Support de Formation en ligne

Configuration, mise en oeuvre et administration de serveurs Internet et Intranet sous Linux.

Support de cours réalisé dans le cadre de formations effectuée au Burkina Faso et au Niger

par Africa Computing

Auteurs : Philippe Drouot, Hédi Kamel et Frédéric Renet

Infos : service−formation@africacomputing.org

Africa Computing est un organisme de formation professionnelle
Déclaration DDTEFP n°93 13 10226 13.

PLAN DE LA FORMATION

1. PRÉAMBULE

Linux est un système d'exploitation moderne bénéficiant de l'ensemble des fonctionnalités d'Unix. Ce n'est pas un produit commercial : c'est un logiciel libre que l'on peut obtenir gratuitement. Il est livré avec toutes les fonctionnalités, les outils et les utilitaires habituellement livrés avec les variantes commerciales d'Unix :

- c'est un système 32 bits (64 bits sur certaines plate−formes) ;
- il est multi−utilisateurs ;
- il est multitâche (multitâche préemptif et non coopératif comme Windows 98) ;
- dans le domaine des réseaux, il prend parfaitement en charge la famille des protocoles TCP/IP et possède bien plus de caractéristiques que la plupart des variantes commerciales d'Unix ;
- il dispose de shells très performants ainsi que de XFree86, une implémentation complète du système X−Window.

Linux possède les caractéristiques idéales pour implémenter un serveur Internet stable, performant, sécurisé et flexible. L'objectif principal de cette formation est d'acquérir les compétences permettant de proposer à ses clients ou partenaires un service d'hébergement de sites Internet ainsi qu'un service de gestion de boîtes aux lettres électroniques.

Les parties pratiques de cette formation s'appuient sur deux distributions de Linux : la debian 3.0 "woody" et la Mandrake 9.0.

Très complète et conçue de façon méticuleuse et efficace, la Debian permet de tout dimensionner selon ses besoins. Sa prise en main est parfois délicate, surtout pour les débutants. Mais une fois passés les premiers obstacles, on ne veut plus en changer.

La Mandrake est faite pour le confort de l'utilisateur final. Elle est à recommandé à tous ceux qui veulent utiliser leur machine rapidement sans passer trop de temps à jouer le rôle de l'ingénieur système.

Vous pouvez téléchargez les distributions de la Debian et de la Mandrake sur les sites debian.org et linux−mandrake.com. Si vous êtes en Afrique et que le débit de votre connection Internet est trop lent pour envisager de télécharger les 3 CDs de chacune des 2 distributions, contactez nous à l'adresse : service−formation@africacomputing.org (mailto:service−formation@africacomputing.org)

Pour vous intitier à l'administration d'un serveur Internet, vous pouvez vous appuyer sur l'utilisation de l'excellent outil graphique d'administration Webmin. Bien que peu connu du grand public, ce logiciel libre permet d'administrer entièrement un serveur Internet à partir de simples pages html : un simple navigateur suffit alors pour paramétrer la plupart des fonctionnalités. Il devient alors possible d'administrer son serveur depuis n'importe quel poste relié à Internet. Attention toutefois aux nombreux trous de sécurité qui sont régulièrement découvert sur Webmin. Pour l'utilisation d'un serveur en production dont la sécurité est un sujet sensible, nous vous déconseillons vivement l'utilisation de Webmin.

Sous Linux comme sur tous les systèmes Unix, l'administration consiste principalement à éditer et modifier des fichiers textes qui font office de fichiers de configuration. Bien que la perspective de passer son temps à éditer des fichiers de configuration en mode texte ne soit pas très alléchante et qu'il soit possible de se contenter d'utiliser l'interface graphique webmin ou linuxconf, il est important d'avoir une bonne vision générale de l'administration en mode texte : ceci afin de pouvoir résoudre correctement les problèmes rencontrés mais aussi pour utiliser des fonctionnalités non paramétrables graphiquement, pour mieux comprendre les mécanismes en jeux ou encore pour être capable d'intervenir sur n'importe quel serveur Unix (directement sur la machine ou depuis n'importe quel poste en utilisant le service telnet). C'est pourquoi nous traitons dans cette formation à la fois l'administration à partir des

fichiers textes de configuration et à partir du puissant outil graphique webmin.

Dans chaque section de ce document, nous ajoutons à la description du paramétrage des fichiers de configuration textuels, une partie décrivant la procédure correspondante avec Webmin (ou d'autres outils graphiques de configuration).

Par convention, les parties décrivant l'administration à partir de l'interface graphique Webmin sont encadrées.

Pour toutes remarques, commentaires ou suggestions sur ce support de cours, vous pouvez envoyer un courrier électronique à l'adresse suivante : service-formation@africacomputing.org (mailto:service-formation@africacomputing.org)

L'équipe d'Africa Computing

2. PREMIERS CONTACTS AVEC LINUX − INSTALLATION

2.1. POURQUOI UTILISER LINUX COMME SERVEUR INTERNET ?

2.1.1. Le système d'exploitation Linux

Linux est le système qui connaît actuellement le plus grand développement sur l'Internet.

Principalement pour les raisons suivantes :

- Linux est le système de prédilection pour l'installation de trois logiciels serveurs leaders sur l'Internet : Apache en serveur Web (60% avec ses dérivés selon Netcraft (http://www.netcraft.com)), Sendmail en serveur courrier et Bind en serveur DNS ;
- Le logiciel Samba qui lui permet d'être serveur de fichier et d'impression en environnement Microsoft ;
- La stabilité et la sécurité que lui confère le développement de son architecture et de ses modules au sein de la communauté Open Source.
- Le large choix d'applications dans de très nombreux domaines. Par exemple, la dernière distribution Debian donne accés à plus de 2000 logiciels différents.
- Moins d'interruptions de service grace à une gestion intelligente de l'installation des logiciels. Une serveur sous Linux ne doit êter redémarré que lors d'une modification matérielle comme l'ajout d'un disque ou d'une carte.
- Logiciel Libre. Linux est gratuit et librement recopiable. Cela signifie que l'on peut télécharger une version de Linux ou l'emprunter et l'installer sur n'importe quel nombre d'ordinateur.
- Accés aux sources des logiciels. Tous les utilisateurs peuvent modifier le fonctionnement des programmes ou engager un programmeur pour le faire.
- Linux est plus efficace et consomme moins de ressources CPU et mémoire que Windows. On peut par exemple faire un serveur d'impression avec un vieux 486.

LINUX COMME SERVEUR = COUTS REDUITS, SECURITE ET PERFORMANCE !

2.1.2. Linux et le serveur Apache

Le serveur Web Apache propose une qualité de service que peu d'offres commerciales peuvent concurrencer, preuve en est la formidable part de marché de cette solution. En Janvier 2002, Apache représente 62% des serveurs Web dans le monde contre 27% pour Microsoft Internet Information Server.

Apache tourne sur Unix, que ce soit Linux ou un UNIX BSD, ainsi que sur WindowsNT, W2K, et WXP. Plus d'informations sur le site d'Apache (http://httpd.apache.org).
Une nette majorité des serveurs web tournent sous Unix (dont une bonne partie sous Linux), pour des raisons de performance et surtout de fiabilité.

Le serveur Web Apache peut être utilisé comme simple serveur web, ou bien comme serveur d'application et interface de base de données avec les logiciels PHP et MySQL.

De plus utiliser des logiciels libres, par opposition à des logiciels payants d'origine US, est d'une part nettement moins cher, et un moyen de préserver l'indépendance technologique des pays.

2.1.3. Les possibilités serveurs de Linux

Linux en tant que serveur Intranet / Internet peut devenir l'ensemble des solutions suivantes et il est bien entendu possible qu'un seul et même ordinateur gère toutes ces possibilités :

- un serveur WEB classique (HTTP) ;
- un serveur FTP ;
- un serveur de mail (SMTP, POP) ;
- un serveur Proxy ;
- un Firewall ;
- un serveur DNS ;
- un routeur, etc…

Linux peut gérer un réseau d'entreprise, comme :

- un serveur de fichiers ;
- un serveur d'impression ;
- un serveur de fax ;
- un serveur de connexion Dial−Up (permet de devenir fournisseur d'accès à Internet)
- un serveur de partage de connexion ;
- un serveur de sauvegarde, etc..

Pour transformer un serveur Linux en serveur de base de données, il suffit de coupler le logiciel de base de données (comme MySQL) avec le serveur Web Apache via un langage comme PHP. Un simple navigateur Web suffit alors pour accéder à l'application voulue, ce qui permet d'alimenter et de consulter très facilement des bases de données.

2.2. INSTALLATION D'UNE DISTRIBUTION DE LINUX

2.2.1. Première étape : vérifier son matériel

Pour installer Linux, une machine de type PC 386 ou plus dotée de 64Mo de mémoire vive est nécessaire. Il faut réserver un espace disque d'au moins 700 Mo (Sans environnement graphique, on peut se contenter de 32 Mo de mémoire vive et 300 Mo d'espace disque).

Avant d'installer Linux, il est recommandé de connaître les désignations de la carte graphique, de la carte Ethernet ainsi que de la carte son. Si vous avez déjà Windows d'installé sur votre machine, nous vous suggérons de vous rendre sur Panneau de Configuration, Système puis Gestionnaire de périphérique et de noter les références de votre carte graphique, de votre carte Ethernet ainsi que de votre carte son. Veillez également à noter votre adresse IP si vous bénéficier d'une adresse IP fixe, ce qui devrait être le cas si vous souhaitez installer un serveur Internet.

2.2.2. Seconde étape : choisir sa distribution Linux

Quand on parle d'un système Linux, on fait un abus de language. En effet Linux désigne seulement le kernel, une distribution englobe à la fois le kernel et les programmes permettants à l'utilisateur d'interagir avec le kernel. Chaque distribution a donc une certaine liberté dans la façon de présenter les commandes et sur le fonctionnement général du système. En particulier les programmes d'installation et de configuration sont souvent spécifiques à une distribution. Néanmoins un administrateur formé sur une distribution sera à même d'utiliser une autre distribution sans problème majeur.

Les distributions les plus connues sont :

- RedHat (http://www.redhat.com),
- Debian (http://www.debian.org),
- Mandrake (http://www.linux−mandrake.com),

- SuSE,
- Slackware,
- Corel Linux.

La Mandrake bénéficie de paquetages plus à jour et de binaires optimisés pour Pentium (et non pour 386) . Cette distribution est plus spécialement adaptée pour une utilisation bureautique.

La Debian est une distribution réalisée entièrement par des volontaires et ne contient que des logiciels libres. Elle est particulièrement adaptée pour une utilisation sur des serveurs et pour construire des ordinateurs spécialisés (routeur, serveur de fax...) en récupérant des 386 ou des 486. Il est bien sur possible de l'utiliser sur un poste de travail, et il faut noter le nombre impressionant de paquetage disponibles (plus de 2000 paquetages sur 7 CDs).

Remarque : qu'appelle t'on au juste un logiciel libre ?

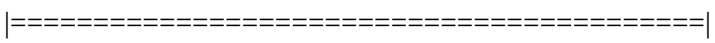
L'expression "Logiciel libre" fait référence à la liberté, et non pas au prix. C'est à dire "la liberté pour les utilisateurs d'exécuter, de copier, de distribuer, d'étudier, de modifier et d'améliorer le logiciel".

Linux est un logiciel libre. Il est donc possible d'acquérir gratuitement Linux mais il également autorisé de vendre un logiciel libre : c'est le cas des CD−Rom de distributions Linux. Rien n'interdit non plus l'acheteur d'une distribution de la redistribuer gratuitement. En bref, la vente des distributions se résume à la vente du support et il est tout à fait possible de télécharger gratuitement Linux sur le site de la distribution choisie.

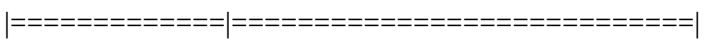
2.2.3. Troisième étape : préparer ses disques durs

On peut installer Linux sur une partition DOS/Windows en utilisant par exemple Linux4win de Mandrake. Cette méthode est pratique car elle permet d'installer Linux sans partitionner son disque dur sur lequel est déjà installé Windows (Linux s'installe dans un unique fichier sur le disque Windows). Cette méthode présente le désavantage de ralentir le temps d'exécution de Linux (au moins d'un facteur 2) et vous risquez donc d'être très déçu par les performances de Linux.

Avant de commencer l'installation, vous allez donc devoir libérer de la place pour les nouvelles partitions Linux. Si vous comptez installer Linux sur un disque dur , pas de problème. Par contre, si vous voulez faire cohabiter Windows et Linux sur le même disque, et que Windows est déjà installé sur la totalité du disque, nous allons devoir passer de cette configuration :



à cette configuration :



Pour libérer de la place, il faut défragmenter au préalable votre disque dur afin que toutes les données soient réunies au début du disque dur (sous Windows : Programme −> Accessoires −> Outil système −> défragmenteur de disque dur). Pour redimensionner votre disque dur, vous pouvez utilisez un logiciel approprié, comme Partition Magique. Nous vous recommandons cependant d'utiliser l'outil DiskDrake de la distribution Mandrake : l'opération de partionnement sera effectué durant l'installation de la Mandrake (cf. étape 7 ci−après).

Une autre possibilité pour repartitionner le disque dur est d'utiliser le programme FIPS qui se trouve dans le répertoire "tools" du premier CD de la distribution Debian.

1 Disquette de boot
sous dos formatter une disquette avec la commande `C:\>FORMAT A: /S` et copier les fichiers FIPS.EXE, ERRORS.TXT et RESTORRB.EXE sur cette disquette

2 Défragmenter le disque dur
Il faut vérifier qu'il n'y pas d'erreurs sur le disque dur avec SCANDIK puis utiliser l'utilitaire de défragmentation du disque dur sous windows, ou utiliser la commande `DEFRAG` sous dos.

3 Exécution de FIPS
Bootez sur la disquette préparée en 1 et à l'invite du dos exécuter la commande `FIPS`

4 Utilisation de FIPS
Après le premier écran tapez n'importe quelle touche. Ensuite la liste des partitions s'affiche. Choisissez la partition que vous voulez réduire.
Ensuite il vous est demandé si vous voulez sauvegarder votre ancienne configuration, répondre :y deux fois
Arrivé à ce point il ne reste plus qu'à utiliser les flèches gauche et droite pour choisir la taille de la nouvelle partition.
Après avoir validé votre choix en appuyant sur la touche "entrée" il faut taper "c" pour continuer et confirmer par "y".

La nouvelle partition est déclarée de type DOS. Lors de l'installation de Linux il faudra supprimer cette partition pour pouvoir créer les partitions Linux.

2.2.4. Installation de la Mandrake 9.0

Pour installer la distribution Mandrake Linux 9.0, vous avez besoin au minimum d'un PC à base de processeur Pentium (ou compatible), d'un lecteur de CD−ROM, de 32 Mo de mémoire au minimum ainsi que d'une carte graphique compatible VESA 2.0. (64 Mo minimum pour une installation graphique).

Pour installer Linux, vous devez démarrer la machine avec une version minimale de Linux. Pour cela trois méthodes sont possibles :

Démarrer l'installation directement depuis le CD−ROM : le CD−ROM étant bootable, si votre Bios le permet vous pouvez booter directement sur le CD d'installation.

Démarrer l'installation à partir de Windows : lors de l'insertion du CD−ROM d'installation depuis Windows, une fenêtre s'ouvre et vous pouvez démarrer l'installation en cliquant sur le bouton "Complete Installation".

Démarrer l'installation à partir d'une disquette de démarrage : sous Windows, lancer le programme rawritewin.exe du répertoire \dosutils du CD−ROM d'installation. Cliquez sur le bouton "…" à coté de "Image file" et sélectionnez le fichier cdrom.img du répertoire \images. Il ne vous reste plus qu'à cliquer sur le bouton "write" pour créer la disquette de boot. Redémarrer ensuite la machine avec la disquette crée.

Une fois l'installation lancée avec la méthode de votre choix, un écran graphique Linux Mandrake apparaît : appuyer sur la touche Entrée pour démarrer l'installation. Au bout de quelques secondes, la procédure d'installation démarre.

1. Choix de la langue : French / France

Vous devez ensuite valider la licence GPL, puis :

2. Classe d'installation : vous avez le choix entre une installation dite recommandée (installation standard), et une installation en mode expert. Si vous avez déjà une version antérieure de la Mandrake, vous pouvez également effectué une mise à jour du système ou uniquement des paquetages. Choisissez impérativement le mode expert afin de rester maître de l'installation (ce qui vous permet de choisir les paquetages à installer et de sélectionner manuellement le modèle de votre carte graphique si sa détection automatique échoue).

3. Détection des disques durs : DrakX, le programme de configuration détecte automatiquement les périphériques PCI et SCSI de votre système. Si certains disques durs n'ont pas été détectés, vous pouvez les sélectionner

manuellement à partir d'une liste.

4. Configuration de la souris : sélectionnez le type de votre souris parmi la liste proposée. Si vous ne savez pas quoi choisir, laisser la sélection proposée par défaut.

5. Choix du clavier : français puis cliquez sur "OK".

6. Sécurité : Laissez les options par défaut. Si vous souhaitez utiliser votre machine en tant que serveur Internet, il est suggéré de choisir comme "niveau de sécurité" le mode "Plus élevé" afin d'obtenir un serveur vraiment sécurisé (dans ce mode, il n'est pas possible entre autre pour des raisons de sécurité de se connecter directement depuis l'extérieur sur la machine avec l'utilisateur root). Cependant si vous débutez il est vivement conseillé de laiser le "niveau de sécurité" proposé par défaut et surtout pas le mode "paranoïaque" qui vous isole du réseau.

7. Système de fichiers : durant cette étape, nous allons créer les partitions nécessaires à Linux à partir de l'utilitaire DiskDrake. Attention à cette étape qui peut s'avérer dangereuse pour vos données si vous partager un même disque dur avec plusieurs systèmes d'exploitation (Windows et Linux par exemple). Il convient donc d'être très vigilant aux choix effectués.

Choisissez alors le disque dur où installer Linux (hda = premier disque dur IDE, hdb = second disque dur IDE, sda = premier disque dur SCSI, sdb = second disque dur SCSI, etc..).

Si vous n'avez pas de disque dur ou de partition disponible pour installer Linux, il vous faut alors redimensionner la partition existante : attention avant de redimensionner la partition Windows, il est nécessaire de fragmenter le disque dur afin que tous les fichiers soient situés au début du disque dur et non en vrac sur l'ensemble du disque dur. Il est également fortement conseillé de sauvegarder ses données au préalable.

Une fois identifié ou créé une partition dédiée à Linux, il vous faut segmenter l'espace disponible pour Linux en trois partitions :

- une partition de type Ext2 (linux native) pour la racine du système avec / comme point de montage (cette partition sert aux fichiers systèmes),
- une partition de type Swap (linux swap) pour la mémoire virtuelle du système (il est recommandé d'allouer une taille équivalente au double de la mémoire vive disponible),
- et enfin une partition de type Ext2 (linux native) pour les répertoires utilisateurs avec /home comme point de montage.

A noter que vous pouvez également utiliser le mode "Partitionnement automatique" si vous préférez laisser au système le choix du paramétrage des partitions.

Si vous avez un quelconque doute durant cette étape et si vous utilisez un même disque dur pour Windows et Linux, il est conseillé de rebooter la machine et d'effectuer une sauvegarde préalable de toutes vos données.

Une fois terminé la configuration des partitions nécessaires à Linux, cliquer sur le bouton "Terminer" pour la création effective des partitions. En fonction des choix effectués, il est possible que le système vous demande alors de redémarrer la machine pour prendre en considération les partitions nouvellement créées (vous devez alors relancer la procédure d'installation à partir du CD−ROM de démarrage, de la disquette de démarrage ou depuis Windows : l'installation redémarre alors et il out faudra à nouveau préciser les choix 1 à 7 et durant l'étape 8 définir les points de montage associées aux 2 partitions non swap c'est à dire / et /home).

8. Formatage des partitions : le programme d'installation vous propose par défaut de formater la partition racine, la partition utilisateurs et la partition de swap nécessaires à Linux. Cliquez sur "OK" pour lancer le formatage de ces partitions. Si vous avez Windows d'installé, ne cliquer surtout pas sur la partition /mnt/windows ce qui aurait pour

effet de supprimer toutes les données de Windows ! La vérification de la présence de blocs endommagés n'est pas indispensable : sélectionner des partitions à vérifier si vous avez des doutes sur l'intégrité de votre disque dur (vous pouvez tout de même vérifier les partitions racine et utilisateurs, ça ne vous coûtera pas plus cher !)

9. Choix des paquetages : durant cette étape, vous pouvez sélectionner les applications que vous souhaitez installer ou non sur votre système. Vous risquez d'être fort impressionné voire désappointé par le grand nombre d'applications disponibles aussi si vous n'avez pas besoin d'applications particulières telles que des logiciels de gravure de CD, vous pouvez vous contenter des choix par défaut.

Pour les besoins serveur, nous vous recommandons de sélectionner les paquetages de la rubique "Serveur".

A noter que si vous utiliser à la fois Windows et Linux sur la même machine, vous pouvez également ajouter Wine de la rubrique Emulators si vous souhaitez exécuter des applications Windows depuis Linux.

A l'issu de la validation des choix, le système copie sur le disque durs tous les paquetages sélectionnées. Cette étape dure une dizaine de minute en fonction du nombre de paquetages sélectionnés.

10. Mot de passe du root : pour d'évidentes raisons de sécurité le mot de passe du root doit être compliqué et il faut à tout pris éviter de se contenter d'un prénom, d'une date de naissance et même de l'assemblage de mots existants dans un dictionnaire (des logiciels sont spécialisés dans la recherche automatique de mots de passe par simple assemblage de mots du dictionnaire). Il est donc conseillé d'utiliser à la fois des caractères en minuscules et en majuscule, des chiffres et des symboles. Un mot de passe tel que KaOuaga32 ! ne pourra pas être déterminé par des programmes de recherche de mot de passe.

11. Ajout des utilisateurs : vous pouvez ajouter des utilisateurs lors de cette étape ou par la suite à tout moment. Nous vous suggérons d'ajouter au moins un utilisateur ne serait−ce que parce que si vous avez opté pour le niveau de sécurité dit "paranoïaque", il ne vous sera pas possible d'effectuer un accès distant sur la machine à partir du compte root (cf. étape 6). Cliquer ensuite sur "Terminer" pour passer à l'étape suivante.

12. Configuration du réseau : avec la version 9.0, cette étape est extrémement simplifié puisqu'il vous suffit de suivre l'assistant automatique et de vérifier que le choix proposé correspond bien à votre configuration !

Configuration du périphérique réseau eth0 : vous devez spécifier l'adresse IP de votre réseau. Etant donnée que vous souhaitez installer un serveur Internet (ou Intranet), vous devriez logiquement avoir une adresse IP fixe. (Vous pouvez sélectionner l'option BOOT/DHCP au lieu de saisir une adresse IP si votre adresse IP n'est pas fixe mais allouée par votre fournisseur d'accès Internet − abonnement au câble ou à l'ADSL par exemple. Toutefois sans adresse IP fixe, il ne vous sera pas réellement possible d'utiliser votre système en tant que serveur Internet).

Vous devez ensuite spécifier le nom de votre machine sous la forme nom−de−la−machine.nom−du−domaine.top−level−domain (par exemple sirius.africacomputing.org pour spécifier que la machine s'appelle sirius et fait partie du domaine africacomputing.org), ainsi qu'éventuellement l'adresse IP de la passerelle si vous utiliser par exemple un routeur pour accéder à l'extérieur, puis du proxy si vous en avez un.

13. Configuration des services : cette étape vous permet de sélectionner les services qui seront lancés automatiquement lors du démarrage du système. Les choix proposés conviennent pour la plupart des besoins. Si vous souhaiter utiliser un serveur LDAP, vous pouvez rajouter ce service dès à présent (mais vous pourrez également l'ajouter à tout moment). De même si vous souhaitez utiliser un serveur DNS, vous pouvez ajouter le service "named" dès à présent.

14. Programmme d'amorçage : si vous utilisez plusieurs systèmes d'exploitation sur la même machine (ce qui est déconseillé pour une utilisation de votre machine en tant que serveur), vous avez la possibilité d'installer un chargeur de démarrage qui vous permettra de choisir au démarrage quel système d'exploitation vous souhaitez lancer. Mandrake vous propose le choix entre Lilo et Grub. Si vous avez plusieurs système d'exploitation nous vous suggérons d'installer Grub (validez alors les options par défaut), sinon cliquez simplement sur "Aucun".

Si vous utilisez Lilo et que vous installé Linux sur le même disque que Windows, spécifiez votre partition root Linux comme support de boot car le Master Boot Record (MBR) de votre disque dur est déjà occupé par celui de Windows.

15. Disquette de démarrage : comme pour l'installation de tout OS, il est fortement recommandé d'effectuer une disquette de démarrage au cas où vous auriez un problème pour démarrer directement à partir de votre disque dur.

16. Configuration X : il s'agit de la configuration de la carte graphique. Nécessaire pour l'exécution de Xfree86, l'implémentation Linux de l'environnement graphique X−Window. Sélectionnez le pilote de votre carte graphique en vous appuyant sur la désignation de votre carte. Puis sélectionnez un pilote pour votre moniteur, puis la résolution graphique (généralement 1024x768 en 4 millions de couleurs) et enfin testez si la configuration fonctionne (si l'écran devient noir et que rien n'apparaît, attendez quelques secondes puis sélectionner une autre carte graphique ou un autre moniteur). Activez enfin le lancement de l'interface graphique au démarrage si vous souhaitez utiliser votre machine en local (c'est à dire autrement qu'à partir d'une connexion distante).

Félicitation, l'installation est maintenant terminée. Retirez le CD−ROM ainsi que l'éventuelle disquette utilisée pour le démarrage de l'installation et rebooter votre machine.

A l'apparition de l'interface graphique X, loguer vous au système Linux avec l'utilisateur root. Lancer ensuite Netscape et saisissez l'adresse http://localhost :10000 Si l'installation s'est déroulée normalement, netscape devrait vous demander votre login et votre mot de passe (utilisez l'utilisateur root) : l'administration graphique de votre serveur à partir de la seule interface Webmin est maintenant possible !

2.2.5. Installation d'une distribution Debian

Voici les menus depuis le démarrage sur le CD1 : 1. Welcome to Debian GNU/Linux 3.0 ! . . .
boot : **F3**
Ceci affiche les différentes saveurs de noyau disponible. Pour le matériel récent et le support de l'ext3, choisir bf24
2. Choose The Language : Fr, **Continue**
3. Choisissez une variété : Français (France)
4. Note sur cette version : **Continuer**
5. Suivant : Configurer le Clavier : Sélectionnez un clavier : azerty/fr−latin1
6. Suivant : Partitionner un disque dur

- Sélectionnez un disque dur : /dev/hda
- Limitations de LILO : **Continuer**
- Note sur l'espace additionnel pour ReiserFs : **Continuer**
- cfdisk : on crée le partitionnement suivant :

Name	Flags	Part Type	FS Type	[Label]	Size(MB)	
hda1	Boot	Primary	Linux		20.48	(cf 20M)
hda2		Primary	Linux Swap		131.61	(cf 128M)
hda3		Primary	Linux		10001.95	(cf 10000M)
hda4		Primary	Linux		29849.55	

Ces valeurs ont titre d'exemples. La première partition de 20Mo est optionnelle, son Boot flag aussi.
Se reporter au manuel complet pour des explications sur le choix des partitions. Pour simplifier, sur une machine de type WorkStation, 3 partitions suffisent : swap (entre 1 à 2 fois la taille de la RAM, type 82 swap linux), / (de 2 `a 5 Go) et /home (le reste du disque).

7. Suivant : Initialiser et activer une partition d'échange

- Sélectionner une partition d'échange : /dev/hda2 : Linux swap

- Faire la recherche des blocs défectueux ? : non
- Ètes−vous sûr ? : oui

8. Suivant : Initialisez une partition Linux.

- Sélectionner une partition : /dev/hda3 : Linux native
- Système de fichiers : ext3
- Faire la recherche des blocs défectueux ? : non −* Ètes−vous sûr ? : oui
- Monter comme système de fichiers racine ? : oui

9. Autre choix : Initialisez une partition Linux.

- Sélectionner une partition : /dev/hda1 : Linux native
- Système de fichiers : ext3
- Faire la recherche des blocs défectueux ? : non
- Ètes−vous sûr ? : oui
- Sélectionnez le point de montage : /boot

10. Autre choix : Initialisez une partition Linux.

- Sélectionner une partition : /dev/hda4 : Linux native
- Système de fichiers : ext3
- Faire la recherche des blocs défectueux ? : non
- Ètes−vous sûr ? : oui
- Sélectionnez le point de montage : /home

La recherche des blocs défectueux peut−être utile, si vous n'avez pas confiance dans le disque, mais, c est une étape très longue. Nous ne le faisons donc pas.

11. Suivant : Installer le noyau et les modules des pilotes CDROM Debian trouvé, **oui**

12. Suivant : Configurer les modules des pilotes matériels

- Notes sur les pilotes chargés : **Continuer**
- On sort de l utilitaire de sélection car aucun module ne nous intéresse !

13. Suivant : Configurer le réseau :

- Choisir un nom de machine : nom_de_ma_machine
- Configuration automatique du réseau : Non
- Adresse IP de la machine : 123.456.789.123
- Netmask : 255.255.255.0
- Gateway : 123.456.789.1 ou laissez le champ vide −* Nom de Domaine : domaine.tld (le .tld est optionnel)
- Adresses des serveurs de noms : 123.456.789.123 (jusqu à 3 adresses) ou laissez le champ vide

14. Suivant : Installer le système de base

- Sélectionner le média d'installation : cdrom
- Veuillez insérer le CD−ROM : **Continuer**
- Sélectionner le chemin de l'archive : /instmnt

15. Suivant : Rendre le système amorçable

- Où faut−il installer le chargeur de démarrage LILO ? : **/dev/hda : Installer dans le MBR**
- Sécurisation de LILO : **Continuer**

16. Créer disquette amorce : non

17. Autre Choix : Réamorcer le système

- Réamorcer le système : (on enlève le CD−ROM ! !) **oui**

Configuration après le Premier Reboot

1. Debian System Configuration

- **Ok** Ce script d'installation peut être relancé à partir de /usr/sbin/base−config
- Time Zone Configuration : Is the hardware clock set to GMT ? : **yes**
- What area do you live in ? : Europe
- Select a city or time zone : Paris

Password setup

- Shall I enable md5 passwords : yes
- Shall I enable shadow passwords : yes
- Enter a password for root : ********
- Re−enter password to verify : *********
- shall I create a normal user account now ? no La création des utilisateurs doit normalement attendre d'avoir fait un /etc/skel/ propre.
- Shall I remove the pcmcia packages ? : yes (sauf portable)
- Do you want to use a ppp connection to install the system ? : no

2. Apt configuration

- Choose the method apt should use to access the Debian archive : cdrom
- Enter CD ROM device file : /dev/cdrom (il faut remettre le CD 1 dans le lecteur, avant de répondre)
- Scan another CD ? : (Pensez à changer le CD avant de répondre) yes (On passe en revue tous les CD de la distribution)
- Add another apt source : no
- Use security updates from security.debian.org ? no (Il faudrait répondre yes mais cela ne fonctionne que si l on dispose d une connection internet à ce niveau de l installation)

3. Run tasksel : no

4. Run dselect ? : no

5. Removing pcmcia ? : yes (sauf portable)

6. Do you want to erase any previously downloaded .deb files ? : yes

7. I can do some automatic configuration of your mail system . . . : Enter

- Enter value : (4) Local delivery only.
- Enter value : none (tous les mails système iront au root)
- Confirm : yes

8. Debian System Configuration : Have fun ! : OK

L'installation Base est terminée. On dispose maintenant d un système Debian GNU/Linux minimum.

3. UTILISATION DE LINUX

3.1. INTRODUCTION

Linux est un système d'exploitation puissant mais son utilisation n'est pas facile pour les débutants non familiarisés avec l'environnement UNIX. L'utilisation de la plupart des applications peut s'effectuer à partir de l'interface graphique X−Window (ou à partir de sur−couches de X−Window telles que les environnements graphiques KDE et GNOME). Cependant pour certains travaux, il est beaucoup plus pratique et plus souple d'utiliser des lignes de commande depuis un environnement shell plutôt que d'utiliser de lourdes solutions graphiques. De plus, si vous devez intervenir sur votre serveur Linux à distance (c'est à dire depuis un poste connecté à Internet), vous allez inévitablement devoir utiliser des lignes de commande.

Qu'appelle t'on un shell ? Un shell est la liaison la plus élémentaire entre l'utilisateur et le système d'exploitation, c'est à dire le programme de gestion de la ligne de commande. Les commandes saisies sont interprétées par le shell et transmises au système d'exploitation.

De nombreuses commandes du shell ressemblent aux commandes MS−DOS : en utilisant la terminologie UNIX, nous pouvons considérer que le programme command.com correspond au shell de MS−DOS. Dans les environnements de type UNIX, il existe plusieurs shells (bash, tcsh, csh, sh, etc..)

3.2. LES COMMANDES DE BASE

Pour toutes les commandes, il est possible d'obtenir de l'aide en tapant `man` suivi du nom de la commande. En tapant une commande suivie du paramètre `−−help`, nous obtenons la liste des paramètres possibles. N'hésitez pas à recourir à la commande `man` ou au paramètre `−−help` dès que vous avez besoin d'aide.

3.2.1. Se déplacer dans les répertoires (cd)

Lorsque vous ouvrez une session Unix avec votre login et votre mot de passe, vous vous retrouvez devant le "prompt" du shell. En fonction du shell employé, le prompt peut avoir la forme suivante :

```
[philippe@sirius essai] $
```

Le mot philippe signifie que vous vous êtes "logué" sur le compte de l'utilisateur philippe, @sirius signifie que vous êtes sur la machine qui porte le nom sirius et essai signifie que vous êtes dans le répertoire essai.

La commande `cd` permet de se déplacer dans les répertoires. La command `ls` permet d'afficher la liste des fichiers d'un répertoire.

Attention, sous les systèmes Unix, un répertoire est désigné par le symbole `/` ou slash et non pas par un \ ou anti−slash comme c'est le cas sous DOS.

Exemples d'utilisation de la commande `cd` :

`$ cd /`	déplacement à la racine du système
`$ cd /essai`	déplacement dans le répertoire essai de la racine
`$ cd essai`	déplacement dans le répertoire courant essai
`$ cd essai/`	déplacement dans le répertoire courant essai
`$ cd /usr/local`	déplacement dans le répertoire apache du répertoire /usr

`$ cd ..`	recule d'une branche vers la racine
`$ cd ~`	déplacement dans son répertoire personnel
`$ cd ~philippe`	déplacement dans le répertoire personnel de l'utilisateur de philipppe

Etant donné que le système mémorise le répertoire courant (répertoire dans lequel on est), on peut utiliser des noms de chemins relatifs :

`$ cd /home/philippe/essai`	chemin absolu
`$ cd essai`	chemin relatif

3.2.2. Lister les fichiers d'un répertoire (ls)

La commande `ls` permet de lister le contenu d'un répertoire.

```
[philippe@sirius essai] $ cd /bin
[philippe@sirius /bin] $ ls
arch        dd          gzip        nisdomainname su
ash         df          hostname    ping          sync
awk         dmesg       kill        ps            tar
cp          fgrep       mount       sh            ypdomain
cpio        gawk        mt          sleep         zcat
csh         grep        mv          sort          zsh
date        gunzip      netstat     stty          ls
```

La commande `ls` sans arguments donne un listing brut difficile à exploiter. Pour obtenir des informations plus précises, il est nécessaire d'utiliser l'argument `−l`.

Exercice : Taper la commande ls avec −l en argument.

Avant de continuer, il est nécessaire de fournir quelques explications sur la gestion des fichiers. Sous Linux, un fichier peut représenter :

- un fichier texte ou un fichier exécutable (on parle alors de fichier binaire) ;
- un répertoire ;
- un périphérique ;
- une référence à un autre fichier (on parle alors de lien).

Linux étant un système multi−utilisateur, les utilisateurs doivent par conséquent être administré. Pour faciliter cette administration, les utilisateurs sont réunis en groupes. Ce qui permet de paramétrer des droits spécifiques à chaque groupe : droits en lecture, mais aussi en écriture et en exécution.

Revenons à l'exemple ci−dessus : les informations fournies sont relativement nombreuses et sont regroupées en colonnes :

- la première colonne fournit des informations sur les droits ;
- la colonne suivante est le nombre de liens pointant sur ce fichier ;
- la suivante donne le nom du propriétaire du fichier.
- la quatrième indique le nom du groupe qui peut accéder au fichier selon les actions autorisées par le

propriétaire ;
- la cinquième nous donne la taille du fichier en octets.
- la sixième donne la date et l'heure de la dernière modification du fichier.
- et enfin vient le nom du fichier. S'il s'agit d'un lien, la référence du fichier est indiqué par un −> [source]

La première colonne constitué de 10 caractères fournit des informations sur le type de fichier et les droits associés. Pour le premier caractère :

- le caractère `−` représente un simple fichier ;
- la lettre `d` représente un dossier (directory) ;
- la lettre `l` représente un lien ;

Il existe d'autres types de fichiers mais nous ne nous en occuperons pas à ce niveau. Les 9 lettres suivantes sont groupées trois par trois et indiquent les droits associés au fichier, c'est à dire par qui et comment un fichier peut−être utilisé :

- le premier triplet correspond aux droits du propriétaire,
- le second triplet correspond aux droits des membres du groupe (un utilisateur intègre un groupe de travail afin de partager des fichiers) ;
- le troisième correspond aux droits des autres utilisateurs du système.

Pour chaque triplet :

- la première lettre indique si le fichier peut être lu ou pas : `r` (readable) si le fichier peut être lu et un tiret sinon ;
- la seconde lettre indique si le fichier peut être écrit ou pas : `w` (writable) ou un tiret ;
- la troisième lettre indique si le fichier peut être exécuté : `x` pour un binaire exécutable ou un tiret.

Ainsi dans l'exemple précédent, le fichier lisezmoi.txt peut être lu et écrit pas son propriétaire, il peut également être lu par les membres de son groupe mais ne peut être modifié. Pour les autres, la lecture et la modification de ce fichier ne sont pas autorisés.

Pour vous aider à retenir l'ordre de présentation des droits (utilisateur / groupe / autres), vous pouvez utiliser l'astuce mnémotechnique suivante : je, nous, ils.

Notons qu'il existe également des fichiers cachés sous Linux : lorsque le nom d'un fichier commence par un point (caractère .), celui−ci n'est visible qu'avec l'option `−a`.

Exercice : Taper ls avec −la en argument depuis votre répertoire personnel.

Les liens ainsi que la modification des droits associés à un fichier sont abordés un peu plus loin.

3.2.3. Retrouver dans quel répertoire je suis (pwd) et créer un répertoire (mkdir)

Lorsque l'on se déplace dans un répertoire, le shell n'affiche que le nom du répertoire dans lequel on se trouve sans préciser le chemin complet. On peut donc très facilement se tromper de répertoire : par exemple penser être dans le répertoire `/bin` alors que l'on se trouve dans le répertoire `/usr/local/bin`. La commande `pwd` permet de connaître le chemin du répertoire dans lequel on se trouve.

```
[philippe@sirius bin]$ pwd
/usr/local/bin
```

Pour créer un répertoire il suffit d'utiliser la commande mkdir avec le nom du répertoire souhaité en paramètre.

Exercice : Créer un répertoire essai dans votre répertoire personnel.

3.2.4. Copier (cp), supprimer (rm), déplacer et renommer un fichier (mv)

La copie de fichier s'effectue avec la commande `cp` (copy). La syntaxe de la commande `cp` est la suivante : `cp source destination` La source et la destination pouvant être un fichier ou un répertoire.

Exemples :

`$ cp lisezmoi.txt essai.txt`	Duplique le fichier lisezmoi.txt en essai.txt
`$ cp lisezmoi.txt essai/`	Copie le fichier lisezmoi.txt dans le répertoire essai
`$ cp essai/ essai2/`	Copie les fichiers du répertoire essai dans le répertoire essai2
`$ cp -R essai/ essai2/`	Copie tous les fichiers du répertoire essai - y compris les sous-répertoires dans le répertoire essai2

La commande `rm` (remove) permet de supprimer un fichier.

`$ rm lisezmoi.txt`	Supprime le fichier lisezmoi.txt

L'option `-R` permet de supprimer récursivement tout le contenu d'un répertoire. Attention, évitez au maximum d'utiliser cette option et surtout ne l'utiliser jamais en tant que root.

La commande `rmdir` (remove directory) permet de supprimer un répertoire.

`$ rmdir essai`	Supprime le répertoire essai

La commande `mv` permet de renommer un fichier.

`$ cp lisezmoi.txt essai.txt`	Duplique le fichier lisezmoi.txt en essai.txt
`$ mv lisezmoi.txt essai2.txt`	Renomme le fichier lisezmoi.txt en essai2.txt

3.2.5. Afficher le contenu d'un fichier (cat et more)

La commande cat permet de visualiser le contenu d'un fichier c'est à dire d'envoyer le contenu du fichier vers une la sortie par défaut : l'écran.

Exercice : Repérer un fichier non exécutable et afficher son contenu (vous pouvez vous rendre dans le répertoire /etc).

La commande `more` permet également de visualiser le contenu d'un fichier. L'affichage s'effectue page par page.

Exercice : Afficher le fichier précédent avec la commande more.

La commande `more` permet également de passer en mode éditeur en tapant `vi` pendant la visualisation du fichier.

3.2.6. Editer un fichier (vi et emacs)

L'éditeur le plus redouté des informaticiens est `vi`. Cet éditeur est l'éditeur élémentaire que l'on retrouve sur la plupart des systèmes d'exploitation et qui n'utilise pas d'interface graphique. Il prend en charge les commandes et les données en même temps. Une fois `vi` lancé, deux modes de fonctionnement se présentent : le mode commandes et le mode édition.

- pour passer du mode édition au mode commande il suffit d'appuyer sur la touche échappement ;
- pour passer du mode commande au mode édition il faut taper la commande d'insertion (ou équivalent)

Une fois lancé : `vi <nom de fichier>` ; vous pouvez employer quelques unes des commandes ci−après les plus courantes :
commandes avec passage en mode insertion

- `i`, `a` : insère une lettre avant ou après le curseur
- `cw` : modifie le mot courant
- `cc` : modifie la ligne courante

commandes sans passage en mode insertion

- `x` , `X` : efface le caractère suivant ou avant les curseur
- `dw` : efface un mot
- `dd` : efface une ligne
- `yw` : copie un mot dans le buffer
- `yy` : copie la ligne courante dans le buffer
- `p` : copie le buffer à la position courante
- `/<mot>` : recherche d'instances de 'mot' dans le fichier ; n pour suivante, N pour précedente
- `:<nombre>` : va a la ligne
- `:q` : quitte le fichier
- `:q!` : quitte le fichier sans sauvegarder
- `:w` : sauvegarde le fichier
- `:x` : sauvegarde et quitte

`emacs` est un autre éditeur standard utilisé dans différents systèmes d'exploitation, il dispose d'un langage qui permet de le personnaliser à souhaits. Il dispose néanmoins de menus 'habituels' tels que la gestion des fichiers, la recherche de caractères, etc.. Ainsi que de règles préprogrammées qui permettent aux développeurs une mise en page dépendante du langage utilisé (C, C++, java ...) reconnaissant les commandes courantes, les chaînes de caractères, etc ..

Exercice : Ajouter les noms des serveurs du réseau local dans le fichier /etc/hosts

3.2.7. Retrouver un fichier (find et which)

Comme l'on s'en doute bien il arrive que l'on ait à retrouver un fichier dont on ne connaît plus l'emplacement ou même le nom ; Linux comprend quelques outils pour ces recherches.

La commande which permet de scruter les répertoires les plus communément utilisés (dont le chemin est indiqué dans la variable d'environnement `PATH`) pour retrouver le nom de fichier indiqué en argument.

Exemple : `which apachectl` recherchera dans tous les répertoires (du PATH) le fichier apachectl

Cette commande est surtout utile pour vérifier que l'on utilise bien la version souhaitée d'un binaire (exécutable). La commande `whereis` est semblable à la commande `which`.

La commande `find` :

Syntaxe : `find <répertoire> <conditions>` Arguments : le répertoire du début de recherche et les conditions sur des attributs du fichier.

Exemple : `find /home/philippe −name \*.txt −size +100k` recherchera dans le sous le répertoire de l'utilisateur Philippe tous les fichiers qui finissent par ".txt" et qui pèsent de plus de 100 kilo−octets.

Le signe `|` appelé pipe (ou tube) permet de relier avantageusement les commandes. L'introduction du pipe (tube) permet de combiner plusieurs commandes parmi lesquelles la commande `grep`.

On peut ainsi combiner la commande `find` avec la commande `grep` afin de retrouver une chaîne de caractère dans un fichier.

Exemple : `find . −name LISEZMOI | xargs grep −n LINUX` recherchera à partir du répertoire courant (noté .) les fichiers de noms LISEZMOI et affichera les lignes numérotées (option −n) contenant la chaîne de caractères Linux.

3.2.8. Trouver du texte dans un fichier (grep)

Syntaxe : `grep [options] <chaîne de caractères> <nom de fichier>`

Exemple : `grep −i −n pacifique *.txt` affichera toutes les lignes numérotées (option −n) contenant le mot pacifique sans prendre en compte les majuscules (option −i) dans les fichiers finissant par .txt

3.2.9. Les liens (ln)

La création de liens symboliques (opposition aux liens physiques) évite la copie de fichiers identiques dans différents répertoires. Par exemple, si une application a besoin d'un fichier volumineux contenant des données relatives à un groupe d'utilisateurs, il est possible de l'avoir virtuellement dans les répertoires courant en créant un lien symbolique : `ln −s <source> <destination>`

Il n'est pas nécessaire que la source existe pour cette création au même titre que sa destruction n'altérera pas le lien mais son appel génèrera un erreur de type fichier introuvable.

3.2.10. Connaître l'espace disque restant (df, du)

Pour contrôler l'espace occupé et l'espace d'un disque dur (en fait d'une partition), il existe deux commandes très utiles.

La commande `df` renseigne sur l'espace disque total, disponible (disk free). Elle s'utilise sur tous répertoires "montés".

Cette commande s'utilise généralement avec en argument le nom d'un fichier pour vérifier le point de montage de son répertoire.

Exemple : `df ~/essai` nous indiquera la partition sur laquelle est sauvegardé le répertoire essai (le ~ représente /home/).

La commande `du` calcule l'espace occupé (disk usage) pour un répertoire (sous entendu le répertoire et ses sous−repértoires).

L'option `−k` permet un affichage en kilo−octets.

Exemple : `du −k −s essai` affichera la liste des sous−répertoires du répertoire essai récursivement sans indiquer tous les fichiers et leur taille (option −s).

3.2.11. Redirections

On notera pour l'occasion, la possibilité de redirection très utile sous linux. Cette technique permet de rediriger la sortie d'une commande ou d'un programme ailleurs que vers l'écran, c'est à dire dans un fichier ou vers un autre programme. Ainsi, nous pouvons envoyer un fichier dans l'entrée d'une commande, mettre l'affichage d'une commande dans un fichier et même envoyer l'affichage d'une commande dans l'entrée d'une autre. Enfin, dans notre cas, nous pouvons envoyer un fichier dans un autre.

Avec le symbole > (signe supérieur), nous pouvons rediriger la sortie d'un programme vers un fichier.

Exemple : `cal > fevrier` Ici, nous envoyons l'affichage de la commande cal − le calendrier pour le mois en cours − dans le fichier nommé fevrier.

Exemple : `cat > essai1.txt` permet de passer dans 'l'editeur' cat que (l'on quitte avec ctrl+d)

Avec le symbole < (signe inférieur), nous redirigerons le contenu d'un fichier vers l'entrée d'une commande.

Exemple : `mail marc < courrier` envoie par courrier électronique à marc le fichier nommé courrier.

>> (supérieur supérieur) : ajoute à la fin Ce symbole permet d'ajouter l'affichage d'une commande à la fin d'un fichier, sans pour autant écraser ce qu'il y avait déjà dans le fichier. Avec un seul supérieur, le contenu du fichier serait remplacé par la sortie de la commande et donc ce contenu aurait été perdu.

Exemple : `cat fichier1 fichier2 fichier3 >> fichier_complet`

Nous ajoutons à la fin (concaténons) du fichier fichier_complet le contenu des fichiers fichier1, fichier2 et fichier3.

Rappel : | (symbole barre), réalise un tube entre deux commandes.

Ce symbole permet de faire en sorte que l'affichage d'une commande soit dirigé dans l'entrée d'une autre commande.

3.2.12. Modification des droits d'accès

La commande `chmod` permet de changer les droits d'accès d'un fichier.

Vous ne pouvez modifier les droits d'un fichier que si vous en êtes le propriétaire. Il existe une exception : l'administrateur système root peut modifier les droits d'accès de tous les fichiers.

Syntaxe : `chmod <modification des droits d'accès> <nom du fichier>`

La partie est elle−même composée de 3 parties :

- les droits d'accès à modifier : `u` (user) pour le propriétaire, `g` (group) pour le groupe, `o` (others) pour les autres et `a` (all) pour tout le monde ;
- les types de modification à effectuer : − pour rajouter de nouveaux droits et + pour supprimer des droits d'accès déjà mis en place ;
- la modification individuelle des droits : `r`,`w` ou `x`

Exemples :

`$ chmod a+r lisezmoi.txt`	Autorise à tout le monde la lecture de lisezmoi.txt
`$ chmod ug+rw,o+r lisezmoi.txt`	Le propriétaire et les membres de son groupe peuvent lire et modifier le fichier lisezmoi.txt alors que les autres personnes ne peuvent que le lire.
`$ chmod a+x mon−script`	Permet de spécifier que le fichier mon−script peut être exécuté

La commande `chown` permet de changer le propriétaire d'un fichier. Syntaxe : `chown utilisateur fichier`

La commande `chgrp` permet de changer le groupe d'un fichier (il faut que le propriétaire appartienne au groupe). Syntaxe : `chgrp groupe fichier`

3.3. ARCHIVER, COMPRESSER ET DÉCOMPRESSER

Archivage de fichiers :

Pour archiver des fichiers, on assemble le groupe de fichiers à archiver :

```
tar <destination> <sources>
```

Assemblage des différents fichiers (fichier i) dans monfichier :

```
$ tar −cf monfichier.tar fichier1 fichier2 ... fichiern
```

Pour assembler en récursif (avec les sous−répertoire) des répertoire :

```
$ tar −cf monfichier.tar rep1 rep2 ... repn
```

Désassemblage :

```
$ tar −xf <monfichier.tar>
```

Compression d'un fichier :

Une commande de compression permettra ensuite de diminuer la taille totale de ces fichiers assemblés : `gzip`

Compresse monfichier et le remplace par le fichier monfichier.gz : `gzip monfichier`

Pour décompresser un fichier archive essayer la commande suivante : `gzip −d fichier.gz`

Ainsi l'on peut assembler et compresser les fichiers à archiver.

Remarque : la commande `tar xvfz` permet de décompresser en même temps que le désassemblage.

3.4. ECRIRE DES SCRIPTS

Ecrit par Frédéric Bonnaud.

Vous aurez envie d'écrire un script (petit programme écrit avec un langage simple : shell, perl ou autre) dès que vous aurez tapez dans un terminal quatre fois la même série de commandes et que vous vous apercevrez que vous êtes

ammené à le refaire de nombreuses fois. Un script est une suite d'instruction élémentaire qui sont éxécutées de façon séquencielle (les unes après les autres) par le langage de script Dans cet article nous nous limiterons à l'utilisation du shell comme langage, et en particulier à bash. En guise de première introduction, vous pouvez lire ce qui concerne les commandes du shell dans l'article Le Shell et les Commandes. Attention, n'espérez pas que ce document constitue un manuel complet de programmation ! C'est une courte introduction qui nous l'espérons, vous permettra d'écrire de petits scripts qui vous rendrons de précieux services.

Notions de base

Pour commancer, il faut savoir qu'un script est un fichier texte standard pouvant être créer par n'importe quel éditeur : vi, emacs ou autre. D'autre part, conventionnellement un script commance par une ligne de commentaire contenant le nom du langage à utiliser pour interpréter ce script, pour le cas qui nous intéresse : /bin/sh. Donc un script élémentaire pourrait être :

```
#!/bin/sh
```

Évidemment un tel script ne fait rien ! Changeons cela. La commande qui affiche quelque chose à l'écran est echo. Donc pour créer le script bonjour_monde nous pouvons écrire :

```
#!/bin/sh
echo "Bonjour, Monde !"
echo "un premier script est né."
```

Comment on l'éxécute ? C'est simple il suffit de faire :

```
[user@becane user]$ sh bonjour_monde
Bonjour, Monde !
un premier script est né.
[user@becane user]$ _
```

C'est pas cool, vous préféreriez taper quelque chose comme :

```
[user@becane user]$ ./bonjour_monde
Bonjour, Monde !
un premier script est né.
[user@becane user]$ _
```

C'est possible si vous avez au préalable rendu votre script éxécutable par la commande :

```
[user@becane user]$ chmod +x bonjour_monde
[user@becane user]$ ./bonjour_monde
Bonjour, Monde !
un premier script est né.
[user@becane user]$ _
```

Résumons : un script shell commence par : `#!/bin/sh`, il contient des commandes du shell et est rendu éxécutable par `chmod +x`.

Quelques conseils concernant les commentaires Dans un shell−script, est considéré comme un commentaire tout ce qui suit le caractère # et ce, jusqu'a la fin de la ligne.
Usez et abusez des commentaires : lorsque vous relirez un script 6 mois après l'avoir écrit, vous serez bien content de l'avoir documenté. Un programme n'est jamais trop documenté. Par contre, il peut être mal documenté ! Un commentaire est bon lorsqu'il décrit pourquoi on fait quelque chose, pas quand il décrit ce que l'on fait.

Exemple :

```
#!/bin/sh
# pour i parcourant tous les fichiers,
for i in * ; do
# copier le fichier vers .bak
   cp $i $i.bak
# fin pour
done
```

Que fait le script ? Les commentaires ne l'expliquent pas ! Ce sont de mauvais commentaire. Par contre :

```
#!/bin/sh
# on veut faire un copie de tous les fichiers
for i in * ; do
# sous le nom $i.bak
  cp $i $i.bak
done
```

Là, au moins, on sait ce qu'il se passe. (Il n'est pas encore important de connaitre les commandes de ces deux fichiers)

Le passage de paramètres

Un script ne sera, en général, que d'une utilisation marginale si vous ne pouvez pas modifier son comportement d'une manière ou d'une autre. On obtient cet effet en "passant" un (ou plusieurs) paramètre(s) au script. Voyons comment faire cela. Soit le script essai01 :

```
#!/bin/sh
echo le paramètre \$1 est \"$1\"
echo le paramètre \$2 est \"$2\"
echo le paramètre \$3 est \"$3\"
```

Que fait−il ? Il affiche, les uns après les autres les trois premiers paramètres du script, donc si l'on tappe :

```
$ ./essai01 paramètre un
le paramètre $1 est "paramètre"
le paramètre $2 est "un"
le paramètre $3 est ""
```

Donc, les variables $1, $2 ... $9 contiennent les "mots" numéro 1, 2 ... 9 de la ligne de commande. Attention : par "mot" on entend ensemble de caractères ne contenant pas de caractères de séparations. Les caractères de séparation sont l'espace, la tabulation, le point virgule.

Vous avez sans doute remarqué que j'ai utilisé les caractères : \$ à la place de $ ainsi que \" à la place de " dans le script. Pour quelle raison ? La raison est simple, si l'on tape : echo "essai" on obtient : essai, si l'on veut obtenir "essai" il faut dire à echo que le caractère " n'indique pas le début d'une chaine de caractère (comme c'est le comportement par défaut) mais que ce caractère fait partie de la chaine : on dit que l'on "échappe" le caractère " en tapant \". En "échappant" le caractère \ (par \\) on obtient le caractère \ sans signification particulière. On peut dire que le caractère \ devant un autre lui fait perdre sa signification particulière s'il en a une, ne fait rien si le caractère qui suit \ n'en a pas.

Maintenant, essayons de taper :

```
$ ./essai01 *
le paramètre $1 est "Mail"
le paramètre $2 est "essai01"
le paramètre $3 est "nsmail"
$ _
```

(Le résultat doit être sensiblement différent sur votre machine). Que c'est−il passé ? Le shell a remplacé le caractère * par la liste de tous les fichiers non cachés présents dans le répertoire actif. En fait, toutes les substitutions du shell sont possible ! C'est le shell qui "substitue" aux paramètres des valeurs étendues par les caractères * (toute suite de caractères) et [ab] (l'un des caractères a ou b). Autre exemple :

```
$ ./essai01 \*
le paramètre $1 est "*"
le paramètre $2 est ""
le paramètre $3 est ""
$ _
```

Et oui, on a "échappé" le caractère * donc il a perdu sa signification particulière : il est redevenu un simple *.

C'est bien, me direz vous, mais si je veux utiliser plus de dix paramètres ? Il faut utiliser la commande shift, à titre d"exmple voici le script essai02 :

```
#!/bin/sh
echo le paramètre 1 est \"$1\"
shift
echo le paramètre 2 est \"$1\"
shift
echo le paramètre 2 est \"$1\"
shift
echo le paramètre 4 est \"$1\"
shift
echo le paramètre 5 est \"$1\"
shift
echo le paramètre 6 est \"$1\"
shift
echo le paramètre 7 est \"$1\"
shift
echo le paramètre 8 est \"$1\"
shift
echo le paramètre 9 est \"$1\"
shift
echo le paramètre 10 est \"$1\"
shift
echo le paramètre 11 est \"$1\" Si vous tapez :
$ ./essai02 1 2 3 4 5 6 7 8 9 10 11 12 13
le paramètre 1 est "1"
le paramètre 2 est "2"
le paramètre 2 est "3"
le paramètre 4 est "4"
le paramètre 5 est "5"
le paramètre 6 est "6"
le paramètre 7 est "7"
le paramètre 8 est "8"
le paramètre 9 est "9"
```

```
le paramètre 10 est "10"
le paramètre 11 est "11"
$ _
```

A chaque appel de shift les paramètres sont déplacés d'un numéro : le paramètre un devient le paramètre deux et c. Évidemment le paramètre un est perdu par l'appel de shift : vous devez donc vous en servir avant d'appeler shift.

Les variables

Le passage des paramètres nous à montrer l'utilisation de "nom" particuliers : $1, $2 etc. ce sont les substitutions des variables 1, 2 et c. par leur valeurs. Mais vous pouvez définir et utiliser n'importe quelle nom. Attention toute fois, à ne pas confondre le nom d'une variable (notée par exemple machin) et son contenue (notée dans cas $machin). Vous connaissez la variable PATH (attention le shell différencie les majuscules des minuscules) qui contient la liste des répertoires (séparés par des " :") dans lesquels il doit rechercher les programmes. Si dans un script vous tapez :

```
1:#!/bin/sh
2:PATH=/bin           # PATH contient /bin
3:PATH=PATH:/usr/bin  # PATH contient PATH:/bin
4:PATH=/bin           # PATH contient /bin
5:PATH=$PATH:/usr/bin # PATH contient /bin:/usr/bin
```

(Les numéros ne sont là que pour repérer les lignes, il ne faut pas les taper).

La ligne 3 est très certainement une erreur, à gauche du signe "=" il faut une variable (donc un nom sans $) mais à droite de ce même signe il faut une valeur, et la valeur que l'on a mis est "PATH :/usr/bin" : il n'y a aucune substitution à faire. Par contre la ligne 5 est certainement correcte : à droite du "=" on a mis "$PATH :/usr/bin", la valeur de $PATH étant "/bin", la valeur après substitution par le shell de "$PATH :/usr/bin" est "/bin :/usr/bin". Donc, à la fin de la ligne 5, la valeur de la variable PATH est "/bin :/usr/bin".

Attention : il ne doit y avoir aucun espace de part et d'autre du signe "=".

Résumons : MACHIN est un nom de variable que l'on utilise lorsque l'on a besoin d'un nom de de variable (mais pas de son contenu) et $MACHIN est le contenu de la variable MACHIN que l'on utilise lorsque l'on a besoin du contenu de cette variable. Variables particulières Il y a un certain nombre de variables particulières, voici leur signification :

- la variable * (dont le contenu est $*) contient l'ensemble de tous les "mots" qui on été passé au script.
- la variable # contient le nombre de paramètres qui ont été passés au programme.
- la variable 0 (zéro) contient le nom du script (ou du lien si le script a été appelé depuis un lien).

Il y en a d'autres moins utilisées : allez voir la man page de bash.

Arithmétique

Vous vous doutez bien qu'il est possible de faire des calculs avec le shell. En fait, le shell ne "sait" faire que des calculs sur les nombres entiers (ceux qui n'ont pas de virgules ;−). Pour faire un calcul il faut encadrer celui−ci de : $((un calcul)) ou $[un calcul]. Exemple, le script essai03 :

```
#!/bin/sh
echo 2+3*5 = $((2+3*5))
MACHIN=12
echo MACHIN*4 = $[$MACHIN*4] Affichera :
$ sh essai03
2+3*5 = 17
```

```
MACHIN*4 = 48
```

Vous remarquerez que le shell respecte les priorités mathématiques habituelles (il fait les multiplications avant les additions !). L'opérateur puissance est "**" (ie : 2 puissance 5 s'écrit : 2**5). On peut utiliser des paranthèses pour modifier l'ordre des calculs.

Les instructions de controles de scripts

Les instructions de controles du shell permettent de modifier l'éxécution purement séquencielle d'un script. Jusqu'à maintenant, les scripts que nous avons créé n'étaient pas très complexe. Ils ne pouvaient de toute façon pas l'être car nous ne pouvions pas modifier l'ordre des instructions, ni en répéter.

L'éxécution conditionnelle

Lorsque vous programmerez des scripts, vous voudrez que vos scripts fassent une chose si une certaine condition est remplie et autre chose si elle ne l'est pas. La construction de bash qui permet cela est le fameux test : if then else fi. Sa syntaxe est la suivante (ce qui est en italique est optionnel) :

```
if <test> ; then
   <instruction 1>
   <instruction 2>
   ...
   <instruction n>
else
   <instruction n+1>
   ...
   <instruction n+p>
fi
```

Il faut savoir que tout les programmes renvoie une valeur. Cette valeur est stockée dans la variable ? dont la valeur est, rapelons le : $?. Pour le shell une valeur nulle est synonyme de VRAI et une valeur non nulle est synonyme de FAUX (ceci parce que, en général les programmes renvoie zéro quand tout c'est bien passé et un numéro non nul d'erreur quand il s'en est produit une).

Il existe deux programmes particulier : false et true. true renvoie toujours 0 et false renvoie toujours 1. Sachant cela, voyons ce que fait le programme suivant :

```
#!/bin/sh
if true ; then
   echo Le premier test est VRAI($?)
else
   echo Le premier test est FAUX($?)
fi

if false ; then
   echo Le second test est VRAI($?)
else
   echo Le second test est FAUX($?)
fi
```

Affichera :

```
$ ./test
Le premier test est VRAI(0)
```

```
Le second test est FAUX(1)
$ _
```

On peut donc conclure que l'instruction if ... then ... else ... fi, fonctionne de la manière suivante : si (if en anglais) le test est VRAI(0) alors (then en anglais) le bloque d'instructions comprises entre le then et le else (ou le fi en l'absence de else) est éxécuté, sinon (else en anglais) le test est FAUX(différent de 0)) et on éxécute le bloque d'instructions comprises entre le else et le fi si ce bloque existe.

Bon, évidemment, des tests de cet ordre ne paraissent pas très utiles. Voyons de vrais tests maintenant.

Les tests

Un test, nous l'avons vu, n'est rien de plus qu'une commande standard. Une des commandes standard est 'test', sa syntaxe est un peu complexe, je vais la décrire avec des exemples.

- si l'on veut tester l'existance d'un répertoire , on tapera : test −d
- si l'on veut tester l''existance d'un fichier , on tapera : test −f
- si l'on veut tester l'existande d'un fichier ou répertoire , on tapera : test −e

Pour plus d'information faites : man test.

On peut aussi combiner deux tests par des opérations logiques : ou correspond à −o, et correspond à −a (à nouveau allez voir la man page), exemple : `test −x /bin/sh −a −d /etc` Cette instruction teste l'existance de l'éxécutable /bin/sh (−x /bin/sh) et (−a) la présence d'un répertoire /etc (−d /etc).

On peut remplacer la commande test par [ ] qui est plus lisible, exemple :

```
if [ −x /bin/sh ] ; then
   echo /bin/sh est executable. C\'est bien.
else
   echo /bin/sh n\'est pas executable.
   echo Votre système n\'est pas normal.
fi
```

Mais il n'y a pas que la commande test qui peut être employée. Par exemple, la commande grep renvoie 0 quand la recherche a réussi et 1 quand la recherche a échoué, exemple :

```
if grep −E "^frederic:" /etc/passwd > /dev/null ; then
   echo L\'utilisateur frederic existe.
else
   echo L'utilisateur frederic n\'existe pas.
fi
```

Cette série d'instruction teste la présence de l'utilisateur frederic dans le fichier /etc/passwd. Vous remarquerez que l'on a fait suivre la commande grep d'une redirection vers /dev/null pour que le résutat de cette commande ne soit pas affichée : c'est une utilisation classique. Ceci explique aussi l'expression : "Ils sont tellement intéressant tes mails que je les envoie à /dev/null" ;−).

Faire quelque chose de différent suivant la valeur d'une variable

Faire la même chose pour tous les éléments d'une liste Lorsque l'on programme, on est souvant ammené à faire la même pour divers éléments d'une liste. Dans un shell script, il est bien évidemment possible de ne pas réécrire dix fois la même chose. On dira que l'on fait une boucle. L'instruction qui réalise une boucle est

```
for <variable> in <liste de valeurs pour la variable> ; do
   <instruction 1>
   ...
   <instruction n>
done
```

Voyons comment ça fonctionne. Supposons que nous souhaitions renommer tous nos fichiers *.tar.gz en *.tar.gz.old, nous taperons le script suivant :

```
#!/bin/sh
# I prend chacune des valeurs possibles correspondant
# au motif : *.tar.gz
for I in *.tar.gz ; do
   # tous les fichier $I sont renommé $I.old
   echo "$I -> $I.old"
   mv $I $I.old
# on fini notre boucle
done
```

Simple, non ? Un exemple plus complexe ? Supposons que nous voulions parcourir tous les répertoires du répertoire courant pour faire cette même manipulation. Nous pourrons taper :

```
1:#!/bin/sh
2:for REP in `find -type d` ; do
3:    for FICH in $REP/*.tar.gz ; do
4:        if [ -f $FICH ] ; then
5:            mv $FICH $FICH.old
6:        else
7:            echo On ne renomme pas $FICH car ce n\'est pas un
répertoire
8:        fi
9:    done
10:done
```

Explications : dans le premier for, on a précisé comme liste : `find -type d` (attention au sens des apostrophes, sur un clavier azerty français on obtient ce symbole en appyant sur ALTGR+é). Lorsque l'on tape une commande entre apostrophes inverses, le shell éxécute d'abord cette commande, et remplace l'expression entre apostrophe inverse par la sortie standard de cette commande. Donc, dans le cas qui nous intéresse, la liste est le résultat de la commande find -type d, c'est à dire la liste de tous les sous répertoires du répertoire courant. Donc en ligne 2 on fait prendre à la variable REP le nom de chacun des sous répertoires du répertoire courant, puis (en ligne 3) on fait prendre à la variable FICH le nom de chacun des fichiers .tat.gz de $REP (un des sous répertoires), puis si $FICH est un fichier on le renomme, sinon on afficher un avertissement.

Remarque : ce n'est pas le même fonctionnement que la boucle for d'autres langage (le pascal, le C ou le basic par exemple).

Faire une même chose tant qu'un certaine condition est remplie.

Pour faire une certaine chose tant qu'une condition est remplie, on utilise un autre type de boucle :

```
while <un test> ; do
   <instruction 1>
   ...
   <instruction n>
```

```
done
```

Supposons, par exemple que vous souhaitiez afficher les 100 premiers nombres (pour une obscure raison), alors vous taperez :

```
i=0
while [ i −lt 100 ] ; do
   echo $i
   i=$[$i+1]
done
```

Remarque : −lt signifie "lesser than" ou "plus petit que".

Ici, on va afficher le contenu de i et lui ajouter 1 tant que i sera (−lt) plus petit que 100. Remarquez que 100 ne s'affiche pas.

Refaire à un autre endroi la même chose

Souvant, vous voudrez refaire ce que vous venez de tapez autre part dans votre script. Dans ce cas il est inutile de retaper la même chose, préférez utiliser l'instruction function qui permet de réutiliser une portion de script. Voyons un exemple :

```
#!/bin/sh
function addpath ()
{
   if echo $PATH | grep −v $1 >/dev/null; then
       PATH=$PATH:$1;
   fi;
   PATH=`echo $PATH|sed s/::/:/g`
}

addpath /opt/apps/bin
addpath /opt/office52/program
addpath /opt/gnome/bin
```

Au debut, nous avons défini une fonciton nommée addpath dont le but est d'ajouter le premier argument ($1) de la fonction addpath à la varaible PATH si ce premier argument n'est pas déjà présent (grep −v $1) dans la variable PATH, ainsi que supprimer les chemins vide (sed s/ : :/ :/g) de PATH.
Ensuite, nous éxécuton cette fonction pour trois arguments : /opt/apps/bin, /opt/office52/bin et /opt/gnome/bin.

En fait, une fonction est seulement un script écrit à l'intérieur d'un script. Elles permettent surtout de ne pas multiplier les petits scripts, ainsi que de partager des variables sans se préocuper de la clause export mais cela constitue une utilisation avancée du shell, nous ne nous en occuperons pas dans cet article.

Remarque : le mot function peut être omis.

Autres types de répétitions

Il existe d'autres type de répétitions, mais nous ne nous en occuperons pas dans cette article, je vous conseille la lecture forcément profitable de la man page de bash.

4. ADMINISTRATION LINUX

4.1. RÔLE DE L'ADMINISTRATEUR SYSTÈME

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 :

- configurer le noyau du système d'exploitation ;
- sauvegarder les données et réparer les systèmes de fichier ;
- gérer les utilisateurs ;
- installer de nouveaux logiciels ;
- intégrer de nouveaux disques durs et de nouvelles partitions ;
- configurer le processus de démarrage de Linux ;
- configurer le réseau

Du fait que le super−utilisateur root possède tous les droits, il doit posséder des connaissances concernant le fonctionnement du système.

4.2. PRINCIPAUX RÉPERTOIRES SYSTÈMES

Répertoires standards :

`/`	Répertoire racine (ou root) contenant tous les répertoires.
`/home`	Répertoire contenant les répertoires personnels de tous les utilisateurs autres que root.
`/root`	Répertoire personnel de l'administrateur système root.

Répertoires système :

`/bin`	Répertoire contenant les commandes et utilitaires employés par tous les utilisateurs (ls, rm, cp, etc..)
`/boot`	Répertoire contenant des informations permettant le chargement de Linux.
`/dev`	Répertoire contenant tous les fichiers périphériques permettant d'accéder aux composants matériels.
`/etc`	Répertoire contenant les commandes et fichiers de paramétrages nécessaires à l'administration système.
`/lib`	Répertoire contenant les bibliothèques communes à tous les utilisateurs
`/proc`	Répertoire spécial utilisé par le système et contenant la liste des processus en cours d'exécution.
`/sbin`	

	Répertoire contenant les commandes et utilitaires utilisées seulement par l'administrateur système.
/tmp	Répertoire contenant les fichiers temporaires.
/usr	Répertoire composé d'un certain nombre de sous répertoires utilisés par l'ensemble des utilisateurs.
/var	Répertoire spécial utilisé par le système pour stocker des données souvent modifiées.

4.3. GESTION DES UTILISATEURS

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 utilisateur afin d'éviter d'effectuer de fausses manips 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
$ whoami
root
$ exit
$ whoami
pdrouot
```

Chaque utilisateur dispose d'un répertoire personnel sous `/home`, par exemple `/home/philippe` pour l'utilisateur philippe. Outres 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.

4.3.1. Principe de l'ajout des utilisateurs

L'ajout d'un utilisateur consiste à :

- associer un mot de passe à l'utilisateur (ajout d'une entrée dans le fichier `/etc/passwd`) ;
- définir à quel groupe appartient l'utilisateur (ajout d'une entrée dans le fichier `/etc/group`) ;
- créer le répertoire personnel de l'utilisateur ;
- créer le fichier de configuration personnel du shell ;

Une entrée (c'est à dire une ligne) du fichier `/etc/passwd` est de la forme :

Nom :mot de passe :numéro d'utilisateur :numéro de groupe :champs spécial :répertoire personnel :shell de démarrage

Exemple d'entrée du fichier /etc/passwd :

```
$ cat /etc/passwd | grep philippe
phlippe:x:501:100:Philippe Drouot:/home/philippe:/bin/bash
```

Une entrée (c'est à dire une ligne) du fichier `/etc/group` est de la forme :

Nom de groupe : champs spécial : numéro de groupe : membre1 , membre2, etc..

Exemple d'entrée du fichier /etc/group :

```
$ cat /etc/group | grep 100
users:x:100:
```

Pour ajouter un utilisateur philippe, vous devez :

1. ajouter l'utilisateur philippe dans le fichier `/etc/passwd`
2. ajouter éventuellement un nouveau groupe dans `/etc/group` (si vous souhaitez créer un groupe spécifique pour philippe)
3. créer le répertoire personnel du nouvel utilisateur (home directory), copier les fichiers de configuration du shell et changer les droits du répertoire philippe afin que l'utilisateur philippe devienne propriétaire de son répertoire personnel :

```
$ mkdir /home/philippe
$ cp /etc/skel/* /home/philippe
$ chown philippe /home/philippe
$ chgrp le_groupe_de_philippe /home/philippe
```

4. donner un mot de passe à l'utilisateur philippe par la commande :

```
$ passwd philippe
```

4.3.2. Ajout d'utilisateur et de groupe avec les commandes useradd et groupadd

Pour faciliter l'ajout d'utilisateurs et de groupe, il existe des commandes spécifiques.

Ajout d'un utilisateur avec la commande useradd :

Syntaxe : `useradd nom-utilisateur -g groupe -d répertoire-personnel -m`
L'option `-m` permet de recopier les fichiers de configuration du shell. On peut remplacer le shell courant par un shell spécifique avec l'option `-s` (par exemple `-s /etc/ftponly`).

```
$ useradd philippe -g users -d /home/philippe -m
$ passwd philippe
Changing password for user philippe
New UNIX password :
Retype new UNIX password :
Passwd : all authentification tokens update successfully
```

Suppression d'un utilisateur avec la commande userdel :

```
$ userdel -r philippe
```

L'option `-r` permet de supprimer le répertoire personnel de l'utilisateur à supprimer.

Ajout d'un groupe avec la commande groupadd :

```
$ groupadd ftpusers
```

Suppression d'un groupe avec la commande groupdel :

```
$ groupdel ftpusers
```

Gestion graphique des utilisateurs et des groupes :

Vous pouvez également ajouter un utilisateur graphiquement à partir de l'utilitaire de configuration DrakeConf fourni avec la distribution ou Mandrake (l'ajout d'un utilisateur s'accompagne de l'ajout d'un groupe du même nom) ou à partir de l'utilitaire linuxconf disponible sur toutes les distributions linux (vous pouvez alors choisir le groupe d'appartenance de l'utilisateur).

Comme pour la plupart des tâches d'administration système, vous pouvez gérer les utilisateurs et les groupes très facilement avec Webmin :

Ajout d'un utilisateur sous 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.

Astuces :

- Si vous êtes amenés à ajouter régulièrement des utilisateurs (pour ajouter par exemple des comptes ftp ou des comptes mails), vous avez intérêt à créer un "utilisateur Webmin" spécialement dédié à l'ajout de comptes utilisateurs et pour lesquels les paramètres par défauts sont déjà pré−réglés (groupe utilisateur et emplacement du répertoire personnel associé au compte).

- Pour des raisons de sécurités, il est important d'interdire l'accès à la machine aux utilisateurs Internet qui n'ont pas d'autres besoins que d'effectuer du ftp ou de consulter des mails. Pour cela l'interpréteur de commande (shell) des utilisateurs doit être remplacé par `/etc/ftponly` (et non par exemple /bin/sh qui autoriserait un accès telnet à la machine).

Exemple de fichier /etc/ftponly :

```
# !/bin/sh
#
# ftponly shell
#
trap "/bin/echo Sorry ; exit 0" 1 2 3 4 5 6 7 10 15
#
Admin=root@your−domain.com
#
/bin/echo
/      b      i      n      /      e      c      h      o
"*********************************************************"
/bin/echo " You are NOT allowed interactive access."
/bin/echo
/bin/echo " User accounts are restricted to ftp and web
access."
/bin/echo
/bin/echo " Direct questions concerning this policy to
$Admin."
/      b      i      n      /      e      c      h      o
"*********************************************************"
/bin/echo
#
#
```

```
exit 0
```

De même pour un utilisateur qui n'a besoin que d'accéder à sa boîte aux lettres, la seule commande que nous lui autorisons consiste à changer son mot de passe. Le shell d'un tel utilisateur est : `/bin/passwd`

Attention, pour que les shells `/etc/ftponly` et `/bin/passwd` soient autorisés par le système, il faut les ajouter dans le fichier `/etc/shell`.

4.4. GESTION DES PROCESSUS

Un processus est un programme chargé en mémoire et en cours d'exécution. Contrairement à Windows 98, les systèmes UNIX sont des systèmes multitâches préemptifs, c'est à dire que chaque programme ou processus tournent indépendamment. Lorsqu'un processus est planté, le système continue à tourner car les processus sont traités indépendamment. La destruction d'un processus n'a pas d'effet sur l'exécution des autres processus.

Informations associées à chaque processus :

Pour chaque processus exécuté, le système d'exploitation stocke un certain nombres d'informations :

- Numéro unique du processus PID (Process IDentification) ;
- Numéro du processus parent PPID (Parent Process Identification) ;
- Numéro d'utilisateur PID (User IDentification) ayant lancé le processus ;
- Numéro du groupe GID (Group IDentification) ayant lancé le processus ;
- Durée de traitement utilisé (temps CPU) et priorité du processus ;
- Référence au répertoire de travail courant du processus ;
- Table de référence des fichiers ouverts par le processus.

Chaque processus peut créer lui−même des processus d'où la notion de processus parent. C'est la cas par exemple du serveur Apache : lors de son lancement, le processus père crée plusieurs processus fils afin de répondre indépendamment à plusieurs clients. La destruction du processus parent (parent process) entraîne la destruction de tous les processus fils (child process).

Afficher les processus avec la commande top :

La commande `top` permet d'afficher des informations en continu sur l'activité du système (quantité de RAM et pourcentage de CPU utilisés par les processus). Pour quitter la commande top, il suffit de taper la lettre p.

Commande ps :

Il s'agit de la commande la plus employée pour obtenir des informations sur les processus exécutés par le système. Cette commande permet de connaître les processus actifs à un moment donné.

```
$ ps
PID TTY TIME CMD
26687 pts/2 00:00:00 bash
26797 pts/2 00:00:00 ps
```

La commande `ps` sans arguments ne fournit que la liste des processus associés au terminal utilisé. Pour connaître tous les processus exécuté par le système, il est nécessaire d'ajouter l'argument `−aux`

```
$ ps
```

	affiche tous les processus lancés par l'utilisateur dans le terminal
`$ ps −x`	affiche tous les processus lancés par l'utilisateur
`$ ps −aux`	affiche tous les processus lancés par tous les utilisateurs
`$ ps −aux | grep httpd`	affiche tous les processus nommés httpd lancés

Commande pstree :

La commande `pstree` permet d'afficher les processus sous forme d'arborescence : pratique pour voir les dépendances entre processus.

Tuer un processus avec la commande kill :

La commande `kill` permet d'envoyer un signal à un processus en cours.

Syntaxe : `kill −Numéro−de−signal PID`

Pour tuer un processus, c'est à dire obliger l'application à se terminer, nous utilisons le signal de numéro 9 (SIGKILL) qui oblige le processus à se terminer : cette option permet de tuer le processus quel que soit son état et même si celui−ci est planté ou instable.

Exemple : lancement de la commande `top`, retour au shell avec les touches Ctrl + Z, recherche du processus correspondant à la commande top et destruction de celui−ci.

```
$ top
(puis appuyez sur les touches Ctrl et Z)
$ ps −aux | grep top
pdrouot 27060 1.9 0.8 1736 1060 pts/1 T 09:02 0:00 top
pdrouot 27064 0.0 0.3 1332 512 pts/1 S 09:02 0:00 grep top
$ kill −9 27060
```

Opérateur : l'opérateur & permet de lancer plusieurs processus en parallèle alors que l'opérateur ; lance les processus en série.

Exercice : comparer la commande netscape &netscape &netscape &avec la commande netscape ; netscape ; netscape ; netscape

Gestion des processus sous Webmin

Rubrique "Système" puis "Gestionnaire de processus" Très pratique car cette fonction permet de connaître les processus qui tournent ainsi que leur occupation mémoire et CPU depuis votre machine ou encore depuis n'importe quel poste connecté à Internet ce qui permet de vérifier si la machine est surchargée ou non sans avoir à être physiquement devant. Pour afficher les processus parents et leurs enfants, cliquez sur "PID".
Pour classer les processus en fonction de leur occupation mémoire ou CPU, cliquez sur "Mémoire" ou sur "Processeur".
Pour tuer un processus, cliquez sur le numéro du processus, puis cliquez sur "Envoyer le signal" avec "TERM" (terminer) en paramètre.

4.5. MONTAGE DE DISQUES

4.5.1. Montage manuel

Dans les systèmes UNIX, les périphériques, les disques durs et les partitions sont gérées sous forme de fichiers contenus dans le répertoire `/dev` : chaque fichier du répertoire `/dev` correspondent à une sorte de driver.

Pour utiliser un périphérique, il faut attacher le fichier périphérique correspondant à un point de montage.

Ainsi `/dev/hda` correspond au premier disque dur IDE, `/dev/hda1` à la première partition du disque dur IDE, `/dev/hda2` à la seconde partition, `/dev/hdb` au second disque dur IDE, `/dev/hdb1` à la première partition du second disque dur, etc..

De même `/dev/sda` correspond au premier disque dur SCSI, `/dev/sda1` à la première partition du disque dur SCSI, etc..

`/dev/cdrom` correspond souvent au lecteur de cd−rom et `/dev/fd0` au premier lecteur de disquette.

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.

Les principaux type de support sont :

> `ext2` : filesystem Linux ; `msdos` : disque de type MS−DOS ; `vfat` : disque de type Windows ; `iso9660` : CD−ROM ;

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`.

Exemple : pour changer de CD, il faut au préalable démonter le CD courant avec la commande :

```
$ umount /mnt/cdrom
```

A noter qu'à l'issu de l'installation de la distribution Mandrake Corporate Server, le lecteur de CD−ROM est monté par défaut ainsi que l'éventuelle partition Windows.

4.5.2. Montage automatique

Le fichier `/etc/fstab` (File System Table) contient toutes les informations concernant le montage automatique des disques au démarrage du système.

```
$ cat /etc/fstab
/dev/hda5 / ext2 defaults 1 1 none
/dev/hda7 /home ext2 defaults 1 2
/dev/cdrom /mnt/cdrom auto user,noauto,nosuid,exec,nodev,ro 0 0
/dev/fd0 /mnt/floppy auto sync,user,noauto,nosuid,nodev 0 0
```

```
/dev/hda1 /mnt/windows vfat user,exec,umask=0 0 0 none /proc proc defaults 0
0
/dev/hda6 swap swap defaults 0 0
```

Liste des paramètres associés à chaque périphérique :

- device (périphérique) de la partition ;
- point de montage de la partition ;
- type de la partition ;
- options (gestion des droits) ;
- sauvegarde (si ce champ est non nul, l'utilitaire dump doit sauvegarder ce système de fichier) ;
- test et réparation (si ce champs est non nul, l'utilitaire fsck doit tester la partition avant de la monter. Le numéro correspond à l'ordre des tests).

L'ajout d'un périphérique dans le fichier `/etc/fstab` permet de le monter automatiquement au démarrage.

4.6. INSTALLATION DE NOUVEAUX LOGICIELS

L'installation de nouveaux logiciels s'effectue soit à partir des sources, soit à partir d'un binaire (application déjà compilée), soit à partir d'un paquetage rpm.

4.6.1. Installation à partir des sources

L'installation à partir des sources consiste à compiler des lignes de code (en C ou en C++) puis à installer le binaire produit. Les avantages de cette méthode sont multiples :

- un même code source peut être compilé sur n'importe quelle machine UNIX et ce quel que soit son processeur (Intel, Alpha, Risc, PowerPC, etc..) ;
- vous pouvez spécifier le répertoire où l'application doit être installée ;
- vous pouvez compiler l'application avec des options spécifiques (ajout de modules particuliers, optimisation du binaire en fonction du processeur, etc…)
- les sources étant moins volumineux que les binaires, le téléchargement des sources d'une application est beaucoup plus rapide que le téléchargement du binaire ou du paquetage rpm correspondant.

Qu'elle que soit l'application, la procédure d'installation est identique :

1. Préparation de la compilation par la commande : `$ ./configure -prefix=répertoire-de-destination`
2. Compilation de l'application par la commande : `$ make`
3. Installation de l'application par la commande : `$ make install`

Il ne vous reste ensuite plus qu'à exécuter le script de lancement de l'application et si la nouvelle application doit être lancée systématiquement au démarrage de la machine (cas des services Internet), il vous faut également copier le script de lancement dans le répertoire `/etc/rc.d/init.d` (cf. chapitre *Lancement de programmes au démarrage*)

4.6.2. Installation à partir d'un binaire

Pour une application donnée, il existe peut être déjà une version binaire compilée pour votre processeur. Il ne vous reste plus qu'à télécharger l'application, la décompresser puis la déplacer dans le répertoire de votre choix. A noter

que dans la désignation employée dans les distributions binaires : Intel−386 désigne un processeur Intel de type 386, intel−486 de type 486, intel−586 de type Pentium, intel−686 de type Pentium II, etc…
Tout comme pour une installation à partir des sources vous devez ensuite lancer l'application et vérifier le cas échéant si celle−ci est lancée au démarrage.

4.6.3. Installation à partir d'un paquetage rpm

RPM (Red Hat Package Manager) est un puissant gestionnaire d'applications permettant d'installer, de mettre à jour, de vérifier ou de désinstaller des composants logiciels.

Pour installer un nouveau paquetage appli.rpm :

```
$ rpm −ivh appli.rpm
```

Attention, si vous installez un paquetage par cette méthode et qu'il existe déjà sur votre système dans une version inférieure, vous risquez d'avoir des problèmes pour le désinstaller (voir plus bas). Les paramètres `−vh` permettent d'ajouter une barre de progression.

Pour mettre à jour (upgrader) un paquetage :

```
$ rpm −Uvh appli.rpm
```

Pour supprimer un paquetage :

```
$ rpm −e appli.rpm
```

Afficher la liste de tous les paquetages installés :

```
$ rpm −qa
```

Vérifier à partir du nom si un paquetage est déjà installé :

```
$ rpm −qa | grep php
```

Lister le contenu d'un paquetage :

```
$ rpm −ql appli.rpm
```

Si vous avez besoin d'une application précise sous le format rpm, vous devriez pouvoir la trouver sans trop de difficultés sur le site http://www.rpmfind.net.

4.6.4. Installation à partir d'un paquetage Debian

La distribution Debian utilise son propre système de paquetage dont les fichiers sont reconnaissables par l'extentions .deb .

Pour gérer les paquetages il existe deux programmes :

- dpkg
- apt−get

Pour simplifier, on peut considérer que `apt` est une surcouche sur `dpkg`.

La première chose à faire est de définir par quel moyen nous allons accèder aux paquetages, soit par CDROM soit par une liaison réseau. Le programme `apt-setup` permet de configurer les sources de paquetage de manière interactive.

Une autre manière possible est d'ajouter directement les informations dans le fichier `/etc/apt/sources.list`.

Même si on utilise les cdroms pour installer les paquetages il est quand même conseiller d'ajouter la ligne :`deb http://security.debian.org/ stable/updates main contrib non-free` au fichier `/etc/apt/sources.list` ce qui permet de télécharger les dernièrs paquetages qui corrigent des problèmes de sécurité. Après une modification de ce fichier il faut lancer la commande `apt-get update` pour que le système puisse construire la base de données des paquetages en prenant en compte la nouvelle source.

Commandes usuelles :

`apt-get install Nom_Du_Paquetage` : Installe un nouveau paquetage
`apt-get remove Nom_Du_Paquetage` : Supprime d'un nouveau paquetage
`dpkg -S nom_du_fichier` : Donne le paquetage auquel appartient le fichier
`dpkg -i nom_du_fichier` : Installe le paquetage précédenment téléchargé
`dpkg -L Nom_Du_Paquetage` : Liste tous les fichier d'un paquetage
`apt-get update` : Mise à jour de la liste des paquetages
`apt-get dist-upgrade` : Mise à jour de tous les paquetages
`dpkg-reconfigure Nom_de_Paquetage` : reconfigure un paquetage déja installé
`apt-cache search XXXXX` : cherche XXXX dans le nom ou la description des paquetages disponnibles

Exemple d'utilisation d'apt-get :

```
debian:/home/fred# apt-get install vim<br>
Reading Package Lists... Done<br>
Building Dependency Tree... Done<br>
The following extra packages will be installed:<br>
 libgpmg1 <br>
The following NEW packages will be installed:<br>
 libgpmg1 vim <br>
0 packages upgraded, 2 newly installed, 0 to remove and 0  not
upgraded.<br>
Need to get 3796kB of archives. After unpacking 12.3MB will be
used.<br>
Do you want to continue? [Y/n]<br>
```

4.6.5. Lancement de programmes au démarrage

Au départ le kernel Linux lance le premier process init. Init va alors lire son fichier de configuration (`/etc/inittab`). Les programmes qui vont être exécutés dépendent du run-level spécifié dans le fichier inittab. Par défaut la distribution Mandrake utilise le niveau 5 alors que Debian utilise le niveau 2.

Le run-level qui va être exécuté peut être fixé au lancement en donnant en paramètre `init=X` au momment du boot sur l'invite de commande lilo.

Chaque run-level correspond à une utilisation particulière, où certains services vont être lancés au démarrage ou non, suivant le choix de l'administrateur.

A chaque run-level est associé un répertoire `/etc/rcX.d/` ou X est le run-level. Dans ce répertoire on trouve les scripts qui vont être exécutés au démarrage. Si on examine le contenu du répertoire `/etc/rc2.d/` on voit qu'il n'y

a aucun "vrai" fichier, mais seulement des liens sur des scripts qui se trouvent dans `/etc/init.d/`. Ceci permet de centraliser les scripts de démarrage qui sont souvent commun à plusieurs run−level.

Pour lancer automatiquement une application au démarrage, il est nécessaire d'écrire un script de lancement de l'application et de le placer dans le répertoire `/etc/init.d/` puis de faire un lien (grace à la commande `ln`) dans le ou les run−level concernés.

Par exemple voici le fichier que l'on va placer dans `/etc/init.d` :

```
#! /bin/sh
#
#Exemple d'execution d'une commande au démarrage
#
#Activation du mode DMA sur /dev/hda
/sbin/hdparm /dev/hda −d 1
```

on sauvegarde ce script sous le nom `hdparm` dans le répertoire `/etc/init.d`. Il faut le rendre exécutable :

```
chmod a+x /etc/init.d/hdparm
```

puis le mettre dans le run−level désiré :

```
ln −s /etc/init.d/hdparm /etc/rc2.d/S80hdparm
```

pour la Debian, ou :

```
ln −s /etc/init.d/hdparm /etc/rc5.d/S80hdparm
```

pour la Mandrake.

5. PRINCIPES FONDAMENTAUX D'UN RÉSEAU

5.1. LE PROTOCOLE TCP/IP

Le principe fondamental de l'Internet a été de créer un mode de transmission par paquet remplaçant les modes en continu utilisés jusque−là pour la transmission de données. Chaque fichier transmit sur Internet est segmenté en paquets de données autonomes pouvant être transmis indépendamment les uns des autres. Pour que cela fonctionne, chaque paquet de données doit contenir des informations de pilotage telles que l'adresse de l'ordinateur émetteur et l'adresse de l'ordinateur récepteur.

Le protocole de communication permettant de transmettre des données sur Internet est le protocole **TCP/IP**. TCP/IP n'est en fait pas un mais deux protocoles, l'un étant superposé sur l'autre.

Le protocole de premier niveau, **IP** (Internet Protocol) s'occupe du routage des informations entre l'expéditeur et le destinataire : il accomplit sa tâche en divisant les informations en paquets (de 1 500 octets) et leur adjoint une adresse de provenance et de destination (exactement comme une enveloppe envoyée par la poste).

TCP (Transport Control Protocol) s'appuie sur IP pour gérer le transfert des données entre l'expéditeur et le destinataire. TCP fournit également les mécanismes permettant d'établir les connections, de vérifier l'arrivée dans le bon ordre des données, de gérer des données perdues, les erreurs et de récupérer des données concernées.

Lors de la transmission de données sous forme de paquet, IP ne vérifiant en aucune manière que le paquet est bien arrivée, TCP exige que le destinataire envoi un accusé de réception ou ACK (Acknowledged). De ce fait, l'hôte expéditeur peut se trouver devant trois situations différentes :

- lorsque le destinataire reçoit un paquet, et si celui−ci est le paquet attendu, il répond par le message ACK ;
- si la somme de contrôle indique une erreur ou si le numéro d'ordre est incorrect, le destinataire envoie un message NAK (Not Acknowledged) ;
- si le destinataire ne répond rien, TCP décide que soit le paquet, soit la réponse s'est perdu et renvoie de ce fait le paquet concerné.

A noter que outre les données, TCP envoie également des informations lors de l'établissement de la connexion et lors de son interruption.

Précisons également que TCP n'est pas le seul protocole utilisant IP : TCP/IP comprend également UDP (Unigram Data Protocol). Il s'agit d'un protocole sans connexion et sans garantie utilisé pour des transmissions de faible importance (comme la vidéo ou le son sur Internet).

5.2. LE MODÈLE RÉSEAU TCP/IP

TCP/IP ne constitue que deux couches dans un ensemble de protocoles allant de la base (matériel) vers le sommet (application). Le **modèle réseau TCP/IP** ressemble au modèle réseau ISO/OSI à 7 couches (couche physique, couche liaison, couche réseau, couche transport, couche session, couche présentation et couche application).

TCP/IP est une hiérarchie réseau à quatre niveaux superposés au matériel :

4	Protocoles d'application	SMTP (Sendmail), HTTP (Apache), telnet, ftp, rlogin, DNS, etc..
3	Protocoles de transport	TCP, UDP, ICMP
2	Protocoles de réseau (Internet)	IP
1	Protocoles d'accès au réseau (liaison de données)	Ethernet, ISDN, SLIP, PPP, etc..

Matériel	Câbles, lignes téléphoniques, etc..

Exemple :

Dans une communication typique entre un serveur web et un client, les différentes couches ressembleraient à cela :

Du coté serveur, relié à un réseau Ethernet :

- niveau 4 : HTTP (Apache)
- niveau 3 : TCP
- niveau 2 : IP
- niveau 1 : Ethernet

Du coté du client, connecté à Internet par modem :

- niveau 4 : HTTP (Netscape)
- niveau 3 : TCP
- niveau 2 : IP
- niveau 1 : PPP (Point−To−Point Protocol) imbriqué sur la connexion série afin de franchir l'étape entre IP et le matériel)

5.3. ADRESSES IP ET CLASSES DE RÉSEAUX

5.3.1. Le futur IPv6

Selon la norme IPv4, une adresse est codée sur 32 bits (soient 4 octets), ce qui permettrait théoriquement d'attribuer 232 adresses. Du fait de la répartition en réseaux de classe A,B et C, le nombre d'adresses possibles est largement inférieur au nombre théorique et il existe aujourd'hui un risque de pénurie d'adresse IP.

La norme IPv6 consiste à utiliser 128 bits pour coder les adresses (soient 16 octets). Cette norme a été adoptée en 1995 après quatre années de discussions dans différentes assemblées et groupes de travail.

La compatibilité avec la norme IPv4 a été préservé afin de permettre une phase de transition suffisante pour le passage de IPv4 vers IPv6. A noter que les versions récentes de Linux prennent déjà en compte la norme IPv6.

5.3.2. Classes de réseaux

Pour que l'acheminement des données fonctionne sur un réseau TCP/IP (Intranet ou Internet), chaque ordinateur doit posséder **une adresse IP unique**. Si en plus, l'ordinateur doit communiquer sur Internet, son adresse IP doit également être unique.

Selon la norme en vigueur actuellement (**IPv4**), une adresse IP est codé sur 32 bits répartis en quatre octets : par exemple 192.168.20.101 (la valeur d'un octet variant de 0 à 255).

L'ensemble des adresses IP est divisé en régions, à l'intérieur desquelles coexistent plusieurs classes de réseaux. Internet considère que les adresses IP à l'intérieur d'une classe de réseau font partie du même réseau : Internet n'attend qu'un point d'entrée, ce que nous appelons une **passerelle**, pour pouvoir router des paquets aux hôtes de ce réseau.

L'espace adresse IP est réparti entre des régions de réseaux de classe A, B et C :

- les réseaux de classe A, en nombre très limité, possèdent une adresse dont le premier nombre est compris entre 1 et 126. Seul ce premier nombre est fixe. Un réseau de classe A peut posséder 16 777 214 hôtes ;

- les réseaux de classe B, possèdent une adresse dont le premier nombre est compris entre 128 et 191. Les deux premiers nombres sont fixes. Il peut ainsi exister 16 382 réseaux de classe B possédant chacun jusqu'à 65 534 hôtes ;

- les réseaux de classe C, possèdent une adresse dont le premier nombre est compris entre 192 et 223. Les trois premiers nombres sont fixes. Il peut ainsi exister plus de 2 millions de réseaux de classe C possédant chacun un maximum de 254 hôtes.

5.3.3. Masque de réseau et routage

Un masque de réseau est un nombre logiquement ajouté (à l'aide de l'opérateur booléen AND) à une adresse IP afin d'obtenir l'adresse réseau.

Exemple :

	198	4	211	127	Adresse IP
	255	255	255	0	Masque de réseau de classe C
donne	198	4	211	0	Adresse de réseau

Le masque de réseau définit les adresses d'une plage adresse IP considérés comme étant directement connectés, c'est à dire appartenant au même segment de réseau. Des adresses différentes − obtenues par addition avec le masque de réseau − sont considérées comme appartenant à un réseau externe, et doivent utiliser les passerelles et les routeurs pour communiquer.

Exemple : considérons les 3 hôtes suivants : hôte 1 d'adresse 192.168.1.1, hôte 2 d'adresse 192.168.1.2 et hôte 3 d'adresse 192.168.2.1

En définissant un masque de réseau de 255.255.255.0 pour tous les hôtes, les hôtes 1 et 2 sont considérés comme appartenant au même réseau : si l'hôte 1 envoie un paquet à l'hôte 2, TCP/IP tentera de l'envoyer directement. En revanche, l'hôte 1 ne peut envoyer de données directement à l'hôte 3 car le masque considère que 192.168.1 et 192.168.2 sont deux réseaux différents : il enverra donc le paquet vers une passerelle. Il en résulte que tout hôte possède l'adresse IP d'au moins une passerelle afin de pouvoir expédier les paquets qu'il ne peut transmettre lui−même.

Quel est l'intérêt de ce système ?

En divisant l'espace adresse en réseaux logiques, trouver un hôte particulier devient une tâche facile. Pas besoin de connaître tous les hôtes de l'Internet car il suffit de disposer d'une liste de passerelles et de sélectionner celle constituant l'étape logique suivant la route. La passerelle suit la même procédure à l'aide de sa propre liste de passerelles et ainsi de suite, jusqu'à ce que le paquet atteigne la passerelle finale et sa destination.

5.3.4. Adresses IP particulières

Il existe des adresses IP appelées adresses de diffusion permettant la réception de données sur l'ensemble des hôtes d'un réseau. Nous ne traiterons pas ici ces adresses particulières.

Certaines adresses sont réservées à un usage personnel. Elles ne sont pas routées sur l'Internet et ne peuvent pas générer de problèmes quand vous les réutilisez. Leurs intervalles sont :

Classe	Masque de réseau	Adresses réseau
A	255.0.0.0	10.0.0.0

B	255.255.0.0	172.16.0.0 à 172.31.0.0
C	255.255.255.0	192.168.0.0 à 192.168.255.0

Quelle adresse choisir pour configurer un réseau local ? Cela n'a pas vraiment d'importance mais il est recommandé d'utiliser pour un même réseau, des nombres consécutifs.

Par exemple, si vous avez deux ordinateurs connectés via Ethernet et vous avez besoin maintenant de deux adresses à assigner aux deux cartes réseau, vous pouvez utiliser simplement 192.168.0.1 et 192.168.0.2

A noter également que l'adresse 127 de réseau de classe A est universellement réservée à la boucle locale du réseau, ce qui permet de tester les fonctionnalités de l'interface réseau de son propre ordinateur (c'est pour cela que l'on retrouve systématiquement la ligne `127.0.0.1 localhost` dans le fichier `/etc/hosts`).

5.3.5. Le concept des ports

Lorsqu'un client contacte un serveur, c'est le plus souvent en vue d'utiliser un service précis, courrier électronique ou FTP par exemple. Afin de différencier ces services, TCP dispose du concept de port qui permet à même interface réseau de fournir plusieurs services différents.

Le port standard pour le protocole HTTP correspond au port 80. Tout service ou protocole réseau standard possède un port associé auquel se connectent les clients pour y accéder : qu'il s'agisse d'HTTP, de FTP, de telnet ou de tout autre standard. La liste des ports standards est défini dans le fichier `/etc/services`. Voici ci−après une liste provenant du fichier `/etc/services` : Les ports `http` (80) et `https` (443) sont les plus répandus. Il est possible de préciser un port particulier dans l'URL d'un navigateur : il suffit de placer " :" ainsi que le numéro de port après l'adresse web. Exemple : `http://localhost:10000` pour accéder à WebMin.

`inetd` (Internet Daemon) est le service chargé d'écouter les différents port. Lorsque `inetd` reçoit une requête sur un port dont il a la charge d'écouter, il exécute le service associé (contrairement à Apache qui s'exécute indépendamment).

5.4. NOMS LOGIQUES ET DNS

5.4.1. Adresses IP et noms logiques d'ordinateurs

Etant donné qu'il est difficile de mémoriser les adresses IP des différents ordinateurs au sein d'un réseau, il est fort pratique d'associer à chaque adresse IP un nom logique. La définition des correspondances entre les noms des machines et les adresses IP se trouve dans le fichier `/etc/hosts`.
Ce fichier doit être présent sur toutes les machines du réseau. Si un ordinateur est ajouté ou retiré du réseau, le fichier `/etc/hosts` doit être modifié en conséquence sur toutes les machines du réseau. Ce type d'administration n'est donc possible que si le réseau ne dépasse pas une certaine taille.

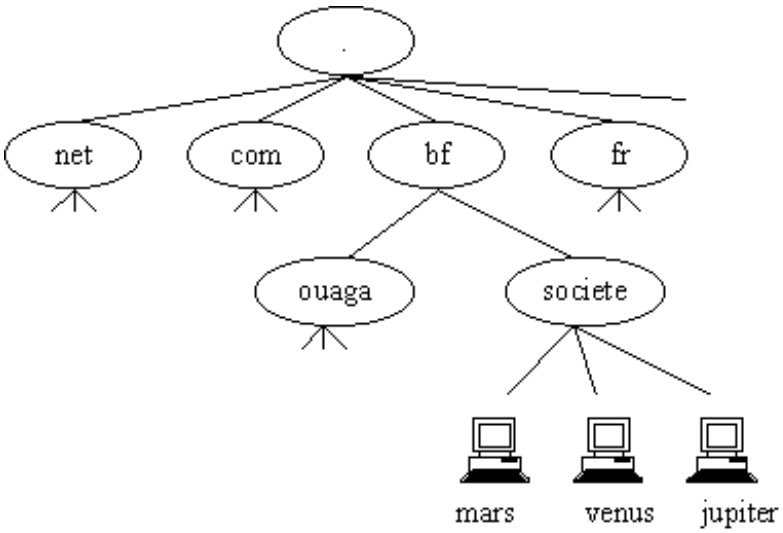
5.4.2. DNS − Domain Name Service

La méthode de mise en correspondance des noms d'ordinateurs et des adresses IP décrite précédemment a été la seule méthode employée sur l'Internet jusqu'en 1984. Jusqu'à cette date, toutes les adresses et noms d'ordinateurs étaient centralisés et gérés aux Etats−Unis par le NIC (Network Information Centre) sous la forme d'un fichier hosts.txt. Face à l'accroissement rapide de l'Internet, cette méthode s'est avérée rapidement impraticable, d'où l'introduction d'un nouveau mode d'adressage des ordinateurs : le DNS (Domain Name Service).

Organisation et structure de DNS :

Le DNS organise les noms d'ordinateurs selon une hiérarchie de domaines semblable à celle du système de fichier Linux. Partant d'une racine commune (domain racine), le service se stratifie en plusieurs couches depuis la couche

supérieur contenant les domaines principaux, vers les couches inférieures divisées en sous−domaines.



Exemple de structure de domaine

Par domaine, on entend une collection d'ordinateurs regroupés selon des critères géographiques ou organisationnels. Ce type de structure permet de satisfaire facilement la contrainte d'unicité des noms d'ordinateurs. On obtient le nom complet d'un ordinateur en commençant par le nom de la machine dans le cadre de son sous−domaine particulier et en suivant le chemin qui remonte vers le niveau supérieur de la hiérarchie (TLD ou Top Level Domain) ; le symbole de séparation entre les différents niveaux étant le point.

L'administration des noms de domaine génériques est effectué par le NIC (http://www.internic.net). L'administration des noms de domaines géographiques a été transférée aux différents pays. En principe, le nombre de sous−domaines placés en dessous de la couche des domaines de second niveau n'est pas limité.

Fonctionnement du DNS :

Avec cette structure, on obtient une gestion décentralisée et délocalisée des domaines. Chaque serveur de nom local gère les données pertinentes de tous les ordinateurs relevant de son domaine de compétence et est en mesure de répondre aux demandes en provenance de l'Internet concernant son domaine.

Le DNS constitue une banque de données mondiale constitué d'un grand nombre de serveurs de noms de domaines. Un serveur de nom de domaine stocke les informations nécessaires relatives à tous les ordinateurs présent dans son domaine de compétence. Cette zone peut comprendre un ou plusieurs domaines. Dans chaque zone, deux serveurs de noms au moins doivent exister pour des raisons de fiabilité (informations accessibles par des voies redondantes).

Rappelons que l'adressage de chaque machine de l'Internet s'effectue exclusivement par l'intermédiaire de l'adresse IP. Lorsqu'une application (Netscape par exemple) veut prendre contact avec un ordinateur dont seul le nom DNS est connu, il est nécessaire de convertir au préalable le nom DNS en une adresse IP. Pour cela une requête est envoyé au serveurs de noms figurant dans le fichier de configuration. Concernant le fichier de configuration de résolution de noms de domaine, il existe deux stratégies différentes :

- la première possibilité consiste à adresser une requête directement à un serveur dont la compétence s'exerce sur les domaines principaux ;
- la seconde possibilité consiste à adresser la requête à un serveur de nom local qui à son tour, s'il ne peut satisfaire lui−même la requête, adresse celle−ci à un autre serveur de nom, etc.. la requête migre ainsi de la base vers le sommet.

La mise en oeuvre d'un serveur DNS sera traité ultérieurement.

5.5. OUTILS RÉSEAUX

Linux contient de nombreux de nombreux utilitaires permettant de faciliter l'administration d'un réseau.

ifconfig : utilitaire standard UNIX permettant d'obtenir des informations sur la configuration de l'interface réseau (carte Ethernet par exemple) : `$ ifconfig −a`
Servez−vous de `man ifconfig` pour connaître les options.

netstat : utilitaire de surveillance d'un réseau sous les systèmes UNIX.

ping : l'outil le plus simple et le plus pratique des outils réseaux. ping permet de vérifier si un nom d'hôte distant ou une adresse IP est accessible.

traceroute : utilitaire trés utile pour diagnostiquer des problèmes réseaux, en particulier si la commande ping ne réussit pas à atteindre le serveur distant. Il existe des traceroute graphiques permettant de visualiser le chemin parcouru par les données entre un client et un serveur.

5.6. CONFIGURATION D'UN RÉSEAU LOCAL SOUS LINUX

Interface réseau : l'interface réseau est représentée physiquement par votre carte réseau mais le terme interface réseau est aussi utilisé pour désigner un nom logiciel auquel assigner une adresse IP (eth0 par exemple). Une adresse IP est toujours assignée à une interface réseau, jamais à un ordinateur. La commande ifconfig sert à afficher la configuration des différentes interfaces réseau actives.

Adresses IP : référez vous au chapitre *Adresses IP particulières* pour décider quelle adresse utiliser pour votre réseau.

Fichiers de configuration :

`/etc/hosts` : ce fichier spécifie comment résoudre les noms des machines du réseau local (inutile de mettre en oeuvre un serveur DNS pour un petit réseau local). La syntaxe des lignes de ce fichier est :

```
     Adresse IP      Nom de l'hôte          Alias
Ex : 127.0.0.1       localhost
     192.168.0.1     sirius.mondomainesirius
```

`/etc/resolv.conf` : ce fichier spécifie où résoudre ce qui ne se trouve pas dans `/etc/hosts`. C'est dans ce fichier que vous devez spécifier les adresses IP des serveurs DNS utilisés pour accéder à Internet en suivant la syntaxe suivante : `nameserver 212.102.31.1`

`/etc/HOSTNAME` (ou `/etc/sysconfig/network` sur certaines distributions) : ce fichier configure le nom de la machine locale. Au démarrage du système, ce fichier est lu et son contenu est envoyé à la commande hostname. Vous pouvez utiliser la commande `hostname` pour changer le nom du serveur.

Exemple : `hostname sirius.mondomaine`

Autres fichiers de configuration du réseau : `/etc/hosts.allow` , `/etc/hosts.deny` et `/etc/hosts.equiv`.

6. MISE EN OEUVRE D'UN SERVEUR APACHE

6.1. APACHE ET L'INTERNET

6.1.1. Pourquoi Apache est−il devenu un standard ?

- Coût nul
- Code source disponible et modifiable permet un développement rapide du serveur, la création de modules spécifiques et une très grande réactivité dans la correction de tout bogue identifié.
- Très grande flexibilité du serveur grâce à sa structure modulaire l'ajout d'un nouveau module permet d'ajouter de nouvelles fonctionnalités.

Sa flexibilité combinée à sa stabilité, à ses performances, ainsi qu'à la disponibilité du code source ont fait du logiciel Apache, le serveur WEB le plus populaire sur Internet.

6.1.2. Quel type de matériel faut−il pour un serveur Apache sous Linux ?

Faibles exigences matérielles :

Apache s'exécute sur n'importe quel type de machine. Pour un serveur de sites WEB peu exigeants, un simple 486 fera parfaitement l'affaire. Pour des sites très exigeants utilisant de nombreuses bases de données, un Pentium multiprocesseur peut être envisagé.

Concernant la mémoire : plus il y a de mémoire vive et plus la quantité de donnée en mémoire est importante ce qui a pour conséquence d'accélérer les accès.

Concernant le disque dur : un disque dur rapide permet d'améliorer les performances d'accès aux données des sites WEB. A noter que dans le cas de sites à grande audience, il est préférable d'utiliser plusieurs disques de tailles moyennes plutôt qu'un seul disque à grande capacité (un disque dur ne pouvant lire qu'à un seul endroit à la fois).

Concernant la carte réseau, une carte Ethernet 100baseT est préférable à une carte 10baseT (du moins en théorie car il faut encore que votre fournisseur d'accès offre une bande passante suffisante). Si le serveur doit également être connecté à un Intranet local, on peut envisager d'utiliser deux cartes réseaux : une dédiée au réseau interne et l'autre au réseau Internet. Ce qui permet d'augmenter la sécurité et d'éviter que l'encombrement du serveur WEB diminue la bande passante du réseau interne.

Exigence fondamentale : utiliser une machine dédiée :

Exécuter Apache sur une machine dédié à l'hébergement Internet. Pour assurer une disponibilité permanente des sites hébergés et compte tenu des faibles exigences matériels d'Apache, il est judicieux d'utiliser une machine spécialement dédiée à Apache et d'éviter le partage des ressources avec d'autres applications.

Connexion Internet :

Pour héberger des sites Internet, une **liaison permanente haut−débit** ainsi qu'une **adresse IP fixe** sont indispensables. Ce sont surtout les caractéristiques de votre connexion Internet plus que les caractéristiques de la machine qui vont définir les performances de votre serveur WEB. Lors du choix du fournisseur d'accès Internet (FAI ou ISP en anglais pour Internet Server Provider), il faut se poser des questions sur sa fiabilité, la qualité de la connexion Internet proposé ainsi que sur le nombre de clients qui se partagent la bande passante disponible (le débit de la bande passante fournie reste purement théorique puisque celle−ci est partagée par les utilisateurs).

6.1.3. Le protocole HTTP

HTTP ou HyperText Transfer Protocol est un protocole de requêtes et de réponses. Le dialogue entre un client web (un navigateur tel que Netscape) et un serveur (Apache) se traduit par une requête du client à laquelle le serveur répond en effectuant le traitement intermédiaire adéquat.

6.2. INSTALLATION ET EXÉCUTION D'APACHE

6.2.1. Tester le serveur Apache

L'approche la plus simple et la plus naturelle pour tester le bon fonctionnement d'Apache consiste à lancer un navigateur web et d'utiliser `http://localhost` comme adresse. Après installation et lancement d'Apache, si tout se passe bien, vous devriez voir apparaître la page web par défaut d'Apache.

Si aucune page n'apparaît vérifier tout d'abord si le serveur est bien lancé en affichant les processus :

```
$ ps − aux | grep httpd
```

Dans le cas où le fichier de configuration d'Apache comporte une erreur, un message signale l'emplacement de l'erreur lors du lancement du serveur.

6.2.2. Installer Apache à partir d'un paquetage préconstruit

Se référer au site http://www.funix.org pour les détails de l'installation.

6.2.3. Installer Apache à partir des sources

Se référer au site http://www.funix.org pour les détails de l'installation.

6.2.4. Lancer, arrêter et redémarrer le serveur

L'exécutable Apache porte le nom `httpd` pour HyperText Transfer Protocol Daemon (selon la terminologie Unix, une application offrant des services au lieu de communiquer directement avec l'utilisateur s'appelle un démon − daemon en anglais).

Pour lancer Apache, il suffit d'exécuter httpd :

```
/etc/init.d/httpd start
```

Une fois lancé, Apache s'exécute en tâche de fond. Pour arrêter Apache :

```
/etc/init.d/httpd stop
```

Si le process apache ne réponds plus aux commandes il faut rechercher les processus `httpd` :

```
$ ps − aux | grep httpd
```

puis les détruire :

```
<blockquote>
$ kill −9 <pid>
```

où est l'identificateur du processus Apache racine.

Dans le cas où on a simplement changé un fichier de configuration on peut exécuter la commande :

`/etc/init.d/httpd graceful`
qui demande au process apache courant de relire ses fichiers de configuration.

Dans ce mode, les transactions client ne sont pas interrompues et les processus Apache sont réactualisés au fur et à mesure que les requêtes en cours sont achevés.

6.2.5. Lancer automatiquement le serveur au démarrage de l'ordinateur

Il est important de s'assurer que les sites hébergés soient accessibles à tout moment ou presque. Dans le cas d'une coupure de courant prolongée dépassant la durée d'efficacité de l'onduleur, le serveur s'éteint. Lorsque le courant est rétabli, la machine redémarre. Il est crucial comme tout autre service réseau que le serveur Apache soit redémarré automatiquement afin que le temps d'indisponibilité des sites hébergés soit le plus court possible.

Si apache a été installé par un paquetage de la distribution le démarrage au boot doit être automatique.

Si apache a été compilé et installé depuis le code source il faut recopier le script `apachectl` dans le répertoire `/etc/init.d/` et créer un lien dans le répertoire /etc/rcX.d correspondant au run−level désiré.

6.3. CONFIGURATION DE BASE DU SERVEUR HTTP

La configuration du serveur Apache peut s'effectuer soit en modifiant manuellement son fichier de configuration avec un éditeur de texte soit en utilisant une interface graphique dédiée (par exemple l'excellent WebMin). Même si la configuration d'Apache à partir d'un fichier texte peut paraître fastidieux, cette méthode présente l'avantage de contrôler exactement et de façon exhaustive le fonctionnement du serveur.

Le fichier de configuration d'Apache se nomme `httpd.conf` et est placé dans le répertoire `conf/` du répertoire d'Apache dans le cas d'une installation manuelle (par exemple `/usr/local/apache/conf/`). Dans le cas d'une installation à partir d'un paquetage, le fichier `httpd.conf` se trouve généralement dans le répertoire `/etc/httpd/conf` A noter que dans les versions anciennes d'Apache, le contenu du fichier `httpd.conf` était réparti dans 3 fichiers différents : `httpd.conf`, `access.conf` et `srm.conf`.

A l'issu de l'installation du serveur Apache, le fichier `httpd.conf` est configuré par défaut. Avant d'envisager de le modifier, assurer au préalable que le serveur fonctionne (cf. le chapitre précédent) et n'oublier pas de dupliquer le fichier de configuration par défaut. En effectuant par exemple :

```
$ cp httpd.conf httpd.conf.defaut
```

Comment Apache structure−t−il sa configuration ?

- les directives de conteneur Apache ont une portée limitée ;
- les directives peuvent être utilisées à portée globale ou locale ;
- il est possible d'outrepasser une directive à l'aide d'une configuration par répertoire.

Nom de serveur : `ServerName www.ouaga.bf` Il ne s'agit pas du nom du serveur pour lequel Apache répond mais du nom avec lequel Apache envoie sa réponse.

Adresse IP du serveur : `BindAdress 192.168.13.11`

Port(s) à écouter : `Port 80` Il est possible de remplacer les configurations Port et Bindadress par la directive `Listen` : Listen 192.168.13.11 :80 (L'avantage est qu'il est possible de spécifier plusieurs fois la directive Listen, ce qui n'est pas le cas pour BindAdress)

Serveur autonome ou non :

`ServerType standalone` : apache est autonome `ServerType inetd` : apache n'est lancé que lorsque inetd reçoit une requête sur les ports pour lequel il est configuré.

Utilisateur et groupe : dans le but de rendre Apache moins vulnérable aux éventuelles attaques, il est possible de le configurer pour qu'il s'exécute sous un utilisateur et un groupe ayant des droits restreints.

```
User nobody
Group nobody
```

Adresse électronique de l'administrateur : `ServerAdmin webmaster@ouaga.bf`

Racine du serveur : `ServerRoot /etc/httpd`

Journal d'erreur par défaut : `ErrorLog logs/error_log`

Emplacement par défaut des pages html : `DocumentRoot /home/httpd/html`

Pages par défaut : `DirectoryIndex index.htm index.html` index.php

Exercice 1 : modifier le fichier de configuration Apache de façon à ce que les pages web se situent dans votre répertoire personnel et afin que les navigateurs pointent par défaut sur la page accueil.htm

Exercice 2 : utiliser la directive ErrorDocument afin de rerouter les erreurs 404 vers une page d'erreur personnalisée renvoyant sur la page d'accueil du site.

Exercice 3 : utiliser la directive CustomLog afin de tracer les connections des clients à partir de leur adresses IP. Puis positionner la variable HostNameLookups à On afin de tracer les clients à partir de leur nom DNS.

6.4. CONFIGURATION AVANCÉE DU SERVEUR HTTP

6.4.1. Les hôtes virtuels

On peut mettre en place des hôtes virtuels, en d'autres termes un utilisateur pour un même serveur Apache croira en voir plusieurs.

Exemple : soit votre serveur Apache sirius (adresse IP 192.168.13.11), vos domaines ouaga.bf et bobo.bf, nous allons créer les hôtes virtuels www.sirius.ouaga.bf, www.mars.ouaga.bf et www.tourisme.bobo.bf qui vont pointer chacun vers un endroit différent du disque.

Editez le fichier `/etc/httpd/conf/httpd.conf` et décommentez les lignes suivantes tout à la fin du fichier (enlevez le # devant la ligne) :

```
# For DynamicVhosts and VirtualHomePages, uncomment those lines:
LoadModule vhost_alias_module modules/mod_vhost_alias.so
AddModule mod_vhost_alias.c
```

Maintenant nous allez éditer le fichier `/etc/httpd/conf/vhosts/Vhosts.conf` et rajouter :

```
NameVirtualHost 192.168.13.11

<VirtualHost 192.168.13.11>
ServerAdmin webmaster@sirius.ouaga.bf
DocumentRoot /home/httpd/html/sirius
ServerName sirius.ouaga.bf
```

```
ServerAlias www.sirius.ouaga.bf sirius.bobo.bf www.sirius.bobo.bf
ErrorLog logs/sirius-error_log
CustomLog logs/sirius -access_log common
ErrorDocument 404 /erreur.html
</VirtualHost>

<VirtualHost 192.168.13.11>
ServerAdmin webmaster@mars.ouaga.bf
DocumentRoot /home/httpd/html/mars
ServerName mars.ouaga.bf
ServerAlias www.mars.ouaga.bf
ErrorLog logs/mars-error_log
CustomLog logs/mars -access_log common
ErrorDocument 404 /erreur.html
</VirtualHost>

<VirtualHost 192.168.13.11>
DocumentRoot /home/httpd/html/tourisme
ServerName www.tourisme.bobo.bf
</VirtualHost>
```

Ce fichier est appelé dans `/etc/http/conf/httpd.conf` à la ligne :

```
Include conf/vhosts/Vhosts.conf
```

Les directives `VirtualHost` peuvent également être placé dans le fichier principal de configration afin de réduire le nombre de fichiers de configuration.

Il faut maintenant créer les hôtes sirius, mars et tourisme, pour cela il existe deux méthodes :

- en éditant le fichier `/etc/hosts` (visibilité des sites restreinte au réseau local)

```
192.168.13.11 sirius sirius.ouaga.bf www.sirius.ouaga.bf
www.mars.ouaga.bf mars.ouaga.bf www.tourisme.bobo.bf
```

- si vous disposez d'un serveur DNS, il faut ajouter les nouvelles entrées et relancer le serveur DNS (nous aborderons cette méthode dans le chapitre consacré à la mise en oeuvre d'un serveur DNS).

6.4.2. Protection d'une page

La protection d'une page pour l'utilisateur phil se fait de manière très simple, tous les fichiers à accès limité devant être concentré dans un même répertoire. Dans ce répertoire, il suffit de créer un fichier nommé `.htaccess` contenant :

```
AuthUserFile auth/phil.users
AuthName "Acces Restreint"
AuthType Basic

<Limit GET POST>
require valid-user
</Limit>
```

Le fichier `phil.users` doit contenir la liste des utilisateurs habilités à accéder au répertoire où se trouve `.htaccess`

A noter que le fichier .htaccess<code> peut être nommé différemment en utilisant la directive <code>AcessFileName.

Pour créer ce fichier il suffit d'une part de créer le répertoire /etc/httpd/auth si celui−ci n'existe pas, puis de taper :

```
htpasswd −c /etc/httpd/auth/phil.users phil
```

L'option −c correspondant à la création du fichier.

Pour que l'utilisateur hedi puisse accéder aussi au répertoire réservé de phil :

```
htpasswd /etc/httpd/auth/phil.users hedi
```

Si vous voulez vous assurez que personne ne puisse consulter les fichiers .htaccess de vos utilisateurs, rajoutez dans le fichier httpd.conf, la directive suivante :

```
<files ~ "/\.ht">
order deny,allow
deny from all
</files>
```

6.5. ANALYSER LES LOGS APACHE

Le premier Webalizer est sûrement le plus connu, le deuxième Awstats est le nouveau venu dans le domaine, il gagne à être connu car il fournit une information plus riche que Webalizer. Les deux outils possèdent l'avantage de présenter les résultats dans une page web.

6.5.1. Utilisation de Webalizer

Cet utilitaire disponible à l'adresse http://www.webalizer.com fonctionne aussi bien sur Windows que sur Linux. L'avantage de la version Linux est que celle−ci permet d'automatiser la génération périodique des statistiques à partir de la commande cron : il est alors envisageable de présenter les pages résultats dans une section de votre site protégée par mot de passe.

La génération des statistiques s'effectue soit en passant le nom du fichier de trace à la commande webalizer soit par simple appel de webalizer après avoir modifié la directive LogFile du fichier de paramètrage webalizer.conf

6.5.2. Utilisation de Awstats

Awestat est un outil plus riche que webalizer. Il est téléchargeable à l'adresse : http://awstats.sourceforge.net

L'installation s'effectue à partir de la commande unzip. Dans un premier temps, il est utile de remplacer dans le fichier httpd.conf la ligne :

```
CustomLog /usr/local/apache/logs/access_log common
```

par :

```
CustomLog /usr/local/apache/logs/access_log combined
```

puis de relancer le serveur Apache.

A présent on va copier un certain nombre de fichiers se trouvant sous le répertoire awstats :

```
cp awstats.pl /usr/local/apache/cgi−bin/
cp −R browser/ /usr/local/apache/icons/
cp −R cpu/ /usr/local/apache/icons/
cp −R flags/ /usr/local/apache/icons/
cp −R os /usr/local/apache/icons/
cp −R other/ /usr/local/apache/icons/
cp −R clock/ /usr/local/apache/icons/
```

Il est ensuite nécessaire de modifier le fichier `awstats.pl` afin de spécifier la variable `LogFile` ainsi que les serveurs virtuels en respectant la syntaxe suivante :

```
@HostAliases=
("sirius.ouaga.bf","mars.ouaga","127.0.0.1","192.168.13.11");
```

puis enfin de modifier les droits :

```
chmod u+w,a+rx awstats.pl
chmod a+wrx /usr/local/apache/cgi−bin
chmod a+wrx /etc/httpd/logs
chmod a+rw /etc/httpd/logs/access_log
```

Pour obtenir les statistiques, tapez l'adresse suivante dans un navigateur : `http://localhost/cgi−bin/awstats.pl?lang=1`

7. LE TRIO GAGNANT APACHE, PHP ET MYSQL

7.1. PHP

7.1.1. PHP pour générer dynamiquement des pages

PHP est un langage de script permettant d'intégrer des instructions de programmation puissantes directement dans du code HTML. Le serveur traite les scripts PHP d'une page et génère dynamiquement la page HTML résultat des fonctions PHP. Le principal intérêt de PHP est que celui−ci permet de créer facilement des pages dynamiques résultats de calculs ou de requêtes SQL effectuées sur une base de données.

Qu'appelle t'on une page dynamique ? Deux appels consécutifs d'une même page dite dynamique peuvent donner deux pages HTML différentes (la page est statique au niveau du client web mais elle est générée dynamiquement au niveau du serveur).

PHP peut également générer des fichiers PDF, s'interfacer avec des serveurs de messagerie, des serveurs LDAP ou encore générer des images et graphiques GIF à la volée, etc..

L'utilisation d'un langage de script tel que PHP est un passage obligé pour réaliser un site à contenu évolutif et riche sans avoir à passer son temps à modifier sans cesse les pages du site : plusieurs personnes alimentent, à partir d'une simple interface Web, une ou plusieurs bases de données qui servent à générer le contenu du site en fonction des demandes des utilisateurs.

7.1.2. Différences avec les autres langages de scripts

Javascript : le code Javascript est interprété par le client Web alors que le code PHP est directement interprété par le serveur Web (avec PHP seul le résultat est visible au niveau du client). De ce fait, Javascript est surtout utilisé pour la présentation et la manipulation d'une page html (menu déroulant, bannière, etc..) alors que PHP sert surtout à générer automatiquement du contenu à partir de bases de données.

Perl : Perl est le langage le plus populaire pour les solutions SGI. Il permet de supporter des technologies complexes tels qu'un moteur de recherche. Contrairement à PHP, Perl ne peut pas s'intégrer directement dans une page web ce qui implique une mise au point beaucoup plus délicate et plus complexe qu'avec PHP. Sans compter que la force de PHP réside dans le fait que celui−ci supporte directement les requêtes SQL.

ASP : ASP ou Active Server Page est la solution proposée par Microsoft avec son serveur Web IIS (Internet Information Server) pour créer des pages dynamiques. L'intérêt de ASP est que celui−ci utilise les fonctionnalités de Perl. Format propriétaire Microsoft, ASP inclue beaucoup moins de fonctionnalités que PHP. De ce fait si le serveur Internet n'est pas sous Windows, nous n'avons aucun intérêt à utiliser des scripts ASP au lieu de scriots ASP.

7.2. MYSQL

La grande force de PHP réside dans son utilisation conjointe avec une base de donnée : ce qui permet de fournir un contenu évolutif sans avoir à passer sans cesse son temps à modifier les pages html. PHP s'interface avec la quasi totalité des SGBD du marché.

7.3. INSTALLATION DE PHP ET MYSQL

Installation de PHP et MySQL : se référer au site funix.org (http://www.funix.org) pour l'installation de PHP et MySQL à partir des sources ou des paquetages rpm.

A l'issu de l'installation de MySQL, le script `mysql_install_db` doit être lancé afin de créer les bases mysql et test.

L'étape suivante consiste à la mise en place des utilisateurs :

```
Mot de passe du root : mysqladmin −u root password 'mot−de−passe'
```

Pour ajouter un utilisateur phil qui sera un super utilisateur avec les mêmes droits que root :

```
$ mysql −u root −p mysql
Enter password:
Reading table information for completion of table and column names
You can turn off this feature to get a quicker startup with −A

Welcome to the MySQL monitor. Commands end with ; or \g.
Your MySQL connection id is 2 to server version: 3.23.23−beta−log

Type 'help' for help.

mysql> INSERT INTO user
−> VALUES('localhost','phil',PASSWORD('mot−de−passe'),
−> 'Y','Y','Y','Y','Y','Y','Y','Y','Y','Y','Y','Y','Y','Y');
Query OK, 1 row affected (0.00 sec)
mysql> FLUSH PRIVILEGES;
```

Attention de ne pas remplacer localhost par le nom de votre machine lors de la saisie de la commande SQL ci−dessus.

Vérification du bon fonctionnement de PHP : écrivez un fichier nommé info.php contenant les lignes suivantes :

```
<?
phpinfo();
?>
```

Installation de phpMyAdmin :

phpMyAdmin est une application indispensable qui permet d'administrer entièrement un serveur MySQL en tant que super−utilisateur. Pour un simple utilisateur, il est possible d'utiliser phpMyAdmin pour administrer la base de donnée qui lui est allouée.

Exercice : téléchargez l'application phpMyAdmin sur le site phpinfo.net (http://www.phpinfo.net) (Rubrique Applications puis catégorie Bases de données) puis installer et mettre en oeuvre l'interface d'administration du serveur MySQL.

7.4. APPLICATIONS PHP ET MYSQL PRÊTES À L'EMPLOI

7.4.1. Pourquoi vouloir réinventer la roue ?

Il existe de très nombreuses applications php prêtes à l'emploi couvrant la plupart des domaines : Annuaire, Bannières, Bases de Données, Livre d'Or, WebMail, Moteur de recherche, Portail, Newsletter, E−Commerce, Petites Annonces, Formation, Forum, Sondage, etc..

Avant de vous lancer dans de longs développements, assurez vous que l'application que vous souhaitez n'existe pas déjà... quitte à y effectuer quelques modifications afin de correspondre exactement à vos besoins.

7.4.2. Exemple 1 : installation d'un système de discussion temps−réel phpMyChat

phpMyChat est une application de Chat en PHP, supportant des bases MySQL, PostgreSQL et ODBC. Disponible en de nombreuses langues.

Exercice : téléchargez l'application phpMyChat sur le site phpinfo.net (http://www.phpinfo.net) (Rubrique Applications puis catégorie Chat / IRC) puis installer et mettre en oeuvre le système temps−réel de discussion accessible sous la forme d'un sous−domaine de type chat.nom−machine

7.4.3. Exemple 2 : mise en oeuvre de forums avec Phorum

Phorum est le plus populaire et le plus complet des forums PHP. Phorum peut s'interfacer avec des bases de données MySQL ou PostgreSQL. Il est permet de mettre rapidement en place des forums de discussion en plusieurs langues sur plusieurs thèmes de discussion.

Exercice : téléchargez l'application Phorum sur le site phpinfo.net (http://www.phpinfo.net) (Rubrique Applications puis catégorie Forum) puis installer et mettre en oeuvre le système temps−réel de discussion accessible sous la forme d'un sous−domaine de type forum.nom−machine.

7.5. APACHE, PHP ET MYSQL SOUS WINDOWS

Le package EasyPhp disponible sur le site easyphp.org (http://www.easyphp.org) permet d'installer très facilement PHP, MySQL et PhpMyAdmin sur une machine à base de Windows 98 ou Windows NT. Les dernières versions de ce package comprennent un programme d'installation qui se charge d'installer à votre place le trio Apache, PHP et MySQL.

La version Windows peut exploitable à la mise en oeuvre d'un serveur d'hébergement professionnel permet de mettre au point en local les sites sous Windows (avec par exemple le logiciel Dreamweaver Macromedia) sans avoir à recourir à de nombreux transfert par ftp.

8. MISE EN OEUVRE D'UN SERVEUR DNS

8.1. INTRODUCTION

Pour résoudre les noms de domaine et les adresses IP, et afin de localiser les machines des réseaux distants, nous avons vu au chapitre *Noms logiques et DNS* qu'il est plus simple de mémoriser des noms que des nombres, surtout lorsqu'on considère le nombre d'adresses sur Internet. Les ordinateurs, au contraire, n'utilisent que les adresses IP pour communiquer via TCP/IP.

Lorsque vous entrez sur Internet en spécifiant une adresse comme, par exemple, http://www.aidburkina.org, votre navigateur envoie une requête au Serveur de Domaine de votre fournisseur d'accès, qui essaie de déterminer l'adresse IP correspondante. Si votre fournisseur n'est pas l'**autorité** pour cette **zone**, il transmet la requête au domaine autorité, jusqu'à ce qu'elle arrive au domaine indiqué.

Cela signifie que chaque serveur de domaine dispose de toutes les informations relatives à la zone qu'il contrôle, et aussi des informations de base sur les autres zones. Quand une requête est envoyée en dehors de la zone d'autorité, le serveur sait au minimum où chercher. Cela signifie que la requête peut avoir à transiter par plusieurs serveurs de domaine avant d'atteindre la destination finale.

Pour l'installation d'un **Serveur de Noms de Domaine (DNS)**, nous utilisons l'application **BIND** (Berkeley Internet Name Daemon). Au préalable, il est nécessaire d'installer les paquetages bind et named. Durant l'installation, le fichier `/etc/named` ainsi que répertoire `/var/named` sont créés.

Nous allons installer le domaine fictif ouaga.bf

8.2. TYPES D'ENREGISTREMENTS ASSOCIÉS À LA DÉFINITION D'UNE ZONE

Enregistrement SOA :	Désigne l'autorité pour le domaine,
Enregistrement NS :	Indique le Serveur de Noms pour ce domaine.

Les enregistrements suivants donnent les informations sur les hôtes du domaine :

A :	translation nom −−> adresse,
PTR :	translation adresse −−> nom (translation inverse),
CNAME :	nom canonique (nom officiel de l'hôte),
TXT :	information libre (texte),
RP :	personne responsable.

Commentaires : les commentaires commencent par un point−virgule et prennent fin avec la ligne.

Enregistrement SOA :

Le premier enregistrement de chaque fichier est l'Autorité Primaire (SOA, Start Of Authority). Cette ligne indique que ce Serveur de Noms est la source primaire d'information sur les hôtes de ce domaine. Notre Serveur de Noms, que nous nommerons ns1, est autorité sur le domaine ouaga.bf du fait de l'enregistrement SOA.

Exemple :

```
ouaga.bf. IN SOA        root.ouaga.bf. (
ns1.ouaga.bf.
                        1 ; Série pour mise à jour
                        10800 ; Mise à jour 3 heures
                        3600 ; Nvelle tentative après 1h
                        604800 ; Expire après 1 semaine
                        86400 ) ; Minimum TTL 1 semaine
```

Le point à la fin des noms est très important !

Enregistrement NS :

L'enregistrement NS (Name Server) de notre domaine ouaga.bf est :

```
ouaga.bf. IN NS ns1.ouaga.bf.
ouaga.bf. IN NS ns2.ouaga.bf.
```

Ces enregistrements indiquent que deux Serveurs de Noms existent pour le domaine ouaga.bf. Les Serveurs de Noms sont installés sur les hôtes ns1 et ns2.

Enregistrements d'adresses et d'alias :

Il reste à indiquer les correspondances adresses −−> noms d'hôtes.

```
A record
; Adresses des hôtes

localhost.ouaga.bf. IN A 127.0.0.1
machine1.ouaga.bf. IN A 192.253.253.2
ns1.ouaga.bf. IN A 192.253.253.3
ns2.ouaga.bf. IN A 192.253.253.4

; Alias

machine2.ouaga.bf. IN CNAME ns1.ouaga.bf.
```

Dans le premier blocs, le "A" indique une adresse. Le deuxième est la table des alias : nous employons un enregistrement CNAME (nom canonique, nom d'hôte complet).

Enregistrements PTR :

Les enregistrements de type PTR (pointeur) correspondent à enregistrement par hôte. Ils permettent la résolution inverse (retrouver un nom de domaine à partir de l'adresse IP). Les adresses sont inversées, puis "in−addr.arpa" est ajouté.

```
; Enregistrements PTR
2.249.249.192.in−addr.arpa. IN PTR ns1.ouaga.bf.
3.249.249.192.in−addr.arpa. IN PTR machine1.ouaga.bf.
```

8.3. CONFIGURATION D'UN SERVEUR DNS

Etape 1 : Modification du fichier `/etc/named.conf` afin de spécifier les zones dont notre serveur a autorité ainsi que pour spécifier les adresses DNS spécifiés par notre fournisseur d'accès.

Extrait du fichier de configuration (jupiter.ouaga.bf étant le serveur DNS administrant la zone ouaga.bf) :

```
options {
directory "/var/named";
forwarders {
212.52.129.34;
};
};

zone "." {
type hint;
file "root.hints";
};

zone "ouaga.bf" {
type master;
file "ouaga.bf";
};

zone "0.0.127.in-addr.arpa" {
notify no;
type master;
file "127.0.0.rev";
};

zone "20.90.196.in-addr.arpa" {
notify no;
type master;
file "196.90.20.rev";
};
```

Pour définir un serveur DNS secondaire, il faudrait spécifier :

```
zone "ouaga.bf" {
type slave;
file "ouaga.bf";
masters {192.253.253.3};
};
```

Etape 2 : Modification du fichier `/var/named/named.local` : on remplace localhost par ns1.ouaga.bf (excepté pour le dernier localhost de ce fichier).

```
@ IN SOA jupiter.ouaga.bf. root.ouaga.bf. (
2001042401 ; Serial
28800 ; Refresh
14400 ; Retry
3600000 ; Expire
86400 ) ; Minimum
IN NS jupiter.ouaga.bf.

1 IN PTR localhost.
```

Etape 3 : Création sous /var/named du fichier ouaga.bf contenant :

```
      @ IN SOA jupiter.ouaga.bf. root.ouaga.bf. (
      2001042502; Serial
      28800 ; Refresh
      14400 ; Retry
      3600000 ; Expire
      86400 ) ; Minimum
      IN NS jupiter.ouaga.bf.
      IN MX 5 196.90.20.102

localhost         IN  A     127.0.0.1
graphisiao        IN  A     196.90.20.101
nasso             IN  A     196.90.20.102
hiper             IN  A     196.90.20.103
onatel            IN  A     196.90.20.104
jupiter           IN  A     196.90.20.105
makaya            IN  A     196.90.20.106
bib               IN  A     196.90.20.107
lonab             IN  A     196.90.20.108
ziga              IN  A     196.90.20.109
cenatrin          IN  A     196.90.20.110
datasys           IN  A     196.90.20.111
mars              IN  A     196.90.20.112

www               IN  CNAME ziga.ouaga.bf.
www.lonab         IN  CNAME lonab.ouaga.bf.
www.chat.lonab    IN  CNAME lonab.ouaga.bf.
chat.lonab        IN  CNAME lonab.ouaga.bf.

www.ziga          IN  CNAME ziga.ouaga.bf.
private.ziga      IN  CNAME ziga.ouaga.bf.
www.private       IN  CNAME ziga.ouaga.bf.
chat.ziga         IN  CNAME ziga.ouaga.bf.
www.chat.ziga     IN  CNAME ziga.ouaga.bf.
forum.ziga        IN  CNAME ziga.ouaga.bf.
www.forum.ziga    IN  CNAME ziga.ouaga.bf.

www.hiper         IN  CNAME hiper.ouaga.bf.
chat.hiper        IN  CNAME hiper.ouaga.bf.
www.chat.hiper    IN  CNAME hiper.ouaga.bf.

www.datasys       IN  CNAME datasys.ouaga.bf.
www.chat.datasys  IN  CNAME datasys.ouaga.bf.
chat.datasys      IN  CNAME datasys.ouaga.bf.

www.cenatrin      IN  CNAME cenatrin.ouaga.bf.
private.cenatrin  IN  CNAME cenatrin.ouaga.bf.
chat.cenatrin     IN  CNAME cenatrin.ouaga.bf.
www.chat.cenatrinIN  CNAME cenatrin.ouaga.bf.

www.planete       IN  CNAME planete.ouaga.bf.
www.chat.planete  IN  CNAME planete.ouaga.bf.
chat.planete      IN  CNAME planete.ouaga.bf.
```

```
www.graphisiao   IN  CNAME graphisiao.ouaga.bf.
www.chat         IN  CNAME graphisiao.ouaga.bf.
chat             IN  CNAME graphisiao.ouaga.bf.
www.mairie       IN  CNAME graphisiao.ouaga.bf.
mairie           IN  CNAME graphisiao.ouaga.bf.
www.mairie       IN  CNAME graphisiao.ouaga.bf.
ouaga            IN  CNAME graphisiao.ouaga.bf.
```

Puis dans le fichier `/etc/nsswitch.conf` on doit avoir à la ligne hosts :

```
hosts: files dns
```

Etape 4 : rédémarrage du serveur DNS par la commande `/etc/rc.d/init.d/named restart`

8.4. TEST DU SERVEUR DNS

Pour terminer la mise en oeuvre du serveur DNS, il est nécessaire de modifier le fichier `/etc/resolve` en y plaçant :

```
domain ouaga.bf
nameserver 192.249.249.2
```

Le test du bon fonctionnement de notre serveur DNS s'effectue avec la commande `nslookup`.

9. MISE EN OEUVRE D'UN SERVEUR DE MESSAGERIE

9.1. INTRODUCTION

La gestion du courrier électronique a longtemps été un problème difficile sous les environnements de type UNIX. Ceci prinicpalement du fait de la complexité de paramétrage du logiciel Sendmail. Heureusement, des logiciels efficaces et plus simples sont apparus pour traiter la gestion du courrier électronique.

Deux protocoles sont utilisés pour le courrier électronique : **SMTP** (Simple Mail Transfer Protocol) pour envoyer du courrier le client de ce protocole est celui qui envoie les courriers) et **POP** (Post Office Protocol) pour en recevoir (le client est celui qui les reçoit). Cet aiguillage permet aux machines qui ne peuvent pas se permettre d'être des serveurs (parce qu'elles sont trop souvent éteintes) de communiquer tout de même par courrier électronique.

9.2. LA ZONE DE STOCKAGE DU COURRIER ÉLECTRONIQUE /VAR/SPOOL/MAIL

Sous Linux, chaque utilisateur dispose d'un fichier à son nom (user) dans le répertoire `/var/spool/mail` : c'est sa boîte aux lettres (mailbox). Pour lire son courrier électronique, de nombreux programmes sont disponibles ; tous lisent ce fichier. Le démon de courrier électronique, quant à lui, y ajoute les courriers reçus.

Postfix

Ce démon gère une file d'attente de courrier électronique. Il permet de recevoir le courrier à destination de la machine locale. Le transfert de courrier de poste à poste peut poser beaucoup plus de problèmes, et nécessite l'emploi d'un service DNS. Cependant, l'installation par défaut permet de gérer le courrier électronique en Intranet : éditer et modifier le fichier `/etc/postfix/main.cf`, démarrer le démon postfix et le tour est joué !

Avec Postfix, finie la syntaxe complexe et difficilement compréhensible du `sendmail.cf` de SendMail. Il n'est plus nécessaire d'utiliser un ensemble de macros pour configurer le serveur de courrier. Le paramétrage de postfix consiste à adapter un seul fichier nommé `main.cf`

9.3. INSTALLATION DE POSTFIX, UNE ALTERNATIVE À SENDMAIL

9.3.1. Cas où sendmail est déjà installé

Si vous ne voulez pas vous retrouvez dans des situations complexes de configuration, la solution consiste à oublier sendmail. Pour que postfix devienne votre agent de transport de courrier, vous devrez tout d'abord désactiver sendmail. Même si vous ne l'avez jamais activé ou configuré, si vous utilisez une distribution autre que la Mandrake, il y a beaucoup de chance pour que sendmail ait été installé par défaut lors de l'installation de votre distribution Linux.

Pour désactiver sendmail, il faut suivre les commandes suivantes :

```
# cd /usr/sbin
# mv sendmail sendmail.OFF
# ./sendmail.OFF −q
# mv /usr/bin/newaliases /usr/bin/newaliases.OFF
# mv /usr/bin/mailq /usr/bin/mailq.OFF
# chmod 0 /usr/sbin/sendmail.OFF /usr/bin/newaliases.OFF \
/usr/bin/mailq.OFF
# ln −s /usr/local/sbin/sendmail /usr/sbin/sendmail
# ln −s /usr/local/sbin/sendmail /usr/bin/mailq
# ln −s /usr/local/sbin/sendmail /usr/bin/newaliases
```

9.3.2. Configuration

Nous supposerons que votre machine s'appelle nasso et que votre domaine s'appelle ouaga.bf.

La majeure partie du travail de configuration consiste à adapter le fichier `/etc/postfix/main.cf` (ou `/usr/local/etc/postfix/main.cf`) à nos besoins. Les modifications doivent être effectuée à partir de l'utilisateur root.

Avant toute modification, il est prudent de sauvegarder le fichier original :

```
# cp /etc/postfix/main.cf /etc/postfix/main.cf.reference
```

Puis, éditer `main.cf` :

Les options par défaut sont déjà configurés et vous n'avez à modifier que les paramètres suivants :

```
# INFORMATIONS SUR LES REPERTOIRES LOCAUX
queue_directory = /var/spool/postfix
command_directory = /usr/local/sbin
daemon_directory = /usr/local/libexec/postfix

# POSSESSION DES FILES D'ATTENTE ET DES PROCESSUS
mail_owner = postfix

# NOMS DE LA MACHINE ET DU DOMAINE
myhostname = nasso.ouaga.bf

# POUR L'ENVOI DU COURRIER
myorigin = $myhostname

# MODE DE TRANSPORT
default_transport = smtp

# GESTION DES ALIAS
alias_maps = hash:/etc/postfix/aliases
alias_database = hash:/etc/postfix/aliases

# DELIVRANCE DU COURRIER
mailbox_command = /usr/local/bin/procmail
```

La première section sert à spécifier les emplacements :

- `/var/spool/postfix` est le répertoire de base pour toutes les files d'attente de postfix, Lors de son premier lancement, postfix créera tous les sous−répertoires pour ses files sous ce répertoire ;

- `/usr/local/sbin<code> est le répertoire où se trouvent les commandes de postfix (les exécutables dont le nom commence par post, et sa version de sendmail) ;`

  `−* <code>/usr/local/libexec/postfix`
  est le répertoire contenant les démons de postfix : c'est là que se trouvent tous les programmes serveurs qu'il utilise.

La deuxième section précise qui est le propriétaire de la file d'attente et de la plupart des processus serveurs de postfix. Ici, nous avons conservé la proposition, après avoir créé l'utilisateur postfix. Voici son entrée dans notre fichier `/etc/passwd` :

```
postfix:x:101:101::/var/spool/postfix:/bin/false
```

Le 'x' dans la partie mot de passe vient du fait que nous utilisons des mots de passe cachés. Le groupe 101 correspond au groupe postfix, lui aussi créé pour l'occasion :

```
# grep 101 /etc/group
postfix:x:101:
```

La troisième section sert à indiquer le nom complet de notre machine.

La section suivante concerne l'envoi du courrier : elle permet de renseigner postfix sur la machine qui a posté.

Il faut ensuite préciser le protocole utilisé pour l'acheminement du courrier. Par défaut, postfix ne reconnaît que smtp et uucp. Mais il est possible de créer des transports dans `/etc/postfix/master.cf`, ce qui permet de changer des paramètres en fonction de multiples critères. Il est ainsi possible dupliquer le transport smtp et en changer les caractéristiques en fonction des courriers entrants ou sortants ce qui est très souple.

La gestion des alias peut s'effectuer par le fichier `/etc/aliases` tout comme pour sendmail. Mais il est préférable d'utiliser le fichier de configuration spécifique à Postfix `/etc/postfix/aliases` (ou `/usr/local/etc/postfix/aliases` (utiliser la commande `man aliases` pour obtenir des détails sur le format de ce fichier). Ce fichier permet de définir des alias entre des noms de destinataires. Le paramétrage des alias tout comme la plupart des paramètres de Postfix peut être effectué à partir de l'interfae Webmin, ce qui évite de connaitre la syntaxe.

Pour des raisons d'optimisation, postfix, tout comme sendmail, exige que les fichiers tels que le fichier d'alias soit traité comme une base de données au format DBM ou DB. Pour générer ces formats, on utilise l'utilitaire `/usr/sbin/postalias`. A noter qu'en utilisant Webmin, cette opération est effectuée automatiquement.

En adoptant le format DB la commande pour générer le fichier `/etc/postfix/aliases.db` à partir du fichier `/etc/postfix/aliases` est :

```
# postalias hash:/etc/postfix/aliases
```

9.3.3. Connexion au serveur pour accéder à sa boîte aux lettres

La consultation du courrier électronique depuis une machine distante s'effectue en utilisant soit le protocole imap soit le protocole pop3.

Le protocole imap présente de nombreux avantages mais étant donné que de nombreux clients ne le supportent pas, nous sommes obligés d'utiliser au moins pop3.

Pour activer les services pop3 et/ou imap, il suffit d'enlever les commentaires devant ces services dans le fichier `/etc/services` et `/etc/inetd.conf` puis de relancer le démon inetd.

Il existe de nombreuses interfaces WebMail en PHP très professionnels. L'application IMP disponible sur phpinfo.net (http://www.phpinfo.net) est la plus utilisée actuellement chez les providers professionnels. IMP étant assez diffcile à configure, on peut trés bien lui préférer une interface WebMail telle que SquirrelMail également téléchargable à partir du site phinfo.net : il s'agit application PHP très simple à installer et sufisamment complète pour couvrir les besoins des utilisateurs.

10. UTILISATION D'EXIM COMME SERVEUR DE MAIL

10.1. INTRODUCTION

L' e−mail est certainement l'application la plus utilisée sur internet. Le fonctionnement repose sur deux programmes distincts, le MUA et le MTA.

Le MUA (Mail User Agent) est un programme chargé de présenter les e−mails à l'utilisateur et ce programme ne s'ocuppe pas de la manière dont voyage les messages sur le réseau.

Le MTA (Mail Transfert Agent) est responsable de l'acheminement des e−mails entres les ordinateurs. Pour ce faire le MTA utilise le protocole SMTP (Simple Mail Transfert Protocol).

Le protocole SMTP spécifie la manière de transferer des e−mails entre des machines, pourtant quand nous envoyons un e−mail à toto@lesite.com le destinataire reçoit le message bien que lesite.com désigne un domaine et non pas un machine. Ceci est possible grace à une astuce au niveau des DNS où est spécifié une machine qui gère le mail pour tout le domaine lesite.com .

10.2. INSTALLATION D'EXIM

Sur une distribution Debian, exim est le MTA par défaut et devrait être installé, on peut le vérifier par la commande :

```
debian:~# exim −bP configure_file
/etc/exim/exim.conf
debian:~#
```

si exim n'est pas installé, il suffit de lancer la commande :
`debian:~# apt−get install exim`

10.3. CONFIGURATION D'EXIM

Il faut maintenant éditer le fichier `/etc/exim/exim.conf` pour configurer le fonctionnement d'exim à notre situation.

Pour l'exemple nous allons mettre en oeuvre un serveur de mail pour un domaine fictif appelé monsite.com.

La première chose à indiquer dans le fichier de configuration est le domaine qui est utilisé par exim pour envoyer les e−mails depuis l'hôte local :
`qualify_domain = monsite.com`

Puis on spécifie la liste des domaines gérés par exim :
`local_domains = localhost:monsite.com` on peut ajouter d'autres domaines en les ajoutants simplement sur la même ligne séparés par : .

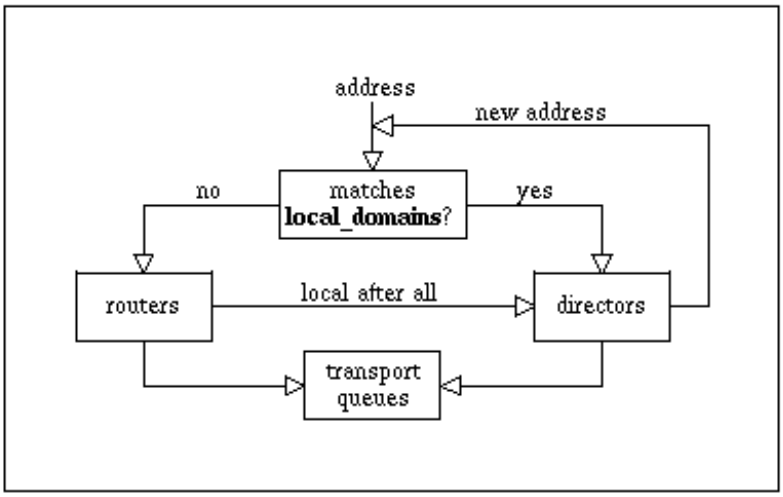
Pour que tout le monde ne puisse pas utiliser notre serveur de mail pour envoyer du SPAM, nous précisons quels sont les machines autorisées à envoyer des e−mails :
`host_accept_relay = 127.0.0.1` Dans cette configuration seul l'ordinateur sur lequel est exécuté exim pourra envoyer des e−mails. Si on veut que les utilisateurs d'un réseau local puisse envoyer des e−mails on utilisera :

```
host_accept_relay = 127.0.0.1:192.168.0.0/24
```

Ainsi toutes les machines dont l'adresse IP est 192.168.0.X pourront utiliser exim pour envoyer des e−mails.

Avec la ligne :
`smtp_accept_queue_per_connection = 100` nous spécifions qu'un maximum de 100 messages seront envoyés pour une conexions, si il y a des messages suplémentaires ils seront simplement mis en attente et seront envoyés plus tard.

Après ces configurations d'ordre général, nous allons voir coment fonctionnne le mecanisme de livraison des messages dans exim.



Driver interactions

Le traitement du courrier se fait par le biais de trois pilotes différents : les transports, les Directors, et les Routers.

Transports

Un transport est un pilote qui prends un message d'une file d'attente pour le transmettre à un destinataire.

Voici les deux Transports les plus importants qui sont définis dans le fichier de configuration par défaut :

```
local_delivery:
 driver = appendfile
 group = mail
 mode = 0660
 mode_fail_narrower = false
 envelope_to_add = true
 return_path_add = true
 file = /var/spool/mail/${local_part}
```

Le Transport `local_delivery` défini la manière de livrer un message à un utilisateur local. Le driver appendfile (ajoute_au_fichier) est utiliser pour ajouter le message dans le fichier de l'utilisateur concerné (`/var/spool/mail/utilisateur`).

```
remote_smtp:
 driver = smtp
```

Le Transport `remote_smtp` permet le traitement des redirections des messages via le protocole SMTP.

La définition des Transports se termine avec la ligne :
`end`

Directors

Un Director détermine comment un message doit être livré, un Director prends une décision mais ne fait pas de livraison lui même. Il désigne un ou plusieurs transports pour effectuer la livraison.

```
real_local:
 prefix = real-
 driver = localuser
 transport = local_delivery
```

Ce director désigne un message réellement local, c'est à dire un message d'un utilisateur local pour un autre utilisateur local. Le transport sélectionné est alors `local_delivery`

```
userforward:
 driver = forwardfile
 file_transport = address_file
 pipe_transport = address_pipe
 reply_transport = address_reply
 no_verify
 check_ancestor
 check_local_user
 file = .forward
 modemask = 002
 filter
```

Ce director est chargé de faire appliquer les directives lues dans le fichier `.forward` de l'utilisateur.

```
localuser:
 driver = localuser
 transport = local_delivery
```

Et le plus utilisé des directors, pour la livraison des messages aux utilisateurs locaux.

Il y a d'autes utilisations des directors, par exemple pour gérer des hôtes virtuels, quand une seule machine gère le mail de plusieurs domaines.

Sur une distribution débian, il est possible d'obtenir une configuration fonctionnelle dans la pluspart des cas en utilisant le script de configuration `eximconfig`.

Utilisation d'eximconfig

En tant qu'utilisateur root, lancer la commande `/usr/sbin/eximconfig`

Dans la liste de 1 à 5 il faut choisir le 1, dans le cas d'une machine avec une connexion permanente. Et le numéro 2 si l'accès au réseau se fait par modem.

Dans la deuxième question il faut indiquer le nom du domaine que vous utilisez. Par exemple monsite.com .

Dans la troisième question vous pouvez spécifier d'autres noms de domaine en plus de monsite.com

La quatrième question vous permet de préciser quels autres domaines ont le droit d'utiliser le serveur de mail. Il est prudent de laisser la valeur par défaut qui est none. Si vous devez modifier ce paramètre relisez bien la documentation d'exim pour être sur de ce vous faites.

La cinquième question est utile dans le cas où vous avez un réseau local dans lequel les ordinateurs peuvent utiliser ce serveur de mail. Il faudra indiquer par exemple 192.168.1.0/24 pour authoriser les machines du réseau local.

La sixième question demande le nom d'un utilisateur à contacter pour tout ce qui concerne le serveur de mail.

11. WEBMIN

11.1. QUELQUES MOTS SUR WEBMIN

Quelques mots sur WebMin pour conclure....

11.2. MODIFICATION DU LANGAGE DE L'INTERFACE

Webmin −> Configuration de Webmin −> Langue

11.3. MISE À JOUR DE WEBMIN

Webmin−> Configuration de Webmin −> Mise à jour de Webmin −> Dernière version sur www.webmin.com.

Puis en cliquant sur "Mettre à jour Webmin", le programme télécharge automatiquement la dernière version de Webmin puis à l'issu de l'installation Webmin vous demande à nouveau votre login et votre de mot de passe afin de se connecter à la nouvelle version. Toutes vos configurations sont conservées !

Cette procédure de mise à jour est vraiment très pratique et ne nécessite aucun effort : gageons que de nombreux programmes s'appuieront sur le même principe pour les mise à jour automatique !

12. MISE EN PLACE D'UN PARE−FEU (FIREWALL) ET D'UN SERVEUR MANDATAIRE (PROXY)

12.1. UN PARE−FEU : POUR QUOI FAIRE ?

Un pare−feu est une structure destinée à empêcher un feu de la traverser. Dans un immeuble, il s'agit d'un mur qui divise complètement des parties de celui−ci. Les pare−feux Internet sont conçus pour isoler votre réseau local privé de l'Internet.

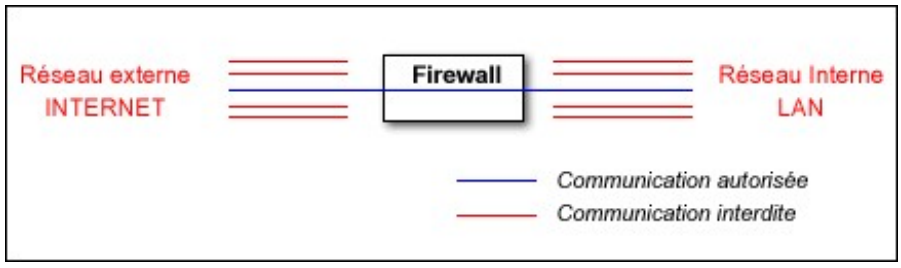
En effet, chaque ordinateur connecté à Internet (ou à n'importe quel réseau d'une manière plus générale) est susceptible d'être victime d'une intrusion pouvant compromettre l'intégrité du système, ou permettre de voler ou d'altérer des données. Cependant, il serait naïf de penser que le danger ne vient que de l'extérieur et dans la réalité les pare−feux ont une double fonction : maintenir les intrus dehors et maintenir les employés dedans.

Il existe deux types de pare−feux :

- les pare−feux IP ou filtrants qui ont pour objectif de bloquer tout le trafic sauf celui sélectionné ;
- les serveurs mandataires qui ont pour objectif de réaliser les connexions résaux pour vous.

12.1.1. Pare−feu filtrant ou firewall

Un pare−feu filtrant fonctionne au niveau du réseau. Les données ne sont autorisées à quitter le système que si les règles du pare−feu le permettent. Lorsque les paquets arrivent, ils sont filtrés en fonction de leurs type, origine, destination et port qui sont décrits dans chacun de ceux−ci.



Un pare−feu pour bloquer le trafic non autorisé

Souvent le routeur − qu'il soit matériel ou logiciel − et le pare−feu ne font qu'un. [A noter que pour désigner un routeur − dont la fonction est de relier deux réseaux − on a tendance à utiliser le terme routeur lorsque la fonction de routage est réalisée matériellement (routeur Cisco par exemple) et passerelle lorsque cette fonction est logicielle (serveur NAT sous Linux)]

Les pare−feux filtrants sont plus transparents que les serveurs mandataires (proxy) pour les utilisateurs. Ceux−ci n'ont en effet pas à configurer des règles dans leurs applications pour utiliser Internet. De plus, les pare−feux filtrants ne fournissent pas de contrôle par mot de passe.

12.1.2. Serveur mandataire ou proxy

Le meilleur exemple du fonctionnement de ceux−ci est celui d'une personne se connectant à un système puis, depuis celui−ci, au reste du monde. C'est seulement avec un serveur mandataire que ce processus est automatique. Lorsque vous vous connectez à l'extérieur, le logiciel client vous connecte en fait d'abord au serveur mandataire. Le serveur mandataire se connecte alors au serveur que vous cherchez à atteindre (l'extérieur) et vous renvoie les données reçues.

Puisque les serveurs mandataires gèrent toutes les communications, ils peuvent enregistrer tout ce qu'ils font (donc ce que vous faites). Pour les mandataires HTTP (web), cela comprend les URL que vous demandez. Pour les mandataires FTP, cela inclut chaque fichier téléchargé. Ils peuvent même expurger les mots "inappropriés" des sites que vous visitez ou analyser la présence de virus.

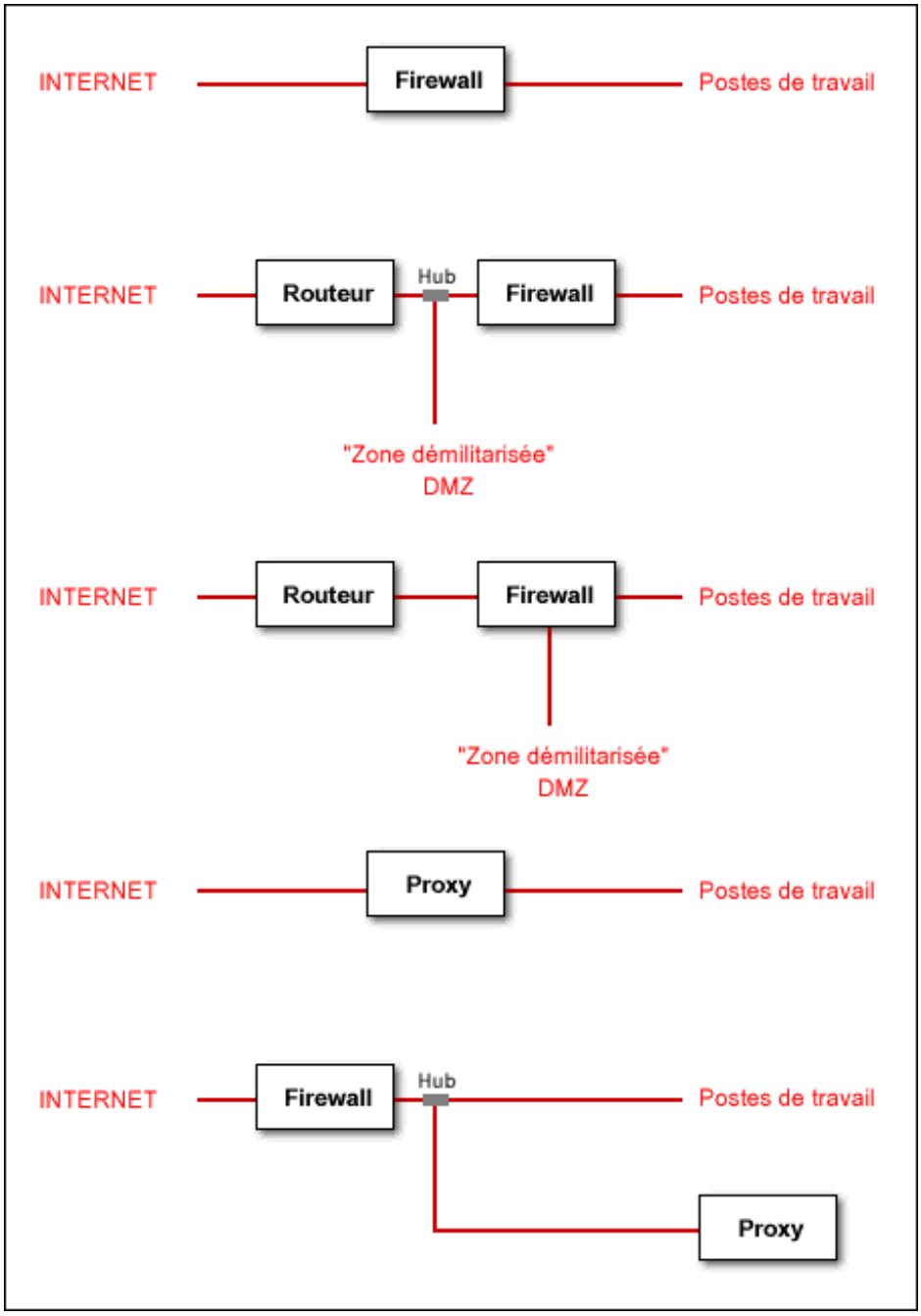
Les serveurs mandataires d'applications peuvent authentifier des utilisateurs. Avant qu'une connexion soit réalisée vers l'extérieur, le serveur peut demander à l'utilisateur de se connecter préalablement. Pour un utilisateur web, cela fonctionnera comme si chaque site requérait une connexion.

Pour résumer, un proxy permet :

- de faire du cache,
- de filtrer certains sites,
- d'interdire l'accès d'Internet à certaines machines (ou certains utilisateurs dans le cas d'utilisateurs itinérants),
- d'interdire le téléchargement,
- de protéger votre réseau,
- de partager l'accès à internet.

12.1.3. Architectures de pare−feu

Il existe de nombreuses manières de structurer un réseau pour protéger des systèmes à l'aide d'un pare−feu.



La stratégie à adopter dépendra de si on fournit par soi−même des services Internet et si on souhaite surveiller les utilisateurs.

L'utilisation d'un pare−feu filtrant et d'un serveur mandataire est complémentaire. Cependant le risque zéro n'existe pas car il faut conserver le contrôle de chaque connexion et il suffit par exemple d'un utilisateur avec un modem pour compromettre tout un réseau local.

12.2. METTRE EN OEUVRE UN FIREWALL FILTRANT

Toutes les fonctionnalités des firewalls filtrants sont directement implémentées dans le noyau Linux. Selon la version du noyau Linux utilisée, il est possible de configurer jusqu'à 3 types de fonctionnalités du firewall filtrant sur un système Linux.

- jusqu'à la version 2.1.102, c'est ipfwadm qui est implémenté,
- depuis la version 2.1.102, on utilise ipchains,
- à partir du noyau 2.4, iptables est implémenté en plus.

Rappelons que pour bénéficier d'un filtrage applicatif, il faut installer sur le système un serveur mandataire ou Proxy tel que Squid.

Contrairement à Ipfwadm et Ipchains, Iptables supporte la translation d'adresse, ce qui permet d'utiliser Iptables à la fois comme serveur NAT (routeur) et comme firewall filtrant.

12.2.1. Avec Ipchains

Ipchains peut filtrer les paquets selon 3 chaînes : ce qui rentre (input), ce qui sort (ouput) et ce qui est transmis (forward). Une chaîne est une vérification de règles. Bien entendu, on peut définir des chaînes différentes suivant l'interface utilisée. Par exemple :

Tout ce qui arrive sur eth0 est filtré d'une manière.
Tout ce qui rentre sur l'interface eth1 est filtré d'une autre manière.
Pour chacune des chaînes, 3 polices peuvent être utilisées :

- on accepte le paquet (ACCEPT),
- on rejette le paquet en prévenant la source qu'on a droppé le paquet (REJECT)
- ou on supprime le paquet directement (DENY).

Note : il n'est pas conseillé d'utiliser REJECT mais plutôt DENY, car lorsque quelqu'un essaye de pirater, il vaut mieux qu'il ne sache pas si le paquet est accepté ou rejeté.

Syntaxe :

Les principales options d'ipchains sont les suivantes :

−P pour policy, sert à redéfinir la police.
−L pour list, sert à lister les règles d'une chaîne
−F pour supprimer les règles d'une chaîne
−A pour ajouter une règle à une chaîne

Pour ce qui est des autres options nous vous invitons à lire le HOWTO d'ipchains.

Syntaxe classique :

```
ipchains −A chaine −i interface −s source −d destination −j police
```

Les chaines : input,output,forward

Les interfaces : eth0,eth1, ppp0,etc..

La source : une adresse ip spécifique (192.168.1.1/24) ou une classe d'adresse ip entière (192.168.1.0/24). Le /24 correspond au masque de sous réseaux (255.255.255.0).

Les polices : accept,deny,reject

Exemples :

Vous voulez mettre une protection « anti−spoofing ». L'anti spoofing est en faite un procédé qui permet à un pirate de changer l'adresse ip source. Donc il pourrait faire croire que c'est quelqu'un qui appartient à notre réseau.

Pour se prémunir de ce type d'attaque, on met des filtres comme celui−ci :

```
ipchains −A input −i ppp0 −s 192.168.1.0/24 −d 0.0.0.0 −j DENY
```

Ce filtre rejette tous les paquets IP qu'y on comme adresse source en 192.168.1.x et qui vient de l'interface eth0.

En théorie ceci est impossible puisqu'une ip en 192.168.1.x est non routable donc ce n'est pas possible que ça vienne de ppp0 (modem) : c'est le principe du spoofing.

Une fois ce filtre validé vous pouvez visualiser s'il a été pris en comptes :

```
ipchains −L input
```

Vous pouvez faire de même pour la sortie. Ipchains est très flexible et très puissant vous pouvez par exemple filtrer suivant un certain protocole ou un certain port.

Pour relancer les règles au démarrage, le mieux est d'écrire un script qui se lance au démarage.

Politiques :

Deux politiques sont envisageables :

1. On rejette tous et on ouvre au fur et à mesure.
2. On ouvre tous et on ferme ce qui est dangereux.

Evidemment la première politique est bien plus sure !

12.2.2. Avec iptables

Configurer le réseau pour le partage de connexion

Si vous désirez mettre en place un partage de connexion Internet, il faut commencer par bien configurer les interfaces réseau du serveur NAT et ne pas oublier d'activer la fonction de forwarding IP au niveau du noyau.

Par exemple, pour un serveur NAT dont l'interface réseau connectée au réseau extérieur est eth0 avec configuration par DHCP et dont l'interface connectée au réseau local est eth1, le fichier /etc/network/interfaces doit ressembler à l'exemple suivant :

```
# /etc/network/interfaces
# Fichier de configuration d'exemple des interfaces réseau
# pour faire un serveur NAT
# Formation Debian GNU/Linux par Alexis de Lattre
# http://www.via.ecp.fr/~alexis/formation−linux/

# Plus d'informations dans "man interfaces"


# L'interface "loopback"
auto lo
iface lo inet loopback
```

```
# L'interface "eth0" connectée à Internet (configuration par DHCP)
auto eth0
iface eth0 inet dhcp

# L'interface "eth1" connectée au réseau local (IP privée fixe)
auto eth1
iface eth1 inet static
   address 192.168.0.1
   netmask 255.255.255.0
   broadcast 192.168.0.255

# Activation de la fonction de forwarding IP au niveau du noyau
up echo "1" >| /proc/sys/net/ipv4/ip_forward
```

Etablir des règles de firewalling

```
#!/bin/sh
# Script "iptables.sh"
# Fichier contenant les règles de filtrage "iptables"
# Formation Debian GNU/Linux par Alexis de Lattre
# http://www.via.ecp.fr/~alexis/formation-linux/

# REMISE à ZERO des règles de filtrage
iptables -F
iptables -t nat -F
iptables -P INPUT ACCEPT
iptables -P FORWARD ACCEPT
iptables -P OUTPUT ACCEPT

# DEBUT des règles de FIREWALLING

# DEBUT des politiques par défaut

# Je veux que les connexions entrantes soient bloquées par défaut
iptables -P INPUT DROP

# Je veux que les connexions destinées à être forwardées
# soient acceptées par défaut
iptables -P FORWARD ACCEPT

# Je veux que les connexions sortantes soient acceptées par défaut
iptables -P OUTPUT ACCEPT

# FIN des politiques par défaut

# J'accepte les packets entrants relatifs à des connexions déjà
établies
iptables -A INPUT -m state --state RELATED,ESTABLISHED -j ACCEPT

# J'autorise les connexions TCP entrantes sur les ports 20 et 21
# (pour que mon serveur FTP soit joignable de l'extérieur)
iptables -A INPUT -p tcp --dport 20 -j ACCEPT
iptables -A INPUT -p tcp --dport 21 -j ACCEPT
```

```
# J'autorise les connexions TCP entrantes sur le port 22
# (pour que mon serveur SSH soit joignable de l'extérieur)
iptables −A INPUT −p tcp −−dport 22 −j ACCEPT

# J'autorise les connexions TCP entrantes sur le port 25
# (pour que mon serveur de mail soit joignable de l'extérieur)
iptables −A INPUT −p tcp −−dport 25 −j ACCEPT

# J'autorise les connexions TCP et UDP entrantes sur le port 53
# (pour que mon serveur DNS soit joignable de l'extérieur)
iptables −A INPUT −p tcp −−dport 53 −j ACCEPT
iptables −A INPUT −p udp −−dport 53 −j ACCEPT

# J'autorise les connexions TCP entrantes sur le port 80
# (pour que mon serveur HTTP soit joignable de l'extérieur)
iptables −A INPUT −p tcp −−dport 80 −j ACCEPT

# J'autorise les connexions TCP et UDP entrantes sur le port 139
# mais uniquement sur l'interface "eth1"
# (pour que mon serveur Samba soit joignable depuis mon LAN
seulement)
iptables −A INPUT −p tcp −−dport 139 −i eth1 −j ACCEPT
iptables −A INPUT −p udp −−dport 139 −i eth1 −j ACCEPT

# J'autorise les connexions UDP entrantes sur le port 177
# (pour que des clients puissent se connecter à mon système par
XDMCP)
iptables −A INPUT −p udp −−dport 177 −j ACCEPT

# J'autorise les connexions TCP entrantes sur le port 6001
# (pour que je puisse me connecter par XDMCP à une machine
distante)
iptables −A INPUT −p tcp −−dport 6001 −j ACCEPT

# J'autorise les connexions TCP entrantes sur le port 2401
# (pour permettre l'accès au CVS à des utilisateurs qui n'ont
# pas de compte sur le système via le "pserver")
iptables −A INPUT −p tcp −−dport 2401 −j ACCEPT

# J'autorise les flux UDP entrants sur le port 1234
# (pour pourvoir reçevoir les flux VideoLAN)
iptables −A INPUT −p udp −−dport 1234 −j ACCEPT

# J'autorise les flux UDP envoyés sur l'adresse multicast
224.2.127.254
# et dont le port destination est 9875 (pour reçevoir les annonces
SAP)
iptables −A INPUT −p udp −d 224.2.127.254 −−dport 9875 −j ACCEPT

# J'autorise les flux TCP et UDP entrants nécessaires au
fonctionnement
# de GnomeMeeting
iptables −A INPUT −p tcp −−dport 30000:33000 −j ACCEPT
iptables −A INPUT −p tcp −−dport 1720 −j ACCEPT
```

```
iptables −A INPUT −p udp −−dport 5000:5006 −j ACCEPT

# J'accepte le protocole ICMP (i.e. le "ping")
iptables −A INPUT −p icmp −j ACCEPT

# J'accepte le protocole IGMP (pour le multicast)
iptables −A INPUT −p igmp −j ACCEPT

# Pas de filtrage sur l'interface de "loopback"
iptables −A INPUT −i lo −j ACCEPT

# La règle par défaut pour la chaine INPUT devient "REJECT"
# (il n'est pas possible de mettre REJECT comme politique par
défaut)
iptables −A INPUT −j REJECT

# FIN des règles de FIREWALLING
# DEBUT des règles pour le PARTAGE DE CONNEXION (i.e. le NAT)

# Je veux que mon système fasse office de "serveur NAT"
# (Remplaçez "eth0" par votre interface connectée à Internet)
iptables −t nat −A POSTROUTING −o eth0 −j MASQUERADE

# FIN des règles pour le PARTAGE DE CONNEXION (i.e. le NAT)


# DEBUT des règles de PORT FORWARDING

# Je veux que les requêtes TCP reçues sur le port 80 soient
forwardées
# à la machine dont l'IP est 192.168.0.3 sur son port 80
# (la réponse à la requête sera forwardée au client)
iptables −t nat −A PREROUTING −p tcp −−dport 80 −j DNAT
−−to−destination 192.168.0.3:80

# FIN des règles de PORT FORWARDING
```

Pour afficher la configuration iptables actuelle, tapez :

- pour la table filter : # iptables −L −v
- pour la table nat : # iptables −t nat −L −v

Dès que les règles iptables en fonctionnement sont satisfaisantes, enregistrez−les comme étant les règles du mode actif :

```
# /etc/init.d/iptables save active
```

12.3. METTRE EN OEUVRE LE PROXY SQUID

Installez Squid avec urpmi squid ou apt−get squid selon que vous soyez sour la Mandrake ou la Debian.

La configuration du proxy s'effectue à partir du fichier /etc/squid.

Pour le détail de la configuration de Squid, se reporter à l'excellent article Configurer Squid comme serveur proxy de LinuxFocus (http://www.linuxfocus.org/Francais/March2002/article235.shtml)

12.4. INSTALLER UNE PASSERELLE, UN PARE−FEU, UN PROXY ET UN SERVEUR D'APPLICATIONS INTRANET EN UNE HEURE ?

Oui c'est possible. Grâce à la E−SMITH (SME), une distribution dérivée de RedHat et qui permet à un débutant d'installer en quelques minutes un PC en serveurs de fichiers, passerelle proxy FTP, firewall, etc..

Pour plus de renseignements et pour graver l'image ISO, rendez vous sur le site http://www.e−smith.org

13. LA SÉCURITÉ

13.1. INTRODUCTION

La sécurité ? C'est probablement le point le plus important pour un administrateur ! LINUX est tellement imposant, constituée de centaines de programmes, qu'il est facile de céder à l'idée qu'il est impossible de sécuriser une machine à 100% et qu'il ne sert à rien d'essayer. Un autre lieu commun et de croire que l'on ne risque rien parce qu'il y a des millions de machines sur Internet et qu'il est peu probable que se soit la notre qui soit attaquée. Il n'y a rien de plus faux, aujourd'hui les crackers disposent d'outils automatiques qui essayent des centaines de failles de sécurité sur plusieurs machines en quelques minutes.

Des vers sont régulièrement lancés, il s'agit de programmes qui cherchent une faille connue sur un logiciel ou un système d'exploitation il s'installent alors sur la machine hôte et continuent à partir de cette machine à explorer le reste du réseau. Ces vers qui sont quasiment aussi vieux qu'Internet même s'ils ne sont pas lancés dans un but destructif peuvent empêcher votre réseau de fonctionner en lançant des millers de requètes par seconde.

Il y a deux grands types d'attaque, celles qui proviennent du réseau et celles qui proviennent d'un utilisateur de la machine.

Nous allons voir comment un système Linux peut être cracké et quelles mesures prendre pour que cela n'arrive pas.

13.2. LES FAMILLES D'ATTAQUE

13.2.1. Permission sur les fichiers

Il est facile de voir qu'un répertoire avec une permission 660 appartenant à root :admin ne peut être écrit par jdupont s'il n'appartient pas au groupe admin. Cela devient moins facile avec 1000 répertoires et des centaines d'utilisateurs et de groupes.

13.2.2. Variables d'environnement

Il y a de nombreux moyens pour créer et lire des variables d'environement pour soit exploiter une faille de sécurité soit accéder à des données sensibles. Les variables d'environement ne doivent contenir aucune information secrète tel que des passwords. D'un autre coté un programme qui lit une ou plusieurs variables d'environement doit vérifier la cohérence des données qu'elles contiennent et traiter les informations comme venant d'une source non sure.

13.2.3. Lecture des passwords sur le réseau

Les programmes tel que `telnet`, `ftp`, `rlogin`, `pop` ou tout autre programme qui s'identifie sur le réseau sans encryption transmet le password en clair. N'importe qui avec un accés physique au réseau peut récupérer des mots de passe sans pourvoir être détecté. Un simple logiciel comme ethereal sur Linux permet de récupérer ces mots de passe.

L'utilisation de tels logiciels est à PROSCRIRE. Il faut utiliser les outils d'open−ssh qui rendent les mêmes services mais qui utilisent une couche de cryptage.

Le seul cas où ces programmes peuvent être utilisé est sur un réseau local qui dispose d'un firewall. De toutes manières aucune machine directement reliée sur internet ne doit accepter un telnet ou un ftp non sécurisé.

13.2.4. Dénie de service

Ces attaques ne sont pas dirigées contre un service ou un ordinateur en particulier mais visent à saturer la bande passante du réseau et par la empécher les utilisateurs légitime d'utiliser les serveurs. Il n'y a malheureusement pas

grand chose à faire contre ce type d'attaque. La meilleur méthode consiste à utiliser un utilitaire de type etherreal pour voir d'où vient l'attaque et tenter de mettre en place au niveau du firewall des règles empéchant les requètes d'arriver au serveur. Dans un deuxième temps il faut essayer de prévenir les utilisateurs/administrateurs de ces machines pour qu'ils arrètent l'attaque.

13.2.5. Les autres types d'attaque

La liste que nous venons de voir est loin d'être exhaustive de nouvelles attaques sont inventées tous les jours. Assurer la sécurité d'un système est un travail quotidien et la prise d'information en est une partie importante.

13.2.6. Buffer overflow

Considérons le programme en C suivant, si vous ne connaisser pas le C ce n'est pas très important c'est l'idée qui est importante.

```
#include <stdio.h>
void do_echo (void){
char buf[256];
gets (buf);
printf ("%s", buf);
fflush (stdout);
}

int main (int argc, char **argv)
{
for (;;) {
do_echo ();
}
}
```

On compile ce programme en faisant : `gcc −o /usr/local/sbin/myechod myechod.c`
Pour en faire un service il faut pour `xinetd`, créer le fichier `/etc/xinetd.d/myechod` qui contient :

```
service myechod
{
flags = REUSE
socket_type = stream
wait = no
user = root
server = /usr/local/sbin/myechod
log_on_failure += USERID
}
```

Tandis que pour `inetd` il faut ajouter la ligne suivante au fichier `/etc/inetd.conf` :

```
myechod stream tcp nowait root /usr/local/sbin/myechod
```

Le service `myechod` n'existant pas il ajouter la ligne suivante dans le fichier `/etc/services` :

```
myechod 400/tcp # Temporary demo service
```

Puis il faut relancer `inetd` ou `xinetd` de la façon habituelle. En faisant un `netstat −na` vous devriez voir la ligne suivante :

```
tcp 0 0 0.0.0.0:400 0.0.0.0:* LISTEN
```

Vous pouvez maintenant faire un `telnet localhost 400` et taper quelques lignes qui devraient être réaffichées par le serveur. En lisant le code on peut s'apercevoir que l'entrée d'une ligne de plus de 256 caractère va joyeusement provoquer l'écrasement d'une zone mémoire qui n'est pas réservée à notre usage.

Comment peut on utiliser cette erreur de programmation pour changer le comportement du programme ? La réponse est simple. Si on écrit des instruction en language machine dans cette mémoire et que cette partie de la mémoire est exécutée plus tard il est possible de faire ce que l'on veut. En général un cracker se contente d'ouvrir un shell ou il pourra se connecter plus tard.

Ce type d'erreur peut être très difficile à détecter dans un très gros programme, il faut avoir pour règle de vérifier systématiquement la longueur des données copiées en provenance de l'extérieur du programme.

D'une manière générale il faut se méfier des fonction du type `strcpy`, `sprintf`, ou `getwd` qui ne font aucune vérification de longueur sur les données entrées.

13.2.7. Les programmes setuid

Un programme comme `su` doit avoir le bit setuid positionné. Un tel programme doit être lancé avec les privilèges de root pour pouvoir changer l'uid d'un utilisateur en un autre. La logique du programme su fait qu'il n'autorise le changement d'uid qu'en donnant le password correspondant, si ce mécanisme pouvait être changer par quelqu'un il pourrait changer son uid et avoir les privilèges de root.

Tous les programmes setuid doivent être considérés avec beaucoup de suspicion, la pluspart des programmes qui doivent être setuid pour fonctionner tendent à être simple et court pour que leur vérification soit simple. Un très bon contre exemple est le programme `sendmail` qui a par ailleur la réputation méritée d'être un trou de sécurité.

13.2.8. Les programmes client

Imaginons qu'un client ftp se connecte à un site inconnu. Si le server de ce site réponds par un message erroné que le client ftp ne sait gérer (un buffer overflow par exemple). Le site distant peut exécuter du code sur la machine cliente.

13.2.9. Fichiers dans /tmp

Si un programme crée un fichier temporaire dans le répertoire `/tmp` et qu'il est possible de prédire le nom de ce fichier alors il est possible de créer ce fichier par avance ou de modifier son contenu sans que le programme ne le sache.

13.3. COMMENT SE DÉFENDRE ?

Par ordre d'importance il convient d'éliminer les risques connus puis les risques possibles puis d'essayer de rendre le vie plus difficile aux crackers et enfin de lutter activement contre les tentatives d'attaque.

13.3.1. Eliminer les risques connus

Se tenir informé :

De nombreuse faille de sécurité sont utilisées bien après que les correctifs ait été publié, il existe un nombre important de serveurs sur internet qui utilisent des version de programmes reconnues comme ayant des problèmes liés à la sécurité. Une installation de linux à partir d'un CD contient toujours un trou de sécurité. Le plus simple est de s'abonner à la mailing list BugTraq qui diffuse en continu les dernières nouveautés en matière de sécurité, ou

d'aller voir chaque matin le site web de Security Focus qui reprends toutes les informations importantes.

Eliminer les packages obsolètes :

La pluspart des distributions Linux tienne à jour un site web qui regroupe les informations ayant trait à la sécurité, c'est une bonne idée d'y aller faire un tour régulièrement pour être tenu au courant des packages à upgrader. Ceci est le minimum que doit faire un administrateur pour sécuriser une machine.

Permissions des fichiers :

Il est facile de trouver qui sont en écriture pour tout le monde. Il ne devrait y en avoir que quelques un dans /dev et /tmp :

```
find / -perm -2 ! -type l -ls
```

Des fichiers sans propriétaire peuvent être une indication qu'un cracker est rentré sur la machine ou que quelquechose est mal configuré :

```
find / -nouser -o -nogroup -ls
```

Désactiver les services insécures :

Les services insécures sont ceux qui permettent que le password soit lu en clair sur le réseau, ou ne fournissent pas une méthode d'autentification sure. Un service qui n'encrypte pas ses données ne doit pas être utilisé pour s'authentifier sur Internet. On peut citer `ftp`, `telnet`, `rlogin`, `uucp`, `imap`, `pop3` et d'autres. Ils ont tous besoin d'un login et d'un password. A la place il faut utiliser `ssh` et `scp`. Il existe des version sécurisées de `pop3` et `imap` (`spop3` et `simap`), mais il est difficile de trouver des clients qui implante ces protocoles. Une alternative à `pop3` est d'utiliser un web-mail avec une connection https pour l'authentification.
Un autre protocole connu pour sa faiblesse au niveau de l'authentification est NFS, il est fortement déconseillé d'utiliser ce protocole sur une machine reliée à Internet.

13.3.2. Eliminer les risques possibles

Réseau :

Enlever tous les services qui ne sont pas utile. En général `xinetd` (ou `inetd`) sont configurés pour un n ombre impressionant de services. Il faut changer ça pour au contraire n'en garder un minimum. Pour xinetd vous pouvez ajouter la ligne disable=yes dans le fichier du service concerné. Si vous utilisez inetd le fichier `/etc/inetd.conf` ne devrait contenir que quelques lignes. Un bon moyen est de partir d'une configuration vide et d'ajouter les services que vous utiliser vraiement. Un serveur web ne devrait avoir que les services http et ssh activés.

La commande `netstat -npl` donne une liste des services actifs sur la machine.

Les programmes setuid :

Il est facile de trouver les programmes setuid sur une machine :

```
find / -type f -perm +6000 -ls
```

Pour chaque programme trouvé posé vous la question un utilisateur a t il une bonne raison d'utiliser ce programme ? Par exemple le programme ping n'est pas du tout nécessaire aux utilisateurs sur un serveur d'hébergement : `chmod -s /bin/ping`

Rendre la vie plus difficile à un cracker :

Le but est de ne pas aider une attaque en ayant une configuration "standard" cela n'ajoute quasiment rien mais pourra empécher une attaque automatique et retarder une attaque manuelle.

Les partitions en lecture seule :

Il est tout a fait possible de monter la partition `/usr` et les répertoires critiques tel que `/bin` et `/sbin` en lecture seule puisque ce sont des programmes qui ne doivent pas être modifier à moins d'ajouter des programmes ou d'upgder des applications.

Les attributs lecture seule :

Linux a la possibilité de mettre des attributs suplémentaires pour rendre un fichier non modifiable en plus des attributs standard. Ces attributs sont controlés par les commandes chattr et lsattr. Vous pouvez les utiliser pour rendre un fichier de log en addition seul (on ne peut qu'ajouter des données) :

```
chatter +a /var/log/messages
```

ou rendre un fichier non modifiable, par exemple :

```
chatter +i /bin/login
```

est une bonne idée, une encore meilleure est :

```
chatter −R +i /bin /boot /lib /sbin /usr
```

Bien sur l'utilisateur root peut remettre les anciens privilèges.

Surveillance périodique du système :

Il est utile d'écrire ses propre script `cron` pour vérifier si des fichiers ont été changé ou utiliser un package qui vérifie l'intégrité du système par des signatures sur chaque fichier.

Un kernel minimal :

Le kernel est un programme comme les autres et il arrive qu'on y trouve des bug, de plus réduire le kernel au minimum pour faire fonctionner les services de la machine augmente les performances du serveur. Il est bon si possible d'enlever le systeme de module qui peut permettre à un hacker de se cacher complètement.

Le projet OpenWall :

Le projet OpenWall est un patch au kernel qui rends la queue (stack) non exécutable et qui arréte la grande majorité des attaques basées sur des buffer overflow.

Il existe bien d'autres moyens de sécuriser un système linux, vous trouverez prochainement sur cette page des liens sur divers projets relatifs à la sécurité.

Fatal error: Maximum execution time of 30 seconds exceeded in
/data/www/org/g/n/africacomputing.org/www/htdocs/mes_fonctions.php3 on line **26**