-Ubuntu SMP"**: The kernel version.
5. **"i686"**: The architecture of the processor.
6. **"GNU/Linux"**: The operating system name.

5.3.1.6. Command: history

The **"history"** command stands for **History (Event) Record**, it prints the history of long list of executed commands in terminal.

```
root@TALLA:~# history
1 sudo add-apt-repository ppa:tualatrix/ppa
2 sudo apt-get update
3 sudo apt-get install ubuntu-tweak
4 sudo add-apt-repository ppa:diesch/testing
5 sudo apt-get update
```

5.3.1.7. Command: sudo

The **"sudo"** (**super user do**) command allows a permitted user to execute a command as the superuser or another user, as specified by the security policy in the sudoers list.

```
root@TALLA:~# sudo add-apt-repository ppa:tualatrix/ppa
```

Note: **sudo** allows user to borrow superuser privileged, while a similar command **'su'** allows user to actually log in as superuser. **Sudo** is safer than **su**. It is not advised to use **sudo** or **su** for day-to-day normal use, as it can result in serious error if accidentally you did something wrong, that's why a very popular saying in Linux community is:

"To err is human, but to really foul up everything, you need root password."

5.3.1.8. Command: mkdir

The **"mkdir"** (**Make directory**) command creates a new directory with name path. However, if the directory already exists, it will return an error message **"cannot create folder, folder already exists"**.

```
root@TALLA:~# mkdir TALLA
```

Note: Directory can only be created inside the folder, in which the user has write permission.

5.3.1.9. Command: touch

The **"touch"** command stands for (Update the access and modification times of each **FILE** to the current time). **touch** command creates the file, only if it doesn't exist. If the file already exists it will update the timestamp and not the contents of the file.


```
root@TALLA:~# touch TALLA_file
```

Note: **touch** can be used to create file under directory, on which the user has write permission, only if the file don't exist there.

5.3.1.10. Command: chmod

The Linux “**chmod**” command stands for (**change file mode bits**). chmod changes the file mode (**permission**) of each given file, folder, script, etc. according to mode asked for. There exist 3 types of permission on a file.

Read (r)=4
Write(w)=2
Execute(x)=1

So if you want to give only read permission on a file it will be assigned a value of ‘4’, for write permission only, a value of ‘2’ and for execute permission only, a value of ‘1’ is to be given. For read and write permission $4+2 = '6'$ is to be given, and so on. Now permission need to be set for 3 kinds of user and usergroup. The first is owner, then usergroup and finally world. For example, let consider the following permission on the file abc.sh.

```
rwxr-x--x abc.sh
```

Here the root's permission is **rw**x (**read**, **write** and **execute**). usergroup to which it belongs, is **r-x** (**read** and **execute** only, no write permission) and for world is **-x** (only **execute**).

The following command changes its permission and provides **read**, **write** and **execute** permissions to owner, group and world.

```
root@TALLA:~# chmod 777 abc.sh
```

The following command allows only **read** and **write** permission to all three.

```
root@TALLA:~# chmod 666 abc.sh
```

Note: one of the most important command useful for sysadmin and user both. On a multi-user environment or on a server, this command comes to rescue, setting wrong permission will either make a file inaccessible or provide unauthorized access to someone.

5.3.1.11. Command: chown

The Linux “**chown**” command stands for (**change file owner and group**). Every file belongs to a user group and an owner. The command “**ls -l**” shows the following contain:

```
root@TALLA:~# ls -l
drwxr-xr-x 3 server root 4096 May 10 11:14 Binary
```

```
drwxr-xr-x 2 server talla 4096 May 15 09:42 Desktop
```

Here the directory **Binary** is owned by user “**server**” and it belongs to usergroup “**root**” where as directory “**Desktop**” is owned by user “**server**” and belongs to user group “**server**”.

This “**chown**” command is used to change the file ownership and thus is useful in managing and providing file to authorised user and usergroup only.

```
root@TALLA:~# chown server:talla Binary
drwxr-xr-x 3 server talla 4096 May 10 11:14 Binary
drwxr-xr-x 2 server talla 4096 May 15 09:42 Desktop
```

Note: “**chown**” changes the user and group ownership of each given **FILE** to **NEW-OWNER** or to the user and group of an existing reference file.

5.3.1.12. Command: **apt**

The Debian based “**apt**” command stands for (**Advanced Package Tool**). **apt** is an advanced package manager for **Debian** based system (**Ubuntu**, **Kubuntu**, etc.), that automatically and intelligently **search**, **install**, **update** and **resolves dependency** of packages on **Gnu/Linux** system from command line.

```
root@TALLA:~# apt-get install mplayer
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following package was automatically installed and is no longer required:
java-wrappers
Use 'apt-get autoremove' to remove it.
---
0 upgraded, 9 newly installed, 0 to remove and 8 not upgraded.
Need to get 3,567 kB of archives.
After this operation, 7,772 kB of additional disk space will be used.
Do you want to continue [Y/n]? y
```

```
root@TALLA:~# apt-get update
Hit http://ppa.launchpad.net raring Release.gpg
Hit http://ppa.launchpad.net raring Release.gpg
Hit http://ppa.launchpad.net raring Release.gpg
Hit http://ppa.launchpad.net raring Release.gpg
Get:1 http://security.ubuntu.com raring-security Release.gpg [933 B]
---
```

Note: The above commands results into system-wide changes and hence requires root password (Check ‘#’ and not ‘\$’ as prompt). **Apt** is considered more advanced and intelligent as compared to **yum command**.

As the name suggest, **apt-cache** searches for package containing sub package **mpalyer**.
apt-get install, updates all the packages, that are already installed, to the newest one.

5.3.1.13. Command: tar

The “**tar**” command is a **Tape Archive**, useful in creation of archive, in a number of file format and their extraction.

```
root@TALLA:~# tar -zxvf abc.tar.gz (Remember 'z' for .tar.gz)
```

```
root@TALLA:~# tar -jxvf abc.tar.bz2 (Remember 'j' for .tar.bz2)
```

```
root@TALLA:~# tar -cvf archive.tar.gz(.bz2) /path/to/folder/abc
```

Note: A ‘**tar.gz**’ means gzipped. ‘**tar.bz2**’ is compressed with bzip which uses a better but slower compression method.

5.3.1.14. Command: cal

The “**cal**” (**Calendar**), is used to display calendar of the present month or any other month of any year that is advancing or passed.

```
root@TALLA:~# cal
```

```
March 2016
```

Su	Mo	Tu	We	Th	Fr	Sa
		1	2	3	4	5
6	7	8	9	10	11	12
13	14	15	16	17	18	19
20	21	22	23	24	25	26
27	28	29	30	31		

To show the calendar of the passed month of February in the year **1835**, one can type the following command.

```
root@TALLA:~# cal 02 1835
```

```
February 1835
```

Su	Mo	Tu	We	Th	Fr	Sa
1	2	3	4	5	6	7
8	9	10	11	12	13	14
15	16	17	18	19	20	21
22	23	24	25	26	27	28

The “**date**” (**Date**) command prints the current date and time on the standard output, and can further be set.

```
root@TALLA:~# date
```

```
Sun May 17 14:13:29 IST 2015
```

```
root@TALLA:~# date --set='11 may 2015 13:57'
```

```
Mon May 11 13:57:00 IST 2015
```

5.3.1.15. Command: cat

The “**cat**” stands for (**C**oncatenation). Concatenate (join) two or more plain file and/or print contents of a file on standard output.

```
root@TALLA:~# cat a.txt b.txt c.txt d.txt >> abcd.txt
root@TALLA:~# cat abcd.txt
```

This command displays the content of the file abcd.txt, which is the initial content of abcd.txt followed by the content of a.txt, followed by the content of b.txt, followed by the content of c.txt and ended with the content of d.txt.

Note: “>>” and “>” are called **append symbols**. They are used to append the output to a file and not on standard output. “>” symbol will delete a file already existed and create a new file; hence for security reason, it is advised to use “>>” that will write the output without overwriting or deleting the file.

Wildcards are a shell feature that makes the command line much more powerful than any **GUI** file managers. To select for example a big group of files in a graphical file manager, the mouse is frequently used. This may seem simple, but in some cases it can be very frustrating. For example, let consider a directory with a huge amount of all kinds of files and subdirectories. Suppose we want to move all the HTML files that have the word “Linux” somewhere in the middle of their names, from that big directory into another directory. What’s a simple way to do this? If the directory contains a huge amount of differently named HTML files, the task is everything but simple!

In the Linux CLI that task is just as simple to perform as moving only one HTML file, and it’s so easy because of the shell wildcards. These are special characters that allow to select file names that match certain patterns of characters. This helps to select even a big group of files by typing just a few characters, and in most cases it’s easier than selecting the files with a mouse. Here’s a list of the most commonly used wildcards:

Wildcard	Matches
*	zero or more characters
?	exactly one character
[abcde]	exactly one character listed
[a-e]	exactly one character in the given range
[!abcde]	any character that is not listed
[!a-e]	any character that is not in the given range
{debian,linux}	exactly one entire word in the options given

5.3.1.16. Command: cp

The “**cp**” command stands for (**C**opy). It copies a file from one location to another location.

```
root@TALLA:~# cp /home/user/Downloads abc.tar.gz /home/user/Desktop
```

This command Returns 0 when success

Note: **cp** is one of the most commonly used command in shell scripting and it can be used with wildcard characters (Describe in the above block), for customised and desired file copying.

5.3.1.17. Command: mv

The “**mv**” command stands for (**move**). It moves a file from one location to another location.

```
root@TALLA:~# mv /home/user/Downloads abc.tar.gz /home/user/talla
```

This command Returns 0 when success

Note: **mv** command can be used with wildcard characters. **mv** should be used with caution, as moving of system/unauthorised file may lead to security as well as breakdown of system.

5.3.1.18. Command: pwd

The command “**pwd**” stands for (**print working directory**). It prints the current working directory with full path name from terminal.

```
root@TALLA:~# pwd
/home/user/talla
```

Note: This command won't be much frequently used in scripting but it is an absolute life saver for newbie who gets lost in terminal in their early connection with Linux.

5.3.1.19. Command: cd

Finally, the frequently used “**cd**” command stands for (**change directory**), it changes the working directory to execute, copy, move, write, read, etc. from terminal itself.

```
root@TALLA:~# cd /home/user/Desktop
server@localhost:~$ pwd
/home/user/Desktop
```

Note: **cd** comes to rescue when switching between directories from terminal. “**Cd ~**” will change the working directory to user's home directory, and is very useful if a user finds himself lost in terminal. “**Cd ..**” will change the working directory to parent directory.

5.3.2. Advanced Commands for Middle Level Linux Users

5.3.2.1. Command: Find

Search for files in the given directory, hierarchically starting at the parent directory and moving to sub-directories.

```
root@TALLA:~# find -name *.sh
./Desktop/load.sh
./Desktop/test.sh
./Desktop/shutdown.sh
./Binary/firefox/run-mozilla.sh
./Downloads/kdewebdev-3.5.8/quanta/scripts/externalpreview.sh
./Downloads/wheezy-nv-install.sh
```

Note: The `-name` option makes the search case sensitive. The option `-iname` can be used to find something regardless of case. (* is a wildcard).

```
root@TALLA:~# find -iname *.SH (find -iname *.Sh / find -iname *.SH)
./Desktop/load.sh
./Desktop/test.sh
./Desktop/shutdown.sh
./Binary/firefox/run-mozilla.sh
./Downloads/kdewebdev-3.5.8/quanta/scripts/externalpreview.sh
./Downloads/wheezy-nv-install.sh
```

```
root@TALLA:~# find -name *.tar.gz
/var/www/modules/update/tests/aaa_update_test.tar.gz
./var/cache/flashplugin-nonfree/install_flash_player_11_linux.i386.tar.gz
./home/server/Downloads/drupal-7.22.tar.gz
./usr/share/gettext/archive.git.tar.gz
./usr/share/doc/apg/php.tar.gz
./usr/share/usb_modeswitch/configPack.tar.gz
```

Note: The above command searches for all the file having extension `'tar.gz'` in root directory and all the sub-directories including mounted devices.

5.3.2.2. Command: grep

The **“grep”** command searches the given file for lines containing a match to the given strings or words. Search `‘/etc/passwd’` for **‘TALLA’** user.

```
root@TALLA:~# grep TALLA /etc/passwd
TALLA:x:1000:1000:TALLA,,,:/home/TALLA:/bin/bash
```

Ignore word case and all other combination with `‘-i’` option.

```
root@TALLA:~# grep -i TALLA /etc/passwd
TALLA:x:1000:1000:TALLA,,,:/home/TALLA:/bin/bash
```

Search recursively (`-r`) i.e. read all files under each directory for a string **“127.0.0.1”**.

```
root@TALLA:~# grep -r "127.0.0.1" /etc/
/etc/vlc/lua/http.hosts:127.0.0.1
/etc/speech-dispatcher/modules/ivona.conf:#IvonaServerHost "127.0.0.1"
/etc/apache2/mods-available/status.conf: Allow from 127.0.0.1 ::1
```

```
/etc/apache2/mods-available/proxy_balancer.conf:# Allow from 127.0.0.1 ::1
/etc/security/access.conf:#+ : root : 127.0.0.1
/etc/dhcp/dhclient.conf:#prepend domain-name-servers 127.0.0.1;
/etc/init/network-interface.conf:    ifconfig lo 127.0.0.1 up || true
/etc/java-6-openjdk/net.properties:# http.nonProxyHosts=localhost|127.0.0.1
/etc/java-6-openjdk/net.properties:# ftp.nonProxyHosts=localhost|127.0.0.1
/etc/hosts:127.0.0.1        localhost
```

Note: The following options can be used along with **grep**.

1. **-w** for word (egrep -w '**word1|word2**' /path/to/file).
2. **-c** for count (i.e., total number of times the pattern matched) (grep -c '**word**' /path/to/file).
3. **-color** for coloured output (grep **-color** server /etc/passwd).

5.3.2.3. Command: man

The '**man**' is the system's manual pager. Man provides online documentation for all the possible options with a command and its usages. Almost all the command comes with their corresponding manual pages. For example,

```
root@TALLA:~# man man
```

```
MAN(1) Manual pager utils MAN(1)
```

NAME

man - an interface to the on-line reference manuals

SYNOPSIS

```
man [-C file] [-d] [-D] [--warnings[=warnings]] [-R encoding] [-L locale] [-m system[...]] [-M path] [-S
list] [-e extension] [-i|-I]
man -k [apropos options] regexp ...
man -K [-w|-W] [-S list] [-i|-I] [--regex] [section] term ...
man -f [whatis options] page ...
[-H[browser]] [-X[dpi]] [-Z] file ...
man -w|-W [-C file] [-d] [-D] page ...
man -c [-C file] [-d] [-D] page ...
man [-hV]
```

5.3.2.4. Command: ps

ps (Process) gives the status of running processes with a unique **Id** called **PID**.

```
root@TALLA:~# ps
```

PID	TTY	TIME	CMD
4170	pts/1	00:00:00	bash
9628	pts/1	00:00:00	ps

To list status of all the processes along with process **id** and **PID**, use option '**-A**'.

```
root@TALLA:~# ps -A
```

PID	TTY	TIME	CMD
-----	-----	------	-----

1	?	00:00:01	init
2	?	00:00:00	kthreadd
3	?	00:00:01	ksoftirqd/0
5	?	00:00:00	kworker/0:0H
7	?	00:00:00	kworker/u:0H
8	?	00:00:00	migration/0
9	?	00:00:00	rcu_bh

Note: This command is very useful when you want to know which processes are running or may need **PID** sometimes, for process to be killed. You can use it with '**grep**' command to find customised output. For example,

```
root@TALLA:~# ps -A | grep -i ssh
1500 ?    00:09:58 sshd
4317 ?    00:00:00 sshd
```

Here '**ps**' is pipelined with '**grep**' command to find customised and relevant output of our need.

5.3.2.5. Command: kill

This command is used to kill process which is not relevant now or is not responding. It is very useful command. Here you can kill a process and start it without restarting the whole system. You need a process' pid (ps) to kill it.

Let suppose you want to kill the program '**apache2**' that might not be responding. Run '**ps -A**' along with **grep** command.

```
root@TALLA:~# ps -A | grep -i apache2
1285 ?    00:00:00 apache2
```

Find process '**apache2**', note its **pid** and **kill** it. For example, in my case, the pid of '**apache2**' is '**1285**'. With the following command, you can kill the process.

```
root@TALLA:~# kill 1285
```

Note: Every time you re-run a process or start a system, a new **pid** is generated for each process and you can know about the current running processes and its **pid** using command '**ps**'.

Another way to kill the same process is.

```
root@TALLA:~# pkill apache2
```

Note: Kill requires **job id** / **process id** for sending signals, where as in **pkill**, you have an option of using pattern, specifying process owner, etc.

5.3.2.6. Command: whereis

The '**whereis**' command is used to locate the **Binary**, **Sources** and **Manual Pages** of the command. For example, to locate the **Binary**, **Sources** and **Manual Pages** of the command '**ls**' and '**kill**'.

```
root@TALLA:~# whereis ls
ls: /bin/ls /usr/share/man/man1/ls.1.gz
root@TALLA:~# whereis kill
kill: /bin/kill /usr/share/man/man2/kill.2.gz /usr/share/man/man1/kill.1.gz
```

Note: This is useful to know where the binaries are installed for manual editing sometimes.

5.3.2.7. Command: service

The '**service**' command controls the **Starting**, **Stopping** or **Restarting** of a '**service**'. This command makes it possible to **start**, **restart** or **stop** a service without restarting the system, for the changes to be taken into effect.

Startting an apache2 server on Ubuntu

```
root@TALLA:~# service apache2 start
* Starting web server apache2      apache2: Could not reliably determine the
server's fully qualified domain name, using 127.0.1.1 for ServerName
httpd (pid 1285) already running      [ OK ]
```

Restarting a apache2 server on Ubuntu

```
root@TALLA:~# service apache2 restart
* Restarting web server apache2      apache2: Could not reliably determine the
server's fully qualified domain name, using 127.0.1.1 for ServerName
... waiting .apache2: Could not reliably determine the server's fully qualified
domain name, using 127.0.1.1 for ServerName      [ OK ]
```

Stopping a apache2 server on Ubuntu

```
root@TALLA:~# service apache2 stop
* Stopping web server apache2      apache2: Could not reliably determine the
server's fully qualified domain name, using 127.0.1.1 for ServerName
... waiting      [ OK ]
```

Note: All the process script lies in '**/etc/init.d**', and the path might needs to be included on certain system, i.e., in spite of running "**service apache2 start**" you would be asked to run "**/etc/init.d/apache2 start**".

5.3.2.8. Command: alias

alias is a built in shell command that lets you assign name for a long command or frequently used command.

I uses '**ls -l**' command frequently, which includes 5 characters including space. Hence I created an alias for this to '**l**'.

```
root@TALLA:~# alias l='ls -l'
```

check if it works or not.

```
root@TALLA:~# l
```

```
total 36
```

```
drwxr-xr-x 3 TALLA TALLA 4096 May 10 11:14 Binary
drwxr-xr-x 3 TALLA TALLA 4096 May 21 11:21 Desktop
drwxr-xr-x 2 TALLA TALLA 4096 May 21 15:23 Documents
drwxr-xr-x 2 TALLA TALLA 4096 May  7 16:58 Music
drwxr-xr-x 2 TALLA TALLA 4096 May 20 16:17 Pictures
```

To remove alias 'l', use the following '**unalias**' command.

```
root@TALLA:~# unalias l
```

```
check, if 'l' still is alias or not.
```

```
root@TALLA:~# l
```

```
bash: l: command not found
```

Making a little fun out of this command. Make alias of certain important command to some other important command.

```
alias cd='ls -l' (set alias of ls -l to cd)
```

```
alias su='pwd' (set alias of pwd to su)
```

```
....
```

```
(You can create your own)
```

```
....
```

Now when your friend types '**cd**', just think how funny it would be when he gets directory listing and not directory changing. And when he tries to be '**su**' the all he gets is the location of working directory. You can remove the alias later using command '**unalias**' as explained above.

5.3.2.9. Command: df

Report disk usages of file system. Useful for user as well as System Administrator to keep track of their disk usages. '**df**' works by examining directory entries, which generally are updated only when a file is closed.

```
root@TALLA:~# df
```

Filesystem	1K-blocks	Used	Available	Use%	Mounted on
/dev/sda1	47929224	7811908	37675948	18%	/
none	4	0	4	0%	/sys/fs/cgroup
udev	1005916	4	1005912	1%	/dev
tmpfs	202824	816	202008	1%	/run
none	5120	0	5120	0%	/run/lock
none	1014120	628	1013492	1%	/run/shm
none	102400	44	102356	1%	/run/user
/dev/sda5	184307	79852	94727	46%	/boot
/dev/sda7	95989516	61104	91045676	1%	/data

```
/dev/sda8    91953192 57032 87218528 1% /personal
```

For more examples of '**df**' command, read the article [12 df Command Examples in Linux](#).

5.3.2.10. Command: du

Estimate file space usage. Output the summary of disk usages by ever file hierarchically, i.e., in recursive manner.

```
root@TALLA:~# du
```

```
8  ./Daily Pics/wp-polls/images/default_gradient
8  ./Daily Pics/wp-polls/images/default
32 ./Daily Pics/wp-polls/images
8  ./Daily Pics/wp-polls/tinymce/plugins/polls/langs
8  ./Daily Pics/wp-polls/tinymce/plugins/polls/img
28 ./Daily Pics/wp-polls/tinymce/plugins/polls
32 ./Daily Pics/wp-polls/tinymce/plugins
36 ./Daily Pics/wp-polls/tinymce
580 ./Daily Pics/wp-polls
1456 ./Daily Pics
36 ./Plugins/wordpress-author-box
16180 ./Plugins
12 ./May Articles 2013/Xtreme Download Manager
4632 ./May Articles 2013/XCache
```

Note: '**df**' only reports usage statistics on file systems, while '**du**', on the other hand, measures directory contents. For more '**du**' command examples and usage, read [10 du \(Disk Usage\) Commands](#).

5.3.2.11. Command: rm

The command '**rm**' stands for remove. **rm** is used to remove files (s) and directories.

Removing a directory

```
root@TALLA:~# rm PassportApplicationForm_Main_English_V1.0
```

```
rm: cannot remove `PassportApplicationForm_Main_English_V1.0': Is a directory
```

The directory can't be removed simply by '**rm**' command, you have to use '**-rf**' switch along with '**rm**'.

```
root@TALLA:~# rm -rf PassportApplicationForm_Main_English_V1.0
```

Warning: "**rm -rf**" command is a destructive command if accidentally you make it to the wrong directory. Once you '**rm -rf**' a directory all the files and the directory itself is lost forever, all of a sudden. Use it with caution.

5.3.2.12. Command: echo

echo as the name suggest echoes a text on the standard output. It has nothing to do with shell, nor does shell reads the output of echo command. However in an interactive script, echo passes the message to the user through terminal. It is one of the command that is commonly used in scripting, interactive scripting.

```
root@TALLA:~# echo "TALLA.com is a very good website"
```

TALLA.com is a very good website

creating a small interactive script

1. create a file, named '**interactive_shell.sh**' on desktop. (Remember '**.sh**' extension is must).
2. copy and paste the below script, exactly same, as below.

```
#!/bin/bash
echo "Please enter your name:"
read name
echo "Welcome to Linux $name"
```

Next, set execute permission and run the script.

```
root@TALLA:~# chmod 777 interactive_shell.sh
root@TALLA:~# ./interactive_shell.sh
```

Please enter your name:

Ravi Saive

Welcome to Linux Ravi Saive

Note: '**#!/bin/bash**' tells the shell that it is an script an it is always a good idea to include it at the top of script. '**read**' reads the given input.

5.3.2.13. Command: passwd

This is an important command that is useful for changing own password in terminal. Obviously you need to know your current password for Security reason.

```
root@TALLA:~# passwd
```

Changing password for TALLA.

(current) UNIX password: *****

Enter new UNIX password: *****

Retype new UNIX password: *****

Password unchanged [Here was password remains unchanged, i.e., new password=old password]

Enter new UNIX password: #####

Retype new UNIX password:#####

5.3.2.14. Command: lpr

This command print files named on command line, to named printer.

```
root@TALLA:~# lpr -P deskjet-4620-series 1-final.pdf
```

Note: The **'lpq'** command lets you view the status of a printer (whether it's up or not), and the jobs (files) waiting to be printed.

5.3.2.15. Command: cmp

compare two files of any type and writes the results to the standard output. By default, **'cmp'** Returns 0 if the files are the same; if they differ, the byte and line number at which the first difference occurred is reported.

To provide examples for this command, let's consider two files:

file1.txt

```
root@TALLA:~# cat file1.txt
```

```
Hi My name is TALLA
```

file2.txt

```
root@TALLA:~# cat file2.txt
```

```
Hi My name is TALLA [dot] com
```

Now, let's compare two files and see output of the command.

```
root@TALLA:~# cmp file1.txt file2.txt
```

```
file1.txt file2.txt differ: byte 15, line 1
```

5.3.2.16. Command: wget

Wget is a free utility for non-interactive (i.e., can work in background) download of files from the Web. It supports **HTTP**, **HTTPS**, **FTP** protocols and **HTTP** proxies.

Download ffmpeg using wget

```
root@TALLA:~# wget http://downloads.sourceforge.net/project/ffmpeg-php/ffmpeg-php/0.6.0/ffmpeg-  
php-0.6.0.tbz2
```

```
--2013-05-22 18:54:52-- http://downloads.sourceforge.net/project/ffmpeg-php/ffmpeg-  
php/0.6.0/ffmpeg-php-0.6.0.tbz2
```

```
Resolving downloads.sourceforge.net (downloads.sourceforge.net)... 216.34.181.59
```

```
Connecting to downloads.sourceforge.net (downloads.sourceforge.net)|216.34.181.59|:80... connected.
```

```
HTTP request sent, awaiting response... 302 Found
```

```
Location: http://kaz.dl.sourceforge.net/project/ffmpeg-php/ffmpeg-php/0.6.0/ffmpeg-php-0.6.0.tbz2  
[following]
```

```
--2013-05-22 18:54:54-- http://kaz.dl.sourceforge.net/project/ffmpeg-php/ffmpeg-php/0.6.0/ffmpeg-  
php-0.6.0.tbz2
```

```
Resolving kaz.dl.sourceforge.net (kaz.dl.sourceforge.net)... 92.46.53.163
```

```
Connecting to kaz.dl.sourceforge.net (kaz.dl.sourceforge.net)|92.46.53.163|:80... connected.
```

```
HTTP request sent, awaiting response... 200 OK
```

```
Length: 275557 (269K) [application/octet-stream]
```

Saving to: 'ffmpeg-php-0.6.0.tbz2'

```
100%[=====>]
2,75,557 67.8KB/s in 4.0s
```

2013-05-22 18:55:00 (67.8 KB/s) - 'ffmpeg-php-0.6.0.tbz2' saved [275557/275557]

5.3.2.17. Command: mount

Mount is an important command which is used to mount a filesystem that don't mount itself. You need root permission to mount a device.

First run '**lsblk**' after plugging-in your filesystem and identify your device and note down you device assigned name.

```
root@TALLA:~# lsblk
```

```
NAME MAJ:MIN RM SIZE RO TYPE MOUNTPOINT
sda 8:0 0 931.5G 0 disk
├─sda1 8:1 0 923.6G 0 part /
├─sda2 8:2 0 1K 0 part
└─sda5 8:5 0 7.9G 0 part [SWAP]
sr0 11:0 1 1024M 0 rom
sdb 8:16 1 3.7G 0 disk
└─sdb1 8:17 1 3.7G 0 part
```

From this screen it was clear that I plugged in a **4 GB** pendrive thus '**sdb1**' is my filesystem to be mounted. Become a **root** to perform this operation and change to **/dev** directory where all the file system is mounted.

```
root@TALLA:~# su
```

Password:

```
root@TALLA:~# cd /dev
```

Create a directory named anything but should be relevant for reference.

```
root@TALLA:~# mkdir usb
```

Now mount filesystem '**sdb1**' to directory '**usb**'.

```
root@TALLA:~# mount /dev/sdb1 /dev/usb
```

Now you can navigate to **/dev/usb** from terminal or **X-windows** system and access file from the mounted directory.

5.3.2.18. Command: gcc

gcc is the in-built compiler for '**c**' language in Linux Environment. A simple **c** program, save it on ur desktop as **Hello.c** (remember '**.c**' extension is must).

```
#include <stdio.h>
int main()
{
    printf("Hello world\n");
    return 0;
}
```

Compile it

```
root@TALLA:~# gcc Hello.c
```

Run it

```
root@TALLA:~# ./a.out
```

Hello world

Note: On compiling a **c** program the output is automatically generated to a new file "**a.out**" and everytime you compile a **c** program same file "**a.out**" gets modified. Hence it is a good advice to define a output file during compile and thus there is no risk of overwrite to output file.

Compile it this way

```
root@TALLA:~# gcc -o Hello Hello.c
```

Here '**-o**' sends the output to '**Hello**' file and not '**a.out**'. Run it again.

```
root@TALLA:~# ./Hello
```

Hello world

5.3.2.19. Command: g++

g++ is the in-built compiler for '**C++**', the first object oriented programming language. A simple **c++** program, save it on ur desktop as **Add.cpp** (remember '**.cpp**' extension is must).

```
#include <iostream>

using namespace std;

int main()
{
    int a;
    int b;
    cout<<"Enter first number:\n";
    cin >> a;
    cout <<"Enter the second number:\n";
    cin>> b;
    cin.ignore();
    int result = a + b;
    cout<<"Result is"<<" "<<result<<endl;
    cin.get();
    return 0;
}
```

```
}
```

Compile it

```
root@TALLA:~# g++ Add.cpp
```

Run it

```
root@TALLA:~# ./a.out
```

```
Enter first number:
```

```
...  
...
```

Note: On compiling a **c++** program the output is automatically generated to a new file “**a.out**” and everytime you compile a **c++** program same file “**a.out**” gets modified. Hence it is a good advice to define a output file during compile and thus there is no risk of overwrite to output file.

Compile it this way

```
root@TALLA:~# g++ -o Add Add.cpp
```

Run it

```
root@TALLA:~# ./Add
```

```
Enter first number:
```

```
...  
...
```

5.3.2.20. Command: java

Java is one of the world’s highly used programming language and is considered fast, secure, and reliable. Most of the the web based service of today runs on java.

Create a simple java program by pasting the below test to a file, named **TALLA.java** (remember ‘**.java**’ extension is must).

```
class TALLA {  
    public static void main(String[] arguments) {  
        System.out.println("TALLA ");  
    }  
}
```

compile it using javac

```
root@TALLA:~# javac TALLA.java
```

Run it

```
root@TALLA:~# java TALLA
```

Note: Almost every distribution comes packed with **gcc compiler**, major number of distros have inbuilt **g++** and **java compiler**, while some may not have. You can **apt** or **yum** the required package.

5.3.3.Advanced Commands for Linux Experts

5.3.3.1. Command: ifconfig

ifconfig is used to configure the kernel-resident network interfaces. It is used at boot time to set up interfaces as necessary. After that, it is usually only needed when debugging or when system tuning is needed.

Check Active Network Interfaces

```
[avishek@TALLA ~]$ ifconfig
```

```

eth0  Link encap:Ethernet HWaddr 40:2C:F4:EA:CF:0E
      inet addr:192.168.1.3 Bcast:192.168.1.255 Mask:255.255.255.0
      inet6 addr: fe80::422c:f4ff:feea:cf0e/64 Scope:Link
      UP BROADCAST RUNNING MULTICAST MTU:1500 Metric:1
      RX packets:163843 errors:0 dropped:0 overruns:0 frame:0
      TX packets:124990 errors:0 dropped:0 overruns:0 carrier:0
      collisions:0 txqueuelen:1000
      RX bytes:154389832 (147.2 MiB) TX bytes:65085817 (62.0 MiB)
      Interrupt:20 Memory:f7100000-f7120000
lo    Link encap:Local Loopback
      inet addr:127.0.0.1 Mask:255.0.0.0
      inet6 addr: ::1/128 Scope:Host
      UP LOOPBACK RUNNING MTU:16436 Metric:1
      RX packets:78 errors:0 dropped:0 overruns:0 frame:0
      TX packets:78 errors:0 dropped:0 overruns:0 carrier:0
      collisions:0 txqueuelen:0
      RX bytes:4186 (4.0 KiB) TX bytes:4186 (4.0 KiB)

```

Check All Network Interfaces

Display details of All interfaces including disabled interfaces using “-a” argument.

```
[avishek@TALLA ~]$ ifconfig -a
```

```

eth0  Link encap:Ethernet HWaddr 40:2C:F4:EA:CF:0E
      inet addr:192.168.1.3 Bcast:192.168.1.255 Mask:255.255.255.0
      inet6 addr: fe80::422c:f4ff:feea:cf0e/64 Scope:Link
      UP BROADCAST RUNNING MULTICAST MTU:1500 Metric:1
      RX packets:163843 errors:0 dropped:0 overruns:0 frame:0
      TX packets:124990 errors:0 dropped:0 overruns:0 carrier:0
      collisions:0 txqueuelen:1000
      RX bytes:154389832 (147.2 MiB) TX bytes:65085817 (62.0 MiB)
      Interrupt:20 Memory:f7100000-f7120000
lo    Link encap:Local Loopback
      inet addr:127.0.0.1 Mask:255.0.0.0
      inet6 addr: ::1/128 Scope:Host
      UP LOOPBACK RUNNING MTU:16436 Metric:1
      RX packets:78 errors:0 dropped:0 overruns:0 frame:0
      TX packets:78 errors:0 dropped:0 overruns:0 carrier:0
      collisions:0 txqueuelen:0
      RX bytes:4186 (4.0 KiB) TX bytes:4186 (4.0 KiB)
virbr0 Link encap:Ethernet HWaddr 0e:30:a3:3a:bf:03
      inet addr:192.168.122.1 Bcast:192.168.122.255 Mask:255.255.255.0

```

```
UP BROADCAST MULTICAST MTU:1500 Metric:1
RX packets:0 errors:0 dropped:0 overruns:0 frame:0
TX packets:0 errors:0 dropped:0 overruns:0 carrier:0
collisions:0 txqueuelen:0
RX bytes:0 (0.0 B) TX bytes:0 (0.0 B)
```

Disable an Interface

```
[avishek@TALLA ~]$ ifconfig eth0 down
```

Enable an Interface

```
[avishek@TALLA ~]$ ifconfig eth0 up
```

Assign IP Address to an Interface

Assign “192.168.1.12” as the IP address for the interface eth0.

```
[avishek@TALLA ~]$ ifconfig eth0 192.168.1.12
```

Change Subnet Mask of Interface eth0

```
[avishek@TALLA ~]$ ifconfig eth0 netmask 255.255.255.
```

Change Broadcast Address of Interface eth0

```
[avishek@TALLA ~]$ ifconfig eth0 broadcast 192.168.1.255
```

Assign IP Address, Netmask and Broadcast to Interface eth0

```
[avishek@TALLA ~]$ ifconfig eth0 192.168.1.12 netmask 255.255.255.0 broadcast 192.168.1.255
```

Note: If using a wireless network you need to use command “**iwconfig**”. For more “**ifconfig**” command examples and usage, read [15 Useful “ifconfig” Commands](#).

5.3.3.2. Command: netstat

netstat command displays various network related information such as network connections, routing tables, interface statistics, masquerade connections, multicast memberships etc.,

List All Network Ports

```
[avishek@TALLA ~]$ netstat -a
```

Active UNIX domain sockets (servers and established)

Proto	RefCnt	Flags	Type	State	I-Node	Path
unix	2	[ACC]	STREAM	LISTENING	13331	/tmp/.winbindd/pipe
unix	2	[ACC]	STREAM	LISTENING	11030	/var/run/mysqld/mysqld.sock
unix	2	[ACC]	STREAM	LISTENING	19308	/tmp/ssh-qnZadSgJAbqd/agent.3221
unix	2	[ACC]	STREAM	LISTENING	436781	/tmp/HotShots
unix	2	[ACC]	STREAM	LISTENING	46110	/run/user/talla/pulse/native
unix	2	[ACC]	STREAM	LISTENING	19310	/tmp/gpg-zfE9YT/S.gpg-agent

....

List All TCP Ports

```
[avishek@TALLA ~]$ netstat -at
```

Active Internet connections (servers and established)

Proto	Recv-Q	Send-Q	Local Address	Foreign Address	State
tcp	0	0	localhost:mysql	*.*	LISTEN
tcp	0	0	*:5901	*.*	LISTEN
tcp	0	0	talla-OptiPl:domain	*.*	LISTEN
tcp	0	0	localhost:ipp	*.*	LISTEN
tcp	0	0	talla-OptiPle:48270	ec2-23-21-236-70.c:http	ESTABLISHED

```
tcp    0    0 talla-OptiPle:48272 ec2-23-21-236-70.c: http TIME_WAIT
tcp    0    0 talla-OptiPle:48421 bom03s01-in-f22.1: https ESTABLISHED
```

...

Show Statistics for All Ports

```
[avishek@TALLA ~]$ netstat -s
```

Ip:

```
4994239 total packets received
3248924 requests sent out
8 outgoing packets dropped
```

Icmp:

```
29460 ICMP messages received
566 input ICMP message failed.
ICMP input histogram:
  destination unreachable: 98
  redirects: 29362
2918 ICMP messages sent
0 ICMP messages failed
ICMP output histogram:
  destination unreachable: 2918
```

IcmpMsg:

```
InType3: 98
InType5: 29362
OutType3: 2918
```

Tcp:

```
94533 active connections openings
23 passive connection openings
5870 failed connection attempts
7194 connection resets received
```

....

OK! For some reason if you want not to resolve host, port and user name as a output of netstat.

```
[avishek@TALLA ~]$ netstat -an
```

Fine, you may need to get the output of netstat continuously till interrupt instruction is passed (**ctrl+c**).

```
[avishek@TALLA ~]$ netstat -c
```

For more “**netstat**” command examples and usage, see the article [20 Netstat Command Examples](#).

5.3.3.3. Command: nslookup

A network utility program used to obtain information about Internet servers. As its name suggests, the utility finds name server information for domains by querying **DNS**.

```
[avishek@TALLA ~]$ nslookup TALLA.com
Server:      192.168.1.1
```

Address: 192.168.1.1#53

Non-authoritative answer:

Name: TALLA.com

Address: 50.16.67.239

Query Mail Exchanger Record

[avishek@TALLA ~]\$ nslookup -query=mx TALLA.com

Server: 192.168.1.1

Address: 192.168.1.1#53

Non-authoritative answer:

TALLA.com mail exchanger = 0 smtp.secureserver.net.

TALLA.com mail exchanger = 10 mailstore1.secureserver.net.

Authoritative answers can be found from:

Query Name Server

[avishek@TALLA ~]\$ nslookup -type=ns TALLA.com

Server: 192.168.1.1

Address: 192.168.1.1#53

Non-authoritative answer:

TALLA.com nameserver = ns3404.com.

TALLA.com nameserver = ns3403.com.

Authoritative answers can be found from:

Query DNS Record

[avishek@TALLA ~]\$ nslookup -type=any TALLA.com

Server: 192.168.1.1

Address: 192.168.1.1#53

Non-authoritative answer:

TALLA.com mail exchanger = 10 mailstore1.secureserver.net.

TALLA.com mail exchanger = 0 smtp.secureserver.net.

TALLA.com nameserver = ns06.domaincontrol.com.

TALLA.com nameserver = ns3404.com.

TALLA.com nameserver = ns3403.com.

TALLA.com nameserver = ns05.domaincontrol.com.

Authoritative answers can be found from:

Query Start of Authority

[avishek@TALLA ~]\$ nslookup -type=soa TALLA.com

Server: 192.168.1.1

Address: 192.168.1.1#53

Non-authoritative answer:

TALLA.com

origin = ns3403.hostgator.com

mail addr = dnsadmin.gator1702.hostgator.com

serial = 2012081102

refresh = 86400

retry = 7200

expire = 3600000

minimum = 86400

Authoritative answers can be found from:

Query Port Number

Change the port number using which you want to connect

```
[avishek@TALLA ~]$ nslookup -port 56 TALLA.com
Server:                TALLA.com
Address:  50.16.76.239#53
Name:      56
Address: 14.13.253.12
```

Read Also : [8 Nslookup Commands](#)

5.3.3.4. Command: dig

dig is a tool for querying **DNS** nameservers for information about host addresses, mail exchanges, nameservers, and related information. This tool can be used from any Linux (**Unix**) or **Macintosh OS X** operating system. The most typical use of **dig** is to simply query a single host.

```
[avishek@TALLA ~]$ dig TALLA.com
;<<>> DiG 9.8.2rc1-RedHat-9.8.2-0.17.rc1.el6 <<>> TALLA.com
;; global options: +cmd
;; Got answer:
;; ->>HEADER<
```

Turn Off Comment Lines

```
[avishek@TALLA ~]$ dig TALLA.com +nocomments
;<<>> DiG 9.8.2rc1-RedHat-9.8.2-0.17.rc1.el6 <<>> TALLA.com +nocomments
;; global options: +cmd
;TALLA.com.                IN      A
TALLA.com.                14400   IN      A      40.216.66.239
;; Query time: 418 msec
;; SERVER: 192.168.1.1#53(192.168.1.1)
;; WHEN: Sat Jun 29 13:53:22 2013
;; MSG SIZE rcvd: 45
```

Turn Off Authority Section

```
[avishek@TALLA ~]$ dig TALLA.com +noauthority
;<<>> DiG 9.8.2rc1-RedHat-9.8.2-0.17.rc1.el6 <<>> TALLA.com +noauthority
;; global options: +cmd
;; Got answer:
;; ->>HEADER<
```

Turn Off Additional Section

```
[avishek@TALLA ~]$ dig TALLA.com +noadditional
;<<>> DiG 9.9.2-P1 <<>> TALLA.com +noadditional
;; global options: +cmd
;; Got answer:
;; ->>HEADER<
```

Turn Off Stats Section

```
[avishek@TALLA ~]$ dig TALLA.com +nostats
;<<>> DiG 9.8.2rc1-RedHat-9.8.2-0.17.rc1.el6 <<>> TALLA.com +nostats
;; global options: +cmd
;; Got answer:
;; ->>HEADER<
```

Turn Off Answer Section

```
[avishek@TALLA ~]$ dig TALLA.com +noanswer
```

```
; <<>> DiG 9.8.2rc1-RedHat-9.8.2-0.17.rc1.el6 <<>> TALLA.com +noanswer
;; global options: +cmd
;; Got answer:
;; ->>HEADER<
```

Disable All Section at Once

```
[avishek@TALLA ~]$ dig TALLA.com +noall
; <<>> DiG 9.8.2rc1-RedHat-9.8.2-0.17.rc1.el6 <<>> TALLA.com +noall
;; global options: +cmd
```

5.3.3.5. Command: uptime

You have just connected to your **Linux Server Machine** and founds Something unusual or malicious, what you will do? Guessing.... NO, definitely not you could run **uptime** to verify what happened actually when the server was unattended.

```
[avishek@TALLA ~]$ uptime
14:37:10 up 4:21, 2 users, load average: 0.00, 0.00, 0.04
```

5.3.3.6. Command: wall

one of the most important command for administrator, **wall** sends a message to everybody logged in with their **mesg** permission set to “**yes**”. The message can be given as an argument to **wall**, or it can be sent to wall’s standard input.

```
[avishek@TALLA ~]$ wall "we will be going down for maintenance for one hour sharply at 03:30 pm"
Broadcast message from root@localhost.localdomain (pts/0) (Sat Jun 29 14:44:02 2013):
we will be going down for maintenance for one hour sharply at 03:30 pm
```

5.3.3.7. command: mesg

Lets you control if people can use the “**write**” command, to send text to you over the screen.

```
mesg [n|y]
n - prevents the message from others popping up on the screen.
y - Allows messages to appear on your screen.
```

5.3.3.8. Command: write

Let you send text directly to the screen of another Linux machine if ‘**mesg**’ is ‘**y**’.

```
[avishek@TALLA ~]$ write talla
```

5.3.3.9. Command: talk

An enhancement to **write** command, **talk** command lets you talk to the logged in users.

```
[avishek@TALLA ~]$ talk talla
```

Note: If **talk** command is not installed, you can always **apt** or **yum** the required packages.

```
[avishek@TALLA ~]$ yum install talk
OR
[avishek@TALLA ~]$ apt-get install talk
```

5.3.3.10. Command: w

what command 'w' seems you funny? But actually it is not. t's a command, even if it's just one letter long! The command "w" is a combination of **uptime** and **who** commands given one immediately after the other, in that order.

```
[avishek@TALLA ~]$ w
15:05:42 up 4:49, 3 users, load average: 0.02, 0.01, 0.00
USER  TTY  FROM      LOGIN@  IDLE  JCPU  PCPU  WHAT
server tty7  :0        14:06   4:43m  1:42  0.08s pam: gdm-passwo
server pts/0  :0.0     14:18   0.00s  0.23s 1.65s gnome-terminal
server pts/1  :0.0     14:47   4:43   0.01s 0.01s bash
```

5.3.3.11. Command: rename

As the name suggests, this command rename files. rename will rename the specified files by replacing the first occurrence from the file name.

Give the file names a1, a2, a3, a4.....1213

Just type the command.

```
rename a1 a0 a?
rename a1 a0 a??
```

5.3.3.12. Command: top

Displays the processes of **CPU**. This command refresh automatically, by default and continues to show **CPU** processes unless interrupt-instruction is given.

```
[avishek@TALLA ~]$ top
top - 14:06:45 up 10 days, 20:57, 2 users, load average: 0.10, 0.16, 0.21
Tasks: 240 total, 1 running, 235 sleeping, 0 stopped, 4 zombie
%Cpu(s): 2.0 us, 0.5 sy, 0.0 ni, 97.5 id, 0.0 wa, 0.0 hi, 0.0 si, 0.0 st
KiB Mem: 2028240 total, 1777848 used, 250392 free, 81804 buffers
KiB Swap: 3905532 total, 156748 used, 3748784 free, 381456 cached
  PID USER  PR  NI  VIRT  RES  SHR  S  %CPU  %MEM  TIME+  COMMAND
23768 ravisai 20   0 1428m 571m 41m  S   2.3 28.9 14:27.52 firefox
24875 ravisai 20   0 263m 14m 10m  S   0.3  0.7  0:02.76 lxterminal
   1 root   20   0 3896 1928 1228  S   0.0  0.1  0:01.62 init
   2 root   20   0   0   0   0  S   0.0  0.0  0:00.06 kthreadd
   3 root   20   0   0   0   0  S   0.0  0.0  0:17.28 ksoftirqd/0
   5 root   0 -20   0   0   0  S   0.0  0.0  0:00.00 kworker/0:0H
```

5.3.3.13. Command: mkfs.ext4

This command create a new **ext4** file system on the specified device, if wrong device is followed after this command, the whole block will be wiped and formatted, hence it is suggested not to run this command unless and until you understand what you are doing.

```
Mkfs.ext4 /dev/sda1 (sda1 block will be formatted)
```

```
mkfs.ext4 /dev/sdb1 (sdb1 block will be formatted)
```

5.3.3.14. Command: vi/emacs/nano

vi (visual), **emacs**, **nano** are some of the most commonly used editors in Linux. They are used oftenly to edit text, configuration,... files. A quick guide to work around vi and nano is, emacs is a.

vi-editor

```
[avishek@TALLA ~]$ touch a.txt (creates a text file a.txt)
```

```
[avishek@TALLA ~]$ vi a.txt (open a.txt with vi editor)
```

[press '**i**' to enter insert mode, or you won't be able to type-in anything]

```
echo "Hello" (your text here for the file)
```

1. **alt+x** (exit insert mode, remember to keep some space between the last letter.
2. **ctrl+x** command or your last word will be deleted).
3. **:wq!** (saves the file, with the current text, remember '**!**' is to override).

nano editor

```
[avishek@TALLA ~]$ nano a.txt (open a.txt file to be edited with nano)
```

edit, with the content, required

ctrl +x (to close the editor). It will show output as:

```
Save modified buffer (ANSWERING "No" WILL DESTROY CHANGES) ?
```

```
Y Yes
```

```
N No      ^C Cancel
```

Click '**y**' to yes and enter file name, and you are done.

5.3.3.15. Command: rsync

Rsync copies files and has a **-P** switch for a progress bar. So if you have rsync installed, you could use a simple alias.

```
alias cp='rsync -aP'
```

Now try to copy a large file in terminal and see the output with remaining items, similar to a progress bar.

Moreover, Keeping and Maintaining backup is one of the most important and boring work a system administrator, needs to perform. Rsync is a very nice tool (there exists, several other) to create and maintain backup, in terminal.


```
[avishek@TALLA ~]$ rsync -zvr IMG_5267\ copy\=33\ copy\=ok.jpg ~/Desktop/
sending incremental file list
IMG_5267 copy=33 copy=ok.jpg
sent 2883830 bytes received 31 bytes 5767722.00 bytes/sec
total size is 2882771 speedup is 1.00
```

Note: -z for compression, -v for verbose and -r for recursive.

5.3.3.16. Command: free

Keeping track of memory and resources is as much important, as any other task performed by an administrator, and 'free' command comes to rescue here.

Current Usage Status of Memory

```
[avishek@TALLA ~]$ free
              total    used    free   shared  buffers   cached
Mem:      2028240 1788272 239968      0   69468   363716
-/+ buffers/cache: 1355088 673152
Swap:      3905532 157076 3748456
```

Tuned Output in KB, or MB, or GB

```
[avishek@TALLA ~]$ free -b
              total    used    free   shared  buffers   cached
Mem:  2076917760 1838272512 238645248      0 71348224 372670464
-/+ buffers/cache: 1394253824 682663936
Swap: 3999264768 160845824 3838418944
```

```
[avishek@TALLA ~]$ free -k
              total    used    free   shared  buffers   cached
Mem:      2028240 1801484 226756      0   69948   363704
-/+ buffers/cache: 1367832 660408
Swap:      3905532 157076 3748456
```

```
[avishek@TALLA ~]$ free -m
              total    used    free   shared  buffers   cached
Mem:         1980    1762    218      0     68     355
-/+ buffers/cache:    1338    641
Swap:        3813    153   3660
```

```
[avishek@TALLA ~]$ free -g
              total    used    free   shared  buffers   cached
Mem:          1      1      0      0      0      0
-/+ buffers/cache:      1      0
Swap:          3      0      3
```

Check Current Usage in Human Readable Format

```
[avishek@TALLA ~]$ free -h
              total    used    free   shared  buffers   cached
Mem:         1.9G    1.7G   208M      0B    68M   355M
-/+ buffers/cache:    1.3G   632M
Swap:        3.7G   153M   3.6G
```

Check Status Continuously After Regular Interval

```
[avishek@TALLA ~]$ free -s 3
              total    used    free   shared  buffers   cached
Mem:      2028240 1824096 204144      0   70708   364180
```

```

-/+ buffers/cache: 1389208 639032
Swap: 3905532 157076 3748456
      total    used    free   shared  buffers   cached
Mem:  2028240 1824192  204048     0    70716   364212
-/+ buffers/cache: 1389264 638976
Swap: 3905532 157076 3748456

```

Read Also : [10 Examples of Free Command](#)

5.3.3.17. Command: mysqldump

Ok till now you would have understood what this command actually stands for, from the name of this command. **mysqldump** commands dumps (backups) all or a particular database data into a given a file. For example,

```
[avishek@TALLA ~]$ mysqldump -u root -p --all-databases > /home/server/Desktop/backupfile.sql
```

Note: **mysqldump** requires mysql to be running and correct password for authorisation. We have covered some useful “**mysqldump**” commands at [Database Backup with mysqldump Command](#)

5.3.3.18. Command: mkpasswd

Make a hard-to-guess, random password of the length as specified.

```
[avishek@TALLA ~]$ mkpasswd -l 10
```

```
zI4+Ybqfx9
```

```
[avishek@TALLA ~]$ mkpasswd -l 20
```

```
w0Pr7aqKk&hmbmqdrImk
```

Note: **-l 10** generates a random password of **10** characters while **-l 20** generates a password of character **20**, it could be set to anything to get desired result. This command is very useful and implemented in scripting language oftenly to generate random passwords. You might need to **yum** or **apt** the ‘expect’ package to use this command.

```
[root@TALLA ~]# yum install expect
```

OR

```
[root@TALLA ~]# apt-get install expect
```

5.3.3.19. Command: paste

Merge two or more text files on lines using. Example. If the content of file1 was:

```
1
```

```
2
```

```
3
```

and file2 was:

```
a
```

b
c
d

the resulting file3 would be:

1 a
2 b
3 c
d

5.3.3.20. Command: lsof

lsof stands for “**list open files**” and displays all the files that your system has currently opened. It’s very useful to figure out which processes uses a certain file, or to display all the files for a single process. Some useful [10 lsof Command](#) examples, you might be interested in reading.

[avishek@TALLA ~]\$ lsof

```
COMMAND PID TID USER FD TYPE DEVICE SIZE/OFF NODE NAME
init 1 root cwd DIR 8,1 4096 2 /
init 1 root rtd DIR 8,1 4096 2 /
init 1 root txt REG 8,1 227432 395571 /sbin/init
init 1 root mem REG 8,1 42672 270178 /lib/i386-linux-gnu/libnss_nis-2.17.so
init 1 root mem REG 8,1 87940 270187 /lib/i386-linux-gnu/libnsl-2.17.so
init 1 root mem REG 8,1 30560 263021 /lib/i386-linux-gnu/libnss_compat-2.17.so
init 1 root mem REG 8,1 124637 270176 /lib/i386-linux-gnu/libpthread-2.17.so
init 1 root 6r 0000 0,9 0 6245 anon_inode
init 1 root 7u unix 0xf5e91f80 0t0 8192 @/com/ubuntu/upstart
init 1 root 8w REG 8,1 3916 394 /var/log/upstart/teamviewer.log.1 (deleted)
```

This is not the end, a **System Administrator** does a lot of stuff, to provide you such a nice interface, upon which you work. System Administration is actually an art of learning and implementing in a very much perfect way. We will try to get you with all other necessary stuff which a linux professional must learn, linux in its basic actually itself, is a process of learning and learning. Your good words are always sought, which encourages us to put in more effort to give you a knowledgeable article. “**Like and share Us, to help Us Spread**”.

Bibliographie

- ❖ Andrew Tanenbaum, « Réseaux », 4ème édition, Pearson Education, ISBN 2-7440-7001-7
- ❖ Claude Servin, Dunod, « Réseaux et Télécoms », ISBN 2-10- 007986-7
- ❖ J. Kurose et K. Ross, « Analyse structurée des réseaux », 2ème édition, Pearson Education, ISBN 2-7440-7000-9
- ❖ W. R. Stevens, « TCP/IP Illustrated Volume 1, The Protocols », Addison Wesley, ISBN 0-201-63346-9
- ❖ D. Comer, « TCP/IP, Architecture, protocoles, applications », 4ème édition, Dunod, ISBN 2-10-008181-0
- ❖ Addison-Wesley, « An Engineering Approach to Computer Networking », ISBN 0-201-63442-6
- ❖ TCP/IP Administration de réseau, C. Hunt (Traduction de Emmanuel Chaput et de Stéphane Larroque), Ed. O'REILLY, 2002.
- ❖ Pratique des réseaux d'entreprise, J. L. Montagnier (Eyrolles)
- ❖ Technologie des ordinateurs et des réseaux, P. A. Goupille (Dunod)
- ❖ Le nouveau guide du PC, Jack Steiner (Osman Eyrolles Multimédia)
- ❖ Gérer et utiliser son réseau pédagogique avec Windows NT, Daniel Le Roizic (Bertrand-Lacoste)

Sites web

- ❖ <http://www.guill.net/>
- ❖ <http://www.courseforge.org/courses/>
- ❖ <http://www.commentcamarche.net/ccmdoc/>
- ❖ <http://www.rfc-editor.org/> (documents normatifs dans TCP/IP)
- ❖ <http://www.icann.org/registrars/accredited-list.html>
- ❖ <http://www.gandi.net/tlds.html.fr>
- ❖ <http://www.linuxfr.org>
- ❖ <http://www.toolinux.org>
- ❖ <http://www.linux-france.org/article/index.html>.
- ❖ <http://lea-linux.org>
- ❖ Ubuntu : <http://www.ubuntu-fr.org/>
- ❖ Redhat : <http://www.redhat.com>
- ❖ Suse : <http://www.suse.com>
- ❖ Debian : <http://www.debian.org>
- ❖ KDE : <http://www.kde.org>
- ❖ GNU : <http://www.gnu.org>
- ❖ GNOME : <http://www.gnome.org>
- ❖ <http://www.esprit-libre.net/configuration/level.php3>
- ❖ <http://www.lea-linux.org>