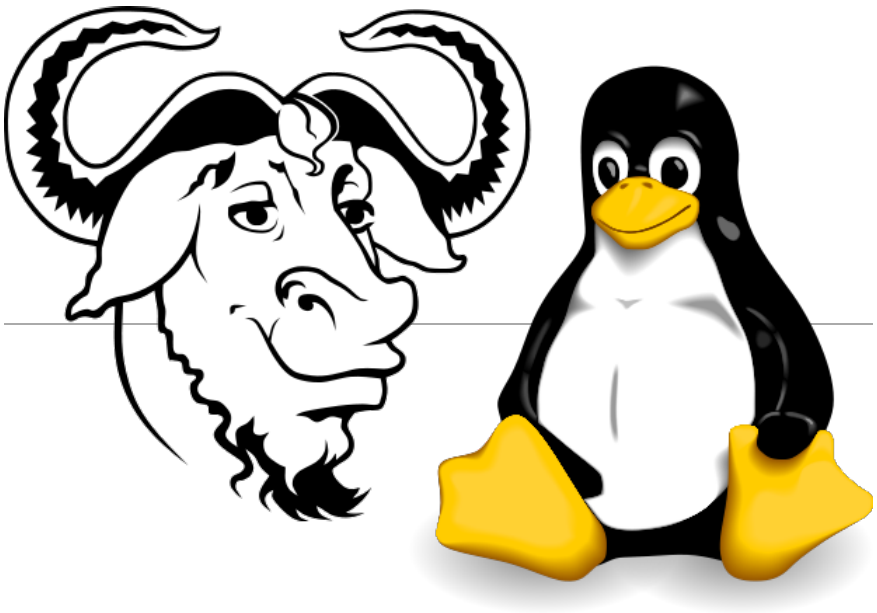


Le système d'exploitation GNU-Linux

Sections

Qu'est-ce qu'un système d'exploitation ?

Linux Is Not UniX.

Objectifs

À la fin de ce chapitre, le lecteur sera en mesure :

- d'expliquer le rôle d'un système d'exploitation,
- de nommer les principales couches de l'architecture d'un système d'exploitation,
- d'associer les générations d'ordinateur aux différents types de systèmes d'exploitation,
- d'expliquer les différentes versions de UNIX et Linux.

Introduction

Un système informatique moderne est composé d'un ou plusieurs processeurs, d'une mémoire principale, de disques durs, d'imprimantes, d'un clavier, d'une souris, d'un écran, d'une carte réseau et de beaucoup d'autres périphériques d'entrée/sortie. En un mot un système complexe. Dans ce contexte, développer des programmes d'application qui doivent tenir compte correctement de toutes ces entrées/sorties n'est pas une mince tâche. C'est pour cette raison que les ordinateurs modernes sont équipés d'un « système d'exploitation ». Une des tâches du système d'exploitation est donc d'offrir aux utilisateurs une interface simple et conviviale avec le matériel.

Dans ce chapitre, nous allons d'abord présenter une brève définition et description des systèmes d'exploitation. Par la suite, nous expliquons l'architecture et les fonctions d'un système d'exploitation. Puis finalement, les différentes versions de Unix et de Linux sont présentées.

Qu'est-ce qu'un système d'exploitation

Un système d'exploitation effectue deux tâches bien distinctes :

- Présenter une machine virtuelle à l'utilisateur.
- Gérer les ressources de l'ordinateur.

Machine virtuelle

Être une machine virtuelle signifie transformer un ensemble de circuits électroniques en un outil moderne qui offre une abstraction simple au niveau des entrées/sorties, de l'utilisation de la mémoire, de la gestion des fichiers, de la protection et du contrôle des erreurs, de l'interaction des programmes entre eux et de leur contrôle.

En deux mots : éviter au programmeur d'avoir à connaître les détails électroniques de tel ou tel microprocesseur et permettre à l'utilisateur de sauvegarder ses fichiers sans se soucier du type de disque utilisé pour stocker les informations.

Gestionnaire de ressources

L'autre fonction du système d'exploitation est le partage des ressources. Le système d'exploitation joue un rôle de policier afin d'éviter les conflits d'utilisation de la mémoire, des périphériques d'entrée/sortie, des interfaces réseau, … On peut facilement imaginer ce qui arriverait si trois programmes essayaient d'imprimer en même temps sans que des priorités aient été préalablement établies.

De plus, lorsque l'ordinateur est utilisé par plusieurs usagers, le partage de la mémoire et surtout sa protection demeurent une priorité absolue. À tout moment, un bon système d'exploitation connaît l'utilisateur d'une ressource, ses droits d'accès et son niveau de priorité.

Architecture d'un système informatique

Tous les systèmes informatiques sont segmentés en « couches » pour permettre un meilleur contrôle de l'ensemble de l'ordinateur. La Figure 1 illustre les différentes couches d'un tel système.

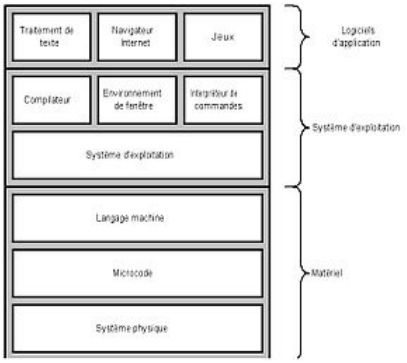


Figure 1 : Architecture d'un système informatique

Matériel

Au plus bas niveau, on retrouve la première couche qui contient les composantes physiques constituées des circuits intégrés, des fils, des sources de courant, …

La couche suivante regroupe des outils logiciels primitifs qui permettent de contrôler directement les composantes physiques sous-jacentes, comme les registres internes du processeur et l'unité arithmétique et logique. Cette couche est appelée le microcode et réside bien souvent dans le processeur de l'ordinateur.

L'autre couche est celle du langage machine qui est interprété par le microcode. Ce langage de bas niveau regroupe 50 à 300 instructions pour permettre de déplacer des bits, de calculer ou de comparer des valeurs à l'aide des registres internes du processeur.

Système d'exploitation

Le système d'exploitation qui se trouve juste au-dessus, offre aux programmeurs et aux utilisateurs un ensemble de fonctions du genre « lire le fichier » ou « afficher à l'écran ». Il s'agit ici d'un niveau d'abstraction élevé qui évite ainsi au programmeur de devoir écrire, par exemple, du code pour déplacer les têtes de lecture d'un disque rigide. Il s'agit du niveau d'exécution des pilotes de périphériques (contrôleurs d'interruptions, de disques, de carte graphique, ...).

En haut de la hiérarchie, il y a la couche où l'on retrouve les interpréteurs de commandes, les compilateurs et les logiciels d'application. Il est clair que ces programmes ne font pas partie du système d'exploitation, même s'ils sont livrés avec celui-ci dans bien des cas.

Logiciels d'application

Finalement, au-dessus de toutes ces couches se trouvent les logiciels d'application qui permettent à un utilisateur d'effectuer des tâches particulières sans qu'il ait à tenir compte des couches inférieures.

Fonctions d'un système d'exploitation

Aujourd'hui, l'informatique, aussi bien dans les entreprises que dans l'enseignement, utilise des machines plus petites fonctionnant avec des systèmes d'exploitation à caractère universel. Parmi ces systèmes d'exploitation, deux se distinguent particulièrement, un système mono-utilisateur, Windows, et un autre multi-utilisateurs et multi-tâches, Unix. D'une manière contestable, on peut affirmer que le premier système est destiné à des ordinateurs individuels, tandis que l'autre est réservé au travail en groupe.

Parmi les nombreux systèmes d'exploitation, Unix/Linux est celui qui offre le plus de richesse, le plus d'homogénéité et le plus de souplesse. Pour cette raison, dans ce livre, Linux a été choisi comme système d'exploitation pour illustrer les concepts théoriques. Par ailleurs, le système MS-DOS puis Windows, en évoluant, ont incorporé beaucoup de caractéristiques de Unix/Linux.

On peut diviser les fonctions des systèmes d'exploitation classiques en quatre parties principales :

1. La gestion des processus (programmes).
2. La gestion de la mémoire.
3. Le système de fichiers.
4. La gestion des entrées/sorties.

Les systèmes d'exploitation modernes intègrent par ailleurs d'autres caractéristiques. Ces dernières concernent notamment deux évolutions majeures des systèmes informatiques. La première est l'interconnexion des différentes machines et des différents systèmes par des réseaux locaux ou étendus. La seconde est la disparition des écrans de texte et leur remplacement par des dispositifs à fenêtres multiples disposant de propriétés graphiques.

Historique des systèmes d'exploitation

L'histoire de l'informatique est très brève – les ordinateurs sont nés avec la seconde guerre mondiale – et pourtant, elle a connu de grandes évolutions. L'historique des systèmes d'exploitation est intimement liée à l'évolution de l'informatique. Cette évolution est séparée en 4 grandes étapes :

1. Première génération (1945-55) : les tubes à vide.
2. Deuxième génération (1955-65) : les transistors et le traitement par lots.
3. Troisième génération (1965-80) : les circuits intégrés et la multiprogrammation.
4. Quatrième génération (1980-aujourd'hui) : les micro-ordinateurs

Première génération (1945-55) : les tubes à vide

En 1946, le premier ordinateur ne comportant plus de pièces mécaniques est créé grâce à J. Mauchly et J. Presper Eckert : l'ENIAC (Electronic Numerical Integrator And Computer). Il est composé de 180 000 tubes à vide et occupe 1500 m². Son principal inconvénient est sa programmation : il était uniquement programmable manuellement avec des commutateurs ou des câbles à enficher.

Note : La première erreur informatique est due à un insecte qui, attiré par la chaleur, était venu se loger dans les tubes à vide et avait créé un court-circuit. Le mot « bug » (insecte en anglais), est resté pour nommer une erreur informatique.

Deuxième génération (1955-65) : les transistors et le traitement par lots

En 1948, le transistor est créé par la firme Bell Labs grâce aux ingénieurs John Bardeen, Walter Brattain et William Shockley. Le transistor permet de rendre les ordinateurs moins encombrants et moins gourmands en énergie électrique : c'est la révolution dans l'histoire de l'informatique. Ce n'est qu'en 1960 qu'IBM commercialise l'un des premiers ordinateurs à base de transistors, l'IBM 7000.

C'est aussi à cette époque que les premiers systèmes d'exploitation sont apparus tel FMS (the Fortran Monitor System) et IBSYS (IBM's operating System), utilisant le traitement par lots pour gérer l'exécution des programmes qui se fait maintenant de façon autonome.

Troisième génération (1965-80) : les circuits intégrés et la multiprogrammation

Le circuit intégré est mis au point en 1958 par Texas Instrument. Il permet de réduire encore la taille des ordinateurs en intégrant plusieurs transistors dans le même composant électronique. Avec cette nouvelle génération d'ordinateurs, une nouvelle génération de systèmes d'exploitation mettant en application le concept de multiprogrammation fait son apparition.

Les premiers travaux sur MULTICS, l'ancêtre d'Unix, sont dus à Ken Thompson pour le compte de Bell Laboratories, AT&T, General Electric et le Massachusetts Institute for Technology.

En 1970, une première version d'Unix voit le jour. Elle fonctionne sur une machine PDP-7 de Digital Equipment. Les principales caractéristiques de ce système sont les suivantes :

- Gestion de fichiers sous forme d'une hiérarchie de répertoires.
- Entrées/sorties gérées de la même façon pour les fichiers, les processus (ou programmes) et les périphériques.
- Gestion multi-tâches de processus indépendants.
- Systèmes à base de commandes pouvant accepter de multiples interpréteurs de commandes, selon les besoins de l'utilisateur.
- Présence d'utilitaires et de compilateurs intégrés dans le système.
- Un système aussi portable que possible pouvant s'adapter à de nombreux types d'ordinateurs. C'est justement pour accroître la portabilité du système que Denis Ritchie inventa le langage de programmation C.

Une grande partie du système Unix fut réécrite en langage C. Le reste du système, en assembleur, dépend du type de machines sur lequel on installe le système. En 1976, la version 6 d'Unix, fut la première version complète. En 1979, ce fut le succès commercial avec la version 7. Cette version a été adoptée par Hewlett Packard et Digital Equipment.

Chaque constructeur eut le droit d'adapter Unix à ses machines. De même, des universités pouvaient travailler à son amélioration. Mais comme le nom d'Unix lui-même

était protégé, chacun dut aussi choisir un nom différent. L'Université de Berkeley en Californie proposa ainsi les versions BSD (Berkeley Software Distribution). Face à cette diversité des versions d'Unix, AT&T proposa, dès 1983, le standard System V. Aujourd'hui, l'X/OPEN group a obtenu le droit de diffuser la marque Unix à tous les systèmes qui se soumettent à un contrôle approprié. Aujourd'hui encore, HP propose HP-UX, IBM propose sa version d'Unix appelée AIX et Sun propose Solaris.

Quatrième génération (1980-aujourd'hui) : les micro-ordinateurs

En 1979, Microsoft prend une licence Unix et développe le système d'exploitation XENIX, disponible jusqu'en 1984. Au moment où le PC fit son apparition, Unix était déjà un système mûr. En 1981, lorsque IBM lança le PC, il choisit le système d'exploitation MS-DOS de Microsoft. MS-DOS, inspiré d'Unix, comportait dès le départ des contraintes liées à son architecture : insuffisance de la mémoire adressable, pas de gestion multi-tâches ou multi-utilisateurs, pas de protection de la mémoire.

Des particuliers eux aussi se sont attelés aux développements d'une version d'Unix : Andrew Tanenbaum, professeur à l'Université d'Amsterdam, développa avec ses élèves une version d'Unix pour PC, appelée Minix, qui vit le jour en 1987. Partant de Minix, Linus Torvalds développa Linux, un Unix pour PC dont la première version, appelée version 0.99, sortit en 1991. Aujourd'hui, Linux est devenu un système d'exploitation stable pour PC. Ce système constitue un logiciel libre, dont chaque utilisateur a le droit de modifier le code source.

Unix et Linux

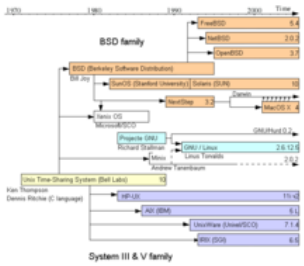
Unix/Linux

Unix est un système d'exploitation très populaire parce qu'il est présent sur un grand nombre de plates-formes, du micro-ordinateur à l'ordinateur central (*mainframe*). L'avantage de cela, c'est que les programmes développés sous Unix peuvent être transférés d'une plate-forme à une autre avec un minimum de modifications.

Ce système est multitâche, c'est-à-dire qu'il est capable de gérer et d'exécuter plusieurs programmes simultanément. De plus, il est multi-utilisateurs, c'est-à-dire que plusieurs personnes peuvent s'y connecter en même temps et travailler ; le système partage alors toutes les ressources logicielles et matérielles de l'ordinateur entre les différents usagers.

L'histoire d'Unix est unique dans le monde des systèmes d'exploitation. En effet, alors que la plupart des systèmes d'exploitation ont été conçus par des fabricants d'ordinateurs pour vendre leurs machines, Unix n'a pas été conçu dans un but commercial. Il l'est devenu parce qu'il constitue une norme en matière de système d'exploitation.

Contrairement à un système d'exploitation commercial complètement contrôlé par son fabricant, le système Unix est aujourd'hui distribué par plusieurs intervenants dont voici les principaux



Historique

- AT&T, à qui on attribue la paternité de Unix ;
- l'université de Berkeley, qui a fait évoluer Unix dans plusieurs domaines ;
- SUN Microsystems, à qui l'on doit les améliorations importantes de l'interface graphique ;
- Santa Cruz Operation et Microsoft, le XENIX/UNIX fut la première version pour PC de Unix.

À cause de cette situation de développement, le système d'exploitation Unix a mis plusieurs années à être standardisé. Actuellement, il en existe deux principales variantes, incompatibles entre elles :

- Unix SYSTEM V,
- Unix BSD.

Il existe aussi une multitude de variations mineures dérivées d'une des deux ou des deux principales variantes ; on a ainsi, en les regroupant :

- les systèmes Unix-Based ;
- les systèmes Unix-Like.

Systèmes Unix-Based

Les systèmes étant "descendants" du système Unix original.

- XENIX/UNIX provenant de SCOMicrosoft ;
- AIX provenant d'IBM ;
- Mac OS X provenant d'APPLE ;
- SunOS/Solaris provenant de SUN MicroSystems ;
- IRIX provenant de Silicon Graphics ;
- ULTRIX provenant de DIGITAL ;
- HP-UX (Hewlett Packard UniX) provenant Hewlett-Packard.
- FreeBSD ;
- OpenBSD ;
- NetBSD ;

Systèmes Unix-Like

Les systèmes Unix-Like reproduisent les mêmes fonctionnalités que la version AT&T, mais le noyau du système est incompatible parce qu'il a été réécrit pour éviter le versement de droit d'auteurs à AT&T. On retrouve dans cette catégorie :

- Minix ;
- GNU/Linux ;

- QNX.

Cette prolifération de produits a fait apparaître un certain nombre de différences entre les systèmes, dont les principales sont :

- les communications inter-programmes ;
- la gestion de la mémoire (segmentation ou pagination) ;
- divers paramètres du système ;
- divers outils qui peuvent être intégrés dans un produit et absents dans un autre.

Principales distributions de Linux

Celui qui s'intéresse aux différentes versions de Linux doit faire la différence entre le noyau du système d'exploitation proprement dit (le *kernel*, en anglais) et la combinaison d'utilitaires qui l'accompagnent. Les distributions se différencient par le choix du noyau et le choix des différents utilitaires disponibles.

Au moment de la rédaction de ce texte, la version actuelle du noyau de Linux porte le numéro 4.2.x, mais les versions suivantes sont probablement prêtes. Vérifiez donc la version du noyau que vous vous procurez. Cette version figure dans de nombreuses distributions et constitue un élément commun.

De nombreux utilitaires sont également communs à toutes les distributions. (Exemples : *Bourne Again Shell* ou l'interface graphique Xfree86).

Voici une liste non exhaustive des distributions disponibles :

- CentOS
- RedHat
- Fedora
- Mandriva
- Debian
- Ubuntu
- OpenSUSE

Voir aussi

Partitionnement du disque

Objectifs

À la fin de ce chapitre, le lecteur sera en mesure :

- d'expliquer le rôle des partitions,
- d'utiliser un outil de partitionnement,
- de créer un système de fichiers.

Introduction

L'utilisation d'une unité de stockage (par exemple un disque dur) soit pour l'installation d'un système d'exploitation ou le stockage de données nécessite que celle-ci soit préalablement préparée d'abord par le partitionnement puis par le formatage logique. Dans ce chapitre, le concept de partition est expliqué avant de présenter un outil de partitionnement. Finalement, le formatage logique est présenté.

Qu'est-ce qu'une partition?

Le partitionnement consiste à créer des zones sur le disque dont les données ne seront pas mélangées. Cela sert donc si l'on veut, par exemple, installer des systèmes d'exploitation différents n'utilisant pas le même système de fichiers. Il y aura donc au minimum autant de partitions que de systèmes d'exploitation utilisant des systèmes de fichiers différents. Dans le cas d'un utilisateur d'un système d'exploitation unique, il y aura une seule partition recouvrant tout le disque, sauf si l'utilisateur désire en créer plusieurs pour, par exemple, séparer les données et les programmes.

Il y a trois sortes de partitions: les partitions principales, la partition étendue et les partitions logiques. Un disque peut contenir jusqu'à quatre partitions principales (dont une seule peut être active), ou trois partitions principales et une partition étendue. Dans la partition étendue, l'utilisateur peut créer des partitions logiques (c'est-à-dire faire en sorte que l'on ait l'impression qu'il y a plusieurs disques durs de taille moindre).

Voyons voir un exemple dans lequel le disque contient une partition principale et une partition étendue composée de trois partitions logiques (nous verrons par la suite les partitions principales multiples):

Figure 1: Exemple de partition

Pour les systèmes Windows, seule la partition principale est amorçable, c'est donc la seule sur laquelle on peut démarrer le système d'exploitation. Sous Linux, toutes les partitions peuvent être amorçables.

On appelle partitionnement le processus qui consiste à écrire les secteurs qui constitueront la table de partitions. La table de partitions est une base de données contenant les informations sur les partitions: taille de celle-ci en terme de nombre de secteurs, position par rapport à la partition principale, types de partitions présentes, systèmes d'exploitation installés, ...

Partition principale

Une partition principale doit être formatée logiquement, puis contenir un système de fichiers correspondant au système d'exploitation installé sur celle-ci. Si jamais vous avez plusieurs partitions principales sur votre disque, une seule sera active et visible à la fois; cela dépendra du système d'exploitation sur lequel vous avez démarré l'ordinateur. En choisissant le système d'exploitation que vous lancez au démarrage, vous déterminez la partition qui sera visible. La partition active est la partition sur laquelle un des systèmes d'exploitation est démarré au lancement de l'ordinateur. Sous Windows, les partitions autres que celle sur laquelle vous démarrez seront alors cachées, ce qui empêchera d'accéder à leurs données. Sous Linux, toutes les partitions sont accessibles.

Partition étendue

La partition étendue a été mise au point pour outrepasser la limite des quatre partitions principales, en ayant la possibilité de créer autant de partitions logiques que vous désirez dans celle-ci. Au moins une partition logique est nécessaire dans une partition étendue, car vous ne pouvez pas y stocker de données directement.

Beaucoup de disques durs d'ordinateur sont formatés en une seule grande partition utilisant l'intégralité de l'espace disponible. Ce n'est pourtant pas la solution la plus avantageuse en terme de performances et de capacité. La solution est de créer plusieurs partitions, ce qui va vous permettre:

- d'installer plusieurs systèmes d'exploitation sur votre disque,
- d'économiser de l'espace disque,
- d'augmenter la sécurité de vos fichiers,
- d'organiser vos données plus facilement.

Secteur de démarrage

Le secteur de démarrage (appelé Master Boot Record ou MBR en anglais) est le premier secteur d'un disque dur (cylindre 0, tête 0 et secteur 1). Il contient la table de partition principale et le code qui, une fois chargé en mémoire, va permettre d'amorcer le système d'exploitation.

Ce programme, une fois en mémoire, va déterminer sur quelle partition le système va s'amorcer, et il va démarrer le programme (appelé « bootstrap ») qui va amorcer le

système d'exploitation présent sur cette partition.

D'autre part, c'est ce secteur du disque qui contient toutes les informations relatives au disque dur (fabricant, numéro de série, nombre d'octets par secteur, nombre de secteurs par « cluster », nombre de secteurs,...). Ce secteur est donc le secteur le plus important du disque dur. Il permet aussi au BIOS de reconnaître le disque dur. Ainsi, sans celui-ci, votre disque dur est inutilisable, c'est donc une des cibles préférées des virus.

Outil de partitionnement

Sous Linux, il existe de nombreux outils de partitionnement. L'utilitaire fdisk est disponible sur la majorité des distributions de Linux. Il permet de créer, d'éditer et de détruire des partitions sur un disque. Le partitionnement avec fdisk entraînera la perte de toutes les données présentes sur le disque sur lequel vous effectuez les opérations.

Syntaxe :

```
fdisk [périphérique]
```

Sans argument fdisk utilisera le premier disque dur qu'il trouve. Il est possible de préciser à fdisk le disque à partitionner, en lui donnant comme paramètre le nom du périphérique.

Exemple :

Partitionner le second disque dur IDE.

```
fdisk /dev/hdb
```

Une fois lancé, le menu suivant apparaît :

```
# fdisk /dev/hda

Commande (m pour aide):
```

Le tableau suivant présente la liste des principales options de la commande fdisk:

Commande	Description
d	**D**estruction d'une partition
l	**L**iste des types de partitions
m	impression du **M**enu en cours
n	création d'une **N**ouvelle partition
p	affichage des **P**artitions
q	Sortie de fdisk sans sauvegarde des paramètres (**Q**uitter)
t	Modification du **T**ype de partition
v	**V**érification de la table des partitions
w	sauvegarde des modifications et sortie de fdisk (**W**rite & exit)

Création des partitions

Voici les étapes pour créer une partition :

1. Démarrer fdisk
2. Taper n pour créer une nouvelle partition.
3. Choisir le type de partition (primaire ou étendu).
4. Choisir le numéro de la partition
5. Ensuite fdisk vous demande l'emplacement du premier cylindre. Par défaut, fdisk affichera toujours le premier cylindre libre trouvé.
6. fdisk vous demande alors l'espace à attribuer à cette partition. Cette taille peut être indiquée en nombre de cylindres, en Octets, en Kilo-octets ou en Mégaoctets.

Afficher les partitions

La commande p du menu principal permet d'afficher les partitions du disque dur sélectionné.

```
Disk /dev/hda : 128 heads, 63 sectors, 623 cylinders
Units = cylinders of 8064 * 512 bytes
```

```
Device     Boot  Start  End     Blocks  Id     System
/dev/hda1  1     254    102400+   83    Linux  native
/dev/hda2  255   309    221760    83    Linux  native
/dev/hda3  310   253    862848    83    Linux  native
```

Formatage logique

Le formatage logique crée un système de fichiers sur le disque, qui va permettre à un système d'exploitation (DOS, Windows 95, Linux, OS/2, Windows NT, ...) d'utiliser l'espace disque pour stocker et utiliser des fichiers. Les systèmes d'exploitation utilisent des systèmes de fichiers différents, ainsi le type de formatage logique dépend du système d'exploitation que vous utilisez.

mkfs, création d'un système de fichiers

Syntaxe :

```
mkfs [-t type] partition
```

Description :

L'utilitaire mkfs permet de créer un système de fichiers sur un disque ou une partition. Ce système de fichiers est de type ext2, ext3 ou msdos.

Exemple 1:

Formater une disquette dos.

```
mkfs –t msdos /dev/fd0
```

Exemple 2:

Formater une disquette Linux.

```
mkfs –t ext2 /dev/fd0
```

Exemple 3 :

Formater la partition /dev/hda3 en ext3.

```
mkfs –t ext3 /dev/hda3
```

e2label, étiquetage d'une partition

Syntaxe :

```
e2label [partition]
```

Description :

La commande e2label affiche ou modifie le nom du système de fichiers spécifié.

Exemple :

Afficher l'étiquette de la partition /dev/hda2

```
e2label /dev/hda2
```

Exercices

1. Nommez trois sortes de partitions?
2. Qu'est-ce qu'une partition?
3. Combien, au maximum, peut-il y avoir de partitions principales?
4. Qu'est-ce qu'un secteur de démarrage?
5. Qu'est-ce que le formatage logique?
6. Formatez une disquette Linux. Écrivez au complet la commande utilisé.

Installation

Installer une distribution Debian

Cette installation a été effectuée à partir du CDROM d'installation via le réseau (Net install) Debian - branche stable (http://www.debian.org/releases/stable/debian-installer/).

La première étape a été de vérifier que l'ordinateur démarrait par défaut sur le CDROM. Si tel est le cas, on doit voir le logo Debian apparaître à l'écran avec un invite "`boot :`".

En appuyant sur les touches de fonction (de F1 à F10), on peut consulter des paramètres optionnels à indiquer au noyau. Ces paramètres sont utiles dans certains cas pour la détection de matériel particulier.

Dans le cas d'une installation normale, on appuie juste sur la touche **Entrée** pour démarrer l'installation.

Le noyau d'installation se charge et détecte le matériel intégré à l'ordinateur.

La première étape du configurateur nous demande de choisir notre langue, notre localisation géographique et la disposition du clavier.

L'étape suivante concerne la détection des paramètres réseaux. Utilisant le protocole DHCP, cette étape s'est déroulée automatiquement. À noter que si cela n'avait pas été le cas, nous aurions du renseigner les paramètres réseaux manuellement : adresse IP, masque de réseau, adresse réseau, passerelle et serveur de noms.

Le processus d'installation détecte automatiquement les périphériques de stockage.

Vient ensuite le partitionnement du disque dur. Nous avons choisi le partitionnement manuel afin de configurer précisément les partitions désirées et leur taille.

Notre disque dur fait 80 Go. Nous avons créé 6 partitions :

- /boot (100Mo) : contient le(s) noyau(x) Linux
- / (20 Go) : la racine du système de fichiers
- /var (20 Go) : les données variables (dont notamment les logs)
- SWAP (2 Go) : la mémoire virtuelle
- /tmp (2 Go) : les fichiers temporaires
- /home (le reste, soit environ 36 Go) : les répertoires utilisateurs

Il est conseillé de créer plusieurs partitions pour des raisons de sécurité. Effectivement, les systèmes d'exploitation ne fonctionnent pas correctement si la partition système est pleine (la racine / sous Unix, ou C: dans le monde Microsoft). Étant donné que l'on fonctionne sur un système multi-utilisateurs, il est donc préférable de créer des partitions dédiées pour /home, /tmp et /var. Effectivement, les utilisateurs peuvent écrire des fichiers dans ces répertoires, et de ce fait les remplir. Si on a prévu une partition dédiée pour ces répertoires, le système va continuer à fonctionner même si ces partitions sont pleines.

L'installateur Debian a ensuite formaté les partitions et installé les paquets essentiels.

Le système nous a ensuite demandé de renseigner les paramètres du compte utilisateur (nom complet, login et mot de passe) et le mot de passe du compte **root**.

Le système nous a demandé ensuite de choisir un miroir Debian afin d'aller télécharger des paquets supplémentaires. Nous avons choisi un miroir en France.

Nous avons eu ensuite la possibilité de configurer un serveur mandataire (un proxy). Nous ne l'avons pas renseigné car notre réseau dispose d'un proxy transparent.

L'installateur Debian a ensuite téléchargé un très grand nombre de paquets supplémentaires.

Une fois le téléchargement terminé (prévoir un certain temps selon le type de connexion à Internet), le système a installé les paquets et nous a demandé de spécifier la résolution d'affichage de l'environnement graphique. Nous avons laissé les paramètres par défaut : 1024x768, 800x600 et 640x480. Il faudra bien entendu choisir la résolution en fonction de l'écran.

Le système nous demande ensuite si on désire installer le programme de boot **GRUB**. Ce programme est effectivement indispensable au bon démarrage du PC. Si GRUB détecte une partition Windows, il va automatiquement la rajouter au multi-boot.

À la suite de ces étapes, l'installation est terminée. Le système nous propose ensuite de redémarrer. Si tout s'est bien passé, l'ordinateur va redémarrer sur le nouveau système fraîchement installé.


Partitionnement


Taille


Caractéristiques

Procédure de récupération du boot

Dans certaines circonstances le boot de Linux peut avoir été supprimé par Windows, il devient alors impossible de démarrer Linux, il est néanmoins possible de le restituer en utilisant la procédure suivante:

En premier lieu il faut démarrer à partir du cdrom d'installation de Linux (assurez vous que le cdrom soit bien en premier dans l'ordre de démarrage des périphériques dans le Bios)

A l'étape ou vous devez donner un nom à la machine ,passez en mode console (ALT + F2)

Faites un fdisk -l /dev/sda puis fdisk -l /dev/sdb pour examiner le contenu de vos partitions et vous assurer de l'emplacement de vos deux systèmes.

Si vous changez le disque dur de place Il faut monter la partition racine contenant le fichier /etc/fstab de façon à pouvoir le modifier ;

```
# mkdir /target
# mount /dev/sdb2 /target
# nano /target/etc/fstab
> remplacer sda par sdb
```

Restitution du boot GRUB

```
# mount /dev/sbd1 /target/boot
```

Faire ensuite :

- Alt+F1 : retour au menu Debian - revenir en arrière (2 fois) - dans le menu choisir **installation Grub** - revenir en arrière (2 fois) - et continuer: OUI (2 fois)

Rebooter et normalement c'est OK, le boot GRUB est de nouveau opérationnel !

Installer une distribution Red Hat

Les sources sont sur https://access.redhat.com/downloads/.

Installer Debian via le réseau

Installation de Debian par le réseau

packets nécessaire

1. dhcp
2. tftp
3. pxe

```
#apt-get install dhcp-server atftpd pxe syslinux
```

Copiez ensuite le fichier /usr/lib/syslinux/pxelinux.0 dans /tftpboot

Configuration du dhcp

```
/etc/dhcp3/dhcpd.conf
```

ddns-update-style none;

option domain-name "diskless.net";

'#'option domain-name-servers gw.diskless.net;

default-lease-time 600; max-lease-time 7200;

log-facility local7;

subnet 192.168.30.0 netmask 255.255.255.0 { range 192.168.30.100 192.168.30.110; '#'option routers rtr-239-0-2.example.org; } next-server 192.168.30.224; option root-path "192.168.30.224:/tftpboot/pxelinux.0";

filename "/tftpboot/pxelinux.0";

```
#redémarer le service /etc/init.d/dchp3-serveur restart
```

tftp

```
/etc/inetd.conf
```

tftp dgram udp nowait root /usr/sbin/tcpd /usr/sbin/in.tftpd /tftpboot

configuration pxe

```
/etc/pxe.conf
```

.# which interface to use

interface=eth0

default_address=192.168.30.210

.# the multicast ip address to listen on

multicast_address=224.0.1.2

.# mtftp info

mtftp_address=192.168.30.210

mtftp_client_port=1758 mtftp_server_port=1759

.# the port to listen on

listen_port=4011

.# enable multicast?

use_multicast=1

.# enable broadcast? use_broadcast=1

.# user prompt

prompt=Press F8 to view menu ... prompt_timeout=10

.# what services to provide, priority in ordering .# CSA = Client System Architecture .# service=<CSA>,<min layer>,<max layer>,<basename>,<menu entry>

service=X86PC,0,0,local,Local boot service=X86PC,0,0,pxelinux,PXELinux

.# tftpd base dir

tftpdbase=/tftpboot

.# domain name .# domain=bla.com

redemarer le service

```
#/etc/init.d/pxe restart
```

Mettre la machine que vous voulez installer en boot pxe et vous devriez avoir l'interface d'installation de debian

Le login

Le login

Les systèmes compatibles Unix sont par définition multi-tâches et multi-utilisateurs, c'est à dire que plusieurs personnes peuvent travailler simultanément sur le même système.

Chaque utilisateur est identifié par un **nom d'utilisateur** (**login**) et un **mot de passe** (**password**).

Le **login** est associé à un numéro unique (le **UID** : **U**ser **ID**entifiant) et permet d'identifier chaque utilisateur. Il ne contient ni espace, ni caractères spéciaux.

Le **mot de passe** doit être choisi judicieusement : il doit mélanger des caractères en minuscule et majuscule, des chiffres et des caractères spéciaux.

Chaque utilisateur dispose d'un répertoire de travail (le Home Directory) dans lequel il a le droit de créer ses propres fichiers et répertoires. Ce répertoire de travail se situe généralement dans le répertoire **home** et porte le nom du **login**. Exemple : le répertoire de travail de l'utilisateur **alex** est **/home/alex**.

Il existe un utilisateur particulier : le **root**. Le **root** est l'administrateur du système, il dispose de tous les droits et s'occupe de la gestion du système : ajout et suppression des utilisateurs, installation et configuration du système et des logiciels ...

Le **root** a tous les pouvoirs sur le système, il peut tout faire, y compris tout casser. De ce fait, il faut donc choisir un mot de passe très sécurisé, et toujours bien vérifier à deux fois avant d'exécuter une opération avec l'utilisateur **root**.

Le **root** n'est pas une personne à proprement parler, il s'agit d'une fonction. Ainsi, il ne faut jamais travailler en permanence avec le compte **root**, mais utiliser son compte habituel, et ne passer **root** (via les commandes **su** ou **sudo**) que si l'on a besoin de réaliser une opération d'administration.

Contrairement aux autres utilisateurs, le répertoire de travail du **root** se situe à la racine du système (**/root**). L'explication est simple : en cas de problème avec la partition **/home**, l'utilisateur **root** pourra quand même accéder à son répertoire de travail.

Commandes de base

Éléments de syntaxe

Les commandes présentées dans les sections qui suivent sont introduites juste après un message d'invite qui dépend de la configuration du shell utilisé. Celui-ci est représenté par un signe dollar $. Il ne doit donc pas être tapé.

Le caractère dièse # marque le début d'un commentaire qui se termine en fin de ligne. Il n'est pas nécessaire de le recopier pour exécuter la commande.

Exemple :

```
$ pwd   # affiche le répertoire courant
```

Pour tester cette commande, il suffit d'entrer p w d ↵

Les lignes qui ne sont pas marquées du signe dollar indiquent ce que la commande précédente doit/peut afficher.

Par ailleurs, il existe deux opérateurs de concaténation des commandes :

- ";" : qui enchaine les commandes quelques soient leurs résultats.
- "&" : qui stoppe l'enchainement si une commande renvoie une erreur.

pwd (print working directory)

Affiche le répertoire courant.

```
$ pwd
/home/alex
```

id

Affiche les informations relatives à l'utilisateur connecté.

```
$ id
uid=1000(alex) gid=1000(alex) groupes=20(dialout), 24(cdrom), 25(floppy), 29(audio), 44(video), 46(plugdev), 106(netdev),
109(powerdev), 1000(alex)
```

passwd

Permet de changer son mot de passe

```
 $ passwd
 Changing password for alex
 (current) UNIX password:
 Enter new UNIX password:
 Retype new UNIX password:
 passwd : le mot de passe a été mis à jour avec succès
```

> **Attention !**
>
> Ne pas utiliser le symbole euros (€) dans les mots de passe, car il est potentiellement mal géré par Linux et pourrait se voir systématiquement refusé.

cd (change directory)

Pour se placer dans un répertoire

```
$ cd .                  # . désigne le répertoire courant
$ cd ..                 # .. désigne le répertoire parent
$ cd /                  # / désigne le répertoire racine
```

```
$ cd /tmp               # désigne le répertoire tmp appartenant à la racine
$ cd tmp                # désigne le répertoire tmp du répertoire courant
```

```
$ cd ../tmp              # désigne le répertoire tmp du répertoire parent du répertoire courant
```

```
$ cd ~                   # permet de revenir dans son répertoire de travail (home directory)
$ cd                     # idem
```

Connecté en **root**, la commande **cd** m'amène au répertoire **/root** qui est le répertoire de travail de l'utilisateur **root**.

Connecté avec l'utilisateur **alex**, **cd** m'amène au répertoire de travail de l'utilisateur **alex**, à savoir **/home/alex**.

ls

Liste les fichiers d'un répertoire

```
$ ls                     # liste les fichiers non cachés du répertoire courant
$ ls -l                  # (l : long) : liste détaillée des fichiers du répertoire courant
$ ls -a                  # (a : all) : liste tous les fichiers, y compris les fichiers cachés
```

On peut combiner plusieurs options, l'ordre n'est pas important. Les quatre commandes suivantes sont identiques :

```
$ ls -a -l
$ ls -l -a
$ ls -la
$ ls -al
```

Lister de façon détaillée (-l) tous les fichiers, même cachés (-a), les plus récents (-t) en derniers (-r), avec leur taille en kilo-octets (-k) lisible facilement avec les unités K pour Kilo, M pour Mega, G pour Giga(-h).

```
$ ls -lartkh
```

cat (con*cat*enate)

Affiche le contenu d'un fichier ou de plusieurs fichiers concaténés sur la sortie standard (l'écran)

```
$ cat /etc/crontab                    # affiche le contenu du fichier /etc/crontab
$ cat /etc/cron.daily /etc/cron.weekly  # concatène les deux fichiers et affiche leur contenu
```

mkdir (make directory)

Permet de créer un répertoire

```
$ mkdir rep1             # crée un répertoire rep1 dans le répertoire courant
$ mkdir /rep1            # tente de créer un répertoire rep1 à la racine,
                         # le système refuse car je ne suis pas connecté en root
```

Exercice : je suis dans le répertoire */var/log*, je souhaite créer un répertoire *rep2* dans le répertoire */home/alex*, comment faire ?

J'ai 3 possibilités:

```
$ mkdir ../../home/alex/rep2    # on utilise un adressage relatif à la position où je suis :
                                # on remonte dans l'arborescence jusqu'à la racine puis
                                # on redescend jusqu'au répertoire alex)
```

```
$ mkdir /home/alex/rep2         # on utilise un adressage absolu en partant de la racine
```

```
$ mkdir ~/rep2                  # on utilise ~ pour désigner le répertoire de travail
```

l'option **-p** permet de créer le(s) répertoire(s) parent(s).

```
$ mkdir  -p rep1/rep2           # crée un répertoire parent rep1 si il n'existe pas,
                                # et crée dans rep1 un répertoire rep2
```

rmdir (remove directory)

Supprimer un répertoire vide

```
$ rmdir rep1
```

cp (copy)

Copier un fichier

```
$ cp /etc/passwd /tmp               # copie le fichier /etc/passwd dans le répertoire /tmp
```

```
$ cp /etc/passwd /tmp/nouveaunom     # copie le fichier /etc/passwd dans le répertoire /tmp
                                     # et le renomme en nouveaunom
```

Attention, si le fichier destination existe déjà, il sera remplacé sans demande de confirmation !

Options courantes :

```
-i : si le fichier destination existe, demande confirmation avant de remplacer le fichier
```

rm (remove)

Effacer un fichier

```
$ rm lefichier                     # efface le fichier lefichier
```

Attention, le fichier est effacé et sans demander confirmation !

Options courantes :

```
-i : demande confirmation avant d'effacer le fichier
-f : ne demande pas de confirmation (annule -i)
-r : supprime les répertoires récursivement
```

Exemples

- Pour supprimer le répertoire "tmp" et son contenu :

```
rm -rf tmp
```

- Pour supprimer tous les fichiers de log de plus de deux jours :

```
find /var/log* -mtime +2 -exec rm {} \;
```

mv (move)

Déplacer ou renommer des fichiers

```
$ mv [Option] Source Destination(répertoire)
$ mv [Option] Répertoire Source
```

```
$ mv fichier_source fichier_cible   # déplacer fichier_source dans fichier_cible

$ mv fichier1 fichier2    # renomme le fichier fichier1 en fichier2
                          # Attention si fichier2 existe, son contenu sera écrasé et
                          # remplacé par celui de fichier1
```

Par précaution, on utilise l'option **-i** qui permet d'être averti par le système avant l'écrasement du fichier destination si celui-ci existe.

```
$ mv -i fichier1 fichier2 # demande la confirmation avant d'effacer la destination
```

```
$ mv rep1/fic1 rep2/fic2  # déplace le fichier fic1 situé dans le répertoire rep1
                          # sous le nouveau nom fic2 situé dans le répertoire rep2
```

```
$ mv rep1 rep2            # déplace le répertoire rep1 dans le répertoire rep2
                          # si rep2 n'existe pas, renomme rep1 en rep2
```

ln (link)

La commande **ln** permet de créer des liens, c'est à dire des raccourcis vers des fichiers ou des répertoires.

```
ln -s destination nom_du_lien
```

Exemple

```
$ ln -s prog1.0 monprogramme
```

Cette commande crée le lien suivant :

```
lrwxrwxrwx 1 alex alex   7 2007-10-26 14:25 monprogramme -> prog1.0
```

L'aide en ligne man

Chaque commande dispose d'une page de manuel en ligne (appelée manpages).

Cette aide en ligne est très utile pour savoir comment utiliser les commandes et connaître la liste exhaustive de toutes les options disponibles.

Pour accéder à cette aide en ligne, il suffit de taper **man <la commande>**. Exemple :

```
$ man ls
```

Les pages de manuel sont réparties en section. Pour connaître l'ensemble des sections, il suffit de consulter la page de manuel de la commande **man** :

```
$ man man
...
1  Programmes exécutables ou commandes de l’interpréteur de  commandes (shell)
 2 Appels système (Fonctions fournies par le noyau)
 3 Appels  de  bibliothèque  (fonctions  fournies  par  les  bibliothèques des programmes)
 4 Fichiers spéciaux (situés généralement dans /dev)
 5 Formats des fichiers et conventions. Par exemple /etc/passwd
 6 Jeux
 7  Divers (y compris les macropaquets et les conventions). Par exemple, man(7), groff(7)
 8 Commandes  de  gestion  du  système (généralement réservées au  superutilisateur)
 9 Sous-programmes du noyau [hors standard]
```

Certaines commandes sont à la fois des commandes systèmes, des appels systèmes ou des fichiers de configuration (exemple : passwd). Il est possible d'indiquer la section que l'on désire consulter :

```
$ man passwd             # la page de man de la commande passwd
$ man 5 passwd           # la page de man du fichier de configuration /etc/passwd
```

Les pages de man sont découpées en différents chapitres (extrait de **man 7 man**) :

Les chapitres des pages de man

Chapitre	Descriptif
SYNOPSIS	Indique brièvement l'interface de la commande ou de la fonction. Pour les commandes, ce paragraphe montre sa syntaxe et ses arguments. Les caractères gras marquent le texte invariable et l'italique indique les arguments remplaçables. Les crochets encadrent les arguments optionnels, les barres verticales (caractère pipe) séparent les alternatives, et les ellipses ... signalent les répétitions. Pour les fonctions, on trouve toutes les déclarations et directives #include, suivies de la déclaration de fonction.
DESCRIPTION	Fournit une explication sur ce que la commande, la fonction ou le format représente. Décrit les interactions avec les fichiers et l'entrée standard, ou ce qui est produit sur la sortie standard ou d'erreur. Ne contient pas les détails d'implémentation internes, sauf s'ils sont critique pour comprendre l'interface. Décrit le cas principal, pour les détails sur les options, on utilise le paragraphe OPTIONS. S'il y a une sorte de grammaire d'entrée, ou un jeu de sous-commandes, on peut les placer dans une section UTILISATION supplémentaire (et placer un bref aperçu dans la section DESCRIPTION).
RETURN VALUE (VALEUR RENVOYÉE)	Donne une liste des valeurs qu'une routine de bibliothèque renverra à l'appelant et les conditions qui provoquent ces retours.
EXIT STATUS (CODE DE RETOUR)	Indique les codes de retour d'un programme et les conditions associées.
OPTIONS	Décrit les options acceptées par le programme et leur influence sur son son comportement.
USAGE (UITILISATION)	Décrit la grammaire de tout sous-langage implémenté.
EXAMPLES (EXEMPLES)	Donne un ou plusieurs exemples d'utilisation de la fonction, du fichier ou de la commande.
FILES (FICHIERS)	Liste les fichiers utilisés par le programme ou la fonction, tels que fichiers de configuration, de démarrage, et les fichiers manipulés directement par le programme. Il faut donner le chemin d'accès complet des fichiers et utiliser le mécanisme d'installation pour modifier le préfixe. Pour la plupart des programmes, l'installation par défaut se fait dans /usr/local, aussi, votre page de manuel de base devrait utiliser /usr/local comme base.
ENVIRONMENT (ENVIRONNEMENT)	Décrit toutes les variables d'environnement qui affectent le programme ou la fonction, ainsi que leurs effets.
DIAGNOSTICS (DIAGNOSTIQUE)	Fournit un survol des messages d'erreurs usuels et comment les considérer. Il n'est pas nécessaire d'indiquer les messages d'erreur système ou les signaux fatals qui peuvent apparaître durant l'exécution du programme, sauf s'ils sont traités spécialement.
SECURITY (SECURITÉ)	Décrit les problèmes de sécurité et leurs implications. Doit contenir les avertissements à propos des configurations ou des environnements à éviter, les commandes ayant des répercussions au niveau sécurité, etc. surtout s'ils ne sont pas évidents. Il n'est pas obligatoire de faire un paragraphe spécifique sur la sécurité. Si l'intelligibilité est améliorée, on peut placer ces informations dans les autres sections (telles que DESCRIPTION ou USAGE (UTILISATION)). Néanmoins, il est important de placer les informations de sécurité quelque part.
CONFORMING TO (CONFORMITÉ)	Décrit les standards ou les conventions suivis par l'implémentation.
NOTES	Contient des notes diverses.
BUGS (BOGUES)	Liste les limitations ou les défauts recensés, ainsi que les sujets à débat.
AUTHOR (AUTEUR)	Liste les auteurs de la documentation ou du programme afin de pouvoir leur envoyer les rapports de bogue.
SEE ALSO (VOIR AUSSI)	Fournit une liste des pages de manuel ayant un rapport, dans l'ordre alphabétique, suivies des autres documents éventuels. Il s'agit d'habitude de la dernière section.

L'éditeur de texte vi

L'éditeur de textes vi

vi' (connu sous ses abréviations anglaises, pronnoncer vie-ail) est un éditeur de texte présent d'office sur la majorité des systèmes Unix actuels depuis 1976, souvent sous la forme d'un clone du logiciel *vi* originel.

vi est un éditeur modal, c'est-à-dire que la signification des boutons et des touches change selon le mode dans lequel il se trouve. L'un de ses intérêts a longtemps été de ne pas nécessiter de souris et de fonctionner avec un nombre réduit de touches. En effet, la présence de deux modes permet d'avoir un mode où les touches lettres servent à la saisie des caractères, en mode commande aux déplacements et autres commandes. De ce fait l'ensemble des nouvelles touches (flèches, clavier numérique et touches de fonction notamment) est totalement superflus. L'interface est donc la même pour tout ordinateur qu'il soit équipé d'un clavier de minitel ou d'un clavier avec d'innombrables touches.

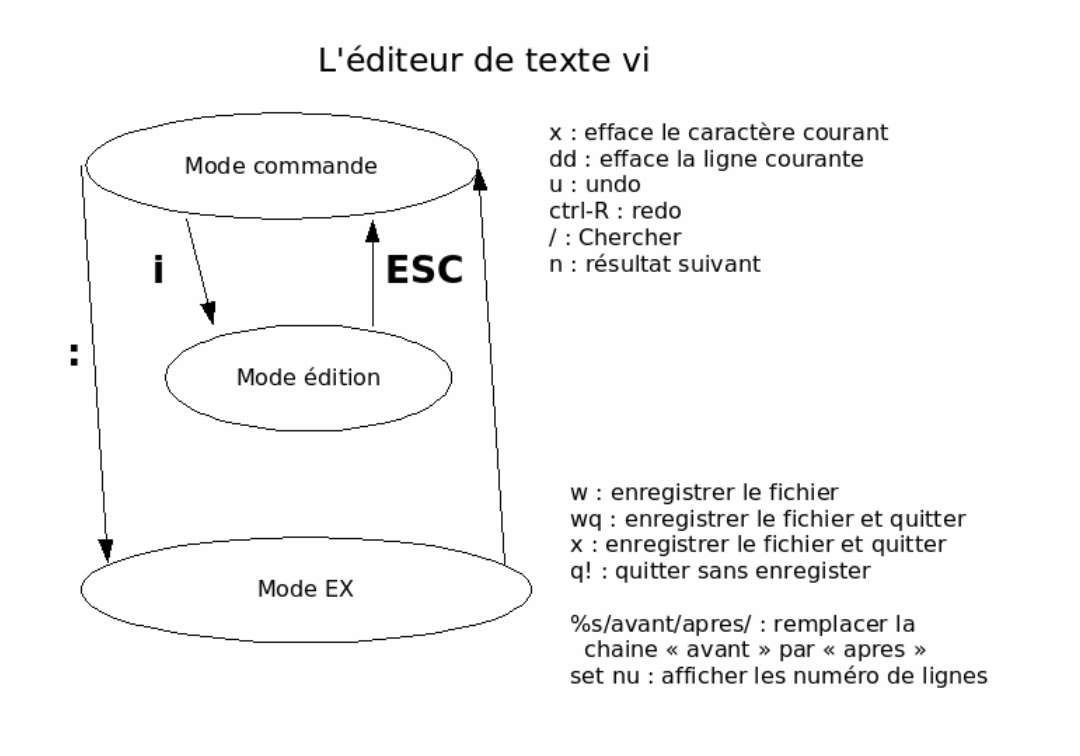
Vi fut écrit par Bill Joy en 1976, « et ben ça ne nous rajeunit pas les enfants.»

```
« Tu connais Notepad ? Ben Vi c'est bien mieux, c'est même super ! »
« Avec vi, tu vas frimer devant tes potes, et te la jouer hackeur. »
```

Derrière un aspect rustique qui nécessites une familiarisation, se cache des fonctionnalités d'édition qui ne sont pas présentes dans d'autres éditeurs de texte. Le minimum à savoir pour utiliser vi est de connaître les deux touches permettant de basculer entre ces deux modes principaux.

```
Alors tape dans un Shell avec tes doigts musclés vi et Entrée.
Là t'as une page blanche, comme pour un nouveau roman d'amitié qui commence entre toi et Linux.
```

```
Tu peux faire plusieurs commandes :
- Tape i : là par miracle tu peux taper des bêtises dans ta page.
- Tape < ESC >  : là tu ne peux plus rien écrire, mais ... tu peux exécuter des commandes.
Et vice versa
```



Il y a donc 2 modes : insertion et commande.

```
En mode insertion ----> "tu peux écrire tout ce que tu veux."
En mode commande ----> "tu peux enregistrer, effacer, quitter, etc."
```

- Pour passer du mode *insertion* au mode *commande* : touche **< ESC >**
- Pour passer du mode *commande* au mode *insertion* : touche **i**. (insert)
- En mode commande, on peut passer en un 3ème mode "mode Execution" en tapant "**:**'"(la ligne de commande est en bas de

l'écran)

Les commandes de base de *vi*

Commande	Fonction de la commande
x	effacer une lettre (xterminator)
dd	effacer la ligne courante (delete)
yy	copier la ligne courante (yank)
p	coller la ligne copiée précédement lors de la commande yy ou de la commande dd
:x	aller à la ligne x (go to line) - Ex. :152 met le curseur à la ligne 152
r	remplace un caractère (replace)
u	annule la dernière commande (undo)
:w	sauvegarder le fichier (write)
:q!	quitter vi sans sauvegarder (quit)
:wq	sauvegarder et quitter (write quit)
:help	pour afficher l'aide dans vi
:q	pour quitter la fenêtre d'aide si elle est ouverte ou l'éditeur
:w *nom_fichier*	enregistre ce qui est saisi dans le fichier *nom_fichier*
/toto	chercher la chaîne *toto* en avant
?toto	chercher la chaîne *toto* en arrière
n	permet de chercher l'occurrence suivante de la chaîne
:s/alex/toto	remplace l'occurrence suivante de la chaîne *alex* par *toto*
:%s/alex/toto/gic	remplace la chaîne *alex* par *toto* dans tout le fichier "**%**" pour tout le fichier, "**s**" search, "**g**" global, "**i**" ignore la casse, "**c**" confirmation

Il est difficile de savoir si on est en mode insertion ou en mode commande, le plus simple est d'appuyer plusieurs fois sur <ESC> pour être sûr d'être en mode commande.

vi est rustique, on le trouve sur tous les systèmes UNIX, même les plus vieux. Il est préférable d'utiliser **vim** (VI iMproved) quand c'est possible, c'est un éditeur vi amélioré

Après l'installation de *vim*, lancez le programme en tapant indifféremment **vi** ou **vim**

Vim est plus convivial, il prend en compte les flèches et d'autres fonctions du clavier récent.

Il existe bien entendu d'autres éditeurs de texte dans Linux, du plus perfectionné au plus basique, mais *vi* est celui qu'on trouve partout (même sur notre vieux minitel). Il est donc impératif d'en connaître les commandes de base.

Quelques autres éditeurs :

```
1. Ed
2. Nano
3. Emacs (editor macro)
4. Joe
5. Pico
6. XEmacs (anciennement Lucid Emacs) est un éditeur de texte pour X-Window, basé sur GNU Emacs.
```

Les shells

Shell veut dire *Coquille*, qui entoure le noyau. C'est un interpréteur de commandes qu'on utilise pour lancer des commandes ou programmer une suite de commandes. L'utilisateur discute avec le Shell, qui discute avec le noyau, qui à son tour discute avec le matériel. Originellement le shell est utilisé sous Unix, il s'est répandu depuis avec différentes versions, la forme la plus simple est sh.

Les versions connues :

- sh : shell Bourne
- ksh : korn shell
- Csh : Shell syntaxe du C
- Tcsh : Csh amélioré
- Bash : Bourne Again Shell
- Zsh : le petit dernier

Le shell est constitué de petits shells spécifiques à chaque travail, chacun ayant un fichier de configuration stocké dans le répertoire /home de chaque utilisateur, lequel peut les modifier à sa convenance :

- `.profile` : exécuté automatiquement lors d'un shell de connexion (à chaque login), quel que soit le shell

Fichiers de configuration spécifiques au shell **Bash** :

- `.bash_profile` : exécuté automatiquement lors d'un shell de connexion (à chaque login)
- `.bashrc` : chargé automatiquement lors d'un shell interactif
- `.bash_logout` : chargé lors de la fermeture du shell (à la déconnexion)

Autres fichiers liés au Bash

- `.bash_history` : c'est un fichier texte contenant l'historique des commandes tapées.

La complétion

« L'art de la saisie des commandes ultra rapide ».

À chaque fois que vous tapez une commande, Linux vous aide à compléter votre commande en appuyant sur la touche TAB ↹.

Tapez le début de votre commande et en appuyant sur la touche TAB ↹, Linux vous la complète ou vous propose les différentes possibilités pour la compléter, à défaut vous aurez droit à un petit bip, s'il existe un trop grand nombre de propositions, Linux vous affiche ce nombre.

exemple :

- `$ cd /h` en tapant TAB ↹ Linux complète la commande en `cd /home`
- `$ cd /` en tapant TAB ↹ Linux me donne la liste de tous les répertoires de la racine afin que j'en choisisse un en tapant ses premiers caractères et que je complète encore par TAB ↹ si besoin est.

Les jokers

Dans une commande sur les fichiers, on peut remplacer une partie d'un nom (un ou plusieurs caractères) de fichier par * ou un seul caractère par ?

Exemple

```
$ ls *.c                        # liste tous les fichiers du répertoire courant dont
                                # l'extension est .c
$ ls fic*.c                     # liste tous les fichiers du répertoire commençant par fic
                                # et dont l'extension est .c
                                # ex : fichier1.c fichier2.c fichierprojet.c ...
$ ls fichier?.c                 # liste tous les fichiers dont le nom est "fichier"
                                # + un caractère quelconque
                                # et dont l'extension est .c
                                # fichier1.c fichier2.c fichier3.c fichier4.c ...
$ ls fichier.[ch]               # liste tous les fichiers commençant par fichier
                                # et dont l'extension est . suivit de c ou h
                                # liste : fichier.c fichier.h
```

La commande zenity : man zenity

Les répertoires importants

Une des premières difficultés pour les débutants Unix est de se repérer dans l'arborescence. Contrairement à un environnement Windows qui regroupe toutes les ressources nécessaires à un programme dans un même répertoire, les ressources dans un système unix (et dans une moindre mesure MacOS) sont réparties dans une hiérarchie générale. Ainsi si le code exécutable d'un programme se trouve dans le répertoire `/bin`, ses libraries se trouveront dans le répertoire `/lib` et ses fichiers de configuration dans le répertoire `/etc`.

La hiérarchie des systèmes unix est normalisée, la spécification étant accessible sur http://www.pathname.com/fhs/

Voici une liste des répertoires à connaître.

Les programmes exécutables

- /bin : les commandes indispensables
- /usr/bin : les autres commandes
- /usr/local/bin : les commandes installées à la main

Les programmes exécutables du super-utilisateur root

- /sbin : les programmes indispensables d'administration
- /usr/sbin : les autres programmes d'administration
- /usr/local/sbin : les programmes d'administration installées à la main

Les fichiers de configuration

- /etc

En particulier, /etc contient les fichiers:

```
/etc/passwd              liste des comptes utilisateurs et leurs répertoires associés.
/etc/shadow              liste des comptes utilisateurs et leurs mots de passe chiffrés.
/etc/X11/xorg.conf       fichier de configuration du serveur graphique.
```

Le(s) noyau(x) Linux

- /boot

Les répertoires de travail des utilisateurs

- /home

Chaque utilisateur du système aura un répertoire dans /home portant son nom.

Le répertoire de travail du super-utilisateur root

- /root

Les librairies partagées

- /lib : les librairies indispensables
- /usr/lib : les autres librairies

Les points de montage

- /mnt : Répertoire dans lequel on pourra créer des points de montage pour des systèmes de fichiers temporaires, destiné à un usage d'appoint. Les composants vitaux du système ne doivent pas dépendre de ce répertoire.
- /media : Répertoire dans lequel des points de montage seront créés automatiquement pour accéder aux supports de stockage amovibles (cdrom, clé usb...)

Les périphériques

- /dev

En particulier, on trouve les fichiers :

```
/dev/hda  ou hdb ou sda etc... qui correspondent aux disques dur. Les disques durs IDE sont notés hd et les SATA ou SCSI sont notés sd
/dev/hda1 ou hda2 etc... qui correspondent aux partitions des disques. hda1 est la 1ere partition du premier disque IDE (hda).
```

```
/dev/eth0 qui correspond à la première carte réseau. Pour plus d'une carte réseau, on verra apparaître d'autres fichiers /dev/eth.
```

```
/dev/audio qui correspond au périphérique son.
```

Les autres programmes et leurs fichiers annexes

- /usr

Ce répertoire est en lecture-seule et est destiné à être partagé. Les programmes installés par le système d'exploitation se trouvent dans /usr/bin. Les programmes installés manuellement par les utilisateurs se trouvent dans /usr/local.

« usr » veut dire *Unix System Ressources* et non pas *user*. Les fichiers relatifs aux utilisateurs du système se trouvent dans le répertoire « /home ».

Le système de fichier virtuel

Donne les infos de la machine

- /proc

sur le processeur

Ex:/proc/cpuinfo

sur la mémoire

Ex:/proc/meminfo

sur les modules

Ex:/proc/modules

sur les montages du systeme

Ex:/proc/mount

sur la swaps

Ex:/proc/swaps

Les fichiers temporaires

- /tmp

Les données variables

- /var

Ce répertoire contient les données variables. En particulier:

```
les logs sont stockés dans /var/log
les mails en attente de livraison sont stockées dans /var/mail
les travaux d'impression seront dans /var/spool ou /var/mail selon le serveur d'impression utilisé.
```

- /var/log, ce répertoire contient les journaux (fichiers détaillant les historiques de programmes).
- /var/mail, ce répertoire contient les emails des utilisateurs du système. Un utilisateur du système peut voir ses emails via la commande « mail ».
- /var/spool, ce répertoire contient les données en attente de traitement (par un programme), et sont en générale supprimé après avoir été traité.

Les programmes et ressources installés à la main

- /usr/local

Ce répertoire contient des données mis en place par l'administrateur du système, comme « /usr » mais dans un contexte plus « isolé » car ces ressources sont géré par ce même administrateur système, et contient des sous-répertoire tel que:

```
/usr/local
├── bin/                    : où se trouvent les programmes.
│   ├── monprogramme        : exemple de programme nommé « monprogramme ».
├── etc/                    : où se trouvent les fichiers de configurations des programmes, comme « /etc ».
│   ├── monprogramme.conf   : exemple de configuration de « monprogramme ».
├── lib/                    : où se trouvent les bibliothèques logiciel.
├── lib64/                  : où se trouvent les bibliothèques logiciel, compilé pour les processeurs à architecture 64 bits.
├── sbin/                   : où se trouvent les programmes utilisable par "root".
```

Redirection des entrées/sorties

Les entrées/sorties des processus

Chaque processus possède 3 flux standards qu'il utilise pour communiquer en général avec l'utilisateur :

- l'entrée standard nommée `stdin` (identifiant 0) : il s'agit par défaut du clavier,
- la sortie standard nommée `stdout` (identifiant 1) : il s'agit par défaut de l'écran,
- la sortie d'erreur standard nommée `stderr` (identifiant 2) : il s'agit par défaut de l'écran.

Ces flux peuvent être redirigés afin que le processus interagisse avec un autre au lieu d'interagir avec l'utilisateur.

Redirection

Rediriger la sortie standard

Quand on exécute une commande, le shell affiche le résultat sur la console de sortie (l'écran par défaut). On peut rediriger cette sortie vers un fichier en utilisant le signe `>`.

Exemple

```
~$ ls>resultat_ls
```

Si le fichier existe déjà, il est écrasé.

Concaténation

Au lieu de créer un fichier, il est possible d'ajouter les sorties d'un processus à un fichier existant en utilisant le double signe `>>`.

Exemple

```
~$ ls>>resultat
```

Si le fichier résultat existe déjà, les affichages sont concaténés.

Syntaxe complète

En fait, les signes `>` peuvent être précédés de l'identifiant du flux à rediriger. Pour la sortie standard, on peut donc utiliser les syntaxes suivantes :

```
~$ ls 1>resultat
~$ ls 1>>resultat
```

Ce qui revient au même que les deux premiers exemples ci-dessus (redirection et concaténation).

Rediriger la sortie d'erreur standard

La redirection du flux de sortie d'erreur standard utilise les même signes, mais précédés de l'identifiant du flux : 2.

Exemples

```
~$ ls 2>erreurs_ls
~$ ls 2>>erreurs_ls
```

Rediriger l'entrée standard

Rediriger l'entrée standard permet d'entrer des données provenant d'un fichier au lieu du clavier.

Exemple

```
~$ cat < mon_fichier.txt
```

Rediriger un flux vers un autre

Il est possible de rediriger un flux vers la sortie standard ou la sortie d'erreur en donnant l'identifiant du flux précédé du caractère & à la place du nom de fichier.

Exemple

```
~$ ls 1>stdout_stderr.txt 2>&1
```

Le fichier `stdout_stderr.txt` contient ce qui a été affiché à la fois sur le flux de sortie standard et le flux de sortie d'erreur.

Échange des deux flux de sortie

L'échange des deux flux de sortie s'effectue en utilisant la syntaxe pour rediriger un flux vers un autre à trois reprises, en utilisant un identifiant de flux fictif comme intermédiaire (3).

Exemple

```
~$ ls 2>&3 1>&2 3>&1
```

Le pipe (un tube)

Redirige la sortie d'une commande vers l'entrée d'une autre commande. Il s'agit donc d'une chaîne de redirection entre deux processus qui ne passe pas par un fichier, mais par une zone mémoire du système.

Exemples

- Afficher la taille des fichiers et répertoires, et les trier du plus grand au plus petit :

```
$ du | sort -rn
```

- Même résultat, mais affiché page par page :

```
$ du | sort -rn | more
```

- Connaître le nombre de fichiers du répertoire /usr/bin :

```
$ ls -1 /usr/bin | wc -l
```

Explications : L'option **-1** de la commande **ls** affiche un fichier ou répertoire par ligne. La commande wc (word count) avec l'option **-l** (line) compte le nombre de lignes.

- Connaître tous les périphériques IDE détectés par le noyau Linux et les afficher page par page :

```
$ dmesg | grep hd | more
```

Explications : la commande **dmesg** affiche les messages du noyau Linux détectés durant le boot. La commande **grep** n'affiche que les lignes contenant le mot **hd**. La commande **more** affiche ces résultats page par page.

Autre exemple : extraire l'adresse IP de la carte réseau eth0 :

```
$ ifconfig eth0 | grep 'inet adr' | cut -f2 -d':' | cut -f1 -d' '
192.168.30.50
```

Invoquer un programme en tâche de fond

Invoquer un programme en tâche de fond

Certains programmes ne rendent pas la main immédiatement (exemple : la compilation d'un gros programme).

Pour récupérer la main, il suffit de rajouter un **&** (ET commercial ou Esperluette) à la fin de la commande :

```
$ xeyes &
```

Si on a lancé une commande qui ne rend pas la main et que l'on a oublié de rajouter le **&**, on peut utiliser la méthode suivante :

```
$ xeyes
<CTRL Z>
[1]+  Stopped                 xeyes
```

Le programme est alors stoppé. Il suffit de taper ensuite la commande **bg** (background) pour qu'il s'exécute en tâche de fond :

```
$ xeyes
<CTRL Z>
[1]+  Stopped                 xeyes
$ bg
[1]+ xeyes &
$
```

La commande **fg** (foreground) permet de refaire passer le programme en premier plan :

```
$ xeyes
<CTRL Z>
[1]+  Stopped                 xeyes
$ bg
[1]+ xeyes &
$ fg
xeyes
```

La commande **jobs** affiche les tâches en cours.

```
$ jobs
[1]+  Running                 xeyes &
```

Propriétaires et droits d'accès

Les droits d'accès

Fonctionnement

Chaque fichier du système est associé à des droits d'accès. Ceux-ci sont affichés par la commande `ls` en utilisant le format long : `ls -l`.

Ces droits d'accès sont résumés en 10 caractères de la forme suivante :

```
type u_read u_write u_exec g_read g_write g_exec o_read o_write o_exec
```

Exemple :

```
drwxr-xr-x
```

Le premier caractère représente le type de fichier :

Types de fichier

Type	Description
-	*Regular file* : fichier normal
d	***D**irectory* : répertoire
l	***L**ink* : **l**ien
b	***B**lock device* : périphérique **b**loc (périphérique à accès direct)
c	***C**haracter device* : périphérique **c**aractère (périphérique à accès séquentiel)
s	***S**ocket*
p	***P**ipe (tube nommé)*

Le reste de la chaîne est scindé en 3 blocs de 3 caractères qui représentent les différents niveaux de droit :

- user(u) : droits concernant le **propriétaire** du fichier,
- group(g) : droits concernant les autres membres du **même groupe** que le propriétaire du fichier,
- other(o) : droits concernant tous les **autres** utilisateurs.

Les trois caractères d'un bloc sont les suivants, dans l'ordre :

- r (***r**ead*) : droit de lire le fichier / lister le contenu du répertoire
- w (***w**rite*) : droit d'écrire dans le fichier / modifier le répertoire (créer/supprimer des fichiers)
- x (e***x**ecute*) : droit d'exécuter le fichier (programme ou script) / ou de traverser le répertoire (changer le répertoire courant).

Si l'un des droits n'est pas accordé, un tiret est affiché à sa place.

Exemple :

```
-rwxrw-r--  ...  script.sh
```

Le fichier `script.sh` est un fichier normal (`-`) et est associé aux droits suivants :

- Le propriétaire peut lire, écrire et exécuter ce fichier (`rwx`),
- Les membres du même groupe que le propriétaire peuvent lire et écrire, mais pas exécuter ce fichier (`rw-`),
- Les autres utilisateurs peuvent seulement lire ce fichier (`r--`).

*Pour plus de détails voir : **w:Permissions_Unix#Droits étendus**.*

Modifier les droits d'accès

La commande `chmod` permet de modifier les droits associés à un fichier.

Enlever le droit de lecture (r) aux autres (o)

```
~$ chmod o-r nom_de_fichier
```

Ajouter un droit d'écriture (w) au groupe (g)

```
~$ chmod g+w nom_de_fichier
```

Combinaison des deux commandes :

```
~$ chmod o-r,g+w nom_de_fichier
```

Spécifier tous les droits avec une valeur numérique :

```
~$ chmod 0754 nom_de_fichier
```

Les droits par défaut et la commande umask

Les droits par défaut d'un nouveau fichier sont définis par rapport à un masque des droits défini pour chaque utilisateur avec la commande `umask`.

Afficher le masque courant :

```
~$ umask
0022
```

Exemples :

- Un fichier est créé avec les droits par défaut 666 (rw-rw-rw-) filtré par le masque 022 :

```
666 & ~022 = 644  rw- r-- r--
```

- Un >répertoire est créé avec les droits par défaut 777 (rwxrwxrwx) filtré par le masque 022 :

```
777 & ~022 = 755  rwx r-x r-x
```

Les propriétaires et les groupes

La commande chown

La commande *chown* permet de changer le propriétaire d'un fichier ou d'un répertoire. Il faut être propriétaire du fichier ou répertoire, ou root selon la ressource que l'on souhaite modifier.

```
La commande chown est suivie du nouveau propriétaire puis du nom du fichier ou du répertoire:
$ chown alex toto  => donne la propriété du fichier toto à alex.
```

La commande chgrp

Comme la commande *chown*, la commande *chgrp* change le groupe propriétaire d'un fichier ou d'un répertoire.

```
La commande chgrp est suivie du nom du groupe puis du nom du fichier ou du répertoire :
$ chgrp etudiant toto  => donne la propriété du fichier toto au groupe etudiant.
```

On peut aussi changer le propriétaire et le groupe en une seule commande :

```
$ chown suivit du nom du propriétaire.nom du groupe suivit du nom du fichier
$ chown alex.prof toto  => donne la propriété du fichier toto à alex et au groupe prof.
```

Le raccourci suivant existe également :

```
$ chown alex. toto  => donne la propriété du fichier toto à alex et au groupe alex.
```

Les Access Control List (ACL)

Afficher les ACL

afficher tous les droits, y compris dans les sous-répertoires : $ getfacl -R *

afficher tous les droits sauf les droits de base : $ getfacl --skip-base -R *

Ajouter une ACL

Modifier une ACL

Supprimer une ACL

Sauvegarder les ACL

Pour les systèmes de fichier XFS, il convient également de sauvegarde les ACLs. La sauvegarde des ACLs doit être effectuée dans un fichier avant la sauvegarde sur bande.

```
$ getfacl --skip-base -R /home/acl.sauv
```

ou

```
$ getfacl --skip-base -absolute_name -R /home > /home/acl.sauv
```

si la sauvegarde se fait en chemin absolu.

Le fichier /home/acl.sauv sera également sauvegardé sur bande.

Processus

Définition d'un processus

Un processus est un programme en cours d'exécution. Par exemple, chaque fois que l'on lance la commande **ls**, un processus est créé durant l'exécution de la commande.

Un processus est identifié par un numéro unique que l'on appelle le PID (**P**rocess **ID**entifiant).

Un processus dispose d'un processus père que l'on appelle le PPID (**P**arent **PID**).

La particularité d'un processus est de s'exécuter avec les droits accordés à l'utilisateur qui a lancé la commande. Ceci participe fortement à la sécurité du système. Ainsi, si un utilisateur contracte un programme malveillant (un virus par exemple), le processus sera soumis au droit d'accès de cet utilisateur, et ne pourra pas effectuer des opérations non autorisées (comme par exemple modifier le fichier de mots de passe).

Au démarrage de l'ordinateur, le système charge le noyau Linux qui se charge de l'initialisation du matériel et de la détection des périphériques. Ceci fait, il démarre ensuite le processus **init** qui a comme particularité d'être le premier processus et de toujours utiliser le PID 1. Ce processus démarre ensuite des processus noyaux (dont le nom est noté entre crochets), et les premiers processus systèmes.

Chaque processus a ainsi un père (sauf init), et peut être à son tour le père d'autres processus, etc.

La commande **pstree** permet de visualiser l'arbre des processus. L'option **-p** permet de visualiser les PID de chaque processus.

Exemple :

```
$ pstree -p
init(1)─┬─atd(2861)
        ├─avahi-daemon(2647)───avahi-daemon(2648)
        ├─cron(2873)
        ├─cupsd(2571)
        ├─dbus-daemon(2579)
        ├─dbus-daemon(3023)
        ├─dbus-launch(3022)
        ├─dcopserver(3054)
        ├─dhcdbd(2628)
        ├─dirmngr(3221)
        ├─events/0(4)
        ├─exim4(2739)
        ├─gconfd-2(3576)
        ├─gdm(2796)───gdm(2802)─┬─Xorg(2805)
        │                       └─startkde(2971)─┬─kwrapper(3063)
        │                                        └─ssh-agent(3019)
        ├─getty(2902)
        ├─getty(2903)
        ├─getty(2904)
        ├─getty(2905)
        ├─getty(2906)
        ├─getty(2907)
        ├─hald(2587)───hald-runner(2588)─┬─hald-addon-keyb(2594)
        │                                └─hald-addon-stor(2621)
        ├─hpiod(2464)
        ├─inetd(2756)
        ├─kaccess(3073)
        ├─kded(3058)
        ├─kdeinit(3051)─┬─artsd(3077)
        │               ├─firefox-bin(3572)─┬─{firefox-bin}(3573)
        │               │                   ├─{firefox-bin}(3574)
        │               │                   ├─{firefox-bin}(3581)
        │               │                   ├─{firefox-bin}(3586)
        │               │                   ├─{firefox-bin}(3587)
        │               │                   └─{firefox-bin}(3588)
        │               ├─kio_file(3071)
        │               ├─klauncher(3056)
        │               ├─konsole(3308)───bash(3309)───pstree(4518)
        │               ├─konsole(3907)───bash(3908)───su(3925)───bash(3926)
        │               └─kwin(3066)
        ├─kdesktop(3068)
        ├─kdesud(3357)
        ├─khelper(5)
        ├─kicker(3070)
        ├─klipper(3085)
        ├─klogd(2446)
        ├─knotify(3082)
        ├─korgac(3086)
        ├─ksmserver(3065)
        ├─ksoftirqd/0(3)
        ├─kthread(6)─┬─aio/0(76)
        │            ├─ata/0(594)
```

```
        |            ├─ata_aux(595)
        |            ├─hda_codec(1575)
        |            ├─kblockd/0(9)
        |            ├─khubd(529)
        |            ├─kjournald(1012)
        |            ├─kjournald(1916)
        |            ├─kjournald(1918)
        |            ├─kjournald(1920)
        |            ├─kjournald(1922)
        |            ├─kmirrord(1874)
        |            ├─kpsmoused(1551)
        |            ├─kseriod(12)
        |            ├─kswapd0(75)
        |            ├─pdflush(73)
        |            ├─pdflush(74)
        |            ├─scsi_eh_0(653)
        |            └─scsi_eh_1(655)
        ├─migration/0(2)
        ├─portmap(2192)
        ├─python(2491)
        ├─rpc.statd(2810)
        ├─soffice(4472)──soffice.bin(4492)─┬─{soffice.bin}(4493)
        |                                  ├─{soffice.bin}(4494)
        |                                  ├─{soffice.bin}(4495)
        |                                  ├─{soffice.bin}(4496)
        |                                  └─{soffice.bin}(4497)
        ├─start_kdeinit(3050)
        ├─syslogd(2440)
        └─udevd(1192)
```

Dans cette arborescence, on constate que le processus **init** est bien le père de tous les processus.

Afficher la liste des processus

La commande **ps** permet d'afficher les processus. Utilisée sans option, la commande **ps** affiche les processus de l'utilisateur courant associé au terminal courant.

Pour connaître tous les processus, y compris ceux qui ne sont pas associés au terminal courant, on utilise les options **aux** (notation BSD) et **-ef** (notation Unix). Pour connaître la totalité des options, consulter la page de manuel de la commande ps (**man ps**).

Exemples :

```
$ ps aux
pixl@nitroglycerine:~$ ps aux
USER       PID %CPU %MEM    VSZ   RSS TTY      STAT START   TIME COMMAND
root         1  0.0  0.0   1944   644 ?        Ss   11:36   0:00 init [2]
root         2  0.0  0.0      0     0 ?        S    11:36   0:00 [migration/0]
root         3  0.0  0.0      0     0 ?        SN   11:36   0:00 [ksoftirqd/0]
root         4  0.0  0.0      0     0 ?        S<   11:36   0:00 [events/0]
root         5  0.0  0.0      0     0 ?        S<   11:36   0:00 [khelper]
...
daemon    2192  0.0  0.0   1684   376 ?        Ss   11:36   0:00 /sbin/portmap
root      2440  0.0  0.0   1628   640 ?        Ss   11:36   0:00 /sbin/syslogd
root      2446  0.0  0.0   1576   376 ?        Ss   11:36   0:00 /sbin/klogd -x
root      2464  0.0  0.0   4884   916 ?        Ss   11:36   0:00 /usr/sbin/hpiod
...
```

```
$ ps -ef
UID        PID  PPID  C STIME TTY          TIME CMD
root         1     0  0 11:36 ?        00:00:00 init [2]
root         2     1  0 11:36 ?        00:00:00 [migration/0]
root         3     1  0 11:36 ?        00:00:00 [ksoftirqd/0]
root         4     1  0 11:36 ?        00:00:00 [events/0]
root         5     1  0 11:36 ?        00:00:00 [khelper]
...
daemon    2192     1  0 11:36 ?        00:00:00 /sbin/portmap
root      2440     1  0 11:36 ?        00:00:00 /sbin/syslogd
root      2446     1  0 11:36 ?        00:00:00 /sbin/klogd -x
root      2464     1  0 11:36 ?        00:00:00 /usr/sbin/hpiod
...
```

Au final, on obtient les mêmes résultats.

Les signaux

Définition

Les signaux offrent un mécanisme permettant d'envoyer un message à un processus en cours d'exécution. On se sert généralement des signaux pour terminer un processus, lui indiquer de relire sa configuration, etc.

Les différents signaux

Pour connaître la liste des signaux, il suffit de consulter la page de manuel de **signal** (section 7) :

```
$ man 7 signal
...
Signal     Valeur    Action   Commentaire
----------------------------------------------------------------------
SIGHUP        1       Term    Déconnexion détectée sur le terminal
                              de contrôle ou mort du processus de
                              contrôle.
SIGINT        2       Term    Interruption depuis le clavier.
SIGQUIT       3       Core    Demande « Quitter » depuis le clavier.
SIGILL        4       Core    Instruction illégale.
SIGABRT       6       Core    Signal d'arrêt depuis abort(3).
SIGFPE        8       Core    Erreur mathématique virgule flottante.
SIGKILL       9       Term    Signal « KILL ».
SIGSEGV      11       Core    Référence mémoire invalide.
SIGPIPE      13       Term    Écriture dans un tube sans lecteur.
SIGALRM      14       Term    Temporisation alarm(2) écoulée.
SIGTERM      15       Term    Signal de fin.
SIGUSR1   30,10,16    Term    Signal utilisateur 1.
SIGUSR2   31,12,17    Term    Signal utilisateur 2.
SIGCHLD   20,17,18    Ign     Fils arrêté ou terminé.
SIGCONT   19,18,25    Cont    Continuer si arrêté.
SIGSTOP   17,19,23    Stop    Arrêt du processus.
SIGTSTP   18,20,24    Stop    Stop invoqué depuis tty.
SIGTTIN   21,21,26    Stop    Lecture sur tty en arrière-plan.
SIGTTOU   22,22,27    Stop    Écriture sur tty en arrière-plan.
```

Les signaux les plus connus sont les trois suivants :

- **SIGHUP** (signal n°1) : pour beaucoup de services réseaux, la réception du signal n°1 lui indique de lire sa configuration. Par exemple, cela permet d'indiquer au processus **apache** (serveur web) de relire sa configuration, sans avoir a arrêter et redémarrer le processus.
- **SIGKILL** (signal n°9) : termine un processus (arrêt brutal). Utile lorsque le SIGTERM ne marche pas (processus planté).
- **SIGTERM** (signal n°15) : demande au processus de s'arrêter (arrêt propre).

Envoyer un signal à un processus

La commande **kill** permet d'envoyer un signal au processus. Syntaxe :

```
$ kill -<numéro du signal ou nom du signal> <PID du processus>
```

Exemples :

```
$ kill -1 12345
$ kill -SIGTERM 12345
```

On peut connaitre le PID du processus en utilisant la commande **ps**, ou bien utiliser la commande **pidof**

```
$ pidof cupsd
2571
$ kill -15 2571
```

La commande **killall** permet d'indiquer le nom du processus plutôt que son PID, et va envoyer le signal à tous les processus possédant ce nom. Exemple :

```
$ xeyes & ; xeyes & ; xeyes &
$ killall xeyes
```

Utilisé sans option, les commandes **kill** et **killall** envoient le signal n°15 (arrêt propre).

Autres commandes affichant les processus

La commande **top** permet d'afficher (entre autres) la liste des processus de manière dynamique (rafraîchit la liste toutes les 3 secondes).

Lorsque la commande **top** fonctionne, les lettres suivantes permettent de changer son fonctionnement :

- **h** : affiche l'aide
- **q** : quitte la commande top
- **M** : tri les processus par utilisation de la mémoire (du plus gourmand au plus sobre)
- **P** : tri les processus par utilisation du processeur (du plus gourmand au plus sobre)
- **s** : permet de changer la durée de rafraîchissement de top
- **k** : permet d'envoyer un signal à un processus

Les processus légers

Les Thread ID (TID ou SPID)

Les processus légers SPID connus aussi sous le nom de TID (Threads ID) sont les numéros des threads. Explication par l'exemple :

Voir aussi : Processus léger sur Wikipédia

Limiter un processus

CPULimit est un utilitaire qui permet de limiter la ressource CPU utilisée par un processus

```
apt-get install cpulimit
```

La syntaxe est la suivante : **cpulimit --pid <pid> --limit <limit>**

<pid> est le pid du processus cible, et <limit> la limitation effective en pourcentage (maximum 100% sur un simple coeur, 200 pour un double ...)

Exemple :

```
cpulimit --pid 2960 --limit 55
```

Exemple avec le nom de processus (au lieu du pid) :

```
cpulimit --exe bigloop --limit 40
```

Conserver un processus en activité

Il est possible de laisser tourner un processus lancé depuis la console même sans être connecté. On utilise la fonction **nohup** qui permet d'ignorer le signal **SIGHUP** lors de la déconnexion.

```
$ nohup application &
```

La sortie standard est automatiquement redirigée vers nohup.out dans le répertoire courant, sauf si une redirection est spécifiée.

```
$ nohup application & > application.log
```

Locale

Sous Linux, les locale sont des variables d'environnement qui ont un impact essentiel sur le bon fonctionnement des logiciels.

Une de ces variable permet notamment de configurer le codage de caractère utilisé par un logiciel. Aujourd'hui, les systèmes utilisent massivement utf-8 depuis le début du siècle. Ceci permet de manipuler des textes ou des fichiers comportant les caractères unicode[1]

Notes

Configuration du réseau

Quelques définitions

Nous avons trois grands axes pour configurer un réseau.

L'adresse IP

Tout d'abord, une adresse IP est unique sur un réseau. C'est une suite de 4 nombres allant de 0 à 255 (inclus) séparés par des points (par exemple 192.168.1.32). Cette adresse IP appartient à une classe réseau. Il existe 5 classes d'IP :

- Classe A -> de 0.0.0.0 à 127.255.255.255
- Classe B -> de 128.0.0.0 à 191.255.255.255
- Classe C -> de 192.0.0.0 à 223.255.255.255
- Classe D -> Réservée à un usage multicast.
- Classe E -> Utilisée à titre expérimental.

De l'adresse IP et de sa classe, on peut déduire trois grandes adresses:

- Le masque réseau (netmask) qui serait dans notre cas 255.255.255.0 (classe C) sert à identifier le réseau associé à une adresse IP.
- L'adresse réseau (network address) : 192.168.1.0, cette adresse ne peut être attribuée à aucun ordinateur sur le réseau.
- L'adresse de diffusion (broadcast address) : permettant d'envoyer un message à toutes les machines situées sur le réseau (ici 192.168.1.255).

Il existe la notation **CIDR**(Classless Inter-Domain Routing) (/24). Sous Linux, on a l'outil **ipcalc** qui permet de voir cette notation.

Exemple :

```
ipcalc 192.168.30.0/24
ipcalc 192.168.30.0/16   (classe B)
ipcalc 192.168.30.0/8    (Classe A)
```

La passerelle

La passerelle ou **GATEWAY** permet de relier deux réseaux informatiques différents. Dans notre cas, elle est du type : 192.168.1.1.

Le serveur DNS

Le serveur **DNS** ou **Domain Name System** établit la liaison entre un nom de domaine et une adresse IP. Grâce au serveur DNS nous pouvons retrouver une machine sur un réseau via son adresse IP ou bien sous sa forme canonique (par exemple www.domaine.fr).

Les fichiers de configuration

/etc/network/interfaces

Sur la distribution Debian, les paramètres réseaux vont être stockés à cet emplacement: **/etc/network/interfaces**. Editons ce fichier avec **vim**. Nous retouvons l'adresse Loopback (127.0.0.1) autrement appelé le localhost qui boucle sur notre propre machine.

Nous allons maintenant attribuer les adresses nécessaires au bon fonctionnement de notre serveur, car celles-ci avaient été attribuées automatiquement lors de l'installation. Nous allons donc enlever la configuration DHCP en nous attribuant une adresse spécifique pour notre serveur (192.168.30.219), ce qui nous donne :

AVANT :

```
allow-hotplug eth0
iface eth0 inet dhcp
```

APRES :

```
auto eth0
iface eth0 inet static
      address 192.168.30.219
      netmask 255.255.255.0
      network 192.168.30.0
      broadcast 192.168.30.255
      gateway 192.168.30.1
```

Il faut maintenant relancer le fichier et vérifier que la configuration a bien été prise en compte.

```
 # /etc/init.d/networking restart
 # ifconfig
```

/etc/resolv.conf

Ce fichier contient les adresses IP des serveurs de noms :

```
$ cat /etc/resolv.conf
search mondomaine.fr autredomaine.fr
nameserver 192.168.30.1
```

Les options **search** et **domain** permettent de spécifier un nom de domaine à rajouter à toute requête portant sur un nom qui ne contient pas de point. Ce qui signifie ici qu'une recherche sur pc235 sera tentée en utilisant pc235.mondomaine.fr. Si aucune correspondance n'est trouvée, une autre recherche sera alors tentée avec pc235.autredomaine.fr, etc.

```
$ ping pc235
PING pc235.mondomaine.fr (192.168.30.235) 56(84) bytes of data.
...
```

L'option **domain** permet d'utiliser un nom (mais un seul) comprenant éventuellement un sous-domaine:

```
$ cat /etc/resolv.conf
domain truc.mondomaine.fr
nameserver 192.168.30.1
```

Dans ce cas, une recherche sera d'abord tentée avec **pc235.truc.mondomaine.fr**, puis, en cas d'échec, avec **pc235.mondomaine.fr**.

Ces deux options ne doivent pas figurer en même temps dans le fichier *resolv.conf*.

L'option **nameserver** permet d'indiquer l'adresse IP du serveur de noms. On peut mettre plusieurs lignes **nameserver** pour indiquer plusieurs serveurs de noms, mais attention, l'ordre à une importance : lors de la résolution d'un nom, le système va contacter le premier serveur DNS, et si il n'a pas obtenu de réponse au bout d'un certain temps (timeout généralement de 30s), il contactera le deuxième.

/etc/hostname

Ce fichier contient le nom de la machine (hostname). Il est utilisé lors du démarrage de la machine pour positionner le nom de la machine.

/etc/hosts

Le fichier **/etc/hosts** est l'ancêtre du DNS. A la création de l'Internet, il y avait très peu d'ordinateurs connectés, et ce fichier contenait la liste de ces ordinateurs et de leurs adresses IP. Chaque fois que de nouveaux ordinateurs étaient connectés à Internet, il fallait mettre à jour ce fichier pour accéder à ces nouveaux ordinateurs. Vu le nombre croissant d'ordinateurs connectés à Internet, ce fonctionnement n'était plus adapté et à donné lieu à l'invention du DNS.

Désormais, ce fichier permet de connaitre les adresses IP d'ordinateurs non présents dans le DNS.

Ce fichier contient une ligne pour l'entrée **loopback** :

```
127.0.0.1       localhost
```

Ce fichier doit aussi contenir une ligne indiquant le nom de l'ordinateur et son adresse IP; ceci est utile en cas de défaillance du serveur DNS :

```
192.168.30.230  pc230.mondomaine.fr pc230 pcalex
```

A la fin de ce fichier, on trouve désormais des entrées pour IPv6 :

```
::1     ip6-localhost ip6-loopback
fe00::0 ip6-localnet
ff00::0 ip6-mcastprefix
ff02::1 ip6-allnodes
ff02::2 ip6-allrouters
ff02::3 ip6-allhosts
```

On peut y ajouter :

```
2620:0:862:ed1a::1 pc230.mondomaine.fr pc230 pcalex
```

Puis tester immédiatement en local :

```
$ host pc230.mondomaine.fr
pc230.mondomaine.fr has address 192.168.30.230
pc230.mondomaine.fr has IPv6 address 2620:0:862:ed1a::1
```

/etc/host.conf

Historiquement, ce fichier contenait l'ordre de recherche pour la résolution des noms : d'abord la consultation du fichier **/etc/hosts**, et ensuite l'interrogation du DNS :

```
order hosts,bind
```

Désormais, ces préférences d'ordre de recherche sont indiqués dans le fichier **/etc/nsswitch.conf** (voir ci-dessous).

/etc/nsswitch.conf

Ce fichier remplace désormais le fichier **/etc/host.conf** et permet d'indiquer au système l'ordre de recherche pour la résolution des noms :

```
more /etc/nsswitch.conf
...
hosts:          files dns
networks:       files
...
```

L'option **files** indique au système de consulter d'abord les fichiers (**/etc/hosts** pour les noms d'hotes et **/etc/networks** pour les noms de domaines) avant d'aller interroger le DNS.

/etc/networks

Ce fichier contient le nom des réseaux et leur adresse réseau :

```
default         0.0.0.0
loopback        127.0.0.0
link-local      169.254.0.0
mondomaine.fr   192.168.30.0
```

Les commandes

hostname

La commande **hostname** permet de connaitre le nom de l'ordinateur :

```
# hostname
pc230
```

L'option **-f** (**-f** ou **--fqdn**) permet de connaitre le nom de l'ordinateur avec son nom de domaine (**F**ull **Q**ualified **D**omain **N**ame) :

```
# hostname -f
pc230.mondomaine.fr
```

La commande **hostname** permet aussi de changer dynamiquement nom de machine, mais attention, cette modification est temporaire et ne sera plus active après un reboot. Pour changer de manière permanente le nom de la machine, il faut modifier le fichier **/etc/hostname** et le fichier **/etc/hosts** (voir ci-dessus) :

```
# hostname
pc230
# hostname pcalex
# hostname
pcalex
```

ifconfig

Utilisé sans argument, la commande **ifconfig** affiche les interfaces réseau actives :

```
# ifconfig
eth0    Lien encap:Ethernet  HWaddr 00:05:5D:E1:F6:11
        inet adr:192.168.30.230  Bcast:192.168.30.255 Masque:255.255.255.0
        adr inet6: fe80::205:5dff:fee1:f611/64 Scope:Lien
        UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
        RX packets:16383 errors:0 dropped:0 overruns:0 frame:0
        TX packets:5998 errors:0 dropped:0 overruns:0 carrier:0
        collisions:0 lg file transmission:1000
        RX bytes:8467107 (8.0 MiB)  TX bytes:810927 (791.9 KiB)
        Interruption:10 Adresse de base:0xc800
```

```
lo      Lien encap:Boucle locale
        inet adr:127.0.0.1  Masque:255.0.0.0
        adr inet6: ::1/128 Scope:Hôte
        UP LOOPBACK RUNNING  MTU:16436  Metric:1
        RX packets:24 errors:0 dropped:0 overruns:0 frame:0
        TX packets:24 errors:0 dropped:0 overruns:0 carrier:0
        collisions:0 lg file transmission:0
        RX bytes:2352 (2.2 KiB)  TX bytes:2352 (2.2 KiB)
```

L'option **-a** permet d'afficher toutes les interfaces, y compris celles qui ne sont pas activées.

On peut aussi spécifier à **ifconfig** l'interface à afficher :

```
# ifconfig eth0
...
```

La commande **ifconfig** permet également de configurer une interface réseau. Attention, cette modification est faite dynamiquement et ne sera plus active après un reboot.

```
# ifconfig eth0 10.0.0.1 netmask 255.0.0.0
```

La commande **ifconfig** peut aussi créer des alias d'interface réseau. On peut ainsi affecter plusieurs adresses à une seule interface réseau.

```
# ifconfig eth0:0 192.168.0.100 netmask 255.255.255.0

# ifconfig
# eth0      Link encap:Ethernet  HWaddr 00:1e:8c:26:af:c5
            UP BROADCAST MULTICAST  MTU:1500  Metric:1
            Packets reçus:0 erreurs:0 :0 overruns:0 frame:0
            TX packets:0 errors:0 dropped:0 overruns:0 carrier:0
            collisions:0 lg file transmission:1000
            Octets reçus:0 (0.0 B) Octets transmis:0 (0.0 B)

  eth0:0    Link encap:Ethernet  HWaddr 00:1e:8c:26:af:c5
            inet adr:192.168.0.100  Bcast:192.168.0.255  Masque:255.255.255.0
            UP BROADCAST MULTICAST  MTU:1500  Metric:1
```

Attention : Si vous supprimez l'interface principale (ici : eth0), tous les alias qui en dépendent seront supprimés.

arp

Le protocole ARP permet de trouver l'adresse MAC d'un ordinateur de mon réseau en fonction de son adresse IP.

La commande **arp** permet d'afficher la table de correspondance **adresses IP => adresses MAC** :

```
# arp -an
fw.mondomaine.fr (192.168.30.1) at 00:10:5A:DC:2B:4B [ether] on eth0
pc235.mondomaine.fr (192.168.30.235) at 00:11:95:DD:FD:F3 on eth0
```

Attention, cette table de correspondance est régulièrement vidée.

route

La commande **route** permet d'afficher la table de routage réseau :

```
# route
```

```
Table de routage IP du noyau
Destination       Passerelle       Genmask         Indic Metric Ref    Use Iface
d12.mondomaine.fr *                255.255.255.0   U     0      0        0 eth0
default           fw.mondomaine.fr 0.0.0.0         UG    0      0        0 eth0
```

L'option **-n** affiche la table de routage réseau sans remplacer les adresses IP par leurs noms canoniques :

```
# route -n
Table de routage IP du noyau
Destination     Passerelle      Genmask         Indic Metric Ref    Use Iface
192.168.30.0    0.0.0.0         255.255.255.0   U     0      0        0 eth0
0.0.0.0         192.168.30.1    0.0.0.0         UG    0      0        0 eth0
```

On peut obtenir le même résultat en utilisant la commande **netstat** :

```
# netstat -rn
Table de routage IP du noyau
Destination     Passerelle      Genmask         Indic   MSS Fenêtre irtt Iface
192.168.30.0    0.0.0.0         255.255.255.0   U         0 0          0 eth0
0.0.0.0         192.168.30.1    0.0.0.0         UG        0 0          0 eth0
```

La commande route permet de modifier la table de routage. On peut ainsi rajouter une route pour contacter un réseau ou une machine particulière.

Exemple : joindre le réseau 10.0.0.0/8

```
pc210:~# route -n
Table de routage IP du noyau
Destination     Passerelle      Genmask         Indic Metric Ref    Use Iface
192.168.30.0    0.0.0.0         255.255.255.0   U     0      0        0 eth0
0.0.0.0         192.168.30.1    0.0.0.0         UG    0      0        0 eth0
pc210:~#
pc210:~# route add -net 10.0.0.0 netmask 255.0.0.0 gw 192.168.30.99 dev eth0
pc210:~#
pc210:~# route -n
Table de routage IP du noyau
Destination     Passerelle      Genmask         Indic Metric Ref    Use Iface
192.168.30.0    0.0.0.0         255.255.255.0   U     0      0        0 eth0
10.0.0.0        192.168.30.99   255.0.0.0       UG    0      0        0 eth0
0.0.0.0         192.168.30.1    0.0.0.0         UG    0      0        0 eth0
```

Exemple : joindre la machine 1.2.3.4 :

```
pc210:~# route -n
Table de routage IP du noyau
Destination     Passerelle      Genmask         Indic Metric Ref    Use Iface
192.168.30.0    0.0.0.0         255.255.255.0   U     0      0        0 eth0
0.0.0.0         192.168.30.1    0.0.0.0         UG    0      0        0 eth0
pc210:~#
pc210:~# route add -host 1.2.3.4 gw 192.168.30.98 dev eth0
pc210:~#
pc210:~# route -n
Table de routage IP du noyau
Destination     Passerelle      Genmask         Indic Metric Ref    Use Iface
1.2.3.4         192.168.30.98   255.255.255.255 UGH   0      0        0 eth0
192.168.30.0    0.0.0.0         255.255.255.0   U     0      0        0 eth0
0.0.0.0         192.168.30.1    0.0.0.0         UG    0      0        0 eth0
```

ping

La commande **ping** permet de tester si on arrive à joindre un serveur donné. Cette commande utilise le protocole **ICMP** et envoi des packets ICMP de type **echo-request**. Si l'ordinateur distant est joignable, il répondra à ce message par un packet ICMP **echo-reply**.

```
$ ping google.fr
PING google.fr (216.239.59.104) 56(84) bytes of data.
64 bytes from 216.239.59.104: icmp_seq=1 ttl=236 time=43.2 ms
64 bytes from 216.239.59.104: icmp_seq=2 ttl=236 time=44.3 ms
64 bytes from 216.239.59.104: icmp_seq=3 ttl=236 time=43.1 ms
64 bytes from 216.239.59.104: icmp_seq=4 ttl=236 time=72.2 ms
...
CTRL-c pour arreter
```

Si l'ordinateur est injoignable, on n'obtiendra pas de réponse à nos packets ICMP :

```
$ ping pc235
PING pc235.mondomaine.fr (192.168.30.235) 56(84) bytes of data.
From pc230.mondomaine.fr (192.168.30.230) icmp_seq=2 Destination Host Unreachable
From pc230.mondomaine.fr (192.168.30.230) icmp_seq=3 Destination Host Unreachable
```

Attention, si l'ordinateur distant ne répond pas, cela ne signifie pas obligatoirement qu'il est indisponible. Le problème peut venir du réseau (la commande **traceroute** permettra de déterminer l'endroit qui bloque), ou il peut être paramétré pour ne pas répondre au protocole ICMP, ou un firewall sur la route peut bloquer le protocole ICMP.

L'option **-c** de la commande **ping** permet d'indiquer le nombre de packets à envoyer :

```
$ ping -c 2 google.fr
PING google.fr (216.239.59.104) 56(84) bytes of data.
64 bytes from 216.239.59.104: icmp_seq=1 ttl=236 time=43.6 ms
64 bytes from 216.239.59.104: icmp_seq=2 ttl=236 time=44.6 ms

--- google.fr ping statistics ---
2 packets transmitted, 2 received, 0% packet loss, time 5104ms
rtt min/avg/max/mdev = 43.643/44.164/44.685/0.521 ms
```

traceroute

La commande **traceroute** permet d'afficher la route empruntée pour atteindre un ordinateur donné :

```
# traceroute www.google.fr
traceroute: Warning: www.google.fr has multiple addresses; using 209.85.135.103
traceroute to www.l.google.com (209.85.135.103), 30 hops max, 40 byte packets
1  fw.mondomaine.fr (192.168.30.1)  1.854 ms  0.323 ms  0.281 ms
2  192.168.10.1 (192.168.10.1)  0.658 ms  0.594 ms  0.522 ms
...
```

mtr

La commande **mtr** (**m**y **tr**aceroute) permet aussi d'afficher la route empruntée pour atteindre un ordinateur donné, mais ré-actualise la liste en permanence :

```
                    My traceroute  [v0.71]
pc230 (0.0.0.0)                            Tue Dec  4 13:13:23 2007
```

```
                         Packets               Pings
Host                      Loss%   Snt   Last   Avg  Best  Wrst StDev
1. fw.mondomaine.fr       0.0%    39    0.7   0.6   0.4   7.4   1.1
2. 192.168.10.1           0.0%    39    0.6   0.8   0.6   7.2   1.1
...
```

nslookup

La commande **nslookup** permet d'interroger un serveur DNS :

```
$ nslookup
> set type=any
> google.fr
Server:         192.168.30.1
Address:        192.168.30.1#53

Non-authoritative answer:
Name:   google.fr
Address: 216.239.59.104
Name:   google.fr
Address: 66.249.93.104
Name:   google.fr
Address: 72.14.221.104
google.fr       nameserver = ns4.google.com.
google.fr       nameserver = ns1.google.com.
google.fr       nameserver = ns2.google.com.
google.fr       nameserver = ns3.google.com.
```

```
Authoritative answers can be found from:
google.fr       nameserver = ns2.google.com.
google.fr       nameserver = ns3.google.com.
google.fr       nameserver = ns4.google.com.
google.fr       nameserver = ns1.google.com.
ns1.google.com  internet address = 216.239.32.10
>
CTRL-d pour sortir
```

Cette commande est intéractive, elle nécessite l'intervention de l'utilisateur. On utilise désormais les commandes **host** et **dig** pour effectuer les mêmes opérations (voir ci-dessous).

host

La commande **host** permet d'effectuer une résolution DNS :

```
$ host pc235
pc235.mondomaine.fr has address 192.168.30.235
pc235.mondomaine.fr mail is handled by 0 pc235.mondomaine.fr.
```

Elle permet aussi d'effectuer une résolution DNS inverse (quel nom canonique est associé à une adresse IP donnée ?) :

```
$ host 192.168.30.235
235.30.168.192.in-addr.arpa domain name pointer pc235.mondomaine.fr.
```

dig

La commande **dig** permet d'interroger les enregistrements DNS (appellé aussi **R**essources **R**ecords ou **RR**) d'un nom de domaine donné :

```
$ dig google.fr
....
;; QUESTION SECTION:
;google.fr.                     IN      A

;; ANSWER SECTION:
google.fr.              205     IN      A       66.249.93.104
google.fr.              205     IN      A       72.14.221.104
...
```

On peut indiquer à la commande **dig** le **R**essource **R**ecord que l'on désire connaitre :

```
$ dig google.fr MX
...
;; ANSWER SECTION:
google.fr.              10800   IN      MX      10 smtp1.google.com.
google.fr.              10800   IN      MX      10 smtp2.google.com.
google.fr.              10800   IN      MX      10 smtp3.google.com.
...
```

On peut indiquer à la commande **dig** le serveur DNS à interroger :

```
$ dig @ns1.google.com google.fr
...
;; ANSWER SECTION:
google.fr.              1800    IN      A       66.249.93.104
google.fr.              1800    IN      A       216.239.59.104
...
```

whois

La commande **whois** permet d'interroger la base de données **whois** contenant les informations sur le propriétaire du domaine et les personnes responsables pour les aspects administratif et technique :

```
$ whois google.fr
...
domain:      google.fr
address:     GOOGLE INC
address:     28, rue Juliette Lamber
address:     75017 Paris
address:     FR
admin-c:     VB2334-FRNIC
tech-c:      NA25-FRNIC
zone-c:      NFC1-FRNIC
nserver:     ns1.google.com
...
```

ip

Les utilisateurs et groupes

Les fichiers de configuration

/etc/passwd

- contient la liste des utilisateurs, un par ligne
- contient d'autres informations relatives aux utilisateurs, dont notamment les **UID** (**U**ser **ID**entifiant) et **GID** (**G**roup **ID**entifiant)
- contenait autrefois les mots de passe chiffrés de chaque utilisateur, mais désormais ces mots de passe sont stockés dans le fichier **/etc/shadow**

Le format de ce fichier est indiqué dans la page de manuel du fichier de configuration passwd (section 5) :

```
$ man 5 passwd
...
- nom de connexion de l'utilisateur (« login »)
- un mot de passe chiffré optionnel
- l'identifiant numérique de l'utilisateur (UID)
- l'identifiant numérique du groupe de l'utilisateur (GID)
- le nom complet de l'utilisateur (champ GECOS)
- le répertoire personnel de l'utilisateur
- l'interpréteur de commandes de l'utilisateur
```

La majorité des comptes définis dans ce fichier sont des comptes administratifs ne servant uniquement à l'exécution de certaines applications (daemons). Ainsi, les processus fonctionnant avec ces comptes sont limités par les droits d'accés de ces derniers.

/etc/shadow

- contient les mots de passe chiffrés
- ne peut être lu uniquement par le root ou les membres du groupe shadow
- contient des informations relatives à l'expiration des mots de passe

Le format de ce fichier est indiqué dans la page de manuel du fichier de configuration shadow (section 5) :

```
$ man 5 shadow
...
- nom de connexion de l'utilisateur (« login »)
- mot de passe chiffré
- nombre de jours, comptés à partir du 1er janvier 1970, depuis le dernier changement de mot de passe
- nombre de jours à attendre avant de pouvoir changer le mot de passe
- nombre de jours après lesquels le mot de passe doit être changé
- nombre de jours avant la fin de validité du mot de passe et pendant lesquels l'utilisateur est averti
- nombre de jours après la fin de validité provoquant la
désactivation du compte
- nombre de jours, comptés à partir du 1er janvier 1970, depuis que le compte est désactivé
- champ réservé
```

La commande **chage** permet de changer les informations relatives à l'expiration des mots de passe (man chage).

/etc/group

- contient la liste des groupes, un par ligne
- pour chaque groupe, le numéro du groupe (**GID**)
- les membres du groupe

Le format de ce fichier est indiqué dans la page de manuel du fichier de configuration group (section 5) :

```
$ man 5 group
...
- nom du groupe
- mot  de passe chiffré du groupe. Si ce champ est vide, aucun mot de passe n'est nécessaire
- GID : identifiant numérique du groupe
- tous les noms des membres du groupe, séparés  par  des  virgules
```

/etc/gshadow

Contient les informations cachées sur les groupes. Il contient des lignes avec les champs suivant séparés par des deux-points :

- nom du groupe
- mot de passe chiffré
- liste d' administrateur du groupe séparés par des virgules
- liste des membres du groupe séparés par des virgules

gpasswd

La commande **gpasswd** administre le fichier /etc/group. Elle permet d' affecter un mot de passe à un groupe.

```
# addgroup testgrp
# Ajout du groupe "testgrp" (identifiant 1001)...
# Terminé.

# gpasswd testgrp
# Changement du mot de passe pour le groupe test
# Nouveau mot de passe : 'mdp'
# Nouveau mot de passe (pour vérification) : 'mdp'
#
```

newgrp

Newgrp change l´identifiant de groupe réel actuel à la valeur du groupe indiqué, ou au groupe par défaut défini dans /etc/passwd si aucun nom de groupe n´est fourni.

La commande groups permet de connaitre les groupes dont je fais partie :

```
$ groups
alex dialout cdrom floppy audio video plugdev netdev powerdev
```

On peut aussi obtenir la liste des groupes (et leurs GID) avec la commande id :

```
$ id
uid=1000(alex) gid=1000(grpalex)
groupes=20(dialout),24(cdrom),25(floppy),29(audio),44(video),46(plugdev),106(netdev),109(powerdev),1000(grpalex)
```

Conversion avec ou sans shadow

Vérification de passwd et group

Gérer les utilisateurs

Ajouter un utilisateur

La commande **useradd** permet d'ajouter des utilisateurs. Le fichier **/etc/default/useradd** contient les paramètres par défaut de la commande.

Invoqué sans option, cette commande crée l'utilisateur, mais le compte n'est pas activé. De plus, elle n'a pas crée le répertoire de travail de l'utilisateur.

```
# useradd paul
```

```
# grep paul /etc/passwd
paul:x:1002:1002::/home/paul:/bin/sh
```

```
# grep paul /etc/shadow
paul:!:13823:0:99999:7:::
```

```
# grep paul /etc/group
paul:x:1002:
```

```
# ls -al /home/paul
ls: /home/paul: Aucun fichier ou répertoire de ce type
```

Sous Debian, on utilisera plutôt la commande **adduser** qui offre plus d'options (positionnement du mot de passe et du champ GECOS à la création du compte, et création du répertoire de travail).

```
# adduser jacques
Ajout de l'utilisateur « jacques »...
Ajout du nouveau groupe « jacques » (1003)...
Ajout du nouvel utilisateur « jacques » (1003) avec le groupe « jacques »...
Création du répertoire personnel « /home/jacques »...
Copie des fichiers depuis « /etc/skel »...
Enter new UNIX password:
Retype new UNIX password:
passwd : le mot de passe a été mis à jour avec succès
Modification des informations relatives à l'utilisateur jacques
Entrez la nouvelle valeur ou « Entrée » pour conserver la valeur proposée
      Nom complet []: Jacques
      N° de bureau []: 123
      Téléphone professionnel []: 01 02 03 04 05
      Téléphone personnel []: 01 06 07 08 09
      Autre []: Chef de projet
Ces informations sont-elles correctes ? [o/N] o
```

```
# grep jacques /etc/passwd
jacques:x:1003:1003:Jacques,123,01 02 03 04 05,01 06 07 08 09,Chef de projet:/home/jacques:/bin/bash
```

```
# grep jacques /etc/shadow
jacques:$1$OWMPkUFD$XnXmTUUJBGoLehbznyXUM/:13823:0:99999:7:::
```

```
# grep jacques /etc/group
jacques:x:1003:
```

```
# ls -al /home/jacques
total 20
drwxr-xr-x 2 jacques jacques 4096 2007-11-06 11:05 .
drwxr-xr-x 6 root    root    4096 2007-11-06 11:05 ..
-rw-r--r-- 1 jacques jacques  220 2007-11-06 11:05 .bash_logout
-rw-r--r-- 1 jacques jacques  414 2007-11-06 11:05 .bash_profile
-rw-r--r-- 1 jacques jacques 2227 2007-11-06 11:05 .bashrc
```

La commande **adduser** dispose du fichier de configuration **/etc/adduser.conf** qui permet de positionner les valeurs par défaut de la commande. La commande **adduser** offrira plus d'options que la commande **useradd**.

Le fichier **/etc/adduser.conf** permet de configurer les options par défaut de la commande **adduser**

Extrait de adduser.conf

```
# La variable DSHELL spécifie le shell par défaut de la session sur le système.
DSHELL=/bin/bash

# La varible DHOME spécifie le répertoire par défaut contenant le dossier HOME de l'utilisateur.
DHOME=/home

# Si GROUPHOMES est à "yes", alors le dossier home sera créé avec le schéma suivant : /home/"nom du groupe"/" nom de
l'utilisateur"
GROUPHOMES=no

# Si LETTERHOMES est à "yes", le dossier home sera créé dans un dossier ayant comme nom la première lettre du nom
d'utilisateur
# Par exemple : /home/u/user
LETTERHOMES=no

# La variable SKEL spécifie le dossier contenant le squelette du fichier utilisateur
# En d'autre terme, elle indique le chemin du fichier d'exemple .profile qui sera copier dans le nouveau dossier home de
l'utilisateur lors de sa création.
SKEL=/etc/skel

# FIRST_SYSTEM_[GU]ID to LAST_SYSTEM_[GU]ID inclusive is the range for UIDs for dynamically allocated administrative and
system accounts/groups.
# Please note that system software, such as the users allocated by the base-passwd package, may assume that UIDs less than
100 are unallocated.
FIRST_SYSTEM_UID=100
LAST_SYSTEM_UID=999

# FIRST_[GU]ID to LAST_[GU]ID inclusive is the range of UIDs of dynamically allocated user accounts/groups.
FIRST_UID=1000
```

```
LAST_UID=29999

# The USERGROUPS variable can be either "yes" or "no".  If "yes" each created user will be given their own group to use as a
default.  If "no", each created user will be placed in the group whose gid is USERS_GID (see below).
USERGROUPS=yes
```

La commande **adduser** permet aussi de rajouter facilement un utilisateur à un groupe donné :

Syntaxe : adduser <*le login*> <*le groupe*>

Exemple :

```
# adduser pierre audio
```

Pour vérifier, on dispose de 3 possibilités : - grep audio /etc/group - id pierre - groups pierre

Le répertoire **/etc/skel** (skeleton : squelette) contient les fichiers par défaut à copier dans les répertoires de travail des utilisateurs nouvellement créés.

Modifier un utilisateur

La commande **usermod** permet de modifier les informations relatives à un utilisateur donné.

Dans certains cas, il sera plus simple de modifier directement les fichiers **/etc/passwd** et **/etc/group** plutôt que d'utiliser cette commande.

Supprimer un utilisateur

La commande **userdel** permet de supprimer des utilisateurs.

Sous Debian, on utilisera plutot la commande **deluser** qui offre plus d'options (notamment la possibilité de créer une archive de tous les fichiers de l'utilisateur avant de supprimer son répertoire de travail).

De la même manière que **adduser**, la commande **deluser** dispose du fichier de configuration **/etc/deluser.conf** qui permet de spécifier son comportement par défaut.

Extrait de **deluser.conf**

```
REMOVE_HOME = 0
REMOVE_ALL_FILES = 0

# Backup files before removing them. This options has only an effect if REMOVE_HOME or REMOVE_ALL_FILES is set.
BACKUP = 0
BACKUP_TO = "."

# delete a group even there are still users in this group
ONLY_IF_EMPTY = 0

# exclude these filesystem types when searching for files of a user to backup
EXCLUDE_FSTYPES = "(proc|sysfs|usbfs|devpts|tmpfs)"
```

Gérer les groupes

Ajouter un groupe

La commande **groupadd** permet d'ajouter un groupe.

```
# groupadd compta
```

Sous Debian, on dispose de la commande **addgroup**, qui est en réalité un lien vers la commande **adduser**.

```
# which addgroup
/usr/sbin/addgroup
```

```
# ls -l /usr/sbin/addgroup
lrwxrwxrwx 1 root root 7 2007-10-24 12:53 /usr/sbin/addgroup -> adduser
```

```
# addgroup compta
```

```
Ajout du groupe « compta » (identifiant 1004)...
Terminé.
```

Modifier un groupe

La commande **groupmod** permet de modifier les informations relatives à un groupe donné.

Dans certains cas, il sera plus simple de modifier directement le fichier **/etc/group** plutôt que d'utiliser cette commande.

Supprimer un groupe

La commande **groupdel** permet de supprimer des groupes.

Sous Debian, on dispose de la commande **delgroup**, qui est en réalité un lien vers la commande **deluser**.

```
# which delgroup
/usr/sbin/delgroup
```

```
# ls -l /usr/sbin/delgroup
lrwxrwxrwx 1 root root 7 2007-10-24 12:53 /usr/sbin/delgroup -> deluser
```

Modifier manuellement les fichiers /etc/passwd, /etc/shadow, /etc/group et /etc/gshadow

Dans certains cas, il est plus simple d'éditer manuellement les fichiers de configuration **/etc/passwd**, **/etc/shadow**, **/etc/group** et **/etc/gshadow** plutôt que d'utiliser les commandes précédement évoquées.

Les commandes suivantes permettent de garantir qu'un seul root est en train de modifier un de ces fichiers :

1. **vipw** : modification du fichier /etc/passwd
2. **vipw -s** : modification du fichier /etc/shadow
3. **vigr** : modification du fichier /etc/group
4. **vigr -s** : modification du fichier /etc/gshadow

Pour plus d'information, se reporter aux pages de manuel (**man vipw** et **man vigr**).

Le processus d'initialisation

Le chargement du noyau Linux

Lorsque l'on allume son ordinateur, la carte mère effectue un test automatique qui se charge de tester le bon fonctionnement des composants de la carte mère (processeur, mémoire, etc.). Ce test s'appelle le POST (Power-On Self Test). Si tout est OK, ce test se termine par un (et un seul) bip.

Ceci fait, le BIOS charge ensuite le programme stocké dans le boot secteur du disque dur.

Sur architecture PC, il existe deux programmes permettant de charger le noyau Linux, ou un autre système d'exploitation.

LILO

LILO (LInux LOader) a été le premier logiciel permettant d'effectuer cette opération.

Exemple de fichier de configuration de Lilo (/etc/lilo.conf) :

```
boot=/dev/hda
map=/boot/map
install=/boot/boot.b
default=linux
prompt
timeout=50
message=/boot/message
# Premiere section : boot Windows
other=/dev/hda1
      label=windows
      table=/dev/hda
# Deuxieme section: boot Linux
image=/boot/vmlinuz
      label=linux
      root=/dev/hda4
      append="acpi=off"
      read-only
# Troisieme section: boot Linux - mode mono-utilisateur
image=/boot/vmlinuz
      label=linux_single_user
      root=/dev/hda4
      append="single acpi=off"
      read-only
```

Ce fichier se compose de paramêtres communs (disque dur de boot, timeout ...) et de sections pour chaque système d'exploitation à booter.

Chaque fois que l'on modifie ce fichier, il faut exécuter la commande **lilo** afin de re-écrire dans le boot secteur l'emplacement exact du noyau sur le disque dur.

```
# lilo
Added windows
Added linux
Added linux_single_user
```

GRUB

Aujourd'hui, on utilise le programme GRUB (GRand Unified Bootloader) car il offre des possibilités supplémentaires.

Contrairement à Lilo, Grub sait *lire* une partition Linux (ext2/3, XFS, reiserFS ...) afin de charger son fichier de configuration (/boot/grub/menu.lst) et le(s) noyau(x) Linux.

Exemple de fichier de configuration de Grub (/boot/grub/menu.lst)

```
default         0
timeout         30
color cyan/blue white/blue
# Premiere section : Linux
title           Debian GNU/Linux, kernel 2.6.18-5-686
     root       (hd1,0)
     kernel     /vmlinuz-2.6.18-5-686 root=/dev/sdb2 ro acpi=off
     initrd     /initrd.img-2.6.18-5-686
     savedefault
# Deuxieme section : Linux mode mono-utilisateur
title           Debian GNU/Linux, kernel 2.6.18-5-686 (single-user mode)
```

```
     root       (hd1,0)
     kernel      /vmlinuz-2.6.18-5-686 root=/dev/sdb2 ro acpi=off single
     initrd      /initrd.img-2.6.18-5-686
     savedefault
# Troisieme section : Windows
title           Windows NT/2000/XP (loader)
root            (hd0,0)
savedefault
makeactive
chainloader     +1
```

A la différence de Lilo, il n'est pas nécessaire d'exécuter une commande particulière lorsque l'on modifie ce fichier.

les messages du noyau Linux

Une fois chargé en mémoire, le noyau va se charger de détecter et d'initialiser les composants de la carte mère et les périphériques présents.

Ces messages sont affichés sur l'écran lors du démarrage.

La commande **dmesg** permet de consulter ces messages ultérieurement. Exemples :

```
# dmesg
Linux version 2.6.18-5-686 (Debian 2.6.18.dfsg.1-13) (dannf@debian.org) (gcc version 4.1.2 20061115 (prerelease) (Debian
4.1.1-21)) #1 SMP Fri Jun 1 00: 47:00 UTC 2007
BIOS-provided physical RAM map:
BIOS-e820: 0000000000000000 - 000000000008f000 (usable)
...
```

A noter que la commande **dmesg** affiche tous les messages détectés durant le boot, ainsi que les messages du noyau affichés par la suite (cad durant le fonctionnement du système).

Dans certains cas, l'affichage de la commande **dmesg** peut être tronqué (notamment les messages affichés durant le boot). Dans ce cas, on peut consulter le fichier **/var/log /dmesg** qui ne contient que les messages du boot.

Le processus init

Une fois que le noyau a détecté l'ensemble des composants de l'ordinateur, il lance le processus du système : *init*.

Le processus init a comme particularité d'avoir le PID (Process IDentifiant) nº 1 (c'est le premier processus), et de fait, il est le père de tous les autres.

Le processus init est lancé par le noyau pour démarrer les autres processus internes à ce dernier (noté entre crochet lorsque l'on fait **ps aux**), comme par exemple les processus kjournald qui gère les journaux des systèmes de fichiers journalisés. }}[2]

Le processus dispose d'un fichier de configuration, c'est le fichier **/etc/inittab**, contenant entre autres les runlevel, paramètres de init.

La syntaxe de ce fichier est la suivante :

code:niveau:action:commande

Exemple de fichier **/etc/inittab**[3] :

```
# Indique le runlevel par défaut (ici le 2)
id:2:initdefault:

# Script d'initiation du boot du système
si::sysinit:/etc/init.d/rcS

# mode mono-utilisateur
~~:S:wait:/sbin/sulogin

# runlevel 0 : arret du système
l0:0:wait:/etc/init.d/rc 0

# runlevel 1 : mono-utilisateur
l1:1:wait:/etc/init.d/rc 1

# runlevel 2 : multitâche sans réseau (runlevel par défaut sur Debian)
l2:2:wait:/etc/init.d/rc 2

# runlevel 3 : multitâche en réseau sans interface graphique
l3:3:wait:/etc/init.d/rc 3

# runlevel 4 : idem 3 mais pour la recherche
l4:4:wait:/etc/init.d/rc 4
```

```
# runlevel 5 : multitâche en réseau avec interface graphique et son
l5:5:wait:/etc/init.d/rc 5

# runlevel 6 : redémarrage du système
l6:6:wait:/etc/init.d/rc 6

# mode emergency
z6:6:respawn:/sbin/sulogin

# lance le reboot quand on appuie sur CTRL-ALT-DEL
ca:12345:ctrlaltdel:/sbin/shutdown -t1 -a -r now

# Pour les onduleurs
pf::powerwait:/etc/init.d/powerfail start
pn::powerfailnow:/etc/init.d/powerfail now
po::powerokwait:/etc/init.d/powerfail stop

# Démarre les ttys en mode console
1:2345:respawn:/sbin/getty 38400 tty1
2:23:respawn:/sbin/getty 38400 tty2
3:23:respawn:/sbin/getty 38400 tty3
4:23:respawn:/sbin/getty 38400 tty4
5:23:respawn:/sbin/getty 38400 tty5
6:23:respawn:/sbin/getty 38400 tty6
```

Les runlevels et les scripts de démarrage

Une fois que le système a effectué le runlevel S (initialisation minimale, comme par exemple vérifier l'intégrité des systèmes de fichiers), il va dans le runlevel 2 (Debian).

Les niveaux de Runlevel

- S : Initialisation commune à tous les runlevels
- 0 : Arrêt de la machine
- 1 : Single-User
- 2,3,4,5 : Fonctionnement normal
- 6 : Reboot de la machine

Les principe de fonctionnement est le suivant :

- tous les scripts de démarrage et d'arrêt sont stockés dans le répertoire **/etc/init.d/**.
- les répertoires rc0.d, rc1.d ... rc6.d contiennent des liens qui vont appeler les scripts d'initialisation.
- Si le lien commence par un **S**, le script va être appelé avec l'option **start**
- Si le lien commence par un **K**, le script va être appelé avec l'option **stop**
- le numéro qui suit le **S** ou le **K** indique l'ordre de démarrage ou d'arrêt.

Exemple, le serveur d'impression **cupsys** :

Le script de démarrage et d'arrêt est situé dans /etc/init.d :

```
$ ls -l /etc/init.d/cupsys
-rwxr-xr-x 1 root root 1977 2007-02-02 14:18 /etc/init.d/cupsys
```

Le lien suivant indique qu'il faut appeller ce script avec l'option start dans le runlevel 2 (runlevel par défaut sous Debian) :

```
$ ls -l /etc/rc2.d/S20cupsys
lrwxrwxrwx 1 root root 16 2007-10-24 15:40 /etc/rc2.d/S20cupsys -> ../init.d/cupsys
```

Le lien suivant indique qu'il faut appeller ce script avec l'option stop dans le runlevel 0 (arrêt de la machine) :

```
$ ls -l /etc/rc0.d/K20cupsys
lrwxrwxrwx 1 root root 16 2007-10-24 15:40 /etc/rc0.d/K20cupsys -> ../init.d/cupsys
```

La commande update-rc.d

Sous Debian, la commande **update-rc.d** permet de gérer les liens des runlevels.

Par exemple, pour enlever le démarrage automatique du serveur web apache :

```
# update-rc.d -f apache remove
Removing any system startup links for /etc/init.d/apache ...
 /etc/rc0.d/K91apache
```

```
  /etc/rc1.d/K91apache
  /etc/rc2.d/S91apache
  /etc/rc3.d/S91apache
  /etc/rc4.d/S91apache
  /etc/rc5.d/S91apache
  /etc/rc6.d/K91apache
```

Pour re-activer le démarrage automatique du serveur web apache :

```
# update-rc.d apache defaults 91
Adding system startup for /etc/init.d/apache ...
  /etc/rc0.d/K91apache -> ../init.d/apache
  /etc/rc1.d/K91apache -> ../init.d/apache
  /etc/rc6.d/K91apache -> ../init.d/apache
  /etc/rc2.d/S91apache -> ../init.d/apache
  /etc/rc3.d/S91apache -> ../init.d/apache
  /etc/rc4.d/S91apache -> ../init.d/apache
  /etc/rc5.d/S91apache -> ../init.d/apache
```

Pour connaitre la liste des options de la commande **update-rc.d**, il suffit de l'appeler sans argument :

```
# update-rc.d
usage: update-rc.d [-n] [-f] <basename> remove
       update-rc.d [-n] <basename> defaults [NN | sNN kNN]
       update-rc.d [-n] <basename> start|stop NN runlvl [runlvl] [...] .
                -n: not really
                -f: force
```

La commande chkconfig

attention : spécifique à REDHAT

Pour savoir en quel(s) runlevel est lancé un service (ie. postgresql)

```
# chkconfig --list postgresql
```

Pour positionner le lancement d'un service à un runlevel donné (ie. 3, 4 et 5)

```
# chkconfig --level 345 postgresql
```

Commandes pour manipuler les runlevel

La commande **runlevel** permet de connaître le runlevel dans lequel on est :

```
# runlevel
N 2
```

Dans cet exemple, on est dans le runlevel n°2 (fonctionnement sous Debian). Le **N** nous indique le runlevel précédent (N (No) : pas de runlevel précédent)

La commande **init** ou la commande **telinit** permet de changer de runlevel. Exemple :

```
# init 6
```

Arrêter ou redémarrer le système

Pour arrêter le système :

- Commun à tous les UNIX : **shutdown**
- Spécifique Linux et systèmes récents : **halt** et **reboot**

```
# shutdown -h now "changement de noyau" &
```

NB : on rajoute un **&** pour garder la main

Il existe différentes options telles que :

- **-h** (h pour halt) : arrêt immédiat
- **-r now** (r pour reboot) : redémarrage immédiat
- **-h +10** : arrêt différé dans 10 minutes
- **-h 10:30** : arrêt à 10h30
- **-c** : annule l'arrêt

Références

1. Consulter par exemple le wikilivre À la découverte d'Unicode.
2. (en) Dr. Sam Liles, *ICCWS2014- 9th International Conference on Cyber Warfare & Security: ICCWS 2014*, Academic Conferences Limited, 24 mars 2014 (lire en ligne (https://books.google.fr/books?id=SaKmAwAAQBAJ&pg=PA190&dq=%22The+kernel+locates+and+runs+the+Init+process+which+launches+all+of+the+other+processes+on+the+device%22&hl=fr&sa=X&ved=0ahUKEwijtsrsgbvQAhXFMhoKHRTVBK0Q6AEIHzAA#v=onepage&q=%22The%20kernel%20locates%20and%20runs%20the%20Init%20process%20which%20launches%20all%20of%20the%20other%20processes%20on%20the%20device%22&f=false))
3. https://linuxonfire.wordpress.com/2012/10/19/what-are-init-0-init-1-init-2-init-3-init-4-init-5-init-6-2/

Les systèmes de fichiers

Les systèmes de fichiers Unix

UNIX gère les inodes (data structure) dans une table qui contient des informations telles que :

- le propriétaire ;
- le groupe de fichiers ;
- les droits d'accès ;
- la date de modification ;
- le type de fichier.

Sous UNIX et LINUX on trouve des partitions, ex. : boot, var, tmp, home qui permettent d'organiser le système de fichiers. Au départ EXT2 sous UNIX, aujourd'hui EXT3 par défaut correspond a EXT2 plus le journal, qui a l'avantage de consigner tout ce que fait le système, et qui en cas de crash lui permet de démarrer plus rapidement sans erreurs. Les deux formats sont compatibles. Il existe toute sorte de systèmes de fichiers.

Non journalisés

```
  * Ext et Ext2 : Extented FS version 2 (Linux, BSD)
  * FAT : File Allocation Table (DOS/Windows, Linux, BSD, OS/2, Mac OS X). Se décompose en plusieurs catégories :
       o FAT12 ;
       o FAT16 ;
       o FAT32 ;
       o VFAT.
  * FFS : Fast File System (BSD, Linux expérimental)
  * HFS : Hierarchical File System (Mac OS, Mac OS X, Linux)
  * HPFS : High Performance FileSystem (OS/2, Linux)
  * minix fs (minix, Linux)
  * S5 (UNIX System V, Linux)
  * UFS : Unix File System (BSD, Linux en lecture seule)
```

Journalisés

```
  * BFS (BeOS, Haiku, Linux en lecture seule et expérimental)
  * Ext3 : Extented FS version 3 - notamment pour l'ajout de la journalisation (Linux, BSD)
  * Ext4 : Extented FS version 4 - notamment pour étendre sa capacité à 1024 peta-octets (Linux expérimental)
  * HFS+ (Mac OS X, Linux)
  * JFS (AIX, OS/2, Linux)
  * JFS2 AIX5
  * LFS : (Linux)
  * NSS : Novell Storage Services (Netware et Suse Linux)
  * NTFS : New Technology FileSystem (Windows NT/2000/XP, Linux (écriture disponible grace au pilote NTFS-3G), Mac OS X en
lecture seule)
  * ReiserFS (Linux, BSD en lecture seule)
  * Reiser4 (Linux expérimental)
  * Spufs : Synergistic processing unit filesystem
  * UFS+ : Unix FS + journal (BSD, Linux en lecture seule)
  * XFS (Irix, Linux, BSD en lecture seule)
  * ZFS : Zettabyte FileSystem (Solaris10, FreeBSD 8.0)
```

Réseau

```
  * AFS Andrew File System : (Aix, Linux)
  * Coda (Systèmes de fichiers informatique) (Linux)
  * NFS (tous les Unix, Linux, Mac OS X) (Windows pour la 4)
  * NCP NetWare Core Protocol (Novell NetWare, Linux en client seul)
  * SSHFS (tous les Unix ?, Linux)
  * SMB ou Server message block (Windows) (Linux, BSD et Mac OS X via Samba)
  * CIFS (Evolution de SMB, supporté par Samba ainsi que par Windows 2000 et XP)
```

Cluster

```
  * GPFS, General Parallel File Sytem : Linux, AIX
  * LustreFS, Compression de Linux et de Cluster : Linux
  * OCFS2, développé par Oracle : Linux
  * PVFS2, Parallel Virtual FileSystem version 2 : Linux, Unix
```

Spécialisés

```
   * CFS Cryptographic File System : FS chiffré (BSD, Linux)
   * cramfs : FS compressé (Linux en lecture seule)
   * EFS Encrypting File System : FS chiffré au dessus de NTFS (Windows)
   * ISO 9660 : en lecture seule sur tous les systèmes lisant les CDROM/DVDROM de données
   * JFFS et JFFS2 : FS pour support physique sans block, typiquement des cartes flash. Il est compressé et journalisé
(Linux)
   * QNX4fs : FS utilisé pour le temps réel (QNX, Linux en lecture seule)
   * UDF : le format de disque universel (système de fichiers des DVD-ROM et des disques optiques réinscriptibles tels les
CD-RW, DVD±RW, etc.)
```

La commande mount

Cette commande indique tous les systèmes de fichiers qui sont montés. On l'utilise pour accéder à une partition (en root pour accéder au CD-ROM ou au lecteur disquette).

Exemple : pour brancher une clé USB il faut la monter. On édite le fichier log par la commande tail-f /var/log/messages afin de savoir sur quel système de fichiers se trouve la clé. Puis on utilise mount pour accéder au contenu de la clé.

Exemple :

```
# mount /dev/sdc1 /mnt   # le repertoire de son choix
```

Il est possible de préciser le systéme de fichier s'il ne le detecte pas

Syntaxe :

```
# mount -t système de fichiers périphériques point de montage
```

Exemple :

```
# mount -t vfat /dev/sdc1 /mnt
```

Lorsque l'on a fini d'utiliser la clé il faut la demonter à cause du cache.

La commande umount

La commande **umount** permet de *démonter* une partition. Cette opération est indispensable si on veut être sûr que les données soient correctement écrites sur la partition.
Exemples :

```
# umount /dev/sdc1
```

ou

```
# umount /mnt
```

Si la partition est en cours d'utilisation, on ne pourra pas la démonter et la commande **umount** va retourner un message d'erreur :

```
# umount /home
umount: /home: device is busy
umount: /home: device is busy
```

Deux commandes permettent de connaître les fichiers ouverts et les processus qui sont en train d'utiliser la partition et qui empeche le *démontage*.

La commande lsof (**l**ist **o**pen **f**iles) permet de connaître les fichiers ouverts :

```
# lsof /home
COMMAND    PID   USER   FD   TYPE DEVICE    SIZE   NODE NAME
gdm       2751   root    9w   REG  22,71     761 964774 /home/alex/.xsession-errors
startkde  2882   alex  cwd    DIR  22,71    4096 964769 /home/alex
...
```

La commande **fuser** permet aussi de connaître les processus qui ont des fichiers ouverts sur la partition.

```
# fuser -m /home
/home:                2751  2882c  2961c  2962c  2965c  2967c  2969c
2971c  2976c  2978c  2979c  2981c  2983c  2984c  2986c  2990c  2992c
2993c  2995c  2997c  2999c  3002c  3037c  3683c  3684c  4102c  4161c
```

Le fichier /etc/fstab

Le fichier /etc/fstab contient la liste des partitions *montées* automatiquement au démarrage du système.

Extrait d'un fichier fstab :

```
# /etc/fstab: static file system information.
#
# <file system> <mount point>   <type>  <options>       <dump>  <pass>
proc            /proc           proc    defaults        0       0
/dev/sdb1       /               ext3    defaults,errors=remount-ro 0       1
/dev/sdb6       /home           ext3    defaults        0       2
/dev/sdb5       none            swap    sw              0       0
/dev/hda        /media/cdrom0   udf,iso9660 user,noauto     0       0
/dev/fd0        /media/floppy0  auto    rw,user,noauto  0       0
```

Pour monter automatiquement une partition au démarrage, il suffit de rajouter une ligne dans le fichier **/etc/fstab**, comme par exemple :

```
/dev/hdc1       /media/winnt    ntfs    defaults        0       0
```

Pour chaque partition, on peut spécifier des options dans le fichier **/etc/fstab**. Options les plus courantes :

- user : permet aux utilisateurs d'effectuer les opérations de montage / démontage
- noauto : ne monte pas automatiquement la partition au boot

La commande **sync** force le noyau à écrire le cache. Pour savoir la taille qu'il reste sur la clé on utilisera la commande **df** ***(disk free) -h (human readable) pour avoir les informations en kilo,mega,giga.*** *df -i* affiche le pourcentage d'occupation de la table des inodes.

Formater un système de fichiers

La commande **mkfs** (**m**a**k**e **f**iles**y**stem) permet de formater un système de fichiers. Exemple :

```
# mkfs -t ext3 /dev/sdc1
```

ATTENTION LE SYSTÈME DE FICHIER NE DOIT PAS ÊTRE MONTÉ

mkfs est un wrapper (aiguillage)

Afin de simuler le formatage, nous allons créer un disque dur virtuel. Pour simuler ceci avec un disque dur virtuel, on utilisera le pseudo périphérique **loop**.

```
# modprobe loop
# dd if=/dev/zero of=hd1 bs=1K count=100000
```

Ceci crée un fichier vide hd1 qui aura une taille de 100 Mo.

```
# losetup /dev/loop0 hd1
```

On *attache* le pseudo-périphérique **/dev/loop0** à notre fichier hd1.

Une fois le disque dur virtuel crée on peut le formater :

```
# mkfs -t ext3 /dev/loop0
# mount /dev/loop0 /media/cleusb
```

Pour le demonter :

```
# umount /media/cleusb
```

CAS PARTICULIER : pour les disquettes, on effectue en général un formatage de bas niveau :

```
# fdformat /dev/fd0        # vérifie l'intégrité de la disquette
```

```
# mkfs -t ext3 /dev/fd0    # formate la disquette
```

ATTENTION A BIEN DEMONTER AU PREALABLE LE PERIPHERIQUE

La commande suivante permet de vérifier l'intégrité du système de fichiers :

```
# fsck -t ext3 /dev/loop0
```

L'option **-c** de **fsck** permet de vérifier les badblocks (en lecture seule par défaut). L'option **-v** (verbose, soit verbeux) pour voir ce qu'il fait.

Le swap

Partition de swap

Le SWAP signifie échange, il s'agit d'une mémoire virtuelle en effet lorsque la mémoire de l'ordinateur arrive à saturation, le systéme écrit sur une partition SWAP de façon à liberer de la mémoire.

Il est possible de créer une partition SWAP grace à la commande **mkswap** :

```
# mkswap /dev/sdb5
```

On utilisera la commande **swapon** /dev/sdb5 pour l'activer :

```
# swapon /dev/sdb5
```

Avec la commande '*free*,' il est possible de vérifier si le swap est activé :

```
# free
...
Swap:      1951856      65080    1886776
```

Pour lister les partitions SWAP :

```
# cat /proc/swaps
...                             Type      Size     Used    Priority
/dev/hda3                       partition  979956  307284  -1
```

Fichier de swap

Il est possible de créer un fichier swap en complément d'une partion SWAP,pour faire ceci nous allons commencer par créer un fichier d'un giga octet :

```
# dd if=/dev/zero of=/ficswap bs=1M count=1000
```

On initialise ensuite le fichier en swap :

```
# mkswap /ficswap
```

On active le swap :

```
# swapon /ficswap
```

Si l'on tape la commande **free** on constate que la taille du swap a augmenté.

```
# cat /proc/swaps
```

Pour désactiver la partition de SWAP :

```
# swapoff /ficswap
```

Utilitaires disques-durs

Technologie S.M.A.R.T.

SMART est une technologie de monitoring des disques dur.

Nom du package apt : **smartmontools**

```
# smartctl -d ata -a /dev/sda
smartctl version 5.36 [i686-pc-linux-gnu] Copyright (C) 2002-6 Bruce Allen
Home page is http://smartmontools.sourceforge.net/

=== START OF INFORMATION SECTION ===
Model Family:     Western Digital Caviar SE (Serial ATA) family
Device Model:     WDC WD800JD-22MSA1
Serial Number:    WD-WMAM9TZ26746
Firmware Version: 10.01E01
User Capacity:    80 026 361 856 bytes
Device is:        In smartctl database [for details use: -P show]
ATA Version is:   7
ATA Standard is:  Exact ATA specification draft version not indicated
Local Time is:    Wed Dec  5 15:16:10 2007 CET
SMART support is: Available - device has SMART capability.
SMART support is: Enabled

=== START OF READ SMART DATA SECTION ===
SMART overall-health self-assessment test result: PASSED

General SMART Values:

(...)

Short self-test routine
recommended polling time:        (   2) minutes.
Extended self-test routine
recommended polling time:        (  33) minutes.
Conveyance self-test routine
recommended polling time:        (   6) minutes.

SMART Attributes Data Structure revision number: 16
Vendor Specific SMART Attributes with Thresholds:
ID# ATTRIBUTE_NAME          FLAG     VALUE WORST THRESH TYPE      UPDATED  WHEN_FAILED RAW_VALUE
  1 Raw_Read_Error_Rate     0x000f   200   200   051    Pre-fail  Always       -       0

(...)
```

Le fichier **/etc/smartd.conf** permet de programmer des tests, et l'envoie de messages d'alertes.

hdparm

hdparm permet de configurer les paramètres d'accès au disque dur.

```
#hdparm /dev/sdb

/dev/sdb:
 IO_support   =  0 (default 16-bit)
 readonly     =  0 (off)
 readahead    = 256 (on)
 geometry     = 9729/255/63, sectors = 156301488, start = 0
```

Pour tester les performances on utilise l'option -tT

```
# hdparm -tT /dev/sdb

/dev/sdb:
 Timing cached reads:   888 MB in  2.00 seconds = 443.99 MB/sec
 Timing buffered disk reads:  174 MB in  3.00 seconds =  57.91 MB/sec
```

hddtemp

hddtemp utilise les données SMART du disque dur spécifié et retourne sa température.

```
#hddtemp /dev/sda
```

```
/dev/sda: WDC WD800JD-22MSA1: 31°C
```

L'option **-n** permet de ne renvoyer que la valeur de la température (utile pour les scripts)

```
# hddtemp -n /dev/sda
31
```

Le système virtuel /proc

Le système de fichiers virtuel /proc

/proc n'existe pas sur le disque dur, il est fournit dynamiquement par le noyau, d'où le nom de **virtuel**.

Il permet de fournir des informations sur ce que voit le noyau.

En outre pour accéder à certains renseignements il sera nécessaire de monter obligatoirement /proc (définit dans **/etc/fstab** et fait automatiquement au boot).

```
# cat /proc/cpuinfo
```

Les commandes **ps**, **top**, **uptime** (et bien d'autres) utilisent **/proc** pour récupérer des informations.

Quelques exemples d'informations :

```
# cat /proc/partitions
major minor  #blocks  name
8     0    312571224  sda
8     1        96358  sda1
8     2    107418622  sda2
8     3    117186142  sda3
```

```
# cat /proc/swaps
Filename                Type         Size     Used      Priority
/dev/sd4          partition      5017592  1694624      -1
```

```
# cat /proc/cpuinfo
processor     : 0
vendor_id     : GenuineIntel
cpu family    : 6
model         : 15
model name    : Intel(R) Core(TM)2 CPU          6600  @ 2.40GHz
stepping      : 6
cpu MHz       : 2402.051
cache size    : 4096 KB
...
processor     : 1
vendor_id     : GenuineIntel
cpu family    : 6
model         : 15
model name    : Intel(R) Core(TM)2 CPU          6600  @ 2.40GHz
stepping      : 6
cpu MHz       : 2402.051
cache size    : 4096 KB
...
```

Certains répertoires commencent par des numéros, il s'agit des PID (Process IDentifiant) des processus en cours d'exécution. A l'intérieur, on peut obtenir des renseignements sur le processus et sur son contexte d'exécution. Exemples :

```
# cat /proc/8595/cmdline
scribus
```

```
# cat /proc/8595/environ
SSH_AGENT_PID=4687
DM_CONTROL=/var/run/xdmctl
SHELL=/bin/bash
...
```

```
# cat /proc/8595/maps
08048000-08945000 r-xp 00000000 fd:00 1519957    /usr/bin/scribus
...
b4ecb000-b4f9e000 r-xp 00000000 fd:00 92595256   /usr/lib/libBLT.2.4.so.8.4
...
```

```
# cat /proc/8595/status
Name:   scribus
State:  S (sleeping)
```

```
SleepAVG:     78%
Tgid:   8595
Pid:    8595
PPid:   4720
TracerPid:    0
Uid:    1000  1000      1000    1000
Gid:    1000  1000      1000    1000
FDSize: 32
Groups: 4 20 24 25 29 44 46 107 109 111 1000 1001
VmPeak:   892432 kB
VmSize:    84236 kB
VmLck:         0 kB
```

Les périphériques /dev

Les fichiers spéciaux

Nous allons nous intéresser plus particulièrement au répertoire **/dev** :

Ce dossier contient tous les périphériques matériels, par exemple : un lecteur CD-ROM, une carte son, une carte réseau, etc.

Il contient également les pseudo-périphériques. Quelques exemples :

- **/dev/zero** génére des zéros
- **/dev/random** génère de l'aléatoire
- **/dev/null** constitue un trou noir à octets, et notamment utilisé pour se débarrasser des fichiers et des affichages
- **/dev/loop0** permet de créer de faux périphériques de type block (stockage) à partir de fichiers créés avec la commande **dd**

Si on liste le contenu de /dev

```
# ls -l /dev | more
```

On s'aperçoit que certains périphériques sont de type **c** (de l'anglais *character*) dans ce cas ils communiquent octet par octet. Ex : un port série.

Alors que d'autres sont de types **b** (blocks) ils communiquent par blocs de données (ex: un disque dur).

Par ailleurs le noyau identifie chaque périphérique au moyen de deux numéros, le majeur (en **vert**) et le mineur (en **rouge**), exemple :

```
# ls -l /dev/sda*
brw-rw---- 1 root disk 8, 0 2007-09-22 18:08 /dev/sda
brw-rw---- 1 root disk 8, 1 2007-09-22 18:08 /dev/sda1
brw-rw---- 1 root disk 8, 2 2007-09-22 18:08 /dev/sda2
brw-rw---- 1 root disk 8, 3 2007-09-22 18:08 /dev/sda3
brw-rw---- 1 root disk 8, 4 2007-09-22 18:08 /dev/sda4
```

Les partitions sda1 à sda4 ont le même **majeur**. Le **majeur** correspond au premier disque dur SCSI ou SATA.

Le noyau identifie ensuite chaque partition grâce au numéro **mineur**.

Les **majeurs** et les **mineurs** sont définit dans la documentation accompagnant le noyau Linux dans le fichier **/usr/src/linux/Documentation/devices.txt** ... et oui! comme vous pouvez vous en douter, le noyau Linux est livré avec une documentation très fournie disponible dans le répertoire **Documentation/** livré avec le noyau, voir chapitre Le noyau Linux et les modules.

Extrait de /usr/src/linux/Documentation/devices.txt :

```
...
8 block     SCSI disk devices (0-15)
              0 = /dev/sda            First SCSI disk whole disk
             16 = /dev/sdb            Second SCSI disk whole disk
             32 = /dev/sdc            Third SCSI disk whole disk
             ...
            240 = /dev/sdp            Sixteenth SCSI disk whole disk
             ...
            Partitions are handled in the same way as for IDE
            disks (see major number 3) except that the limit on
            partitions is 15.
...
```

La commande **mknod** sert à créer de nouveaux périphériques :

```
# mknod /dev/bidon b 42 0
```

On peut également directement lui affecter les droits d'accès :

```
# mknod -m 660 /dev/bidon b 42 0
```

udevd

Cette section est vide, pas assez détaillée ou incomplète.

dmidecode

Cette commande affiche les informations sur les périphériques hardware. Par exemple pour déterminer la version et puissance de la RAM :

```
sudo dmidecode --type 17

 Getting SMBIOS data from sysfs.
 SMBIOS 3.0 present.

 Handle 0x0019, DMI type 17, 40 bytes
 Memory Device
        Array Handle: 0x0018
        Error Information Handle: Not Provided
        Total Width: 64 bits
        Data Width: 64 bits
        Size: 4096 MB
        Form Factor: SODIMM
        Set: None
        Locator: ChannelA-DIMM0
        Bank Locator: BANK 0
        Type: DDR3
        Type Detail: Synchronous
        Speed: 1600 MHz
        Manufacturer: 0215
        Serial Number: 00000000
        Asset Tag: 9876543210
        Part Number: CMSO8GX3M2C1600C11
        Rank: 2
        Configured Clock Speed: 1600 MHz
        Minimum Voltage: Unknown
        Maximum Voltage: Unknown
        Configured Voltage: 1.35 V
```

L'ordonnanceur de travaux cron

cron est un daemon qui permet de programmer des tâches répétitives.

Configuration de cron

Le fichier de configuration de **cron** est **/etc/crontab**, éditable uniquement par le *root*. Exemple :

```
# /etc/crontab: system-wide crontab
# Unlike any other crontab you don't have to run the `crontab'
# command to install the new version when you edit this file
# and files in /etc/cron.d. These files also have username fields,
# that none of the other crontabs do.

SHELL=/bin/sh
PATH=/usr/local/sbin:/usr/local/bin:/sbin:/bin:/usr/sbin:/usr/bin

# m h dom mon dow user  command
17 *    * * *   root    cd / && run-parts --report /etc/cron.hourly
25 6    * * *   root    test -x /usr/sbin/anacron || ( cd / && run-parts --report /etc/cron.daily )
47 6    * * 7   root    test -x /usr/sbin/anacron || ( cd / && run-parts --report /etc/cron.weekly )
52 6    1 * *   root    test -x /usr/sbin/anacron || ( cd / && run-parts --report /etc/cron.monthly )
#
```

Si on veut recevoir un mail à chaque échec d'une tâche, ajouter :

```
MAILTO=Mon@Email.com
```

Voici la syntaxe de la programmation de la fréquence de répétition :

m h dom mon dow user command

code	signification	valeur ou note
m	minutes	0 à 60
h	heure	0 à 23
dom	day of month / jour du mois	1 à 31
mon	month / mois	1 à 12
dow	day of week / jour de la semaine	0 à 7, 0= dimanche, 7= dimanche, on peut aussi mettre les trois premières lettres du jour en anglais
user	user / utilisateur	utilisateur appelant la commande
command	command / commande	commande appelée par l'évènement

Tous les champs peuvent être remplacés par "*" si on souhaite toutes les valeurs correspondantes.

Notation avancée :

On peut définir des valeurs multiples ou des plages de valeurs pour les heures, les jours etc... :

Par exemple, au lieu de faire plusieurs lignes de cron pour plusieurs fois le même évènement, on peut mettre plusieurs valeurs séparées par "**,**"

On peut aussi indiquer une plage de valeur dont les limites sont séparées par "**-**".

De même, on peut définir des fréquences avec "**/**". Par exemples, toutes les 2h s'écrira ***/2** dans la deuxième colonne.

Exemples

```
0 * 7 3 * root beep
```

L'utilisateur root appelle la commande beep, toutes les heures à 0 minute le 7 mars.

```
15,45 8 * * * root beep
```

L'utilisateur root appelle la commande beep, à 8h15 et 8h45 tous les jours.

```
0 8-18 * * * toto beep
```

L'utilisateur toto appelle la commande beep toutes les heures de 8h à 18h tous les jours.

```
0 12 * * mon,wed,fri titi beep
```

L'utilisateur titi appelle la commande beep à midi tous les lundi, mercredi et vendredi.

```
0 */2 * * 1-5 tata beep
```

L'utilisateur tata appelle la commande beep toutes les 2h du lundi au vendredi.

Répertoires de cron

En plus de la commande cron, il existe des répertoires spécifiques pour définir des commandes cron à fréquences précises. Ces répertoires particuliers sont localisés dans le repertoire **/etc**

Titre du tableau

cron.hourly	toutes les heures
cron.daily	tous les jours
cron.weekly	toutes les semaines
cron.monthly	tous les mois

Il suffit de mettre un fichier script exécutable dans le répertoire choisi pour qu'il s'exécute.

Dans le même ordre d'idée, on peut rajouter des répertoires particuliers reliés à des fonctions spécifiques d'une entreprise ou d'une personne, pour exécuter des cron. Il suffit de mettre des fichiers contenants des lignes **cron** dans le répertoire **/etc/cron.d/**.

exemple : /etc/cron.d/comptabilite

Une fois le crontab modifié, il faut lui faire prendre en compte les modifications. Pour cela, on dispose de trois méthodes :

```
# kill -1 <PID du processus cron>
```

ou

```
# killall -1 cron
```

ou

```
# /etc/init.d/cron restart
```

Les crontabs utilisateurs

Un utilisateur peut programmer ses propres crontabs. Ces crontabs sont stockés dans /var/spool/cron/crontabs/**<login>**

Titre du tableau

crontab -l	donne la liste des crontabs actifs
crontab -e	permet d'éditer son propre crontab

En utilisant la commande **crontab** , il n'est pas nécessaire de relancer le daemon **cron** pour que les modifications soient prises en compte.

Droits d'accès

L'horloge Linux

Avant de planifier ses opérations il est primordial de bien s'assurer que les informations de temps système sont correctes.

Réglage manuel

Pour connaître la date actuelle on utilise la commande **date** :

```
# date
mer aoû 19 12:31:00 CEST 2009
```

Cette même commande accepte des arguments pour forcer la mise à jour de l'horloge système au format "MMJJHHmmYYYY". Il faut disposer des droits super utilisateur pour cette action.

```
# date 081912372009
Wed Aug 19 12:37:00 CEST 2009
```

Enfin, il existe deux horloges dans le système, l'horloge matérielle et l'horloge logicielle. Pour synchroniser ces deux horloges utilisez la commande **hwclock**.

```
# hwclock --systohc  //initialise l'heure matérielle à partir de l'horloge système
```

```
# hwclock --hctosys  //initialise l'heure système à partir de l'horloge matérielle
```

Réglage automatique (NTP)

Attention !

Lors du changement d'heure hiver/été où on passe de 2 h à 3 h, les tâches programmées entre 2 h 00 et 2 h 59 ne seront donc pas exécutées. À l'inverse, lors du changement d'heure été/hiver, ces mêmes *tâches* seront exécutées deux fois (à 3 h, il est encore 2 h). Afin d'éviter ce désagrément, la plupart des systèmes Unix sont configurés pour utiliser le temps universel coordonné (UTC), qui n'est pas sujet à ces règles de changement d'heure.

Le backup : tar et gzip

Régulièrement, il vous faudra archiver des informations, en particulier des données utilisateurs. Pour cela, le système GNU/Linux intègre 2 outils: **TAR** et **GZIP**. Comme vous allez le voir, ces 2 outils sont complémentaires dans le procéssus d'archivage des données, ou **backup**.

Archiver des données avec tar

Le nom de cette commande vient du système de sauvegarde sur bande: Tape ARchive. Le principe est simple : prendre un grand nombre de fichiers (et/ou de répertoires) et les regrouper en un seul gros. tar est une commande récursive. Elle va archiver toute l'arborescence en partant du répertoire que vous lui spécifiez.

Archivage

Syntaxe :

```
$ tar cf <nom du fichier tar> <nom du répertoire à archiver>
```

Si vous rajoutez **v** après **c**, la commande va afficher à l'écran toutes ses actions.

Exemple:

```
# tar cvf etc.tar /etc
tar: Removing leading `/' from member names /etc
# ls -l etc.tar
-rw-r--r-- 1 root root 2829320 Nov 12 18:41 etc.tar
```

Note: Comme vous le voyez sur notre exemple, la commande tar retire le **/** du répertoire parent. En faisant cela, elle évite les écrasements de fichiers dans le cas ou vous restituez votre fichier au même endroit.

Test de l'archive

Une fois l'archivage effectué, vous pouvez vérifier votre archive avec la commande :

```
$ tar tf <nom du fichier tar>
```

Restitution

```
# tar xf <nom du fichier tar>
```

De même, si vous rajoutez **v** après **x**, la commande affichera les informations.

Exemple:

```
# tar xvf etc.tar
etc/
etc/GNUstep/
etc/GNUstep/Defaults/
etc/GNUstep/Defaults/WMGLOBAL
etc/GNUstep/Defaults/WMRootMenu
etc/GNUstep/Defaults/WMState
...
```

Compresser un fichier avec gzip

La commande **gzip** est un compresseur de données. Il va partir d'un fichier et compresser l'information pour en diminuer la taille.

Compresser un fichier

```
$ gzip <nom du fichier à compresser>
```

Exemple:

```
# ls -l images.tar*
-rw-r--r-- 1 root      root      10240 Nov  8 20:00 images.tar
# gzip images.tar
# ls -l images.tar*
-rw-r--r-- 1 root      root       4662 Nov  8 20:00 images.tar.gz
```

En utilisant ls -l , on voit trés bien la taille du fichier images non compressé, qui est passé de 10240 octets à 4662 octets.

Décompresser un fichier

```
$ gzip -d <nom du fichier gzip>
```

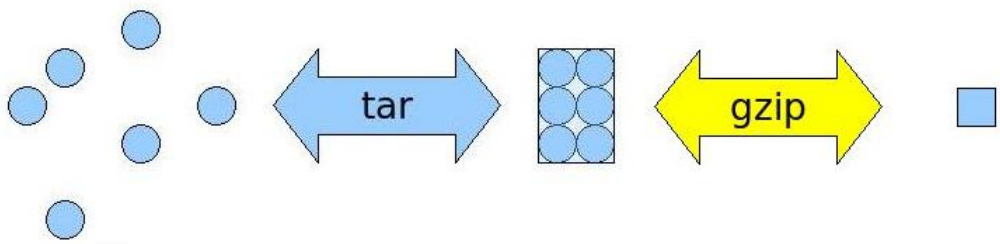
Exemple:

```
# ls -l images.tar*
-rw-r--r-- 1 root      root       4662 Nov  8 20:00 images.tar.gz
# gzip -d images.tar.gz
# ls -l images.tar*
-rw-r--r-- 1 root      root      10240 Nov  8 20:00 images.tar
```

Combiner tar et gzip

On peut combiner tar et gzip pour archiver et compresser en une action (option z) :

```
# tar cvzf <nom du fichier archivé compressé>.tar.gz <nom du répertoire à archiver et compresser>
```



Les alternatives à gzip

bzip2

bzip2 est un autre algorithme de compression. Il est plus performant mais beaucoup moins rapide que gzip.

Pour compresser un fichier avec **bzip2** :

```
# bzip2 unfichier
```

Pour le décompresser :

```
# bzip2 -d unfichier.bz2
```

Tout comme gzip, on peut indiquer à tar d'utiliser bzip2 pour compresser l'archive (option **j**):

```
# tar cjf cible.tar.bz2 /repasauver
```

Comparaison des logiciels de compression

Comparaison des logiciels de compression (http://rlwpx.free.fr/WPFF/comploc.htm)

Un shell script de sauvegarde journalière de /etc

Voici un shell script qui permet de sauvegarder quotidiennement le répertoire /etc :

```
#!/bin/bash
#
# Ne pas oublier :
# - de creer le repertoire /backup : mkdir /backup
# - de rendre ce script executable : chmod +x backup_etc
#
# Tester le script une fois manuellement : ./backup_etc
#
# Le rajouter dans la crontab
```

```
DATEJOUR=$(date +"%Y-%m-%d-%H-%M-%S")
REPABACKUPER="/etc"
REPBACKUP="/backup"
NOMFIC=$REPBACKUP/etc-$DATEJOUR.tgz
```

```
tar czf $NOMFIC $REPABACKUPER
```

Les sauvegardes incrémentales

La commande **find** permet de trouver des fichiers correspondants à des critères donnés. On peut par exemple connaître la liste des fichiers modifiés depuis 1 jour :

```
# find / -mtime -1 -type f
```

En combinant la commande **find** et la commande **tar**, on peut ainsi facilement sauvegarder les fichiers modifiés depuis la veille, et ainsi réaliser une sauvegarde incrémentale.

La commande suivante retourne la liste de tous les fichiers du système modifiés depuis hier (en excluant les fichiers contenus dans /proc, /dev et /sys), et stocke la liste de ces fichiers dans le fichier **/tmp/fic_du_jour**:

```
# find / \( -path /proc -o -path /dev -o -path /sys \) -prune -o
   -mtime -1 -type f -print > /tmp/fic_du_jour
```

On peut ensuite appeler la commande **tar** avec l'option **-T** qui permet d'indiquer la liste des fichiers à archiver :

```
# tar cz -T /tmp/fic_du_jour -f backup.tgz
```

Grâce aux tubes (pipe), on peut s'affranchir de passer par un fichier temporaire et directement enchaîner les deux commandes :

```
# find / \( -path /proc -o -path /dev -o -path /sys \) -prune -o
   -mtime -1 -type f -print | tar cz -T - -f backup.tgz
```

Toujours grâce aux tubes, on peut découper le fichier obtenu en plusieurs fichiers d'une taille donnée :

```
# find / \( -path /proc -o -path /dev -o -path /sys \) -prune -o
   -mtime -1 -type f -print | tar cz -T - | split - -b 5m backup_

# ls -lh backup_*
-rw-r--r-- 1 root root 5,0M 2007-12-06 14:30 backup_aa
-rw-r--r-- 1 root root 5,0M 2007-12-06 14:30 backup_ab
-rw-r--r-- 1 root root 5,0M 2007-12-06 14:30 backup_ac
-rw-r--r-- 1 root root 3,5M 2007-12-06 14:30 backup_ad
```

Pour re-assembler les fichiers découpés, on utilise la commande cat :

```
# cat backup_* > backup.tgz
```

Les logiciels spécialisés

ghost avec partimage

installation d'un serveur d'images

sur le serveur

```
apt-get install partimage partimage-server
```

```
apt-get install ssh
```

ensuite il faut faire plusieurs modifications.

Allez dans le répertoire partimaged et modifier le fichier partimagedusers afin de donner l'autorisation de se connecter a votre utilisateur.

```
cd /etc/partimaged/
echo "mon user">>partimagedusers ou vi partimagedusers
```

il faut créer un dossier qui contiendra les images.

```
mkdir /opt/img
```

partimage fonctionnant avec l'utilisateur partimag il faut lui donner les droits.

```
chown -R partimag:partimag /opt
```

redemmarer le service

```
/etc/init.d/partimaged restart
```

sur le client

installation de partimage

```
 #apt-get install partimage ssh
```

avec l'utilisateur root on lance partimage

```
 #partimage
```

la vous tombez sur un interface graphique

1. sélectionner la partition à sauvegarder.
2. saisissez le nom de l'image ici: /opt/img/mondebian.
3. cocher la case connexion au serveur entrer l'adresse ip puis F5.

il devrait vous demander un login et un mot de passe défini dans /etc/partimaged/partimagedusers.

On donne une description de l'image On sélectionne le type de compression et ensuite " OK ".

sauvegarder une partition NTFS

telechargement de KNOPPIX ici (http://knoppix.com/).

rebooter votre machine avec knoppix une fois démarré si il n'y a pas de DHCP sur le réseaux il faut attribuer un adresse IP.

```
 # ifconfig eth0 192.168.30.210 255.255.255.0
```

et lancer partimage

```
# partimage
```

restauration

A faire

sauvegarde de fichiers avec rsync

Serveur de sauvegardes avec rsync ssh

sur le serveur

```
#apt-get install rsync ssh
```

sur le client

```
#apt-get install rsync ssh
```

il est nécessaire de générer une paire de clés se reporter a la partie ssh-keygen ici

sauvegarder des fichiers

```
rsync -P -av /repertoire a sauver/  utilisateur@192.168.30.210:/repertoire de destination sur le
serveur/
```

options rsync

1. a : permet de copier tous les fichiers, y compris les fichiers et dossiers cachés (fichiers commençant par .)
2. -c : active la compression de type gzip pendant le transfert.
3. -t Conserver la date
4. -p Conserver les permissions
5. -o Conserver le propriétaire
6. -g Conserver les groupes
7. --delete Effacer sur la destination
8. -v affiche les opérations avant de les effectuer
9. --ignore-existing Ignorer les fichiers existants
10. -x Ne pas quitter le système de fichier
11. --progress Montrer la progression
12. --size-only Taille seulement
13. --delete : si le fichier « linux » existe dans « destination » et pas dans « source », il sera supprimé.

Les fichiers journaux syslog

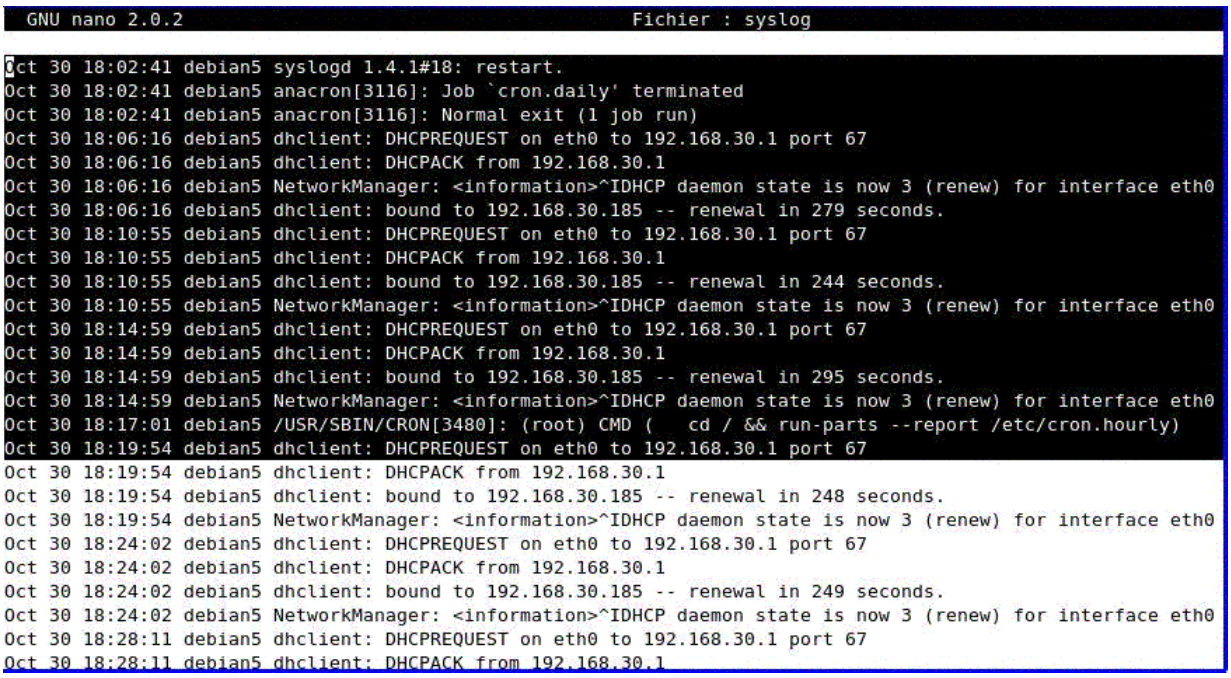
Syslog

syslog est un daemon dédié à l'enregistrement des journaux (*log*) Les journaux *log* sont stockés dans le répertoire **/var/log/**

Voici le contenu du repertoire */var/log*

un journal log est un fichier texte dont les évènements sont enregistrés, un par ligne.

« Extrait de fichier syslog - les messages sont enregistrés avec la date et l'heure de l'évènement »



```
 GNU nano 2.0.2                                Fichier : syslog

Oct 30 18:02:41 debian5 syslogd 1.4.1#18: restart.
Oct 30 18:02:41 debian5 anacron[3116]: Job `cron.daily' terminated
Oct 30 18:02:41 debian5 anacron[3116]: Normal exit (1 job run)
Oct 30 18:06:16 debian5 dhclient: DHCPREQUEST on eth0 to 192.168.30.1 port 67
Oct 30 18:06:16 debian5 dhclient: DHCPACK from 192.168.30.1
Oct 30 18:06:16 debian5 NetworkManager: <information>^IDHCP daemon state is now 3 (renew) for interface eth0
Oct 30 18:06:16 debian5 dhclient: bound to 192.168.30.185 -- renewal in 279 seconds.
Oct 30 18:10:55 debian5 dhclient: DHCPREQUEST on eth0 to 192.168.30.1 port 67
Oct 30 18:10:55 debian5 dhclient: DHCPACK from 192.168.30.1
Oct 30 18:10:55 debian5 dhclient: bound to 192.168.30.185 -- renewal in 244 seconds.
Oct 30 18:10:55 debian5 NetworkManager: <information>^IDHCP daemon state is now 3 (renew) for interface eth0
Oct 30 18:14:59 debian5 dhclient: DHCPREQUEST on eth0 to 192.168.30.1 port 67
Oct 30 18:14:59 debian5 dhclient: DHCPACK from 192.168.30.1
Oct 30 18:14:59 debian5 dhclient: bound to 192.168.30.185 -- renewal in 295 seconds.
Oct 30 18:14:59 debian5 NetworkManager: <information>^IDHCP daemon state is now 3 (renew) for interface eth0
Oct 30 18:17:01 debian5 /USR/SBIN/CRON[3480]: (root) CMD (   cd / && run-parts --report /etc/cron.hourly)
Oct 30 18:19:54 debian5 dhclient: DHCPREQUEST on eth0 to 192.168.30.1 port 67
Oct 30 18:19:54 debian5 dhclient: DHCPACK from 192.168.30.1
Oct 30 18:19:54 debian5 dhclient: bound to 192.168.30.185 -- renewal in 248 seconds.
Oct 30 18:19:54 debian5 NetworkManager: <information>^IDHCP daemon state is now 3 (renew) for interface eth0
Oct 30 18:24:02 debian5 dhclient: DHCPREQUEST on eth0 to 192.168.30.1 port 67
Oct 30 18:24:02 debian5 dhclient: DHCPACK from 192.168.30.1
Oct 30 18:24:02 debian5 dhclient: bound to 192.168.30.185 -- renewal in 249 seconds.
Oct 30 18:24:02 debian5 NetworkManager: <information>^IDHCP daemon state is now 3 (renew) for interface eth0
Oct 30 18:28:11 debian5 dhclient: DHCPREQUEST on eth0 to 192.168.30.1 port 67
Oct 30 18:28:11 debian5 dhclient: DHCPACK from 192.168.30.1
```

Dans chaque ligne d'évènement on distingue :

- La date à laquelle l'évènement a été déclenché
- Le processus déclencheur de l'évènement
- Le processus ayant demandé l'ajout du message correspondant au log
- Le niveau de gravité du message (priority)

TP : afficher les dernières procédures de login, l'heure des tentatives, si elle ont échoué ou réussi.

```
#tail -f /var/log/auth.log
```

le fichier log */var/log/auth.log* est le journal des authentifications.

```
Important : l'heure du système doit être à la bonne heure et à la bonne date, sinon la datation
des messages est éronnée, ce qui complique, si besoin est, la recherche d'anomalies de fonction-
nement du système dans les messages enregistrés dans les fichiers log.
```

syslog possède un fichier de configuration *syslog.conf*, il est stocké dans le répertoire /etc . On peut modifier ce fichier pour l'adapter à nos besoins en messages d'évènements survenus sur le système : envoi de mail, authentification, etc...

Ce fichier est un fichier texte, dont chaque ligne est séparée en deux parties :

- 1ère partie : (le ou) les processus demandeurs (séparés par un point virgule) suivi d'un point et de leur niveau de priorité : <dispositif>.<niveau>

<dispositif> est appelé *facility* <niveau> est appelé *priority*, c'est le niveau de criticité du log. Exemple : panic, error, warning, debug, info...

- 2ème partie : le fichier log correspondant (qui reçoit le message et l'ajoute à la liste de ses messages) : <fichier de log>

« Voici un exemple de fichier /etc/syslog.conf »

```
# /etc/syslog.conf     Configuration file for syslogd.
#
#                      For more information see syslog.conf(5)
#                      manpage.

#
# First some standard logfiles.  Log by facility.
#

auth,authpriv.*                 /var/log/auth.log
*.*;auth,authpriv.none          -/var/log/syslog
#cron.*                         /var/log/cron.log
daemon.*                        -/var/log/daemon.log
kern.*                          -/var/log/kern.log
lpr.*                           -/var/log/lpr.log
mail.*                          -/var/log/mail.log
user.*                          -/var/log/user.log
uucp.*                          /var/log/uucp.log

#
# Logging for the mail system.  Split it up so that
# it is easy to write scripts to parse these files.
#
mail.info                       -/var/log/mail.info
mail.warn                       -/var/log/mail.warn
mail.err                        /var/log/mail.err

# Logging for INN news system
#
news.crit                       /var/log/news/news.crit
news.err                        /var/log/news/news.err
news.notice                     -/var/log/news/news.notice
```

Swatch

L'utilitaire **swatch** peut surveiller un fichier de log et réaliser une action s'il voit passer un mot-clé.

Exemple de fichier de configuration */root/.swatchrc* :

```
#
# A appeler avec la ligne suivante :
#
# swatch --config-file=/root/.swatchrc --tail-file=/var/log/auth.log
#

watchfor        /FAILED/
                echo red
                #mail addresses=alex\@localhost,subject=Alerte AUTH
                exec /usr/bin/zenity --error --text "$_"

watchfor        /Successful/
                echo green
```

Le Serveur de log

On peut être amené à créer un serveur de log si on possède plusieurs serveurs dont on souhaite centraliser les log, par mesure de sécurité ou par commodité (facilité de consultation, d'archivage, etc...)

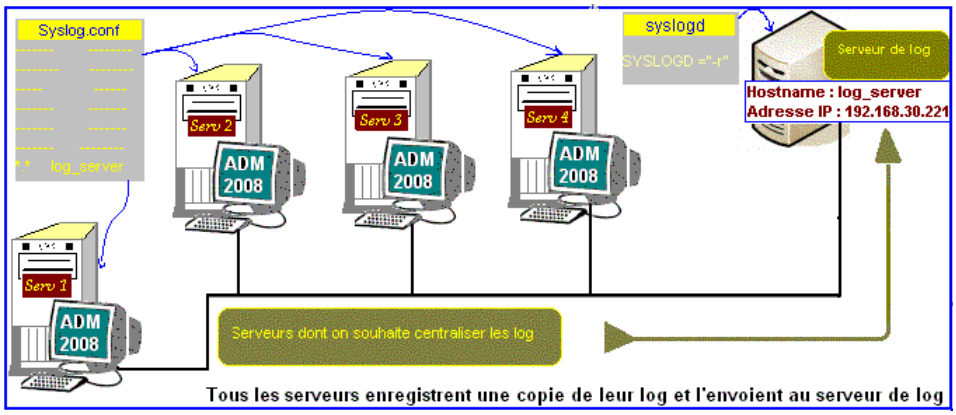
Si on monte un serveur de log, il suffit d'ajouter :

- côté serveurs envoyant les messages de log :----> une ligne dans le fichier syslog.conf de chaque serveur en mentionnant les processus envoyant les messages (exemple : *.* pour tous) et le nom du serveur de log ou son adresse IP précédée de **@**, comme ainsi :

```
*.*         @log_server ou
*.*         @192.168.30.221  ou
mail.inf    @log_server   (stocke les log d'envoi de mail)
```

- côté serveur de log :----> l'option **SYSLOGD ="-r"** dans le fichier **/etc/default/syslogd**

Schéma de réseau avec Serveur de log



Extrait du fichier syslogd d'un serveur non dédié au log - la commande SYSLOGD n'est pas activée

```
 GNU nano 2.0.2                              Fichier : /etc/default/syslogd

#
# Top configuration file for syslogd
#

#
# Full documentation of possible arguments are found in the manpage
# syslogd(8).
#

#
# For remote UDP logging use SYSLOGD="-r"
#
SYSLOGD=""
```

La commande logger

La commande logger permet d'envoyer un message à syslog même connecté en utilisateur

```
$ logger -p auth.info -t unnom "mon message à envoyer"
```

-t permet d'ajouter **unnom**, un *tag* c'est un mot quelqconque pour signer le message

Le programme logrotate

logrotate est un fichier texte situé dans **etc/**, il sert à configurer 'la politique des rotations' des logs. Effectivement, il faut supprimer les anciens fichiers log, sinon on risque une saturation du disque. Pour cela on peut programmer des rotations de logs qui se traduisent par l'archivage des fichiers log assez récents et la suppression des anciens, et ceci avec une périodicité bien définie.

La page man de logrotate donne toute la syntaxe et les mots clé.

Les lignes importantes à configurer du fichier **/etc/logrotate.conf**:

- **monthly** : tourner les logs tous les mois (enlever le **weekly** existant dans le fichier de configuration par défaut Debian)
- **rotate n** : conserver **n** fichiers (ici un fichier par mois [mettre **60**, soit 5 ans])
- **create** : créer un nouveau fichier de log
- **compress** : compresser le fichier tourné

Pour tester vos fichiers de configuration logrotate.conf (et donc de logrotate.d/*) taper : **/usr/sbin/logrotate -dv /etc/logrotate.conf**.

Installation de nouveaux logiciels

Au début de linux, installer un logiciel libre nécessitait de récupérer les sources et de les compiler. Cette étape de compilation pouvait être fastidieuse car il fallait disposer de toutes les librairies utilisées par le logiciel.

Redhat a énormément simplifié ce processus en inventant le format RPM (Redhat Package Manager), format qui propose les logiciels open-source pré-compilés.

Debian s'en est inspiré pour créer le format DEB.

- REDHAT = logiciel.*x*.*y*.rpm
- DEBIAN = logiciel.*x*.*y*.deb

x et *y* correspondent aux numéros de version du logiciel.

dpkg

dpkg est le programme qui permet d'installer, mettre à jour et supprimer un logiciel en format DEB. Sur Redhat, la commande **rpm** fait la même chose.

Installer un fichier DEB

L'option **-i** (i: install) de dpkg (debian package) permet d'installer un fichier DEB :

```
# dpkg -i logiciel-x.y.deb
```

Connaitre la liste de tous les logiciels installés

L'option **-l** (l : list) permet de connaitre la liste de tous les logiciels installés :

```
# dpkg -l
ii  iamerican                         3.1.20.0-4.3                   An American English dictionary for ispell
ii  ibritish                          3.1.20.0-4.3                   A British English dictionary for ispell
ii  icedax                            1.1.2-1                        Creates WAV files from audio CDs
ii  iceweasel                         2.0.0.6-0etch1                 lightweight web browser based on Mozilla
ii  iceweasel-gnome-support    2.0.0.6-0etch1                 Support for Gnome in Iceweasel
...
```

Savoir quel package a installé tel fichier

L'option **-S** permet de savoir quel package a installé tel fichier :

```
# dpkg -S /etc/crontab
cron: /etc/crontab
```

Connaitre le descriptif d'un package installé

L'option **--status** permet de consulter le descriptif d'un package installé :

```
# dpkg --status dosfstools
Package: dosfstools
Status: install ok installed
Priority: optional
Section: otherosfs
Installed-Size: 144
Maintainer: Roman Hodek <roman@hodek.net>
Architecture: i386
Source: dosfstools (2.11-2.1)
Version: 2.11-2.1+b1
Replaces: mkdosfs
Depends: libc6 (>= 2.3.6-6)
Conflicts: mkdosfs
Description: Utilities to create and check MS-DOS FAT filesystems
Inside of this package there are two utilities (mkdosfs alias
mkfs.dos, and dosfsck alias fsck.msdos) to create and to check MS-DOS
FAT filesystems on either hard disks or floppies under Linux. This
version uses the enhanced boot sector/superblock format of DOS 3.3+
as well as provides a default dummy boot sector code.
```

On apprend notamment :

- la priorité : s'agit-il d'un package indispensable ou optionnel ?
- la taille disque occupée par le package installé
- le nom et l'email du mainteneur Debian
- l'architecture
- le numéro de verstion
- les dépendances
- une description courte et longue

Autres options de dpkg

La commande **dpkg** dispose d'une page de manuel détaillant toutes les options disponibles :

```
# man dpkg
```

apt-get

La commande **apt-get** permet d'installer, mettre à jour et supprimer les logiciels. Le principal avantage d'apt-get par rapport à dpkg est sa capacité à télécharger les packages sur les mirroirs Debian et de résoudre les dépendances nécessaires au logiciel, et ceci de façon récursive.

Installer un logiciel

L'option **install** de la commande **apt-get** permet d'installer un logiciel :

```
# apt-get install scribus
Lecture des listes de paquets... Fait
Construction de l'arbre des dépendances... Fait
Les paquets supplémentaires suivants seront installés :
blt python-imaging python-imaging-tk python-tk tk8.4
Paquets suggérés :
blt-demo python-imaging-doc tix scribus-template scribus-doc
ttf-bitstream-vera
Les NOUVEAUX paquets suivants seront installés :
blt python-imaging python-imaging-tk python-tk scribus tk8.4
0 mis à jour, 6 nouvellement installés, 0 à enlever et 3 non mis à jour.
Il est nécessaire de prendre 8467ko dans les archives.
Après dépaquetage, 24,2Mo d'espace disque supplémentaires seront utilisés.
Souhaitez-vous continuer [O/n] ?
```

Note: **apt-get install** va télécharger tous les packages nécessaires au bon fonctionnement du logiciel, et apeller **dpkg -i** sur chaque fichier DEB téléchargés.

Il existe plus de 300 miroirs DEBIAN

Rajouter des mirroirs

On peut rajouter des miroirs à apt-get en modifiant le fichier **/etc/apt/sources.list**. Ce fichier contient la liste des mirroirs officiels DEBIAN, et on peut rajouter d'autres mirroirs.

```
# cat /etc/apt/sources.list
deb cdrom:[Debian GNU/Linux 4.0 r1 _Etch_ - Official i386 NETINST Binary-1 2007$
deb http://ftp.fr.debian.org/debian/ etch main contrib non-free
deb-src http://ftp.fr.debian.org/debian/ etch main contrib non-free
deb http://security.debian.org/ etch/updates main contrib
deb-src http://security.debian.org/ etch/updates main contrib
```

On peut rajouter à ce fichier un mirroir contenant des fichiers DEB non-intégrés dans les mirroirs officiels Debian :

```
deb http://www.virtualbox.org/debian etch non-free
```

Mettre à jour la liste des logiciels disponibles

Chaque fois que l'on modifie le fichier **/etc/apt/sources.list**, il faut lancer la commmande **apt-get update** pour récupérer la liste des nouveaux logiciels :

```
# apt-get update
# apt-get install virtualbox
```

Note: apt-get utilise la commande wget pour télécharger les fichiers.

Mettre à jour tous les logiciels installés

L'option **upgrade** de **apt-get** prend la liste des paquets installés, regarde sur le dépots si ces logiciels sont disponibles dans une version plus récentes. Si c'est le cas, cette commande va mettre à jour ces logiciels.

Effacer les fichiers DEB installés

Par défaut, la commande **apt-get** conserve tous les fichiers DEB installés dans le répertoire **/var/cache/apt/archives**.

L'option **clean** permet d'effacer les fichiers DEB que l'on a installé :

```
# apt-get clean
# ls -l /var/cache/apt/archives
total 0
```

Autres options de apt-get

La commande **apt-get** dispose d'une page de manuel détaillant toutes les options disponibles :

```
# man apt-get
```

apt-cache

Chercher un package

L'option **search** de la commande **apt-cache** permet de rechercher un terme sur les paquets installés et non-installés.

```
# apt-cache search vob
gaupol - subtitle editor for text-based subtitle files
gopchop - Fast, lossless cuts-only editor for MPEG2 video files
gstreamer0.8-mpeg2dec - MPEG1 and MPEG2 video decoder plugin for GStreamer
libogg-vorbis-header-pureperl-perl - A pure Perl interface to Ogg Vorbis information fields
livemedia-utils - multimedia RTSP streaming tools
mplayer - The Movie Player
python-vobject - parse iCalendar and VCards in python
vobcopy - A tool to copy DvD VOBs to hard disk
```

Voir les informations d'un package

L'option **show** de la commande **apt-cache** permet de voir les informations d'un paquet (installé ou non).

```
# apt-cache show k3b
...
```

Autres options de apt-cache

La commande **apt-cache** dispose d'une page de manuel détaillant toutes les options disponibles :

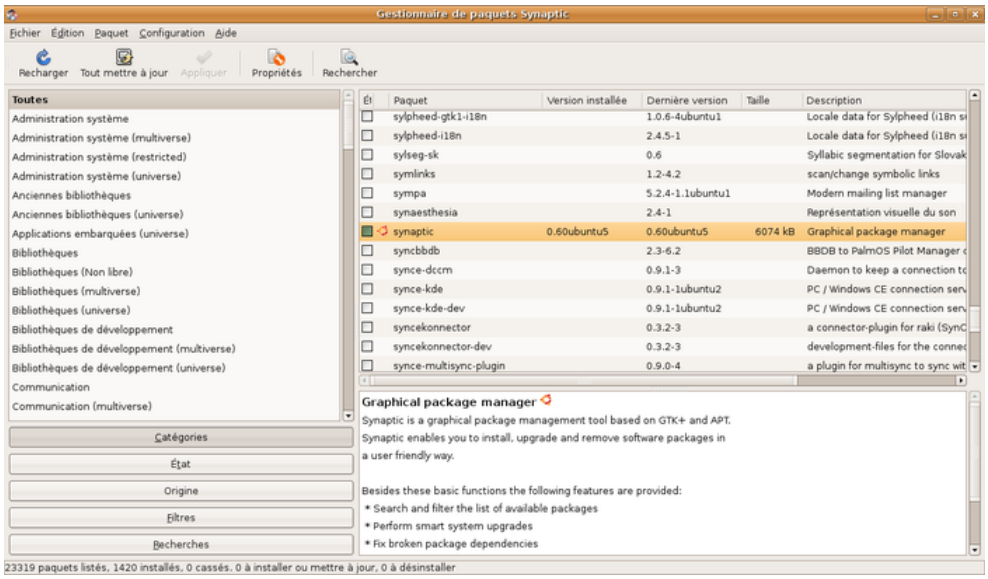
```
# man apt-cache
```

aptitude

La commande **aptitude** est un remplacant d'apt-get, son principal avantage est de désinstaller les dépendances.

synaptic

Le programme **synaptic** permet d'effectuer les mêmes opérations que la commande **apt-get** depuis l'interface graphique.



Les documentations des packages

Chaque package Debian est accompagné d'une documentation. Ces documentations se trouvent dans le répertoire **/usr/share/doc** :

/usr/share/doc/<*nom du package*>/

Certaines documentations sont compressées en gzip afin d'économiser de la place disque. On peut utiliser les commandes **zcat**, **zmore** ou **zless** pour visualiser ces fichiers sans avoir à les décompresser :

```
# zmore /usr/share/doc/python/python-policy.txt.gz
```

Installer un logiciel à partir des sources

Dans certains cas, on peut être amené à devoir installer un logiciel à partir des fichiers sources :

- le logiciel n'est pas disponible sous Debian en .DEB
- on a besoin d'une version plus récente du logiciel
- on a besoin d'une option uniquement activable durant la compilation
- ...

Pour faire ceci, on télécharge les sources du logiciel et on va les compiler.

Debian fournit un répertoire particulier destiné à accueillir les programmes installés à la main : **/usr/local**.

Si on regarde le contenu de ce répertoire, on constate qu'il contient des sous-répertoires déjà présents à la racine :

```
# ls /usr/local/
bin  etc  games  include  lib  man  sbin  share  src
```

Ces répertoires vont accueillir tous les fichiers des programmes installés à la main :

- les exécutables : **/usr/local/bin** ou **/usr/local/sbin**
- les fichiers de configuration : **/usr/local/etc**
- les librairies : **/usr/local/lib**
- les pages de man : **/usr/local/man**
- les fichiers communs (icones, traductions ...) : **/usr/local/share**
- ...

Pour illustrer cette méthodologie, nous allons installer pidgin (http://www.pidgin.im/).

Tout d'abord, on télécharge les fichiers sources du logiciel dans le répertoire **/usr/local/src** :

```
# cd /usr/local/src
# wget http://downloads.sourceforge.net/pidgin/pidgin-2.3.0.tar.bz2
# tar xjf pidgin-2.3.0.tar.bz2
# cd pidgin-2.3.0
```

La première chose à faire est de lire le fichier **README** :

```
# more README
```

On apprend dans ce fichier qu'il faut lire le fichier **INSTALL** pour connaître les dépendances et la procédure pour compiler le logiciel :

```
# more INSTALL
```

Comme dans la majorité des logiciels écrits en langage C, il faut effectuer la procédure standard : **./configure; make; make install** .

On lance donc la première commande :

```
# ./configure
```

Cette étape se solde par une erreur : **pidgin** a besoin de la librairie perl XML parser. On cherche donc le packet Debian contenant cette librairie :

```
# apt-cache search perl xml parser
...
libxml-parser-perl - Perl module for parsing XML files
...
```

On installe donc le packet libxml-parser-perl :

```
# apt-get install libxml-parser-perl
```

On relance le **./configure** :

```
# ./configure
```

Cette étape se solde à nouveau par une erreur : **pidgin** a besoin de la librairie de développement **glib2**. On cherche donc le packet Debian contenant cette librairie et on l'installe :

```
# apt-cache search glib 2 dev
...
libglib2.0-dev - Development files for the GLib library
...
# apt-get install libglib2.0-dev
```

On relance le **./configure** :

```
# ./configure
```

Cette étape se solde à nouveau par une erreur : **pidgin** a besoin de la librairie de développement **xml2**. On cherche donc le packet Debian contenant cette librairie et on l'installe :

```
# apt-cache search lib xml2 dev
...
libxml2-dev - Development files for the GNOME XML library
...
# apt-get install libxml2-dev
```

On relance le **./configure** :

```
# ./configure
...
Pidgin will be installed in /usr/local/bin.
```

```
configure complete, now type 'make'
```

Cette fois-ci, l'étape s'est terminée sans erreur et on peut lancer la compilation :

```
# make
...
# make install
```

```
...
```

Ca y est (enfin!), le logiciel s'est correctement installé, et l'exécutable a été copié dans le répertoire **/usr/local/bin** :

```
# ls -l /usr/local/bin
total 2280
-rwxr-xr-x 1 root staff 2329930 2007-12-07 10:06 pidgin
```

On peut désormais lancer le logiciel et l'utiliser :

```
$ pidgin
```

Le noyau Linux et les modules

Dans certains cas, on peut être amené à recompiler un noyau Linux :

- support d'un périphérique (driver)
- activation d'une option
- ...

Les sources du noyau Linux sont disponibles sur kernel.org (http://www.kernel.org) dans le répertoire /pub/linux/kernel/v2.6/ (http://www.kernel.org/pub/linux/kernel/v2.6/).

Les sources doivent être installées dans le répertoire **/usr/src** et il faut un lien **linux** vers la version du noyau que l'on désire compiler :

```
# cd /usr/src
# wget http://www.kernel.org/pub/linux/kernel/v2.6/linux-2.6.23.9.tar.bz2
...
# tar xjf linux-2.6.23.9.tar.bz2
# ln -s linux-2.6.23.9 linux
# cd linux
```

Si on désire utiliser les mêmes options de compilation du noyau actuel, on copie le fichier **/boot/config-2.6.18-5-686** dans le répertoire **/usr/src/linux** sous le nom **.config** :

```
# cp /boot/config-2.6.18-5-686 .config
```

Pour lancer la configuration (en mode texte) du noyau, on tape ensuite **make menuconfig**. Il faut avoir précédement installé la librairie de développement ncurses et tous les outils de compilation :

```
# apt-get install libncurses5-dev build-essential
```

On installe ensuite **kernel-package** qui contient les outils Debian permettant de fabriquer un packet **.deb** contenant le noyau Linux, les modules, le initrd et un script configurant Grub pour booter sur le nouveau noyau (cf explications ci-dessous) :

```
# apt-get install kernel-package
```

On lance ensuite la configuration du noyau via **make menuconfig** :

```
# make menuconfig
```

On configure le noyau, on quitte en enregistrant la configuration.

Debian fournit une méthode particulière pour lancer la compilation du noyau. il faut auparavant installer le packet **kernel-package**.

La procédure Debian consiste à fabriquer un fichier **.DEB** contenant le noyau et les modules compilés. L'avantage de cette méthode est de faciliter la mise à jour du noyau. De même, elle va automatiquement mettre à jour GRUB ou LILO pour démarrer sur le nouveau noyau.

Si on veut que Debian fabrique le fichier initrd automatiquement, il suffit de l'indiquer avec l'option **--initrd** de **make-kpkg**.

On lance la compilation du noyau et des modules avec la commande **make-kpkg** :

```
# make-kpkg clean

# make-kpkg --initrd --rev custom.1 kernel_image
...
# dpkg -i ../linux-image-2.6.23.9_custom.1_i386.deb
```

On peut également fabriquer un initrd manuellement, voici la procédure :

```
# mkinitramfs -o /boot/initrd.img-2.6.23.9 2.6.23.9
```

On le rajoute dans /boot/grub/menu.lst :

```
title           Debian GNU/Linux, kernel 2.6.23.9
root            (hd1,0)
kernel          /vmlinuz-2.6.23.9 root=/dev/hdd2 ro
```

```
initrd          /initrd.img-2.6.23.9
savedefault
```

Il ne reste plus qu'à rebooter pour démarrer sur le nouveau noyau.

Si tout c'est bien passé, le système va démarrer sur le nouveau noyau. Sinon, le noyau va s'arrêter sur un **kernel panic** et il faudra rebooter sur le noyau précédent, reprendre la configuration du noyau, le recompiler (avec un numéro de version différent, comme par exemple **custom.2).**

On vérifie que l'on a bien démarré sur notre nouveau noyau avec la commande **uname** :

```
# uname -r
2.6.23.9
```

Pour plus d'informations sur la compilation d'un noyau selon Debian, on peut consulter le Debian Linux Kernel Handbook (http://kernel-handbook.alioth.debian.org/index.html).

Modules

lsmod

Cette commande permet de visualiser les modules chargés en mémoire.

Syntaxe:

```
lsmod
```

modinfo

Cette commande permet de visualiser les informations du module comme le nom du créateur, et les options de chargements.

Syntaxe:

```
modinfo chemin_du_module
```

insmod et modprobe

Bien que ces deux commandes permettent de charger un module, la différence tient du fait que certains modules ont besoins que d'autres modules soient chargés.

insmod tente de charger le module demandé et si celui-ci a besoin d'un autre module pour se charger et que ce dernier ne l'est pas, **insmod** renvoie un message d'erreur et ne charge pas le module.

Syntaxe:

```
insmod chemin_du_module
```

modprobe, lui, charge les modules dépendant au modules avant de lancer celui-ci.

Syntaxe:

```
modprobe chemin_du_module
```

rmmod

Cette commande décharge le module dont le nom est indiqué. Contrairement aux autres commandes, il ne faut pas lui indiquer le chemin du module, mais son nom tel qu'il apparaît listé par la commande **lsmod**.

Syntaxe:

```
rmmod nom_du_module
```

rmmod décharge le module *nom_du_module*

depmod

Autres commandes utiles

Manipulation des flux et des fichiers textes

awk

La commande **awk** permet d'effectuer des manipulations sur des fichiers texte (ou un flux redirigé en entrée), dont notamment afficher une colonne particulière. Exemple :

```
$ dpkg -l | awk '{print $2}'
```

Cette commande dit à dpkg que l'on ne veut que la deuxième colonne.

sed

sed : utilitaire de traitement de données très puissant, capable d'utiliser les expressions régulières.

Pour substituer toute « chaîne1 » dans le fichier « chemin1 » avec « chaine2 » et envoyer le résultat dans le fichier « chemin2 » :

```
sed 's/chaine1/chaine2/g' chemin1 > chemin2
```

Exemple de changement de format de date :

```
echo "03/11/2015 23:54:03" | sed -r "s/([0-9]+)\/([0-9]+)\/([0-9]+)/\3-\2-\1/g"
```

donne :

```
2015-11-03 23:54:03
```

Trouver les commandes et les programmes

which

La commande **which** permet de connaitre dans quel répertoire se situe une commande (présente dans le PATH) :

```
 $ which ls
 /bin/ls
```

updatedb et locate

La commande locate (ou slocate) permet une recherche rapide sur le système de fichier (suite à une indexation réalisée par updatedb).

```
$ locate maillog
```

```
warning: locate: warning: database /var/lib/slocate/slocate.db' is more than 8 days old
/etc/log.d/conf/logfiles/maillog.conf
/var/log/maillog
```

Lorsque l'indexation a eu lieu il y a plus de 8 jours, la commande locate l'indique par un message.

Pour réindexer le système de fichier :

```
$ sudo updatedb
ou
# updatedb
```

Note sur les droits : updatedb utilise par défaut une base d'indexation globale. Il faut donc avoir les droits super-utilisateur pour mettre à jour l'indexation globale.

Outils réseaux

wget

La commande **wget** permet de télécharger un fichier depuis la ligne de commande :

```
$ wget http://www.kernel.org/pub/linux/kernel/v2.6/linux-2.6.23.tar.bz2
--13:16:02--  http://www.kernel.org/pub/linux/kernel/v2.6/linux-2.6.23.tar.bz2
          => `linux-2.6.23.tar.bz2'
Résolution de www.kernel.org... 204.152.191.5, 204.152.191.37
Connexion vers www.kernel.org[204.152.191.5]:80...connecté.
requête HTTP transmise, en attente de la réponse...200 OK
Longueur: 45,488,158 [application/x-bzip2]

9% [===>                                     ] 4,245,225    630.29K/s    ETA 01:28
```

Pour utiliser un proxy, il faut définir la variable d'environnement http_proxy. Exemple :

```
http_proxy=monproxy:3128 wget http://apache.cict.fr/ant/source/apache-ant-1.7.0-src.tar.gz
```

wget sur ftp

wget est également compatible avec le protocole ftp. Il est ainsi capable de télécharger un fichier ou un ensemble de fichier depuis un ftp distant.

```
# wget ftp://login:passwd@ftp.host.net/thedir/*
```

Pour télécharger les sous-répertoire et fichiers on utilise l'option "**-r**".

L'option "**-nH**" permet de ne pas sauvegarder le nom d'hôte, autrement wget fera la copie vers un nouveau dossier créé portant le nom de l'hôte du serveur ftp.

Utilisation de base

Typiquement Wget s'utilise par la ligne de commande, avec un ou plusieurs URLs passés en paramètres. De nombreuses options permettent d'affiner le comportement souhaité (téléchargement multiple, suivre les liens...).

```
# Télécharge la page à la racine du site exemple.fr dans un ficher
# nommé "index.html".
wget http://www.exemple.fr/
```

```
# Télécharger les sources de Wget depuis le serveur FTP de GNU.
wget ftp://ftp.gnu.org/pub/gnu/wget/wget-latest.tar.gz
```

Il est possible de télécharger automatiquement plusieurs URLs dans une hiérarchie de dossiers.

```
# Télécharge tous les fichiers *.gif d'un serveur web
# (Les syntaxes de type glob, comme "wget http://www.serveur.com/dir/*.gif", ne marche que avec FTP)
wget -e robots=off -r -l1 --no-parent -A.gif http://www.serveur.com/dir/
```

```
# Télécharge la page racine de exemple.fr, avec les images et les
# feuilles de styles utilisées pour afficher la page, et convertit les
# URLs internes pour fonctionner avec les copies locales.
wget -p -k http://www.exemple.fr/
```

```
# Télécharge le contenu entier de exemple.fr
wget -r -l 0 http://www.exemple.fr/
```

Divers

file

La commande "**file**" permet de connaître le type du fichier indiqué en paramètre.

```
# file TmDedicatedServer_2006-05-30.tgz
TmDedicatedServer_2006-05-30.tgz: gzip compressed data, was "fr.22968.0.TmDedicatedServer_20",
from Unix, last modified: Tue May 30 13:41:35 2006
```

du

La commande "**du**" sert à afficher la taille des grandes espaces (partitions de disque et répertoires)

```
# du -sh /var/temp
```

L'option "**s**" affiche seulement un total pour chaque type d'argument.

L'option "**h**" permet l'affichage automatique de l'unité adaptée (Ko, Mo, Go...).

df

La commande **df** affiche l'espace occupé par les systèmes de fichiers.

```
# df -h
```

L'option "**h**" permet l'affichage automatique de l'unité adaptée (Ko, Mo, Go...).

Installation RAID1 logiciel + LVM + XFS

Voici la configuration à obtenir lors de l'installation de Debian :

Scripts de surveillance

Ce chapitre contient différents scripts de surveillance de l'activité d'un serveur Linux

En langage Python

alimon.py (A LInux MONitor)

Lien direct : alimon.py (http://www.euronode.org/alimon.py).

```
#!/usr/bin/python
# -*- coding: utf-8 -*-
#
###############################################################################
#
# ALiMon.py : A LInux MONitor
#
###############################################################################
#
# Ce script réalise différentes opérations de monitoring et met en évidence
# certains points importants comme une partition disque bientôt pleine ou une
# charge processeur trop élevée.
#
# Ce script a été réalisé durant une scéance de travaux pratiques et a des fins
# didactiques. Il est issu du travail collectif des personnes citées ci-dessous
# en auteurs et a nécessité uniquement deux heures de développement.
#
###############################################################################
#
# Auteurs :
#
# David BISPO, Christophe CARLIER, Jonathan DUHAIL, Jonathan GAULUPEAU,
# Lahoucine HAMOUCHE, Hicham OUHNA, Manuel PIRES, Yann VAITILINGOM,
# Jérémy PELLAUMAIL et Alexandre GUY
#
# Nous remercions également les connectés du canal #afpy du réseau Freenode
# pour leur aide concernant l'unicode et l'encodage utf-8.
#
# Version : 0.3
#
###############################################################################
#
# Ce script est diffusé sous la licence EUPL v1.1
#
# This script is released under EUPL v1.1
#
# http://ec.europa.eu/idabc/eupl
#
###############################################################################
#
# Ce programme est un logiciel libre ; vous pouvez le re-distribuer et/ou le
# modifier au titre des clauses de la European Union Public Licence (EUPL)
# version 1.1, telle que publiée par l'Union Européenne.
#
# Ce programme est distribué dans l'espoir qu'il sera utile,
# mais SANS AUCUNE GARANTIE ; sans même une garantie implicite de
# COMMERCIABILITÉ ou DE CONFORMITÉ À UNE UTILISATION PARTICULIÈRE.
# Voir la EUPL version 1.1 pour plus de détails.
#
# This program is free software; you can redistribute it and/or modify it
# under the terms of the European Union Public Licence (EUPL) version 1.1
# as published by the European Union.
#
# This program is distributed in the hope that it will be useful, but
# WITHOUT ANY WARRANTY; without even the implied warranty of
# MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.
# See the EUPL version 1.1 for more details.
#
###############################################################################
#
# Utilisation manuelle : ./alimon.py
#
# Utilisation automatisée : Rajouter dans le cron :
# 55 23 * * * root alimon.py
#
###############################################################################
```

```
########################################
#### Configuration & pré-traitements ####
########################################


#### Modules importés ####
import commands, unicodedata, os, sys, getopt


#### Fonction Affichage du titre formaté ####
def titre(message):
  message = unicode(message,'utf-8')
  print "\n", "#" * 59
  print "######", message.center(45).encode('utf-8'), "######"
  print "#" * 59, "\n"


#### Fonction affichant l'aide ####
def usage():
  print "###### Aide de alimon ######"
  print "./alimon.py [-f] [-d] [-h]"
  print "   -f  Lance le programme en mode dégradé. Utile si certaines commandes"
  print "       ne sont pas installées sur le serveur."
  print "   -d  Lance le programme en mode debug."
  print "   -h  Affiche l'aide du programme."


#### Variables globables ####
# Active / désactive le mode debug
debug = 0
# Active / désactive le mode dégradé
force = 0
# Seuil maxi du loadavg
seuil_maxi_loadavg = 1.0
# Température maxi des disques durs
temperature_hdd_maxi = 45
# Taux d'occupation maxi des disques durs
taux_occupation_maxi = 80
# Logins autorisés
loginsok = ['alex', 'jo']
# Liste des daemons qui doivent être en cours d'exécution
proclist = ['sshd', 'apache2', 'mysqld', 'named', 'master', 'murmurd', 'pop3-login', 'teamspeak-serve', 'couriertcpd']
# Pourcentage maxi de swap
pourcentage_mem_maxi = 10
# Mémoire mini disponible en Mo
mem_mini = 200
# Liste des sites à 'pinguer'
url_ping = ['www.google.fr']
# Liste des périphériques RAID (exemple : ['md0', 'md1'])
raidlist = []


#### Récupère les arguments ####
# Récupère la liste des arguments
try:
    optlist, list = getopt.getopt(sys.argv[1:], 'dhf')
# Si un argument ne figure pas dans la liste prédéfinie, affiche la fonction 'usage' (aide) puis quitte
except getopt.GetoptError:
    usage()
    sys.exit(1)
# Traite les arguments donnés
for opt in optlist:
  if opt[0] == '-h':
    usage()
    sys.exit(0)
  if opt[0] == '-d':
    debug = 1
  if opt[0] == '-f':
    force = 1


#### Test de présence des commandes shell et récupération de leur chemin ####
commandes_utilisees = ['cat','hostname','last','hddtemp','df','ps','free','ping','grep','uniq','who','uname']
# Contrôle la commande 'mdadm' uniquement si une liste de disques raid est définie
if raidlist:
  commandes_utilisees.append('mdadm')
commandes = {}
for comm in commandes_utilisees:
  (res,commande) = commands.getstatusoutput("/usr/bin/which %s" % comm)
  # Si une commande n'existe pas mais que le mode Force est activé
  if res and force:
```

```
    print "La commande", comm, "n'est pas présente sur votre système, fonctionnement en mode dégradé."
    commandes[comm] = ""
  # Si une commande n'exite pas
  elif res:
    print "La commande", comm, "n'est pas présente sur votre système. Arrêt du programme."
    print "Tapez './alimon.py -h' pour en savoir plus."
    sys.exit(1)
  else:
    commandes[comm] = commande


#### Récupération de la liste des disques durs ####
result = commands.getoutput("%s /proc/partitions" % commandes['cat']).split("\n")
result.pop(0)
hddlist = []
for ligne in result:
  if debug: print "debug> Ligne du fichier '/proc/partitions' en cours de traitement :", ligne
  hdd = ''.join(ligne.split()[-1:])[:3]
  if debug: print "debug> Disque dur à ajouter à la liste :", hdd
  if hdd and hdd not in hddlist:
    hddlist.append(hdd)



############################
#### Programme principal ####
############################


#### INFOS SERVEUR ####
if commandes['hostname'] and commandes['uname'] and commandes['cat'] and commandes['grep'] and commandes['uniq']:
  titre("Informations sur le serveur")
  # On récupère la première ligne du fichier /etc/issue qui contient généralement le nom et la version de la distribution
Linux
  version = open("/etc/issue", "r").readlines()[0][:-1]
  if debug: print "debug> Version de la distribution :", version
  # On récupère le nom du serveur
  hostname = commands.getoutput(commandes['hostname'])
  if debug: print "debug> Hostname :", hostname
  print "Serveur %s sous %s" % (hostname, version)
  print
  # On récupère et affiche la version du noyau par la commande 'uname -r'
  print "Noyau Linux   :", commands.getoutput("%s -r" % commandes['uname'])
  # On récupère et affiche le(s) processeur(s) installé(s) sur le serveur
  cpu = commands.getoutput("%s /proc/cpuinfo | %s \"model name\" | %s" % (commandes['cat'], commandes['grep'],
commandes['uniq']))
  if debug: print "debug> Liste des processeurs non post-traitée :", cpu
  print "Processeur(s) :", cpu.split(': ')[1]
  # On récupère et affiche la quantité de mémoire installée sur le serveur
  mem = commands.getoutput("%s /proc/meminfo | %s \"MemTotal\"" % (commandes['cat'], commandes['grep']))
  if debug: print "debug> Quantité de mémoire non post-traitée :", mem
  print "Mémoire vive  :", ' '.join(mem.split()[1:])


#### DUREE DE FONCTIONNEMENT DU SERVEUR ####
if commandes['cat']:
  titre("Durée de fonctionnement")
  # On récupère la durée de fonctionnement en secondes dans /proc/uptime
  result = commands.getoutput("%s /proc/uptime" % (commandes['cat'])).split()[0]
  duree = float(result)
  if debug: print "debug> Résultat de cat '/proc/uptime' :", duree
  # On calcule le reste en secondes
  secondes = duree%60
  if debug: print "debug> Secondes :", secondes
  # On calcule le reste en minutes
  minutes = duree/60%60
  if debug: print "debug> Minutes :", minutes
  # On calcule le reste en heures
  heures = duree/60/60%24
  if debug: print "debug> Heures :", heures
  # On calcule le nombre de jours
  jours = int(duree/60/60/24)
  if debug: print "debug> Jours :", jours
  print "Serveur lancé depuis %i jour(s), %i heure(s), %i minute(s) et %i seconde(s)." % (jours, heures, minutes, secondes)


#### VERIFICATION DU LOAD AVERAGE ####
if commandes['cat']:
  titre("Vérification du load average")
  # On appelle une commande Unix et on récupère le résultat dans la variable result
  result = commands.getoutput("%s /proc/loadavg" % (commandes['cat']))
  if debug: print "debug> Résultat de 'cat /proc/loadavg' :", result
  # On découpe la chaine de caractères en une liste selon le caractère espace
```

```
  liste_result = result.split()
  if debug: print "debug> La liste découpée :", liste_result
  if debug: print "debug> La première valeur de la liste :", liste_result[0]
  # On compare le load avg de la dernière minute avec le seuil maxi
  # Attention, on converti la chaine de caractère liste_result[0] en nombre flottant via float()
  if float(liste_result[0]) > seuil_maxi_loadavg:
    print "Alerte ! charge CPU supérieure à", seuil_maxi_loadavg, "!!!"
  else:
    print "Charge CPU %s normale car inférieure à %.2f" % (liste_result[0],seuil_maxi_loadavg)


#### CONNECTIONS DU JOUR ####
if commandes['last'] and commandes['grep']:
  titre("Connections du jour")
  # On récupère les dernières connections avec la commande 'last', on filtre avec 'grep'
  login = commands.getoutput("%s | %s \"$(LANG=C date +\"%%a %%b %%-d\")\"" % (commandes['last'], commandes['grep']))
  if login == "":
    login = commands.getoutput("%s | %s \"$(LANG=C date +\"%%a %%b  %%-d\")\"" % (commandes['last'], commandes['grep']))
  print login


#### TEMPERATURE DES HDD ####
if commandes['hddtemp']:
  titre("Températures des disques durs")
  # On vérifie la température pour chaque disque dans la liste 'hddlist' avec la commande 'hddtemp'
  for disque in hddlist:
    if debug: print "debug> Disque en cours de contrôle :", disque
    # On récupère la température du disque
    temperature = commands.getoutput("%s -n /dev/%s" % (commandes['hddtemp'], disque))
    if debug: print "debug> Température du disque en cours :", temperature
    # On vérifie que la température du disque n'est pas supérieure au seuil de tolérance, sinon on imprime un message
d'alerte
    # S'il y a une erreur (disque non compatible SMART), on passe au disque suivant
    try:
      if int(temperature) > temperature_hdd_maxi:
        print "Alerte ! Le disque dur /dev/%s a dépassé %s°C, il est actuellement à %s°C !!!" % \
          (disque, temperature_hdd_maxi, temperature)
      else:
        print "Le disque dur /dev/%s est à %s°C et inférieur au seuil de %s°C" % \
          (disque, temperature, temperature_hdd_maxi)
    except:
      pass


#### VERIFICATION DE L'ESPACE DISQUE ####
if commandes['df'] and commandes['grep']:
  titre("Vérification de l'espace disque")
  # On récupère le pourcentage d'espace libre sur toutes les partitions dont le périphérique commence par '/dev' avec la
commande 'df'
  result = commands.getoutput("%s -P | %s -e '^/dev'" %(commandes['df'], commandes['grep']))
  if debug: print "debug> Résultat du 'df' :", result
  # On fait un contrôle sur chaque ligne du résultat du 'df'
  for disque in result.split('\n'):
    taux_occupation = disque.split()[4][:-1]
    if debug: print "debug> Disque en cours de contrôle :", disque
    if debug: print "debug> Taux d'occupation du disque en cours :", taux_occupation
    # On vérifie que le taux d'occupation n'est pas supérieur au seuil de tolérance, sinon on imprime un message d'alerte
    if int(taux_occupation) > taux_occupation_maxi:
      print "Attention la partition %s est rempli à %s !!!" %(disque.split()[0], disque.split()[4])
    else:
      print "La partition %s est pleine à %s." %(disque.split()[0], disque.split()[4])


#### VERIFICATION ETAT RAID ####
# Se lance si une liste de périphériques a été définie
if raidlist:
  if commandes['mdadm']:
    titre("Vérification de l'état du RAID")
    # On teste chaque périphérique de la liste
    for hddraid in raidlist:
      # On récupère le status de la commande qui vérifie l'état d'un RAID
      (raidstatus, raid) = commands.getstatusoutput("%s --detail -t /dev/%s" %(commandes['mdadm'], hddraid))
      # On teste si le status est différent de 0 (donc status en erreur)
      if raidstatus!=0:
        print "Attention, le périphérique RAID /dev/%s a au moins un disque en dysfonctionnement !" % hddraid
      else:
        print "Le périphérique RAID /dev/%s fonctionne normalement." % hddraid


#### VERIFICATION CONNECTIONS EN COURS ####
if commandes['who']:
  titre("Vérification des logins actuellement connectés")
  # On récupère les noms des utilisateurs actuellement connectés avec la commande 'who'
```

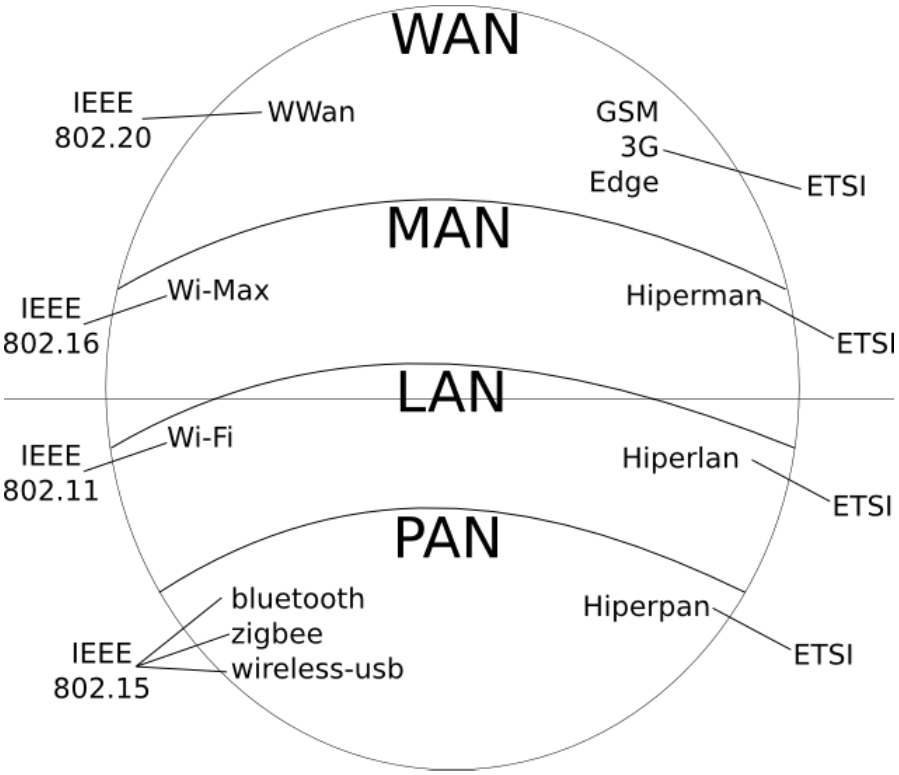
```
  wholiste = commands.getoutput(commandes['who']).split('\n')
  if debug: print "debug> Liste des utilisateurs connectés :", wholiste
  liste_logins_connectes = []
  # On récupère uniquement la première colonne de chaque ligne
  for ligne in wholiste:
    if ligne:
      if debug: print "debug> Ligne en cours de traitement :", ligne
      user = ligne.split()[0]
      if debug: print "debug> Utilisateur en cours de traitement :", user
      # On ajoute l'utilisateur à une liste s'il n'y est pas déjà (évite les doublons)
      if user not in liste_logins_connectes:
        liste_logins_connectes.append(user)
  # On vérifie que les utilisateurs précédemment récupérés sont bien dans la liste des utilisateurs autorisés, sinon on
imprime un message d'alerte
  for user in liste_logins_connectes:
    if user in loginsok:
      print "utilisateur", user, "OK"
    else:
      print "ATTENTION L'utilisateur", user, "est connecte MAIS n'est pas dans la liste"


#### VERIFICATION DES PROCESSUS ####
if commandes['ps']:
  titre("Vérification des processus")
  # On récupère les processus lancés avec la commande 'ps'
  result = commands.getoutput("%s -e" % (commandes['ps'])).split('\n')
  # On supprime la première ligne
  result.pop(0)
  if debug: print "debug> Résultat de la commande 'ps' :", result
  # On récupère uniquement le nom de chaque processus (dernière colonne)
  psliste=[]
  for processus in result:
    if debug: print "debug> Processus à ajouter :", processus
    psliste.append(processus.split()[-1])
  if debug: print "debug> Contenu de la liste des processus :", psliste
  # On vérifie que chaque processus de la liste définie au début du script est présent dans la liste récupérée précedemment
  for processus in proclist:
    if debug: print "debug> Processus en cours de vérification :", processus
    if not processus in psliste:
      print "Attention : le process %s n'est pas lancé actuellement !!!" %(processus)
    else:
      print "Le processus", processus, "est bien en cours d'exécution."


#### VERIFICATION DE LA MEMOIRE ####
if commandes['free']:
  titre("Vérification de la mémoire")
  # On récupère la quantité de mémoire libre avec la commande 'free'
  result = commands.getoutput(commandes['free'])
  if debug: print "debug> Résultat de la commande 'free' :", result
  listefree = result.split("\n")
  if debug:
    print "debug> On truque le résultat de la commande free pour tester"
    listefree = ['             total       used       free     shared    buffers     cached', 'Mem:       2066032     850780
1215252          0     114496     368584', '-/+ buffers/cache:     367700    1698332', 'Swap:      2048276          350500
2048276']
    print "debug> Free truqué :", listefree
  # On récupère la quantité de mémoire totale
  ligne1=listefree[1].split()
  memoire=float(ligne1[1])
  if debug: print "debug> Mémoire totale :", memoire
  # On récupère la quantité de swap totale
  ligne3=listefree[3].split()
  swap=float(ligne3[2])
  if debug: print "debug> Swap total :", swap
  # On calcule le rapport de swap par rapport à la mémoire installée
  pourcentage = (swap/memoire)*100.0
  if debug: print "debug> Pourcentage de Swap :", pourcentage
  # On vérifie que le rapport ne dépasse pas le seuil limite autorisé, sinon on imprime un message d'alerte
  if pourcentage > pourcentage_mem_maxi:
    print "Alerte ! memoire swap supérieur a %d%% de la mémoire (utilisation %.2f%%) !!!" % (pourcentage_mem_maxi,
pourcentage)
  else:
    print "Mémoire swap inférieure à %d%% (utilisation %.2f%%)" % (pourcentage_mem_maxi, pourcentage)
  # On récupère la quantité de mémoire libre en Mo avec la commande 'free -m'
  result = commands.getoutput("%s -m" % (commandes['free']))
  result = result.split('\n')
  if debug: print "debug> Résultat de la commande 'free -m' :", result
  mem = result[2].split()[3]
  if debug: print "debug> Quantité de mémoire libre en Mo :", mem
  # On vérifie que la quantité de mémoire libre ne soit pas inférieure au seuil limite, sinon on imprime un message d'alerte
  if int(mem) < mem_mini:
    print "Attention !! il reste moins de %sMo de mémoire libre !!! (%sMo)" %(mem_mini, mem)
```

```
  else:
    print "Il y a", mem, "Mo de mémoire libre."


#### TEMPS DE REPONSE SERVEUR WEB ####
if commandes['ping']:
  titre("Temps de réponse du serveur")
  # On lance la commande 'ping' sur chaque url définie dans 'url_ping'
  for url in url_ping:
    print "Résultat de la commande ping sur %s" % (url)
    print commands.getoutput("%s -c 1 %s" % (commandes['ping'], url)).split("\n")[-1]

print
```

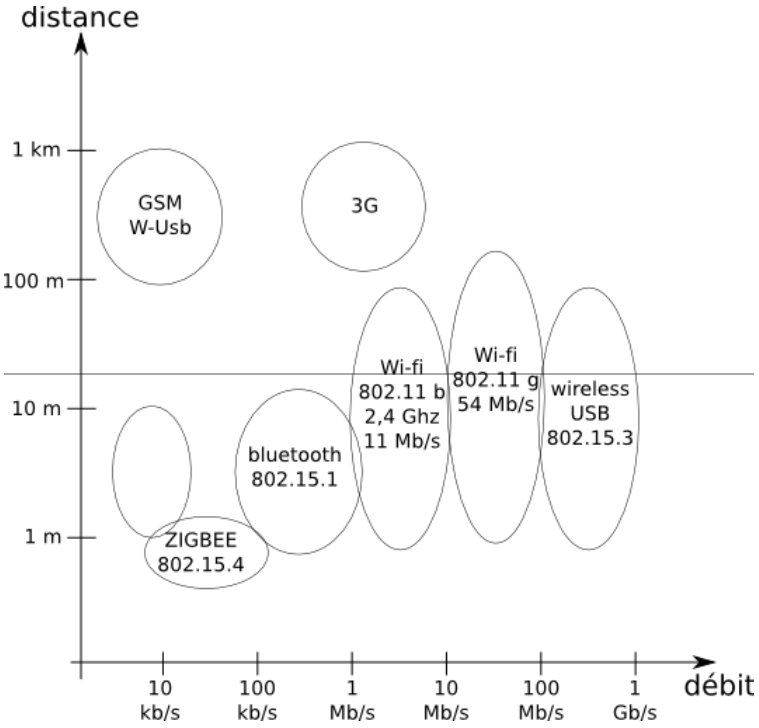
Réseaux sans fil

Théorie des réseaux sans fil

Réseaux sans fil de type PAN, LAN, MAN et WAN



Distances et débits



Les réseaux sans fil PAN

PAN : Personal Area Network

La norme ZBEE

- Norme 802.15.4
- Consomme très peu d'énergie
- Bas débit (maximum 250 Kbit/s)
- Idéal pour les applications de type domotique et automatisme
- Faible coûts de fabrication du composant (1$)

La norme définit trois débits différents :

- 250 Kbit/s en 2.4 GHz (international)
- 20 Kbit/s en 868 MHz (Europe)
- 40 Kbit/s en 915 MHz (USA)

La norme Wireless USB

Objectif : remplacer l'USB

Autres noms : W-USB (Wireless USB), UWB (Ultra-Wide Band), Wimedia

- Norme 802.15.3
- Très haut débit (jusqu'à 480 Mbit/s)
- Idéal pour les applications multimédia (audio, vidéo) et pour le transfert de données
- Possibilité de fonctionner en réseau ou en mode ad-hoc
- Authentification et chiffrement possible

Débits théoriques :

- 54 Mbit/s à 2.4 GHz
- jusqu'à 480 Mbit/s de 3.1 à 10.7 GHz

La norme Bluetooth

- Norme 802.15.1
- Inventé par Ericsson, aujourd'hui groupement de 2500 sociétés
- Technologie peu onéreuse (composant à partir de 3$) et fortement intégré (puce de 9 mm x 9 mm)
- Débits théoriques : de 1 à 3 Mbit/s selon la version de la norme

La norme Bluetooth utilise la fréquence 2.4 GHz et définit trois puissances d'émission :

- 1 mW : portée de quelques mètres
- 2.5 mW : portée de 10 à 20 mètres
- 100 mW : portée de 100 mètres

Un réseau Bluetooth, un **piconet**, est composé d'un maître et de 7 esclaves maximum. Le débit sera partagé entre les différents membres du piconet.

On peut interconnecter des piconets pour former un **scatternet** (scatter : dispersion)

Les communications peuvent être symétrique (vitesses d'émission et de réception identiques) ou asymétrique (vitesses d'émission et de réception différentes).

Les réseaux sans fil LAN

LAN : Local Area Network

La norme Wifi

Principales normes 802.11 :

- 802.11a : fréquence d'émission à 5 GHz, débit jusqu'à 54 Mbit/s

- 802.11b : fréquence d'émission à 2.4 GHz, débit jusqu'à 11 Mbit/s
- 802.11g : fréquence d'émission à 2.4 GHz, débit jusqu'à 54 Mbit/s
- 802.11n : fréquence d'émission à 2.4 GHz ou 5 GHz, débit jusqu'à 300 Mbit/s en utilisant les technologies MIMO (Multiple-Input Multiple-Output) et OFDM (Orthogonal Frequency Division Multiplexing)

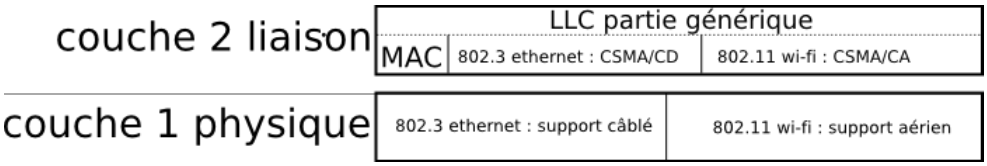
Très proche de l'ethernet câblé (802.3), utilise la sous-couche LLC générique de la couche 2, et ré-implémente la sous-couche MAC.

Contrairement à ethernet câble qui est basé sur la technique d'accès CSMA/CD, la norme Wi-Fi utilise le CSMA/CA (Carrier Sense Multiple Access / Collision Avoidance).

Il y a plusieurs modes de fonctionnement, dont voici les principaux :

- mode **Ad-Hoc** permet d'établir des connexions sans passer par un point d'accès
- mode **Infrastructure** permet d'établir des connexions en passant par un point d'accès, et également de gérer un système de distribution basé sur plusieurs points d'accès utilisant le même ESSID.
- mode **monitor** permet de passer la carte Wi-Fi en mode **promiscuous**, c'est à dire qu'elle voit passer toutes les trames, et pas seulement celles qui lui sont destinées.

Le Wi-Fi utilise des trames de synchronisation temporelle, appelée les **beacon frames**. Ces trames contiennent l'heure du point d'accès.



Sécurité WEP

- Premier protocole d'authentification et de chiffrement
- Signifie : Wired Equivalent Privacy
- Aujourd'hui totalement obsolète car non sécurisé
- Utilise des clés de 64 bits (clé secrète de 40 bits + vecteur d'initialisation -IV- de 24 bits) ou des clés de 128 bits (clé secrète de 104 bits + vecteur d'initialisation -IV- de 24 bits)
- Définit deux techniques d'authentification : **Open System Authentication** (diffusion du ESSID) et **Shared Key Authentication** (pas de diffusion d'informations avant l'authentification)

Sécurité WPA

- Inventé pour pallier les failles de sécurité du WEP
- Signifie : Wi-Fi Protected Access
- La norme WPA1 fonctionne sur du matériel existant (mais est cassable). La norme WPA2 nécessite du matériel plus puissant (authentification et chiffrement plus complexe) et est la méthode la plus sécurisée à l'heure actuelle
- Utilise le standard 802.1x qui se charge de l'authentification et de la génération de la clé
- Le modèle 802.11i assure la sécurité au niveau de la sous-couche MAC via le protocole TKIP (Temporal Key Integrity Protocol) ou via le protocole CCMP (**C**ounter with **C**ipher Block Chaining **M**essage Authentication Code **P**rotocol) (plus sécurisé)

Les réseaux sans fil MAN

MAN : Metropolitan Area Network

La norme Wimax

Les réseaux sans fil WAN

WAN : Wide Area Network

La norme 802.20

Mise en pratique

Les commandes Wi-Fi

Utilisateurs de Gnome, Attention ! les commandes ci-dessous ne fonctionneront que si vous avez désactivé au préalable Network Manager :

Ubuntu 9.10 et supérieur :

```
# service network-manager stop
```

Ubuntu 9.04 :

```
# service NetworkManager stop
```

iwconfig

Invoquée seule, la commande **iwconfig** affiche toutes les interfaces réseaux :

```
# iwconfig
lo        no wireless extensions.

eth0      no wireless extensions.

wmaster0  no wireless extensions.

wlan0     IEEE 802.11bg  ESSID:""
        Mode:Managed  Frequency:2.412 GHz  Access Point: Not-Associated
        Tx-Power=20 dBm
        Retry  long limit:7   RTS thr:off   Fragment thr:off
        Encryption key:off
        Power Management:off
        Link Quality:0  Signal level:0  Noise level:0
        Rx invalid nwid:0  Rx invalid crypt:0  Rx invalid frag:0
        Tx excessive retries:0  Invalid misc:0   Missed beacon:0
```

On peut indiquer à la commande iwconfig l'interface à afficher :

```
# iwconfig wlan0
wlan0     IEEE 802.11bg  ESSID:""
        Mode:Managed  Frequency:2.412 GHz  Access Point: Not-Associated
        Tx-Power=20 dBm
        Retry  long limit:7   RTS thr:off   Fragment thr:off
        Encryption key:off
        Power Management:off
        Link Quality:0  Signal level:0  Noise level:0
        Rx invalid nwid:0  Rx invalid crypt:0  Rx invalid frag:0
        Tx excessive retries:0  Invalid misc:0   Missed beacon:0
```

On peut également utiliser la commande **iwconfig** pour configurer les paramètres du réseau sans fil :

- Pour définir le nom du réseau :

```
# iwconfig wlan0 essid "MON RESEAU"
```

- Pour définir le canal à utiliser :

```
# iwconfig wlan0 channel 3
```

on peut également spécifier la fréquence avec l'option freq :

```
# iwconfig wlan0 freq 2.412G
```

- pour définir le débit

```
# iwconfig wlan0 rate auto
```

- pour fixer le débit :

```
# iwconfig wlan0 rate 11M
```

Le 802.11b définit les débits suivants : 1, 2, 5.5 et 11 Mbit/s.

Le 802.11g définit en plus les débits suivants : 6, 9, 12, 18, 24, 36, 48 et 54 Mbit/s.

- Pour activer ou non le chiffrement WEP :

```
# iwconfig wlan0 key off
```

- pour spécifier la clé WEP :

```
# iwconfig wlan0 key 1234567890
```

on peut gérer de trousseau de quatre clés. On indique le numéro de clé entre crochets :

```
# iwconfig wlan0 key [1] 1234567890
```

- pour changer le mode (ad-hoc, managed, monitor ...) de la carte :

Attention, il faut que la configuration coté IP soit arrêtée :

```
# ifconfig wlan0 down
# iwconfig wlan0 mode Ad-Hoc
# ifconfig wlan0 up
```

- pour changer la puissance d'émission (si la carte le permet) :

```
# iwconfig eth0 txpower 60mW
```

On peut également spécifier plusieurs paramètres sur la même ligne :

```
# iwconfig wlan0 channel 11 key off essid NETGEAR
```

iwlist

La commande **iwlist** permet d'obtenir des informations complémentaires. Invoquée seule, elle affiche la liste des options disponibles :

```
# iwlist
Usage: iwlist [interface] scanning [essid NNN] [last]
              [interface] frequency
              [interface] channel
              [interface] bitrate
              [interface] rate
              [interface] encryption
              [interface] keys
              [interface] power
              [interface] txpower
              [interface] retry
              [interface] ap
              [interface] accesspoints
              [interface] peers
              [interface] event
              [interface] auth
              [interface] wpakeys
              [interface] genie
              [interface] modulation
```

On peut optionnellement spécifier l'interface réseau (inutile si il n'y en a qu'une).

- Afficher la liste des réseaux détectés :

```
# iwlist scan
...
wlan0     Scan completed :
         Cell 01 - Address: 00:09:5B:6F:3C:38
                   Channel:11
```

```
                  Frequency:2.462 GHz (Channel 11)
                  Quality=57/70  Signal level=-53 dBm
                  Encryption key:off
                  ESSID:"NETGEAR"
                  Bit Rates:1 Mb/s; 2 Mb/s; 5.5 Mb/s; 11 Mb/s
                  Mode:Master
                  Extra:tsf=0000001241f8f0da
                  Extra: Last beacon: 192ms ago
                  IE: Unknown: 00074E455447454152
                  IE: Unknown: 010482840B16
                  IE: Unknown: 03010B
```

- afficher les canaux et fréquences disponibles, et également le canal/fréquence actuellement utilisé :

```
# iwlist freq        # OU iwlist channel
...
wlan0    13 channels in total; available frequencies :
         Channel 01 : 2.412 GHz
         Channel 02 : 2.417 GHz
         Channel 03 : 2.422 GHz
         Channel 04 : 2.427 GHz
         Channel 05 : 2.432 GHz
         Channel 06 : 2.437 GHz
         Channel 07 : 2.442 GHz
         Channel 08 : 2.447 GHz
         Channel 09 : 2.452 GHz
         Channel 10 : 2.457 GHz
         Channel 11 : 2.462 GHz
         Channel 12 : 2.467 GHz
         Channel 13 : 2.472 GHz
         Current Frequency:2.447 GHz (Channel 8)
```

- afficher la liste des clés WEP :

```
# iwlist encryption        # OU iwlist keys
...
wlan0    2 key sizes : 40, 104bits
         4 keys available :
                [1]: 1234-5678-90 (40 bits)
                [2]: off
                [3]: off
                [4]: off
         Current Transmit Key: [1]
```

- afficher les possiblités d'authentification du driver de la carte :

```
# iwlist auth
...
wlan0     Authentication capabilities :
                WPA
                WPA2
                CIPHER-TKIP
                CIPHER-CCMP
          Current Authentication algorithm :
                open
                shared-key
```

- afficher la liste des points d'accés détectés :

```
# iwlist ap        # OU iwlist accesspoints
ath0      Peers/Access-Points in range:
    5A:1D:14:86:97:BC : Quality=34/70  Signal level=-61 dBm  Noise level=-95 dBm
    5A:1D:14:86:97:BE : Quality=34/70  Signal level=-61 dBm  Noise level=-95 dBm
    5A:1D:14:86:97:BF : Quality=33/70  Signal level=-62 dBm  Noise level=-95 dBm
...
```

iwevent

La commande **iwevent** permet d'afficher en temps réel les évènements Wi-Fi de la carte.

On lance **iwevent** sur une console, et sur une autre, on passe des commandes iwconfig. On constate que **iwevent** affiche les évènements générés :

```
# iwevent
...
09:15:01.261767   wlan0    New Access Point/Cell address:Not-Associated
09:15:01.261858   wlan0    Set Frequency=2.412 GHz (Channel 1)
...
09:15:40.112202   wlan0    Set Frequency=2.462 GHz (Channel 11)
09:16:01.762209   wlan0    Custom driver  event:ASSOCINFO(ReqIEs=00074e455447454152010402040b16 RespIEs=010482840b16)
09:16:01.762247   wlan0    New Access Point/Cell address:00:09:5B:6F:3C:38
```

iwpriv

La commande **iwpriv** permet de configurer des paramêtres propres à la carte Wi-Fi.

```
# iwpriv wlan0
wlan0     no private ioctls.
```

Par exemple, le driver de cette carte ne fournit de paramêtres spécifiques à configurer.

Avec un autre driver, on obtient la liste des ioctl configurables :

```
# iwpriv ath0
ath0      Available private ioctls :
          setoptie         (8BEE) : set 256 byte  & get   0
          getoptie         (8BEF) : set   0       & get 256 byte
          setkey           (8BF2) : set  64 byte  & get   0
          delkey           (8BF4) : set   7 byte  & get   0
          setmlme          (8BF0) : set  42 byte  & get   0
```

Certaines cartes ethernet câble permettent également de configurer des paramêtres internes :

```
# iwpriv eth0
eth0      Available private ioctl :
                setqualthr (89F0) : set   1 byte & get   0
                gethisto (89F7) : set   0      & get  16 int

# iwpriv eth0 setqualthr 20
# iwpriv eth0 gethisto
```

iwspy

Si la carte et le driver le permettent, la commande **iwspy** permet d'afficher des statistiques en temps réel sur la liaison Wi-Fi :

```
# iwspy ath0
ath0      Statistics collected:
   00:15:6D:D0:E3:E0 : Quality=22/70  Signal level=-74 dBm  Noise level=-96 dBm
   Link/Cell/AP      : Quality=22/70  Signal level=-74 dBm  Noise level=-96 dBm
   Typical/Reference : Quality:0  Signal level:0  Noise level:0
```

Par exemple, la carte (et/ou driver) ci-dessous ne permet pas de collecter des informations.

```
# iwspy wlan0
wlan0     Interface doesn't support wireless statistic collection
```

Autres commandes utiles

lspci

```
# lspci
...
00:1f.1 IDE interface: Intel Corporation 82801G (ICH7 Family) IDE Controller (rev 01)
00:1f.2 IDE interface: Intel Corporation 82801GB/GR/GH (ICH7 Family) SATA IDE Controller (rev 01)
00:1f.3 SMBus: Intel Corporation 82801G (ICH7 Family) SMBus Controller (rev 01)
04:04.0 Ethernet controller: Atheros Communications Inc. Atheros AR5001X+ Wireless Network Adapter (rev 01)
04:08.0 Ethernet controller: Intel Corporation PRO/100 VE Network Connection (rev 01)
```

lsusb

```
# lsusb
...
BUS 005 Device 003: ID 0846:6a00 Netgear, Inc. WG111 WiFi (v2)
```

lshw

```
# lshw -class network
...
 *-network:0
      description: Wireless interface
      product: Atheros AR5001X+ Wireless Network Adapter
      vendor: Atheros Communications Inc.
      physical id: 4
      bus info: pci@0000:04:04.0
      logical name: wmaster0
      version: 01
      serial: 00:24:01:13:fe:de
      width: 32 bits
      clock: 33MHz
      capabilities: pm bus_master cap_list logical ethernet physical wireless
      configuration: broadcast=yes driver=ath5k latency=168 maxlatency=28 mingnt=10 multicast=yes wireless=IEEE 802.11bg
      resources: irq:18 memory:50000000-5000ffff
```

Test du réseau Ad-Hoc

Il faut tout d'abord passer la carte en mode **Ad-Hoc** :

```
# ifconfig wlan0 down
# iwconfig wlan0 mode Ad-Hoc
# ifconfig wlan0 up
```

On ajuste ensuite les paramêtres Wifi :

```
# iwconfig wlan0 essid "UPVD" channel 3 key off
```

On spécifie ensuite une adresse MAC commune à tout le réseau Ad-Hoc :

```
# iwconfig wlan0 ap 00:11:22:33:44:55
```

On spécifie les paramêtres IP :

```
# ifconfig wlan0 10.0.0.1 netmask 255.255.255.0
```

Désormais, les différents membres du réseau Ad-Hoc doivent pouvoir se pinguer.

Si ce n'est pas le cas, on peut essayer de changer la fréquence et de la remettre pour forcer une ré-detection / re-configuration.

Connexion à un réseau sans chiffrement

```
# iwconfig wlan0 essid "NETGEAR" channel 11 key off
# dhclient wlan0
```

Connexion à un réseau WEP

```
# iwconfig wlan0 essid "NETGEAR" channel 11 key CAFECAFECA
# dhclient wlan0
```

Connexion à un réseau WPA

Pour se connecter à un réseau WPA depuis Linux, il faut avoir installé le package **wpasupplicant** :

```
# apt-get install wpasupplicant
```

On crée le fichier de configuration suivant :

```
# cat /etc/wpa_supplicant.conf
network={
   ssid="RESEAUWPA"
   scan_ssid=1
   proto=WPA
   key_mgmt=WPA-PSK
   psk="supermotdepassetressecure"
}
```

On lance la commande **wpa_supplicant** :

```
# wpa_supplicant -D wext -i wlan0 -c /etc/wpa_supplicant.conf
```

Cette commande ne rend pas la main, donc soit on la lance en tâche de fond avec **&**, soit on la laisse tourner et on ouvre une nouvelle fenêtre.

On lance ensuite le client DHCP pour obtenir une adresse IP :

```
# dhclient wlan0
```

Le serveur de noms BIND

Historique

À la création d'Internet, chaque ordinateur du réseau contenait un fichier **/etc/hosts** qui listait le nom de toutes les machines du réseau et leurs adresses IP. À chaque fois que l'on rajoutait une machine sur Internet, il fallait mettre à jour ce fichier.

Le nombre de machines connecté à Internet s'étant rapidement accru, cette solution de fichier **/etc/hosts** communs est devenu ingérable, et il a fallu inventer un procédé capable de palier ce problème.

La solution qui s'est imposée fut la création d'une base de données distribuée, et ainsi est né le principe de serveur DNS.

Un serveur DNS permet de faire la correspondance entre un nom canonique (ex : www.google.fr) et son adresse IP.

Le premier serveur DNS fut créé par l'université de Berkeley et s'appelle BIND (Berkeley Internet Name Domain). BIND est le serveur DNS le plus utilisé et le plus populaire, environ 79 % d'Internet fonctionne avec ce logiciel[1].

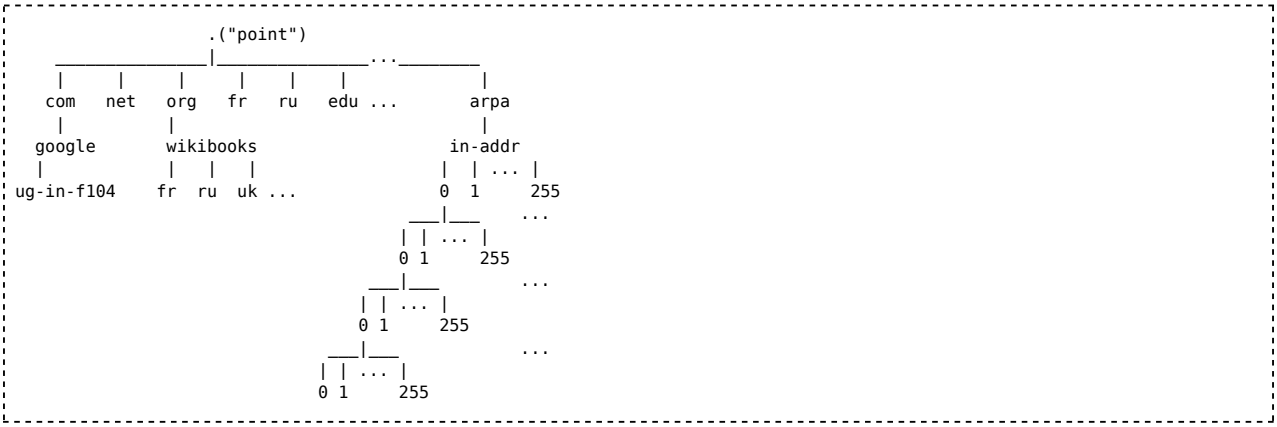
Configuration du client DNS

Sur un serveur Unix, la liste des serveurs DNS est définie dans le fichier **/etc/resolv.conf**.

```
$ cat /etc/resolv.conf
search mondomaine.fr
nameserver 192.168.30.1
```

Principe de fonctionnement du DNS

Schéma : l'arbre à l'envers



Au sommet de l'arbre on trouve des serveurs root qui aiguille vers les *top level domain* (com, net, org, fr, etc.) Il existe une branche spéciale ARPA avec un sous domaine in-addr qui sert à gérer le reverse DNS.

La commande host

La commande host permet d'obtenir l'adresse IP d'un ordinateur :

```
$ host www.google.com
www.google.com is an alias for www.l.google.com.
www.l.google.com has address 209.85.135.147
www.l.google.com has address 209.85.135.99
www.l.google.com has address 209.85.135.103
www.l.google.com has address 209.85.135.104
```

La commande **host** permet également de consulter le DNS inverse, c'est à dire quel nom canonique est associé à une adresse IP donnée :

```
$ host 66.249.93.104
104.93.249.66.in-addr.arpa domain name pointer ug-in-f104.google.com.
```

La commande dig

La commande **dig** permet d'interroger un serveur DNS.

Voici quelques exemples :

En interrogeant le sommet de l'arbre **.**, on obtient la liste des serveurs racines du DNS, appelés les root-servers[2] :

```
$ dig . NS
...
;; ANSWER SECTION:
.                       419748  IN      NS      a.root-servers.net.
.                       419748  IN      NS      b.root-servers.net.
.                       419748  IN      NS      c.root-servers.net.
.                       419748  IN      NS      d.root-servers.net.
...
```

En interrogeant la branche **com**, on obtient la liste des serveurs DNS gérant les noms de domaines en .com :

```
$ dig com. NS
...
;; ANSWER SECTION:
com.                    172800  IN      NS      i.gtld-servers.net.
com.                    172800  IN      NS      j.gtld-servers.net.
com.                    172800  IN      NS      k.gtld-servers.net.
com.                    172800  IN      NS      l.gtld-servers.net.
...
```

Si on interroge la branche **fr**, on obtient la liste des serveurs DNS gérant les noms de domaines en .fr. On constate que les extensions nationales sont gérés par des organismes nationaux (dans le cas de la France, le NIC France) :

```
$ dig fr. NS
...
;; ANSWER SECTION:
fr.                     172800  IN      NS      f.ext.nic.fr.
fr.                     172800  IN      NS      a.ext.nic.fr.
fr.                     172800  IN      NS      a.nic.fr.
fr.                     172800  IN      NS      b.ext.nic.fr.
...
```

En indiquant un nom de domaine, **dig** permet de connaitre différentes informations, comme par exemple :

La liste des serveurs DNS gérant le nom de domaine :

```
$ dig google.fr NS
...
;; ANSWER SECTION:
google.fr.              175462  IN      NS      ns3.google.com.
google.fr.              175462  IN      NS      ns4.google.com.
google.fr.              175462  IN      NS      ns1.google.com.
google.fr.              175462  IN      NS      ns2.google.com.
...
```

La liste des serveurs de mails :

```
$ dig google.fr MX
...
;; ANSWER SECTION:
google.fr.              10800   IN      MX      10 smtp4.google.com.
google.fr.              10800   IN      MX      10 smtp1.google.com.
google.fr.              10800   IN      MX      10 smtp2.google.com.
google.fr.              10800   IN      MX      10 smtp3.google.com.
...
```

Les Ressources Records (RR)

Les informations stockées dans un serveur DNS sont classifiées à l'aide des **Ressources Records** (RR).

Il existe de nombreux Ressources Records, voici les plus courants :

- NS (Name Server) indique les serveurs DNS gérant le nom de domaine. Exemple : **dig google.com NS** donne les name server de google.com
- A (Adresse IPv4) indique l'adresse IPv4 associée à un FQDN (Full Qualified Domain Name). Exemple : **dig www.google.com A** donne les adresses IPv4 de www.google.com

- AAAA (Adresse IPv6) indique l'adresse IPv6 associée à un FQDN (Full Qualified Domain Name). Exemple : **dig www.google.com AAAA** donne les adresses IPv6 de www.google.com
- MX (Mail eXchanger) indique le(s) serveur(s) de mail à contacter pour délivrer les emails du domaine. Exemple : **dig google.fr MX** donne les serveurs de mails acceptant des emails destinés à <un nom>@google.fr.
- CNAME (Canonical NAME) permet de créer des Alias (des noms étant des raccourcis vers d'autres noms). Exemple : **host www.google.fr** nous indique que **www.google.fr** est alias vers **www.google.com**.
- PTR (PoinTeuR) est utilisé par le reverse DNS pour effectuer la résolution d'une adresse IP vers un nom (FQDN). Exemple : **host 72.14.207.99** nous indique que l'adresse IP **72.14.207.99** est associé au nom **eh-in-f99.google.com**

Installation de BIND

Pour installer le serveur BIND sous Debian, on utilise la commande suivante :

```
# apt-get install bind
```

A partir de Debian Lenny, le package contenant le serveur BIND s'appelle **bind9**

Configuration de BIND

Les fichiers de configuration de BIND sont situés dans le répertoire **/etc/bind**.

Le fichier principal de configuration de BIND est **/etc/bind/named.conf**. Debian a choisit de découper ce fichier en 3 fichiers afin de faciliter les mises à jour.

A noter que dans ces fichiers, les lignes en commentaire commencent par **//** et non le **#** habituel des .conf, que l'on retrouve dans la syntaxe Apache.

/etc/bind/named.conf

```
# cat /etc/bind/named.conf
// Documentation : /usr/share/doc/bind/README.Debian

// Inclusion du fichier /etc/bind/named.conf.options
include "/etc/bind/named.conf.options";

// Configuration des logs
logging {
      category lame-servers { null; };
      category cname { null; };
};

// La zone définissant les root servers
zone "." {
      type hint;
      file "/etc/bind/db.root";
};

// La zone localhost
zone "localhost" {
      type master;
      file "/etc/bind/db.local";
};

// La zone inverse localhost
zone "127.in-addr.arpa" {
      type master;
      file "/etc/bind/db.127";
};

// La zone inverse réseau
zone "0.in-addr.arpa" {
      type master;
      file "/etc/bind/db.0";
};

// La zone inverse broadcast
zone "255.in-addr.arpa" {
      type master;
      file "/etc/bind/db.255";
};

// Inclusion du fichier /etc/bind/named.conf.local
include "/etc/bind/named.conf.local";
```

/etc/bind/named.conf.options

```
# cat /etc/bind/named.conf.options
options {
      // Emplacement des zones si on ne spéficie pas de chemin absolu
      directory "/var/cache/bind";

      // Option désormais obsolète depuis BIND 8
      fetch-glue no;

      // Option pour changer le port par défaut
      // query-source address * port 53;

      // Option pour indiquer un DNS à qui on va renvoyer
      // les demandes de résolution
      // forwarders {
      //      0.0.0.0;
      // };
};
```

Dans ce fichier, il est possible de préciser dans la section **forwarders** l'adresse IP du DNS à qui l'on souhaite renvoyer les demandes de résolutions de noms. Par exemple, ceci est utile lorsque notre serveur DNS ne peut pas accéder directement à Internet.

Par défaut BIND écoute sur le port 53 en UDP. On peut également changer ce port dans ce fichier, mais ceci est délicat car il faudra accorder la configuration des clients en conséquence. A noter que le dossier **/etc/services** contient le numéro par défaut des ports de tous les services.

/etc/bind/named.conf.local

On va définir dans ce fichier nos zones locales.

```
 # cat /etc/bind/named.conf.local
 zone "mondomaine.fr" {
       type master;
       file "/etc/bind/db.mondomaine.fr";
 };

 zone "mondomaine2.fr" {
       type master;
       file "/etc/bind/db.mondomaine2.fr";
 };
```

Le fichier définissant la zone

On crée ensuite le fichier de zone **/etc/bind/db.mondomaine.fr**

```
# cat /etc/bind/db.mondomaine.fr
;
; BIND data file for mondomaine.fr
;
$TTL    604800
@       IN      SOA     dns.mondomaine.fr. root.mondomaine.fr. (
                              1         ; Serial
                         604800         ; Refresh
                          86400         ; Retry
                        2419200         ; Expire
                         604800 )       ; Negative Cache TTL
;
@       IN      NS      dns.mondomaine.fr.
dns     IN      A       192.168.30.210
;
srv1    IN      A       192.168.30.211
;
@       IN      MX      0       mail.mondomaine.fr.
;
mail    IN      A       192.168.30.210
alex    IN      CNAME   mail
guillaume       IN      CNAME   srv1
```

A noter que dans les fichiers de zone, les lignes en commentaire commencent par **;** et non le **#** habituel.

Points importants de ce fichier :

Le caractère @ (arobas) remplace le nom de la zone.

Lorsque l'on définit un nom canonique, on a deux possibilités :

- soit on donne le nom en entier (ex: pc210.mondomaine.fr.) . Dans ce cas-là, il ne faut pas oublier le point final, sinon le système rajoute automatiquement le nom de la zone (mondomaine.fr).
- soit on ne donne que le nom "court" (ex: alex). Dans ce cas-là, il ne met pas le point final afin que le système rajoute le nom de la zone.

Test de fonctionnement

Une fois que l'on a modifié ces fichiers, on relance le serveur DNS :

```
# /etc/init.d/bind restart
```

On modifie ensuite le fichier /etc/resolv.conf pour lui indiquer d'utiliser le DNS que l'on vient de configurer :

```
# vi /etc/resolv.conf
search mondomaine.fr
nameserver 127.0.0.1
```

Pour tester, on essaye de pinguer un nom définit dans le DNS :

```
# ping pc210.mondomaine.fr
```

Si tout se passe bien, le DNS doit effectuer la résolution.

On peut aussi utiliser les commandes **host** et **dig** pour vérifier :

```
# host pc210.mondomaine.fr
...
# host guillaume.mondomaine.fr
...
# dig mondomaine.fr MX
...
```

Le fichier définissant la zone inverse

Maintenant que l'on a configuré le DNS de la zone **mondomaine.fr**, on va créer la zone inverse qui va permettre d'associer un nom à une adresse IP.

On rajoute tout d'abord la zone inverse dans le fichier **named.conf.local** :

```
// La zone reverse DNS
zone "30.168.192.in-addr.arpa" {
      type master;
      file "/etc/bind/db.192.168.30";
};
```

Le nom de la zone est composé de l'adresse réseau (à l'envers) associé à **in-addr.arpa**.

On crée ensuite le fichier **/etc/bind/db.192.168.30** :

```
# cat /etc/bind/db.192.168.30
;
; BIND data file for 192.168.30
;
$TTL    604800
@       IN      SOA     dns.mondomaine.fr. root.mondomaine.fr. (
                              1         ; Serial
                         604800         ; Refresh
                          86400         ; Retry
                        2419200         ; Expire
                         604800 )       ; Negative Cache TTL
;
@       IN      NS      dns.mondomaine.fr.
210     IN      PTR     pc210.mondomaine.fr.
211     IN      PTR     srv1.mondomaine.fr.
```

Le ressource record PTR permet de définir le nom associé à l'adresse IP.

Pour vérifier, on relance le DNS et on lui demande quelle nom est associé à une adresse IP donnée :

```
 # /etc/init.d/bind restart

 # host 192.168.30.210
 ...
```

Gestion des zones

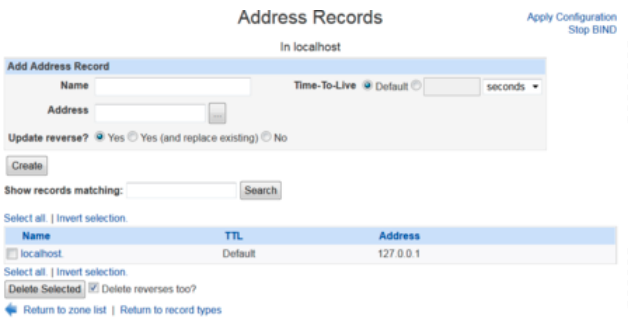
Pour modifier les redirections DNS d'un domaine, il faut modifier sa zone :

```
vim /etc/bind/db.mondomaine.fr
vim /var/lib/bind/example.com.hosts
```

Après modification des zones DNS, il s'avère nécessaire de demander sa propagation en cliquant tout en haut à droite sur *Apply Configuration*. Cela équivaut à la commande RNDC pour "Remote Name Daemon Control" :

```
rndc reload
```

Et éventuellement vérifier qu'il n'y a pas eu d'erreur :



Ajout d'un enregistrement IN A dans Webmin.

```
tail -300 /var/log/syslog
```

Cela permet par exemple de s'apercevoir que le numéro de série de la version de la zone doit être changé à chaque modification :

```
zone serial (999) unchanged. zone may fail to transfer to slaves.
```

Ce que L'interface graphique Webmin, plus ergonomique pour mettre à jour les zones DNS, incrémente automatiquement.

Problèmes connus

SERVFAIL

```
dig @localhost example.com

; <<>> DiG 9.8.1-P1 <<>> @localhost example.com
; (1 server found)
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: SERVFAIL, id: 28241
;; flags: qr rd ra; QUERY: 1, ANSWER: 0, AUTHORITY: 0, ADDITIONAL: 0

;; QUESTION SECTION:
;example.com.             IN      A

;; Query time: 0 msec
;; SERVER: 127.0.0.1#53(127.0.0.1)
;; WHEN: Mon May 25 16:37:31 2015
;; MSG SIZE  rcvd: 35
```

Le serveur DNS mentionné (localhost dans l'exemple) ne connait pas le domaine, mais il peut parfois le résoudre avec `host`.

Si le localhost est censé être autoritaire et que la zone y semble bien définie, ajouter un point à la fin de chaque URL, pour éviter qu'il les interprète en ajoutant le domaine après. En effet :

- `example.com. IN NS ns.example2.com` pourra être interprété à tort :
- `example.com. IN NS ns.example2.com.example.com.`, mais pas :
- `example.com. IN NS ns.example2.com.`

NXDOMAIN

```
host example.com
Host example.com not found: 3(NXDOMAIN)
```

Le domaine est inconnu des DNS, en général il suffit d'attendre la propagation 24 h.

Mais parfois, il manque juste un `rndc reload` ou un `/etc/init.d/bind9 restart`.

found SPF/TXT record but no SPF/SPF record found

Ces enregistrements vont par deux : `IN TXT` et `IN SPF`.

Le domaine ne se propage pas, telnet localhost 53 fonctionne en local mais pas de l'extérieur

Les symptômes sont les mêmes que si le port 53 était bloqué par le pare-feu, mais en fait il faut ajouter les IP publiques dans "/etc/bind/named.conf.options" :

```
listen-on-v6 { ::1; MonIPv6; };
listen-on { 127.0.0.1; MonIPv4; };
```

Puis redémarrer le service :

```
/etc/init.d/bind9 restart
```

query (cache) '...' denied

Retirer les IP locales de :

```
vim /etc/resolv.conf
```

Références

1. (en) Bryan J. Hong, *Building a Server with FreeBSD 7: A Modular Approach*, No Starch Press, 2008 (lire en ligne (https://books.google.fr/books?id=0yOzQrMYLJQC&pg=PA73&lpg=PA73&dq=%22slightly+over+79+percent+of+DNS+servers+use+BIND%22&source=bl&ots=XX6JD4C7Jn&sig=VvK0fHHO1YApBiYHBGoHzqZMRLQ&hl=fr&sa=X&ei=JwxaVZ2_GYO-sAWE6YGwBg&ved=0CCEQ6AEwAA#v=onepage&q=%22slightly%20over%2079%20percent%20of%20DNS%20servers%20use%20BIND%22&f=false))
2. http://www.root-servers.org/ www.root-servers.org

Le serveur de configuration réseau DHCP

Fonctionnement du serveur DHCP

DHCP : Dynamic Host Configuration Protocol. Ce service permet de configurer automatiquement les paramètres IP des machines du réseau local. Le principe est simple. Une machine se connecte sur le réseau local et envoie une requête DHCP Discover pour demander à un éventuel serveur DHCP une adresse IP. Si le serveur est présent, il renvoie un message DHCP Offer. Si plusieurs serveurs DHCP sont présents sur le réseau, le client retient une offre d'un des serveurs et diffuse une demande DHCP : DHCP Request. Le serveur choisit par le client renvoie alors un message de validation DHCP ACK. Le client reçoit donc : Adress IP, Masque de réseau, Adresse de passerelle et Adresse des serveurs DNS.

Installation du serveur DHCP (Sur Debian)

apt-get install dhcp3-server

Configuration du serveur DHCP

Le fichier de configuration du serveur DHCP est **/etc/dhcp3/dhcpd.conf**.

```
$ more /etc/dhcp3/dhcpd.conf

# Indique à DHCP de ne pas faire la mise à jour dans le DNS
ddns-update-style none;

# Le nom de domaine du réseau
option domain-name "exemple.org";

# Le(s) nom(s) des serveurs de noms
option domain-name-servers ns1.exemple.org, ns2.exemple.org;

# Durée par défaut du bail
default-lease-time 600;

# Durée maximum du bail
max-lease-time 7200;

# Pour les logs, on utilise la facilité local7
log-facility local7;
```

```
subnet 192.168.30.0 netmask 255.255.255.0 {

  # La plage d'attributions des adresses IP
  range 192.168.30.20 192.168.30.240;

  # la liste des passerelles
  option routers fw.exemple.org, fw2.exemple.org;

  # L'adresse broadcast
  option broadcast-address 192.168.30.255;
}
```

Pour certains ordinateurs, on peut leur attribuer tout le temps la même adresse IP en les identifiant avec leurs adresses MAC. Il suffit de rajouter la section suivante :

```
host monordi {
  hardware ethernet 0:0:c0:5d:bd:95;
  fixed-address 192.168.30.200;
}
```

Chaque fois que l'on modifie ce fichier, ne pas oublier de relancer le serveur DHCP afin qu'il prenne en compte les modifications :

```
# /etc/init.d/dhcp3-server restart
```

ou

```
# /etc/init.d/isc-dhcp-server restart
```

Chaque fois que DHCP attribue une adresse IP, il enregistre un message dans **/var/log/syslog** :

```
...
```

```
Feb 11 10:44:34 fw dhcpd: DHCPREQUEST for 192.168.30.33 from 00:50:ba:2d:d9:17 via eth0
Feb 11 10:44:34 fw dhcpd: DHCPACK on 192.168.30.33 to 00:50:ba:2d:d9:17 via eth0
...
```

DHCP stocke les adresses IP attribués dans le fichier **/var/lib/dhcp/dhcpd.leases**. Ceci lui permet notamment d'attribuer à un ordinateur la même adresse IP (même si ceci n'est pas obligatoire).

```
$ cat /var/lib/dhcp/dhcpd.leases
...
lease 192.168.30.182 {
      starts 1 2008/02/11 09:46:23;
      ends 1 2008/02/11 09:56:23;
      hardware ethernet 00:1c:c0:0c:b6:25;
      client-hostname "pc231";
}
...
```

Le serveur de shell distant SSH

Le serveur SSH

SSH (Secure SHELL) permet de se connecter à un ordinateur distant et de disposer d'un shell sécurisé. Par défaut, le serveur SSH attend les connexions distantes sur le port 22 / protocole TCP.

Pour installer un serveur SSH, on utilise la commande suivante :

```
# apt-get install ssh
```

La connexion et le transfert de données via SSH est sécurisée par un système de chiffrement utilisant soit l'algorithme RSA (Rivest Shamir Adleman), soit l'algorithme DSA (Digital Signature Algorithm).

Lors de son installation, le serveur SSH génère des clés de chiffrement RSA et DSA. Ces clés sont composées d'une partie privée et d'une partie publique. Elles sont stockées dans le répertoire **/etc/ssh/** :

```
$ ls -l /etc/ssh/
...
-rw------- 1 root root    672 2007-11-05 17:37 ssh_host_dsa_key
-rw-r--r-- 1 root root    600 2007-11-05 17:37 ssh_host_dsa_key.pub
-rw------- 1 root root   1675 2007-11-05 17:37 ssh_host_rsa_key
-rw-r--r-- 1 root root    392 2007-11-05 17:37 ssh_host_rsa_key.pub
```

Chaque tentative de connexion est enregistrée dans le fichier **/var/log/auth.log**.

Fichier de configuration

Le fichier de configuration du serveur SSH est **/etc/ssh/sshd_config** :

```
# cat /etc/ssh/sshd_config

# Port de fonctionnement du serveur SSH
Port 22

# Permet de spécifier sur quelle interface SSH écoute
#ListenAddress ::
#ListenAddress 0.0.0.0

# On utilise exclusivement la version 2 du protocole SSH
Protocol 2

# Emplacement des clés RSA et DSA
HostKey /etc/ssh/ssh_host_rsa_key
HostKey /etc/ssh/ssh_host_dsa_key

# On active la séparation des privilèges
UsePrivilegeSeparation yes

# Durée de vie et taille de la clé
KeyRegenerationInterval 3600
ServerKeyBits 768

# Pour syslog
SyslogFacility AUTH
LogLevel INFO

# Authentification
LoginGraceTime 120
PermitRootLogin no
StrictModes yes

RSAAuthentication yes
PubkeyAuthentication yes
#AuthorizedKeysFile    %h/.ssh/authorized_keys

# Ignore les fichiers ~/.rhosts et ~/.shosts
IgnoreRhosts yes

# Ignore l'authentification RhostsRSA
RhostsRSAAuthentication no
# Même principe pour la version 2 du protocole
HostbasedAuthentication no
```

```
# A décommenter sur on ne veut pas se fier au fichier ~/.ssh/known_hosts pour RhostsRSAAuthentication
#IgnoreUserKnownHosts yes

# Empêche la connexion des utilisateurs qui n'ont pas de mot de passe (PAS RECOMMANDE)
PermitEmptyPasswords no

# Activer les mots de passes par Challenge / Réponse
ChallengeResponseAuthentication no

# Permet de supprimer l'authentification par mot de passe et n'utiliser que l'authentification par clé partagée
#PasswordAuthentication yes

# Options Kerberos
#KerberosAuthentication no
#KerberosGetAFSToken no
#KerberosOrLocalPasswd yes
#KerberosTicketCleanup yes

# Options GSSAPI
#GSSAPIAuthentication no
#GSSAPICleanupCredentials yes

# Activer la redirection X11
X11Forwarding yes
X11DisplayOffset 10

# Afficher le message du jour (Message Of the Day)
PrintMotd no

# Afficher la date et heure de la dernière connexion
PrintLastLog yes

# Maintient la connexion TCP
TCPKeepAlive yes

#UseLogin no

#MaxStartups 10:30:60
#Banner /etc/issue.net

# Permet à un client de passer des variables locales d'environnement
AcceptEnv LANG LC_*

Subsystem sftp /usr/lib/openssh/sftp-server

# Utilise l'authentification via PAM
UsePAM yes
```

Les options de ce fichier de configuration sont décrites dans la page de manuel de **sshd_config** :

```
$ man sshd_config
```

Le fichier de configuration par défaut proposé par Debian nécessite quelques ajustements.

La ligne **PermitRootLogin yes** autorise les connexions distantes à partir du compte **root**. Cette option est dangereuse car elle permet à un attaquant distant de scanner le mot de passe du super-utilisateur **root**.

La bonne pratique est de se connecter avec son compte utilisateur (ex: alex), et ensuite de passer **root** avec la commande **su**, ou d'effectuer les opérations d'administration avec la commande **sudo**.

On remplace donc cette ligne par :

```
PermitRootLogin no
```

Afin de renforcer la sécurité, on peut limiter les connexions SSH à une liste d'utilisateurs donnés. Ceci est réalisé avec l'option **AllowUsers** :

```
AllowUsers alex pierre
```

Sur le même principe, on peut restreindre les connexions à un ou plusieurs groupes Unix :

```
AllowGroups admin sshusers
```

Afin que ces modifications soient prises en compte, il faut relancer le serveur SSH :

```
# /etc/init.d/ssh restart
```

Le client SSH

Utilisation

Pour se connecter à un serveur SSH, on utilise la commande **ssh**. Sa syntaxe est la suivante :

```
$ ssh <login>@<nom ou adresse IP du serveur>
```

Exemple :

```
$ ssh alex@pc210
```

ou

```
$ ssh pc210 -l alex
```

Si on possède le même login sur la machine locale et distante, il est inutile de spécifier le login :

```
$ ssh pc210
```

Si ssh ne fonctionne pas sur le port standard 22, l'option **-p** permet d'indiquer le port à utiliser :

```
$ ssh pc210 -p 2222
```

L'option **-X** du client ssh permet de rediriger l'affichage graphique (le DISPLAY) via le tunnel ssh, et ainsi lancer un programme graphique distant et l'afficher sur notre ordinateur :

```
$ ssh -X pc210
...
pc210$ xeyes &
```

Il faut toutefois que l'option **X11Forwarding** soit positionnée à **yes** sur le serveur pour que la redirection graphique fonctionne.

Vérification du fingerprint

Lors de la première connexion, **ssh** affiche le **fingerprint** du serveur SSH et demande confirmation :

```
$ ssh pc210
The authenticity of host 'pc210 (192.168.30.210)' can't be established.
RSA key fingerprint is 8e:c6:f0:b5:e6:71:c9:20:ec:5d:ed:d4:e1:fc:fb:16.
Are you sure you want to continue connecting (yes/no)?
```

Si on veut être certain de l'authenticité du serveur distant, on peut contacter l'administrateur et vérifier avec lui que le fingerprint indiqué est le bon. Pour faire ceci, taper la commande **ssh-keygen -l** sur le serveur ssh et indiquer le chemin vers la clé RSA du serveur ssh :

```
# ssh-keygen  -lv
Enter file in which the key is (/root/.ssh/id_rsa): /etc/ssh/ssh_host_rsa_key
2048 8e:c6:f0:b5:e6:71:c9:20:ec:5d:ed:d4:e1:fc:fb:16 /etc/ssh/ssh_host_rsa_key.pub
+--[ RSA 2048]----+
|                 |
|                 |
|          . .    |
|         oEo     |
|    .   S.o      |
|o. . + +o=.      |
|=.+   E..+o      |
|oo . . oo.       |
|    . o..        |
+-----------------+
```

Par la suite, le fichier /home/<user>/.ssh/known_hosts stoque l'identité chiffrée de la machine et **ssh** ne nous demande plus confirmation.

Authentification automatique

Si on se connecte souvent sur le même serveur, on peut générer une paire de clés afin de ne pas avoir à saisir le mot de passe à chaque connexion.

Sur la machine cliente

On génére une paire de clés avec la commande **ssh-keygen** :

```
$ ssh-keygen -t rsa
Generating public/private rsa key pair.
Enter file in which to save the key (/home/alex/.ssh/id_rsa):
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /home/alex/.ssh/id_rsa.
Your public key has been saved in /home/alex/.ssh/id_rsa.pub.
The key fingerprint is:
41:ab:25:09:eb:ad:41:66:2d:d6:85:e3:73:02:40:e3 alex@pc210
```

On ne saisi pas de passphrase, sinon le système va demander à saisir la passphrase à chaque connexion, ce qui est aussi contraignant que la saisie du mot de passe.

Sur la machine distante

On se connecte sur la machine distante et on copie le contenu de la clé publique précédement générée (/home/alex/.ssh/id_rsa.pub) dans le fichier **/home/alex /.ssh/authorized_keys**.

```
   ssh-copy-id -i /home/alex/.ssh/id_rsa.pub alex@192.168.30.190
```

Note: Si on ne dispose pas de la commande **ssh-copy-id** (anciennes versions de ssh), on peut utiliser la commande suivante :

```
   cat ~/.ssh/id_rsa.pub | ssh alex@192.168.30.190 "cat - >> ~/.ssh/authorized_keys"
```

Ceci fait, on peut se connecter sur la machine distante sans avoir à saisir le mot de passe.

La commande scp

La commande **scp** (Secure Copy) permet de copier un fichier d'un ordinateur vers un autre en utilisant SSH.

Par exemple, la commande suivante permet de copier le fichier **fichier.txt** vers le répertoire **/tmp** de l'ordinateur **pc211** :

```
$ scp fichier.txt alex@pc211:/tmp
```

On peut également copier un fichier distant sur la machine locale :

```
$ scp alex@pc211:/etc/passwd /tmp
```

L'option **-r** permet de copier un répertoire de manière récursive.

Les clients SSH sous Windows

Il existe plusieurs clients SSH pour Windows, dont notamment les logiciels libres suivants :

- PuTTY[1]
- Winscp[2]

Problèmes connus

Le mot de passe est toujours demandé malgré la clé SSH

Quand on teste la clé SSH depuis le client :

```
$ ssh -i .ssh/id_rsa <login>@<serveur>
```

si les logs d'authentification du serveur acceptent la clé :

```
$ tail /var/log/auth.log
```

cela donne :

```
Accepted publickey for root from xxxx port yyyy ssh2: RSA SHA256:zzzz
```

ou :

```
Found matching RSA key:zzzz
```

au lieu de :

```
Accepted password for root from xxxx port yyyy ssh2
```

Il faut donc réinstaller les clés.

Authentication refused: bad ownership or modes for directory

Il faut s'approprier le dossier :

```
chown MonUtilisateur MonDossier
```

Could not create directory '/c/UsersUtilisateur/.ssh' ... Failed to add the host to the list of known hosts

Se produit sous Windows quand le chemin défini n'est pas celui retenu par SSH. Par exemple, quand :

```
$ setx HOME %USERPROFILE%
```

ou

```
$ setx HOME "C:/Users/Utilisateur/"
```

ou

```
$ setx HOME "C:\\Users\\Utilisateur\\"
```

donnent tous :

```
$ echo $HOME
/c/UsersUtilisateur
```

Peu importe la casse.

On peut aussi utiliser *CMD* au lieu de *Bash* pour modifier la variable d'environnement "home", ce qui aboutit à un résultat équivalent :

```
> setx HOME %USERPROFILE%
> set home
home=C:\Users\Utilisateur
HOMEDRIVE=C:
HOMEPATH=\Users\Utilisateur
```

Attention : sous *Git CMD* cela ne fonctionne pas, il faut vraiment utiliser la console DOS du système :

```
> set home
home=c:UsersUtilisateur
HOMEDRIVE=C:
HOMEPATH=\Users\Utilisateur
```

Mais surtout il faut utiliser *CMD* pour mettre à jour le contenu du dossier .ssh sans obtenir cette erreur, par exemple en installant OpenSSH for Windows (https://sourceforge.net/projects/sshwindows/files/OpenSSH%20for%20Windows%20-%20Release/) :

```
>"C:\Program Files (x86)\OpenSSH\bin\ssh.exe" depot.example.com
RSA key fingerprint is zzzz
Are you sure you want to continue connecting (yes/no)? yes
Warning: Permanently added 'depot.example.com' (RSA) to the list of known hosts.
```

Could not open a connection to your authentication agent

Lancer :

```
eval `ssh-agent -s`
```

et réitérer.

Enter passphrase for key

Il faut ajouter la clé privée depuis le serveur avec :

```
ssh-add ~/.ssh/id_rsa
```

Sinon, spécifier la clé privée depuis le client :

```
ssh -i .ssh/id_rsa <login>@<serveur>
```

Sinon, voir #Permission denied (publickey,hostbased).

error: Received disconnect from x.x.x.x port yyyyy:13: Unable to authenticate [preauth]

Se produit sur un serveur quand on essaie de se connecter automatiquement sans spécifier de clé privée.

Permission denied (publickey,hostbased)

Il manque l'enregistrement de la clé publique (ssh -i).

Sinon, utiliser Putty.

Server refused our key

Vérifier les permissions, par exemple pour root :

```
 ls -alh .ssh
total 24K
drwxr--r--  2 root     root     4,0K mars   1 17:57 .
drwxrwxr-x 25 root     root     4,0K mars   6 12:41 ..
-rw-------  1 root     root      209 mars   1 17:57 authorized_keys
-rw-------  1 root     root      951 mars   1 17:57 id_rsa
-rw-r--r--  1 root     root      397 mars   1 17:57 id_rsa.pub
-rw-r--r--  1 root     root      869 mars   1 17:57 private.ppk
```

Si le problème persiste, la cause est détaillée dans :

```
tail /var/log/auth.log
```

WARNING: UNPROTECTED PRIVATE KEY FILE!

```
Permissions 0644 for '/cygdrive/c/Users/Utilisateur/.ssh/id_rsa' are too open.
It is recommended that your private key files are NOT accessible by others.
This private key will be ignored.bad permissions: ignore key: /cygdrive/c/Users/Utilisateur/.ssh/id_rsa
```

Sous Linux il est facile d'appliquer les permissions sur la clé :

```
chmod 700 ~/.ssh
chmod 600 ~/.ssh/id_rsa
```

Sous Windows par contre, la commande `chmod` n'a aucun effet depuis Bash, même lancée en tant qu'administrateur. Et une modification des droits après suppression de

l'ACL, via Cygwin ou l'explorateur Windows fonctionne chez eux, mais n'a guère plus d'effet pour SSH :

```
$ setfacl -b ~\.ssh
$ setfacl -b ~\.ssh\id_rsa
$ chgrp -R Utilisateurs ~\.ssh
$ chmod -Rv 600 ~\.ssh\id_rsa
mode of ‘C:\\Users\\Utilisateur\\.ssh\\id_rsa’ changed from 0644 (rw-r--r--) to 0600 (rw-------)
$ ls -alh ~\.ssh\id_rsa
-rw-r--r-- 1 Utilisateur 1049089 843 sept. 19  2011 C:\Users\Utilisateur\.ssh\id_rsa
```

Cette section est vide, pas assez détaillée ou incomplète.

Références

1. http://www.putty.org/
2. http://winscp.net

Le partage de fichiers Samba

Introduction

Samba est l'implémentation du protocole SMB (Server Message Block) sous Unix / Linux. Il sert à partager des fichiers et des imprimantes avec les réseaux Microsoft.

Site officiel : www.samba.org (http://www.samba.org)

Samba lance deux services : **smbd** et **nmbd**.

Le protocole SMB est tout sauf performant : il pollue le réseau par l'utilisation intensive du broadcast.

Samba utilise les ports **137** (netbios name service, nbname, en UDP), **138** (netbios datagram service nbdatagram, en UDP), **139** (netbios session service, nbsession, en TCP), et **445** ("direct-hosted" tcp, en tcp et udp).

La méthode de connexion traditionnelle smb utilise les ports 137, 138 et 139 tandis que la nouvelle méthode (CIFS, sur Windows XP) n'utilise que le port 445.

Installation

Pour installer Samba, on tape la commande suivante :

```
# apt-get install samba smbclient smbfs winbind
```

Configuration

Le fichier de configuration de Samba est **/etc/samba/smb.conf**.

Le programme **testparm** analyse le fichier smb.conf et signale les erreurs éventuelles.

Après avoir modifié la configuration, il faut relancer le service :

```
# /etc/init.d/samba restart
```

Le fichier **/etc/samba/smb.conf** se divise en différentes sections :

- La section **[global]** : configuration globale de samba
- La section **[homes]** : cette section particulière permet de remonter une ressource qui correspond au répertoire de travail (home directory) de l'utilisateur qui s'est authentifié.

On peut ensuite créer différentes sections, une par partage voulu.

Samba peut fonctionner de différentes manières. Voici quelques cas de figures courants.

Le partage par ressource sur un réseau Workgroup

Voici un fichier **/etc/samba/smb.conf** permettant de partager une ressource (répertoire ou imprimante) sur un réseau Workgroup :

```
# PARTAGE PAR RESSOURCES (share)

# Il y a 2 cas possibles :
#
# CAS n°1 : on partage une ressource totalement anonyme
# CAS n°2 : on partage une ressource avec un mot de passe associé à la ressource

# Section GLOBAL
# Configuration globale de Samba
[global]
      # Nom du groupe de travail
      workgroup = WORKGROUP

      # Nom Netbios de la machine (identification réseau)
      netbios name = PC230

      # Chaine de commentaire associé au serveur (voisinage réseau)
      server string = %h Serveur (Samba %v)

      # Utilisateurs interdits
```

```
	invalid users = root

	# Enregistre un fichier de log par machine cliente du réseau MS
	log file = /var/log/samba/log.%m

	# Taille maximale des logs : 1 Mo
	max log size = 1000

	# On n'utilise pas Syslog pour enregistrer les logs
	syslog = 0

	# On fait un partage par ressources
	security = share

	# On utilise les mots de passe encryptés
	# (attention, W95 et W98a fonctionnent avec les mdp en clair)
	encrypt passwords = true

	# Accélère les transferts réseaux
	socket options = TCP_NODELAY

	# Empeche nmbd de chercher à résoudre le nom netbios via le DNS
	dns proxy = no

	# Nom du compte invité qui va permettre de créer une
	# ressource partagée par mot de passe (celui du compte invite)
	guest account = invite

	# Emplacement du fichier contenant les logins et mdp samba
	smb passwd file = /etc/samba/smbpasswd

	# Pour récupérer les imprimantes définies dans CUPS
	printing = cups


# Section HOMES
# Cette section est inutile ici car on utilise le mode share


# Section PRINTERS
# Cette section permet de partager les imprimantes définies sur le serveur
[printers]
	# Le commentaire associé à l'imprimante
	comment = Les imprimantes

	# Affiche les imprimantes dans la liste des partages du serveur
	browseable = yes

	# Dans le cas d'une imprimante, c'est l'emplacement
	# des fichiers temporaires
	path = /tmp

	# Spécifie qu'il s'agit d'une imprimante et non un répertoire
	printable = yes

	# Partage l'imprimante de manière anonyme
	public = yes

	# Logique
	writable = no

	# Empeche d'autres utilisateurs de supprimer mes impressions
	create mode = 0700


# Sections REPERTOIRES PARTAGES

# CAS n°1: un CDROM partagé anonymement
[cdrom]
	# Le commentaire associé au CDROM
	comment = Le CDROM

	# Point de montage du CDROM
	# synonyme de directory =
	path = /cdrom

	# Empeche le bloquage d'un fichier par un utilisateur
	locking = no

	# Logique
	writable = no

	# Partage le CDROM de manière anonyme
```

```
      public = yes


# CAS n°2 : un répertoire partagé mais protégé par mot de passe
[partage]
      # Le commentaire associé au répertoire
      comment = Un répertoire partagé mais protégé par mot de passe

      # Emplacement du répertoire partagé
      path = /home/partage

      # Support en lecture / écriture
      # synonyme de read only = no
      writable = yes

      # Partage anonyme désactivé (cas par défaut)
      # synonyme de guest ok = no
      public = no

      # Voir explication ci-dessous
      valid users = invite

      # Masque de création des fichiers et répertoires
      create mask = 0644
      directory mask = 0755
```

Dans le cas du mode share, pour pouvoir positionner un mot de passe sur une ressource, il faut créer un compte UNIX 'invite' :

```
# adduser --shell /bin/false --disabled-login invite
```

On crée ensuite une entrée dans le fichier /etc/samba/smbpasswd :

```
# smbpasswd -a invite
```

Le mot de passe saisi correspondra à celui du répertoire partagé.

NB : dans les versions récentes de Samba, il faut utiliser la commande **pdbedit** à la place de **smbpasswd** :

```
# pdbedit -a invite
```

Il faut ensuite créer le répertoire partagé et donner l'arborescence partagée à l'utilisateur invite du groupe invite :

```
# mkdir /home/partage
# chown nobody.invite /home/partage
```

NB : il faut que l'utilisateur **nobody** ait le droit d'écriture sur le répertoire partagé.

Petit rappel : il est toujours utile de faire un **testparm** pour vérifier la cohérence du fichier de configuration, et ne pas oublier de relancer le service pour que les modifications soient prises en compte.

On peut tester la configuration avec la commande **smbclient** :

```
# smbclient -L PC230
Password:
Domain=[WORKGROUP] OS=[Unix] Server=[Samba 3.0.24]

      Sharename       Type      Comment
      ---------       ----      -------
      cdrom           Disk      Le CDROM
      partage         Disk      Un répertoire partagé mais protégé par mot de passe
      IPC$            IPC       IPC Service (pc230 Serveur (Samba 3.0.24))
      HP_LaserJet_1200_LPT_parport0_HPLIP Printer   HP LaserJet 1200

Domain=[WORKGROUP] OS=[Unix] Server=[Samba 3.0.24]

      Server               Comment
      ---------            -------
      PC230                pc230 Serveur (Samba 3.0.24)
```

```
      Workgroup            Master
      ---------            -------
      ...                  ...
```

La commande **smbclient** demande un mot de passe, on tape **entrée**.

Le partage par utilisateur sur un réseau Workgroup

Voici un fichier **/etc/samba/smb.conf** permettant de partager une ressource (répertoire ou imprimante) avec une authentification **Utilisateur / Mot de passe** sur un réseau Workgroup :

```
# PARTAGE PAR UTILISATEUR (user) - Groupe de travail

# Il y a 2 cas possibles :
#
# CAS n°1 : on partage une ressource totalement anonyme
# CAS n°2 : cas normal du mode user, on fait une authentification
#           par login et mot de passe


# Section GLOBAL
# Configuration globale de Samba
[global]
      # Nom du groupe de travail
      workgroup = WORKGROUP

      # Nom Netbios de la machine (identification réseau)
      netbios name = PC230

      # Chaîne de commentaire associé au serveur (voisinage réseau)
      server string = %h Serveur (Samba %v)

      # Utilisateurs interdits
      invalid users = root

      # Enregistre un fichier de log par machine cliente du réseau MS
      log file = /var/log/samba/log.%m

      # Taille maximale des logs : 1 Mo
      max log size = 1000

      # On n'utilise pas Syslog pour enregistrer les logs
      syslog = 0

      # On fait un partage par utilisateur
      security = user

      # On utilise les mots de passe encryptés
      # (attention, W95 et W98a fonctionnent avec les mdp en clair)
      encrypt passwords = true

      # Accélère les transferts réseaux
      socket options = TCP_NODELAY

      # Empêche nmbd de chercher à résoudre le nom netbios via le DNS
      dns proxy = no

      # Nom du compte invité qui va permettre de créer une
      # ressource partagée par mot de passe (celui du compte invite)
      # POUR LE CAS n°1
      guest account = invite

      # Emplacement du fichier contenant les logins et mdp samba
      # concerne le CAS n°1 et n°2
      smb passwd file = /etc/samba/smbpasswd


# Section HOMES
# Cette section particulière permet de remonter une ressource
# qui correspond au répertoire de travail (home directory)
# de l'utilisateur qui s'est authentifié
[homes]
      # Correspond au home directory de l'utilisateur authentifié
      # %U sera remplacé par le nom d'utilisateur (login)
      comment = Le répertoire personnel de %U

      # Si on active cette option, on dispose d'une ressource 'homes'
      # redondante avec la ressource 'login'
      browseable = no

      # Il faut que l'utilisateur soit authentifié pour accéder
      # à la ressource (son home directory sur le serveur)
```

```
      public = no

      # Autorise l'écriture dans le répertoire personnel
      writable = yes

      # Droits d'accés des fichiers et répertoires crées
      create mask = 0644
      directory mask = 0755


# Section PRINTERS
# Cette section permet de partager les imprimantes définies sur le serveur
[printers]
      # Le commentaire associé à l'imprimante
      comment = Les imprimantes

      # Affiche les imprimantes dans la liste des partages du serveur
      browseable = yes

      # Dans le cas d'une imprimante, c'est l'emplacement
      # des fichiers temporaires
      path = /tmp

      # Spécifie qu'il s'agit d'une imprimante et non un répertoire
      printable = yes

      # Partage l'imprimante de manière anonyme
      public = yes

      # Logique
      writable = no

      # Empêche d'autres utilisateurs de supprimer mes impressions
      create mode = 0700


# Sections REPERTOIRES PARTAGES

# CAS n° 1 : une ressource anonyme sans mot de passe
# Un CDROM
[cdrom]
      # Le commentaire associé au CDROM
      comment = Le CDROM

      # Point de montage du CDROM
      # synonyme de directory =
      path = /cdrom

      # Empêche le bloquage d'un fichier par un utilisateur
      locking = no

      # Logique
      writable = no

      # Partage le CDROM de manière anonyme
      public = yes


# CAS n°2 : un répertoire partagé qu'à certains utilisateurs
[prive]
      # Le commentaire associé au répertoire
      comment = Un répertoire partagé qu'a certains utilisateurs (CAS n°2)

      # Emplacement du répertoire partagé
      path = /home/prive

      # Support en lecture / écriture
      # synonyme de read only = no
      writable = yes

      # Partage anonyme désactivé (cas par défaut)
      # synonyme de guest ok = no
      public = no

      # Voir explication ci-dessous
      # le @prive désigne le groupe Unix 'prive'
      valid users = @prive

      # Masque de création des fichiers et répertoires
      create mask = 0644
      directory mask = 0755
```

Dans le cas du mode user, pour ne partager une ressource partagée qu'à certains utilisateurs, il faut tout d'abord créer un compte UNIX pour chacun d'eux :

```
# adduser --shell /bin/false --disabled-login paul
# adduser --shell /bin/false --disabled-login pierre
```

On crée ensuite une entrée dans le fichier /etc/samba/smbpasswd :

```
# smbpasswd -a paul
# smbpasswd -a pierre
```

NB : dans les versions récentes de Samba, il faut utiliser la commande **pdbedit** à la place de **smbpasswd** :

```
# pdbedit -a paul
# pdbedit -a pierre
```

Coté Linux, les utilisateurs doivent avoir le droit d'écriture sur le répertoire partagé. La meilleure méthode consiste à créer un groupe Unix contenant les utilisateurs autorisés, et d'attribuer le répertoire partagé au groupe en question :

```
# addgroup prive
# adduser paul prive
# adduser pierre prive
# mkdir /home/prive
# chgrp prive /home/prive
# chmod 770 /home/prive
```

Avantage : pour partager cette ressource à de nouveaux utilisateurs, il suffit de leur créer le compte Unix, une entrée dans smbpasswd et de les ajouter au groupe prive (il n'est plus nécessaire de modifier le fichier smb.conf).

```
# chmod g+s /home/prive
```

Permet de définir le propriétaire des fichiers créés avec l'option +s. Ici les propriétaires seront les membres du groupe **prive** au lieu de l'utilisateur.

Connexion à un Active Directory Windows 2012

Serveur AD : SERV2012 (192.168.10.254)

Domaine : DOMWIN.LAN

Configuration de /etc/resolv.conf

Il faut indiquer dans le fichier /etc/resolv.conf d'utiliser le serveur AD en tant que serveur DNS :

```
# cat /etc/resolv.conf
nameserver 192.168.10.254
```

Configuration de Kerberos

On installe Kerberos pour l'authentification sur AD :

```
# apt-get install krb5-{admin-server,user}
```

On configure Kerberos avec notre domaine AD (DOMWIN.LAN) :

```
# cat /etc/krb5.conf

[logging]
Default = FILE:/var/log/krb5.log

[libdefaults]
ticket_lifetime = 24000
clock-skew = 300
default_realm = domwin.lan
# dns_lookup_realm = false
# dns_lookup_kdc = true

[realms]
domwin.lan = {
```

```
kdc = 192.168.10.254:88
admin_server = 192.168.10.254:464
default_domain = domwin.lan
}

[domain_realm]
.domwin.lan = domwin.lan
domwin.lan = domwin.lan
```

Configuration de Samba

```
...
[global]
    workgroup = DOMWIN
    netbios name = ALEX
    wins support = no
    security = ads
    realm = DOMWIN.LAN
...
```

Connexion au domaine AD

On obtient un ticket Kerberos :

```
# kinit Administrateur@DOMWIN.LAN
# klist
```

On récupére le SID du serveur AD (nom netbios de AD : SERV2012) :

```
# net rpc getsid -S SERV2012
```

On rejoint le domaine AD :

```
# net ads join -U Administrateur -S SERV2012
```

On vérifie en interrogeant les partages du serveur AD. Pour cela, on utilise un compte définit sur AD :

```
# smbclient -L SERV2012 -U etudiant
```

On essaye d'accéder à un partage définit sur AD :

```
# smbclient ''\\SERV2012\partage' -U etudiant
```

Fichiers /etc/krb5.conf et /etc/samba/smb.conf : Voir pastebin.com/EDxh7yqH

Le partage sur un domaine Microsoft

Voici un fichier **/etc/samba/smb.conf** permettant de rejoindre un domaine existant afin de partager des ressources :

```
 # PARTAGE PAR UTILISATEUR (server) - Domaine avec authentification
 # déléguée à un controleur de domaine (PDC : Primary Domain Controller)

 # Il y a 2 cas possibles :
 #
 # CAS n°1 : on partage une ressource totalement anonyme
 # CAS n°2 : cas normal du mode user, on fait une authentification
 #           par login et mot de passe


 # Section GLOBAL
 # Configuration globale de Samba
 [global]
       # Nom du groupe de travail
       workgroup = DOMAINE

       # Nom Netbios de la machine (identification réseau)
       netbios name = PC230

       # Chaine de commentaire associé au serveur (voisinage réseau)
```

```
	server string = %h Serveur (Samba %v)

	# Utilisateurs interdits
	invalid users = root

	# Enregistre un fichier de log par machine cliente du réseau MS
	log file = /var/log/samba/log.%m

	# Taille maximale des logs : 1 Mo
	max log size = 1000

	# On n'utilise pas Syslog pour enregistrer les logs
	syslog = 0

	# On fait un partage par utilisateur, l'authentification est
	# déléguée au controleur de domaine (PDC)
	security = server
	password server = PDC

	# On utilise les mots de passe encryptés
	# (attention, W95 et W98a fonctionnent avec les mdp en clair)
	encrypt passwords = true

	# Accélère les transferts réseaux
	socket options = TCP_NODELAY

	# Empeche nmbd de chercher à résoudre le nom netbios via le DNS
	dns proxy = no

	# Nom du compte invité qui va permettre de créer une
	# ressource partagée par mot de passe (celui du compte invite)
	# POUR LE CAS n°1
	guest account = invite

	# Emplacement du fichier contenant les logins et mdp samba
	# concerne le CAS n°1 et n°2
	smb passwd file = /etc/samba/smbpasswd


# Section HOMES
# Cette section particulière permet de remonter une ressource
# qui correspond au répertoire de travail (home directory)
# de l'utilisateur qui s'est authentifié
[homes]
	# Correspond au home directory de l'utilisateur authentifié
	# %U sera remplacé par le nom d'utilisateur (login)
	comment = Le répertoire personnel de %U

	# Si on active cette option, on dispose d'une ressource 'homes'
	# redondante avec la ressource 'login'
	browseable = no

	# Il faut que l'utilisateur soit authentifié pour accéder
	# à la ressource (son home directory sur le serveur)
	public = no

	# Autorise l'écriture dans le répertoire personnel
	writable = yes

	# Droits d'accés des fichiers et répertoires crées
	create mask = 0644
	directory mask = 0755


# Section PRINTERS
# Cette section permet de partager les imprimantes définies sur le serveur
[printers]
	# Le commentaire associé à l'imprimante
	comment = Les imprimantes

	# Affiche les imprimantes dans la liste des partages du serveur
	browseable = yes

	# Dans le cas d'une imprimante, c'est l'emplacement
	# des fichiers temporaires
	path = /tmp

	# Spécifie qu'il s'agit d'une imprimante et non un répertoire
	printable = yes

	# Partage l'imprimante de manière anonyme
	public = yes
```

```
      # Logique
      writable = no

      # Empeche d'autres utilisateurs de supprimer mes impressions
      create mode = 0700


# Sections REPERTOIRES PARTAGES

# CAS n° 1 : une ressource anonyme sans mot de passe
# Un CDROM
[cdrom]
      # Le commentaire associé au CDROM
      comment = Le CDROM

      # Point de montage du CDROM
      # synonyme de directory =
      path = /cdrom

      # Empeche le bloquage d'un fichier par un utilisateur
      locking = no

      # Logique
      writable = no

      # Partage le CDROM de manière anonyme
      public = yes


# CAS n°2 : un répertoire partagé qu'à certains utilisateurs
[prive]
      # Le commentaire associé au répertoire
      comment = Un répertoire partagé qu'à certains utilisateurs (CAS n°2)

      # Emplacement du répertoire partagé
      path = /home/prive

      # Support en lecture / écriture
      # synonyme de read only = no
      writable = yes

      # Partage anonyme désactivé (cas par défaut)
      # synonyme de guest ok = no
      public = no

      # Voir explication ci-dessous
      # le @prive désigne le groupe Unix 'prive'
      valid users = @prive

      # Masque de création des fichiers et répertoires
      create mask = 0644
      directory mask = 0755
```

Remarque : Pour que le serveur Linux puisse rejoindre le domaine existant, il faut se connecter au moins une fois avec le compte **Administrateur** défini sur le contrôleur de domaine, afin que ce dernier crée un compte machine :

```
# net join -U Administrateur
Administrateur's password:
[2008/02/12 11:58:41, 0] utils/net_ads.c:ads_startup(289)
ads_connect: Chaîne multi-octets ou étendue de caractères invalide ou incomplète
ADS join did not work, falling back to RPC...
Joined domain DOMAINE.
```

Dans le cas du mode server, les utilisateurs (login et mot de passe) sont définis sur un serveur existant.

Coté Linux, les utilisateurs doivent avoir le droit d'écriture sur le répertoire partagé. Cependant, les utilisateurs ne sont pas définis sur le serveur Linux mais sur le contrôleur de domaine. La solution consiste à permettre à tout le monde d'écrire dans ce répertoire, Samba se chargeant de n'autoriser que les utilisateurs autorisés sur le contrôleur de domaine.

```
# mkdir /home/prive
# chmod 777 /home/prive
```

Samba en contrôleur de domaine Microsoft

Voici un fichier **/etc/samba/smb.conf** permettant de faire un contrôleur de domaine Microsoft :

```
# Controleur de domaine PDC (Primary Domain Controller)

# Section GLOBAL
# Configuration globale de Samba
[global]
      # Nom du groupe de travail
      workgroup = DOMAINE

      # Nom Netbios de la machine (identification réseau)
      netbios name = PC230

      # Chaine de commentaire associé au serveur (voisinage réseau)
      server string = %h Serveur (Samba %v)

      # Utilisateurs interdits
      invalid users = root

      # Enregistre un fichier de log par machine cliente du réseau MS
      log file = /var/log/samba/log.%m

      # Taille maximale des logs : 1 Mo
      max log size = 1000

      # On n'utilise pas Syslog pour enregistrer les logs
      syslog = 0

      # On est PDC, on fait l'authentification par utilisateur
      security = user

      # On utilise les mots de passe encryptés
      # (attention, W95 et W98a fonctionnent avec les mdp en clair)
      encrypt passwords = true

      # Accélère les transferts réseaux
      socket options = TCP_NODELAY

      # Empeche nmbd de chercher à résoudre le nom netbios via le DNS
      dns proxy = no

      # Nom du compte invité qui va permettre de créer une
      # ressource partagée par mot de passe (celui du compte invite)
      guest account = invite

      # Emplacement du fichier contenant les logins et mdp samba
      smb passwd file = /etc/samba/smbpasswd

      # Permet de devenir le master browser du réseau
      # Le master browser est responsable de l'état du réseau
      # C'est en général le PDC qui effectue cette tache
      local master = yes

      # Permet de devenir le DOMAIN master browser du réseau
      domain master = yes

      # Cette option permet de déclencher une election sur le réseau
      # afin de déterminer qui sera le master browser
      # Le fait de déclencher cette election me donne le plus de chance
      # devenir ce master browser
      preferred master = yes

      # Si ce chiffre est le plus grand du réseau, je deviens le PDC
      os level = 255

      # Permet aux utilisateurs de changer leur mot de passe sur ce serveur
      # depuis leur poste
      update encrypted = yes

      # Définit que le daemon nmbd doit agir en tant que serveur WINS
      wins support = yes

      # Winbind permet de placer des machines dans un domaine controlé
      # par un PDC et d'autoriser les utilisateurs a accéder
      # à ces machines en utilisant les informations du PDC
      winbind separator = +
      winbind enum users = no
      winbind enum groups = no
      winbind uid = 10000-20000
      winbind gid = 10000-20000

      # désactive le support des ACL (Access Control List)
      # fonctionnalité utilisée par Active Directory
      nt acl support = no
```

```
      # Logins autorisés à se connecter aux ressources administratives
      # comme C$, ADMIN$ et IPC$
      admin users = admin

      # PRIMORDIAL !, sinon ca marche pas :
      # il faut que la ressource [netlogon] existe, meme si on n'utilise
      # pas un script logon
      domain logons = yes

      # Nom du fichier netlogon
      logon script = logon.bat

      # Pour autoriser les utilisateurs a changer leurs mots de passe
      passwd program = /usr/bin/passwd %u
      passwd chat = *Enter\snew\sUnix\spassword:* %n\n *Retype\snew\sUnix\spassword:* %n\n

# Section NETLOGON
# Permet de faire exécuter un script d'initialisation réseau par le poste
# client du domaine. Ceci permet par exemple connecter un lecteur réseau (Z:) ou de synchroniser l'heure
[netlogon]
      # Il n'apparaitra pas dans la liste des ressources partagées
      browsable = no

      # Répertoire contenant le netlogon
      path = /etc/samba/netlogon

      # Le netlogon n'a pas besoin d'etre public
      public = no

      # Pour la sécurité, cette ressource est exportée en read only
      writable = no

      # Permet à plusieurs utilisateurs d'utiliser le meme fichier
      # en meme temps
      locking = no

### Ce serveur effectuant la mission cruciale de PDC, il n'est pas
### recommandé de partager des ressources, bien que cela soit possible

[homes]
      path = %H
      browsable = no
      writable = yes
      public = no

[partage]
      path = /home/partage
      public = yes
      browsable = yes


[profiles]
      comment = Partage pour stocker les profils itinérants
      path =  /home/samba/profiles
      writable = yes
      browseable = no
      create mode = 0644
      directory mode = 0755
```

On crée ensuite le compte de l'utilisateur **admin**. il faudra utiliser ce compte lors de la première connexion d'un ordinateur au domaine.

```
# adduser --shell /bin/false --disabled-login admin
# smbpasswd -a admin
```

NB: Sur une Ubuntu, le groupe **admin** existe déjà, donc, soit on rajoute l'option **--ingroup users**, soit on utilise un compte différent pour l'administration (ex: Administrateur).

NB2 : dans les versions récentes de Samba, il faut utiliser la commande **pdbedit** à la place de **smbpasswd** :

```
# pdbedit -a admin
```

Pour chaque membre du domaine, il faut créer un compte unix correspondant à la machine et rajouter une entrée de type machine dans smbpasswd.

Attention : ce login doit porter le nom netbios de la machine et se terminer par un dollar (d'ou le --force-badname) :

```
# adduser --shell /bin/false --disabled-login --force-badname nompc$
```

```
# smbpasswd -a -m nompc$       # ou pdbedit -a -m nompc$
```

NB : On peut automatiser la création des comptes machines en rajoutant la ligne suivante dans la section **[global]** de smb.conf :

Sous Debian :

```
add machine script = /usr/sbin/adduser --shell /bin/false --disabled-login --force-badname %u
```

Sous Redhat :

```
add machine script = /usr/sbin/useradd -d /var/lib/nobody -g 100 -s /bin/false -M %u
```

Il faut aussi créer le répertoire pouvant accueillir le **netlogon** :

```
# mkdir /etc/samba/netlogon
```

On crée également le répertoire stockant les profils itinérants :

```
# mkdir -p /home/samba/profiles
# chmod 777 /home/samba/profiles
```

Le **netlogon** peut servir par exemple à créer automatiquement un volume réseau (ex: **Z:**), ou ajuster l'heure de l'ordinateur avec celle du contrôleur de domaine.

On peut optionnellement créer un fichier logon.bat :

```
# echo '@echo off' > /etc/samba/netlogon/logon.bat
# echo 'echo Bienvenue sur le domaine' >> /etc/samba/netlogon/logon.bat
# echo 'net use T: \\PC230\partage' >> /etc/samba/netlogon/logon.bat
# echo 'net time \\PC230 /set /yes' >> /etc/samba/netlogon/logon.bat
# echo 'pause' >> /etc/samba/netlogon/logon.bat
```

Il faut ensuite relancer samba et **winbind** :

```
# /etc/init.d/samba restart
# /etc/init.d/winbind restart
```

Utilisation de smbclient

smbclient est un client pour les réseaux samba.

Exemple :

```
>smbclient -L PC230
Password:
Domain=[WORKGROUP] OS=[Unix] Server=[Samba 3.0.24]

      Sharename       Type      Comment
      ---------       ----      -------
      cdrom           Disk      Le CDROM
      partage         Disk      Un rÃ©pertoire partagÃ© mais protÃ©gÃ© par mot de passe
      IPC$            IPC       IPC Service (pc230 Serveur (Samba 3.0.24))
      HP_LaserJet_1200_LPT_parport0_HPLIP Printer   HP LaserJet 1200

Domain=[WORKGROUP] OS=[Unix] Server=[Samba 3.0.24]

      Server               Comment
      ---------            -------
      BLACKPEARL
      CHRIS-B48CAF99F
      D
      DEBIANTOINE          Debiantoine server
      PC230                pc230 Serveur (Samba 3.0.24)
      PCPROF-LDKBQYFP
      Workgroup            Master
      ---------            -------
      ATELIER              BOOWISS-MOBIL
      CIS                  KALMIYA
      HOUCINE              ORDI-PARIS02
      KOOKYOO.LAN          KOOKYNUX
```

```
      LECHAT              CREUFOP12
      NONOLEROBOT         SERVERPARIS01
      PC237'S DOMAIN      PC237
      WORKGROUP           DEBIANTOINE
```

Monter un répertoire réseau

Manuellement :

```
# mount -t ntfs -o username=invite,password=invite //PC230/partage /mnt
# ...
# umount /mnt
```

Automatique à chaque démarrage :

Rajouter dans **/etc/fstab** :

```
# <file system> <mount point> <type> <options> <dump> <pass>
...
//PC230/partage /mnt smbfs username=invite,passwd=invite 0 0
```

Utilisation de SWAT : Samba Web Administration Tools

SWAT est une page web permettant de configurer samba de façon plus "simple", mais comme souvent avec les interfaces graphiques, de manière moins pointue.

Pour les amateurs du clickodrôme:

```
# apt-get install swat
```

<Image de SWAT>

Une fois installé, on accède à SWAT avec un navigateur sur l'adresse localhost:901 Swat va lire le fichier smb.conf et le présenter dans une page html. Les onglets permettent de configurer les partages, les imprimantes, de voir les docs etc... Par défaut, l'accès distant à SWAT est interdit, mais il est activable. Toutefois, il est déconseillé d'utiliser cette option.

Le partage de fichiers NFS

Le protocole NFS (Network File System) permet de partager des fichiers dans les réseaux Unix.

Installation du serveur NFS

Pour installer le serveur NFS sous Debian, il suffit de taper la commande suivante :

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

Configuration du serveur NFS

La configuration du serveur NFS est stockée dans le fichier **/etc/exports**.

Ce fichier de configuration dispose d'une page de manuel :

```
# man exports
```

Voici quelques exemples extrait du manpage :

```
$ cat /etc/exports

# fichier exemple /etc/exports
/               master(rw) trusty(rw,no_root_squash)
/projects       proj*.local.domain(rw)
/usr            *.local.domain(ro) @trusted(rw)
/home/joe       pc001(rw,all_squash,anonuid=150,anongid=100)
/pub            (ro,insecure,all_squash)
```

- La première ligne exporte l'ensemble du système de fichiers vers les machines « master » et « trusty ». En plus des droits d'écriture, toute transformation d'UID est désactivée pour l'hôte « trusty ».
- Les deuxième et troisième lignes montrent des exemples de noms de machines avec caractères jokers, et de groupes de machines (« @trusted »).
- La quatrième ligne montre comment mapper tous les utilisateurs vers UID et GID particulier.
- La dernière ligne partage un répertoire **/pub**, à toutes les machines dans le monde, en effectuant les requêtes sous le compte anonyme. L'option insecure permet l'accès aux clients dont l'implémentation NFS n'utilise pas un port réservé.

A chaque modification du fichier **/etc/exports**, il faut relancer le serveur NFS pour que les modifications soient prises en compte :

```
# /etc/init.d/nfs-kernel-server restart
```

On peut également utiliser la commande **exportfs** :

```
# exportfs -ra
```

Options d'exportation

Voici les options d'exportation les plus courantes :

- **secure** : cette option impose l'utilisation d'un port réservé (inférieur à 1024) comme origine de la requête.
- **rw** : exporte le répertoire en lecture / écriture
- **ro** : exporte le répertoire en lecture seule
- **async** : le serveur NFS va pouvoir répondre que le fichier a été écrit sur le support de stockage, même si cela n'a pas encore été fait. Améliore les performances du serveur.
- **sync** : le serveur NFS va écrire physiquement les fichiers sur le support de stockage avant de répondre. Réduit les performances du serveur.

Options liées aux correspondances de UID et de GID (UID et GID mapping)

Le principal problème avec NFS est la correspondance des UID et des GID. Effectivement, l'utilisateur **alex** peut avoir le UID 1000 sur le client et un UID différent sur le

serveur. NFS travaille avec les UID et GID numérique, il va donc par défaut enregistrer sur le serveur que le fichier appartient à l'utilisateur 1000, et non à l'utilisateur alex.

Pour pallier ces problèmes, NFS fournit des mécanismes pour transformer les UID et les GID.

Le problème se pose également avec le super-utilisateur **root** qui dispose du UID 0. Pour des raisons de sécurité, NFS va transformer par défaut les fichiers posés par le root vers le UID et le GID du compte anonyme (nobody.nogroup).

- **root_squash** : option par défaut. transforme les requètes provenant de l'UID 0 / GID 0 vers le UID et GID du compte anonyme.
- **no_root_squash** : ne transforme pas les requètes provenant de l'UID 0 / GID 0. A utiliser avec précaution.
- **all_squash** : transforme tous les UID/GID vers le UID/GID de l'utilisateur anonyme.
- **anonuid** : permet de spécifier le UID de l'utilisateur anonyme.
- **anongid** : permet de spécifier le GID de l'utilisateur anonyme.

Voir les répertoires exportés

```
# showmount
192.168.30.10:/home
192.168.30.12:/home
...
```

Utilisation de NFS depuis un poste client

La commande **mount** permet à un poste client d'accéder à un répertoire partagé par un serveur NFS.

Synopsis :

```
# mount -t nfs <nom ou adresse IP du serveur NFS>:<répertoire partagé> <point de montage>
```

Exemple :

```
# mount -t nfs pc230:/home/partage /mnt
```

Si on veut que ce répertoire soit accessible à chaque boot, il suffit de rajouter la ligne suivante dans le fichier /etc/fstab :

```
$ cat /etc/fstab
...
pc230:/home/partage  /mnt  nfs  defaults  0  0
...
```

Authentification centralisée avec NIS

Pour résoudre le problème des UID / GID non identiques, la solution consiste à déployer NIS.

Pour installer NIS sous Debian, utiliser la commande suivante :

```
# apt-get install nis
```

Lors de l'installation il faut indiquer le domaine NIS, ce dernier peut être différent du domaine internet.

Pour le déploiement du serveur et des clients NIS, on va suivre la procédure Debian décrite dans le fichier **/usr/share/doc/nis/nis.debian.howto.gz** :

```
# zmore /usr/share/doc/nis/nis.debian.howto.gz
```

Configuration du serveur NIS

Il faut tout d'abord vérifier que le fichier **/etc/hosts** contient l'adresse IP et le nom FQDN du serveur :

```
# cat /etc/hosts
...
192.168.30.230  pc230.mondomaine.fr pc230
```

Il faut ensuite vérifier que le fichier **/etc/defaultdomain** contienne le domaine NIS :

```
# cat /etc/defaultdomain
DOMAINENIS
```

Dans le fichier **/etc/ypserv.securenets**, on restreint l'utilisation du domaine NIS au domaine du réseau local. On remplace la ligne **0.0.0.0 0.0.0.0** par : **255.255.255.0 192.168.30.0**.

```
# vi /etc/ypserv.securenets
...
255.255.255.0   192.168.30.0
```

On modifie ensuite **/etc/default/nis** pour indiquer qu'il s'agit du serveur NIS :

```
# vi /etc/default/nis
...
NISSERVER=master
...
```

On relance le serveur NIS :

```
# /etc/init.d/nis restart
```

Cette étape peut prendre un certain temps ...

On lance ensuite la création des bases de données NIS avec la commande suivante :

```
# /usr/lib/yp/ypinit -m
```

Ceci va créer les fichiers partagés dans le répertoire **/var/yp/DOMAINENIS**.

Configuration du client NIS

On installe nis (c'est le même paquet qui fait office de client et de serveur) :

```
# apt-get install nis
```

On vérifie que le fichier **/etc/defaultdomain** contienne le domaine NIS :

```
# cat /etc/defaultdomain
DOMAINENIS
```

Dans le fichier **/etc/yp.conf**, on indique l'adresse IP du serveur NIS :

```
# vi /etc/yp.conf
...
ypserver 192.168.30.230
```

On démarre le client NIS :

```
# /etc/init.d/nis restart
```

On vérifie que le fichier **/etc/nsswitch.conf** contient bien les lignes suivantes :

```
# cat /etc/nsswitch.conf
...
passwd:         files nis
group:          files nis
shadow:         files nis
...
```

```
netgroup:       nis
```

L'option **files** indique au système d'aller voir les fichiers classiques /etc/passwd, /etc/group et /etc/shadow.

L'option **nis** indique au système d'aller ensuite interroger le serveur NIS.

A la fin du fichier **/etc/passwd**, on rajoute la ligne suivante :

```
# echo "+::::::" >> /etc/passwd
```

A la fin du fichier **/etc/shadow**, on rajoute la ligne suivante :

```
# echo "+::::::::" >> /etc/shadow
```

A la fin du fichier **/etc/group**, on rajoute la ligne suivante :

```
# echo "+:::" >> /etc/group
```

Pour tester, on lance la commande suivante :

```
# ypcat passwd.byname
```

On doit voir afficher le fichier **/etc/passwd** partagé.

Sur le serveur NIS, on rajoute un utilisateur :

```
# adduser jack
...
```

Si tout fonctionne, on doit pouvoir se logger sur le client NIS avec l'utilisateur **jack**.

Le serveur d'impression CUPS

Introduction

CUPS (Common Unix Printing System) est un serveur d'impression. CUPS permet ainsi d'imprimer sur une imprimante accessible sur le réseau ou locale (généralement connectée avec un câble USB). La communication avec les imprimantes utilise le protocole IPP (Internet Printing Protocol).

CUPS est très simple à configurer. De ce fait, il est de plus en plus souvent utilisé en remplacement de LPR (premier serveur sous UNIX).

Site Internet : www.cups.org (http://www.cups.org)

Le site linuxprinting.org (http://www.linuxprinting.org) liste les imprimantes compatibles avec CUPS.

Généralement, un système linux moderne est désormais livré avec les utilitaires nécessaire pour détecter, configurer et utiliser des imprimantes.

Installation (Debian)

L'installation initiale de CUPS sous Debian peut se faire avec la commande suivante :

```
$ sudo apt install cups cups-pdf cups-bsd foomatic-db-compressed-ppds openprinting-ppds
```

Le paquet `cups-pdf` fournit une imprimante virtuelle qui permet de générer des fichiers PDF.

Le paquet `cups-bsd` fournit les commandes traditionnelles `lpr`, `lpq` et `lprm`.

Les paquets `foomatic-db-compressed-ppds` et `openprinting-ppds` contiennent des descriptions PPD (PostScript Printer Description) pour un très grand nombre d'imprimantes.

D'autres paquets de PPD sont également disponibles comme `hp-ppd` ou `hpijs-ppds`. Mais aussi le meta-paquet `printer-driver-all-enforce` qui englobe une vingtaine de paquets de différents drivers CUPS.

Configuration côté serveur

CUPS se configure via une interface web accessible par l'adresse http://localhost:631/ .

- On Clique sur **Add Printer** pour ajouter une imprimante
- On spécifie son nom, sa localisation physique et sa description. Exemple : HP1200, Salle D11, L'imprimante laser de la salle D11
- On choisit ensuite le périphérique correspondant au mode de connexion de l'imprimante avec le PC (ex: port parallèle, USB, ipp, http...etc)
- Si l'imprimante est partagée par un autre serveur CUPS, il faut spécifier l'adresse IPP de l'imprimante, comme par exemple : **ipp://192.168.30.210/printers/hp1200**
- On choisit ensuite le constructeur de l'imprimante. Dans le cas où votre constructeur n'est pas dans la liste, il faut télécharger sur Internet le fichier PPD (Printer Postscript Description) correspondant et utiliser le bouton **Parcourir** pour uploader le fichier
- On choisit ensuite le driver correspondant au modèle de l'imprimante. Certains drivers sont les mêmes pour des imprimantes différentes. Si on ne trouve pas son imprimante, il faut aller sur www.linuxprinting.org (http://www.linuxprinting.org) pour connaitre la compatibilité de l'imprimante avec un autre driver
- Pour valider les modifications, on s'authentifie avec le compte **root** pour que le logiciel puisse écrire le fichier de configuration de CUPS

Fichier de configuration

Le fichier principal de configuration de CUPS est **/etc/cups/cupsd.conf**. Il faut le modifier pour indiquer à quel réseau on partage les imprimantes.

Tout d'abord, on indique à CUPS d'écouter sur toutes les interfaces réseaux. On remplace **Listen localhost:631** par :

```
Listen *:631
```

Dans la balise <Location />, il faut rajouter "Allow" suivi de l'adresse du réseau sur lequel se trouve l'imprimante afin de permettre à mon réseau d'accéder à celle-ci.

```
# Restrict access to the server...
```

```
<Location />
  Order allow,deny
  Allow localhost
  Allow 192.168.30.0/24
</Location>
```

Lorsque ce fichier est modifié, il faut relancer le service :

```
# /etc/init.d/cupsys restart
```

Configuration côté client / utilisateur

Interface(s) graphiques

Des outils d'interface graphique permettant d'accéder au système CUPS existent :

- gtklp (Gtk/Gnome)
- system-config-printer-gnome (Gtk/Gnome)

Interface web

Comme pour la configuration côté serveur, on configure CUPS côté client via l'interface WEB.

Dans la procédure décrite ci dessus, au lieu de spécifier un périphérique local (port parallèle, USB ou autre), on spécifie l'adresse IPP de l'imprimante.

Exemple:

```
ipp://192.168.30.210/printers/HP1200
```

ou (sous Windows) :

```
http://192.168.30.210:631/printers/HP1200
```

Impression

Les imprimantes sont normalement utilisables via les applications qui ont besoin d'imprimer (menu «Imprimer» !) : suite bureautique, navigateur internet...

Néanmoins, il est possible de les utiliser sans passer par une application, et même de contrôler directement l'état des imprimantes, en utilisant des commandes spécifiques du système.

Les commandes d'impression

Ces commandes sont fournies par le paquet **cupsys-bsd** installé précédement.

La commande lpr

permet d'imprimer un fichier :

```
$ lpr nom_fichier
```

Pour imprimer sur une imprimante en particulier :

```
$ lpr -P nom_imprimante nom_fichier
```

Pour spécifier le nombre de copies (ici 2 copies) :

```
$ lpr -#2 nom_fichier
```

La commande lpq

permet d'afficher la file d'impression (queue) et connaître les numéros de "job" :

```
$ lpq
HP1200 is ready and printing
```

```
Rank    Owner   Job     File(s)                         Total Size
active  alex    12      fstab                           1024 bytes
1st     root    13      fstab                           1024 bytes
```

L'imprimante HP1200 est opérationnelle et en cours d'impression. C'est l'impression de l'utilisateur alex qui est en cours.
L'impression de l'utilisateur root est en première position de la file d'attente.

Pour voir la file d'attente d'une imprimante en particulier :

```
$ lpq -P nom_imprimante
```

La commande lprm

(lp remove) permet de supprimer une impression de la file d'attente à l'aide de son numéro de "job" :

```
$ lprm num_job
```

Pour supprimer toute la file d'attente (cette action n'est permise que pour le compte root) :

```
# lprm -a
```

La commande lpstat

(lp statistiques) permet d'obtenir diverses statistiques

- La liste des imprimantes installées et l'imprimante par défaut

```
$ lpstat -s
```

- etc...

Supervision

La supervision s'intéresse au suivi du bon fonctionnement et à la détection (et correction!) des problèmes.

Logs

Le premier outil de supervision est l'examen des fichiers «journaux» (logs) de fonctionnement du système CUPS. Ces fichiers sont localisés dans le répertoire

```
/var/log/cups
```

On y trouve trois type d'enregistrements

access.log : enregistrement des accès d'impressions
error.log : enregistrement des erreurs
page.log : enregistrement des pages pour statistiques

Pou obtenir le maximum d'information, il faut positionner le mode Debug dans le fichier de configuration du serveur CUPS

```
LogLevel debug
```

Une surveillance simple dans un terminal :

```
sudo tail -f /var/log/cups/*_log
```

Le serveur de fichiers FTP

Introduction

Le protocole FTP (File Transfer Protocol) permet d'échanger des fichiers volumineux. Il utilise le port 21 (canal de contrôle) et le port 20 (canal des données) en protocole TCP.

Il existe plusieurs serveurs FTP : ProFTPD, Pure-FTPd, VsFTPd, wu-ftpd, muddleftpd... Chacun se différencie des autres par ses fonctionnalités et sa facilité (ou non) a être configuré.

Généralement ces serveurs peuvent également fournir un serveur SFTP (FTP sécurisé, sur le port 22).

ProFTPD

*Pour plus de détails voir : **Administration réseau sous Linux/ProFTPD**.*

- Installation :

```
# apt-get install proftpd
```

- Configuration :

```
# vim /etc/proftpd/proftpd.conf
```

Une fois les modifications faites sur le fichier de configuration, ne pas oublier de relancer le service :

```
# /etc/init.d/proftpd restart
```

Exemple de configuration :

```
# Nom du serveur
ServerName                      "Serveur FTP du Creufop"

# Fonctionne en mode autonome (daemon) et non via inetd
ServerType                      standalone

# Affiche le message de bienvenue que lorsque l'utilisateur s'est authentifié
DeferWelcome                    on

# Permet de voir les liens symboliques
ShowSymlinks                    on

# a commenter
MultilineRFC2228                on
DefaultServer                   on

# Efface un fichier si il est déjà présent
AllowOverwrite                  on

# Options liés aux expirations avant déconnexion
TimeoutNoTransfer               600
TimeoutStalled                  600
TimeoutIdle                     1200

# Fichier à afficher lors de la connexion
DisplayLogin                    welcome.msg

# Affiche le fichier .message si celui-ci est présent dans le répertoire visité
DisplayFirstChdir               .message

# Affiche par défaut le format long de la commande ls
ListOptions                "-l"

# A commenter
DenyFilter                      \*.*/

# A décommenter si on utilise NIS ou LDAP
#PersistentPasswd               off

# Le port standard de contrôle
Port                            21
```

```
# Nombre maximum de connexions simultannées
MaxInstances                    30

# Login et groupe utilisé par le daemon proftpd
User                            nobody
Group                           nogroup

# Masque par défaut des fichiers et des répertoires
Umask                          022  022

# On autorise explicitement les logins autorises au FTP :
<Limit LOGIN>
      AllowUser alex
      AllowUser pierre
      AllowUser alain
      DenyAll
</Limit>
```

Le serveur Web Apache

Installation de Apache2

LAMP

Logiciel tout-en-un pour Linux (Apache + MySQL + PHP), comme WAMP pour Windows.

commande nécessitant les privilèges root

```
# apt-get install tasksel
# tasksel install lamp-server
```

Installation manuelle

Apache sur Debian / Ubuntu

commande nécessitant les privilèges root

```
# apt-get install apache2
```

Le service peut ne pas être lancé par défaut, mais même s'il l'est on peut quand-même essayer de l'activer avec :

commande nécessitant les privilèges root

```
# /etc/init.d/apache2 start
```

On peut ensuite tester le serveur, pour voir si une page s'affiche ou s'il refuse la connexion :

commande

```
$ lynx http://localhost/
```

Cette adresse est le rebouclage, elle peut aussi être rentrée directement dans tout navigateur web.

Si Apache était déjà installé vérifier le fichier pour indiquer le démarrage automatique d'Apache 2 **/etc/default/apache2** :

```
# vi /etc/default/apache2
 ...
NO_START=0
```

PHP

PHP peut-être installé avec toutes les déclinaisons de la distribution Debian (stable, testing, unstable). Il suffit pour cela d'insérer vos lignes préférées dans le fichier */etc/apt/sources.list* :

```
deb http://ftp.fr.debian.org/debian/ stable main non-free contrib
deb-src http://ftp.fr.debian.org/debian/ stable main non-free contrib
```

Ce qui suit suppose que le serveur Web a bien été installé : exécuter les commandes suivantes :

```
sudo apt-get update && apt-get install php7.0 && apt-get install libapache2-mod-php7.0
```

Une fois ces commandes exécutées, redémarrer le serveur Web. Dans le cas d'Apache cela s'effectue avec la commande suivante :

```
/etc/init.d/apache2 restart
```

Si tout s'est bien passé, vous disposez maintenant d'un serveur Web qui a la capacité d'exécuter des scripts PHP dans votre navigateur.

Testons :

commande

```
$ lynx http://localhost/test.php
```

Pour débugger :

commande

```
$ tail /var/log/apache2/error.log
```

Mise à jour

Pour la v7.2 :

```
sudo add-apt-repository ppa:ondrej/php
sudo apt update
sudo apt install php7.2 php7.2-common php7.2-cli php7.2-fpm
sudo a2enmod php7.2
sudo a2dismod php7.0
```

Attention !

Une fois les serveurs Web installés, ils se lancent automatiquement à chaque démarrage de la machine, ce qui est souhaitable pour un serveur, mais pas toujours pour un PC. Pour éviter cela, il suffit d'y désactiver les daemons :

```
sudo update-rc.d apache2 disable
sudo update-rc.d mysql disable
```

Apache sur Gentoo

Premièrement il faut installer Apache si ce n'est pas déjà fait :

```
emerge apache
```

Ensuite, il faut installer PHP :

```
emerge dev-lang/php
```

Puis il faut qu'apache utilise PHP dans sa configuration.

Code: Configuration de apache

```
# nano -w /etc/conf.d/apache2
APACHE2_OPTS="-D PHP5"
```

MySQL seul

MySQL est disponible sur http://dev.mysql.com/downloads/gui-tools/5.0.html au format :

1. .msi (Windows)
2. .dmg (Mac)
3. .rpm (Linux)
4. .tar

En l'absence de gestionnaire de paquets, utiliser le .tar ainsi :

```
shell> groupadd mysql
shell> useradd -r -g mysql mysql
shell> cd /usr/local
shell> tar zxvf /path/to/mysql-VERSION-OS.tar.gz
shell> ln -s full-path-to-mysql-VERSION-OS mysql
```

```
shell> cd mysql
shell> chown -R mysql .
shell> chgrp -R mysql .
shell> scripts/mysql_install_db --user=mysql
shell> chown -R root .
shell> chown -R mysql data
shell> bin/mysqld_safe --user=mysql &
```

APT

```
$ sudo apt-get install mysql-server mysql_secure_installation
```

Puis, modifier PHP pour qu'il supporte MySQL :

```
$ sudo apt-get install php4-mysql
```

Variante

La dénomination des paquets mentionnés peut varier légèrement selon la version. Dans un terminal, entrez :

```
$ sudo apt-get install mysql-server
```

et confirmez.

(Remarque : il semblerait qu'en installant le paquet "mysql-server-5.0", au lieu du paquet mentionné plus haut, certaines personnes rencontrent des problèmes. Il est donc préférable d'installer ce paquet, ou d'installer la dernière version 4 stable avec : $ sudo apt-get install mysql-server-4.1. Consultez le forum pour plus d'informations : [1] (http://forum.ubuntu-fr.org/viewtopic.php?id=15352))

Lancez ensuite la commande :

```
cd && sudo mysql_secure_installation
```

Appuyez sur Entrée lorsqu'il vous demande le mot de passe root MySQL : pour le moment il n'y en a pas.

***NB :** MySQL a ses propres utilisateurs, avec leurs propres privilèges. Le root MySQL n'est donc pas le root système. Il est conseillé de ne pas mettre les mêmes mots de passes pour les utilisateurs MySQL et les utilisateur du système.*

Le script vous demande alors si vous voulez mettre un mot de passe pour l'utilisateur root. Répondez Y, et entrez (2 fois le nouveau mot de passe du root MySQL). Il vous pose ensuite une série de questions. Si vous ne savez pas quoi répondre, acceptez les choix par défaut en appuyant simplement sur Enter.

Votre serveur MySQL est prêt. Par défaut il se lance à chaque démarrage du système, si vous ne le souhaitez pas, il vous suffit de lancer :

```
$ sudo dpkg-reconfigure mysql-server
```

et de répondre "Non" à la question du démarrage systématique de MySQL.

Sur Gentoo

```
emerge mysql
```

Installer PhpMyAdmin

Depuis un tout-en-un, il suffit de créer un chemin accessible depuis le serveur Web :

```
sudo ln -s /usr/share/phpmyadmin /var/www/phpmyadmin
```

Sinon :

```
sudo apt-get install phpmyadmin php5
```

Installer Apache et PHP avec PhpMyAdmin

Grâce aux dépendances des paquets, cette opération peut se faire en une seule fois : *Remarque : Vérifiez que la case "Traiter les paquets recommandés comme des*

dépendances" soit cochée dans Synaptic, configuration, préférences.

```
$ sudo apt-get install phpmyadmin
```

Cela installera automatiquement apache2 + php + modules d'apache pour PHP et MySQL + PhpMyAdmin. Pour accéder à PhpMyAdmin, il faut se rendre à la page http://localhost/PhpMyAdmin.

Note : En cas de problème d'authentification (erreur 2002 notamment) installer le paquet mysql-server peut résoudre ce dernier.

Après l'installation, il vaut mieux modifier les droits d'accès de root, et ajouter un mot de passe pour un peu plus de sécurité. Pour cela, il faut se rendre à la page *privilèges* de PhpMyAdmin.

Remarque pour Ubuntu 5.04 (Hoary Hedgehog) : Afin que cette commande fonctionne il est nécessaire d'avoir effectué les modifications suivantes : dans /etc/apt/ éditer le fichier sources.list supprimer les # des lignes suivantes :

```
# deb http://fr.archive.ubuntu.com/ubuntu hoary universe
```

(cette ligne est dans certain cas '# deb http://archive.ubuntu.com/ubuntu/ hoary universe main restricted multiverse')

```
# deb-src http://fr.archive.ubuntu.com/ubuntu hoary universe
```

Pour la version d'Ubuntu 5.10 (Breezy), vous pouvez effectuer ces changements avec le gestionnaire de paquets synaptic (apt) : Système ---> Administration ---> Gestionnaire de paquets Synaptic

```
           Catégories ---> Dépôts ----> Ajouter et ensuite, sélectionner : maintenu par la communauté universe...
```

Lancer le chargement des nouvelles sources :

```
$ sudo apt-get update
```

Puis lancer l'installation de PhpMyAdmin comme décrit ci-dessus.

Extensions

Pour activer des modules complémentaires :

```
a2enmod Nom_du_module # passe dans /etc/apache2/mods-enabled/
```

Ex :

```
a2enmod rewrite
```

Pour les désactiver :

```
a2dismod Nom_du_module # passe dans /etc/apache2/mods-available/
```

Pour activer des sites :

```
a2ensite Nom_du_site # passe dans /etc/apache2/sites-enabled/
```

Pour les désactiver :

```
a2dissite Nom_du_site # passe dans /etc/apache2/sites-available/
```

Les extensions PHP nécessitent une autre commande. Ex :

```
phpenmod mbstring
```

Problème d'encodage d'Apache2

Si vous rencontrez un problème d'encodage des caractères de vos pages, par exemple les caractères accentués apparaissant sous la forme "" (<?>), c'est probablement

parce qu'Apache2 déclare dans les en-têtes HTTP qui accompagnent les pages visionnées un encodage par défaut en Unicode (UTF-8) :

```
Content-Type: text/html; charset=UTF-8
```

Tandis que les pages visionnées utilisent un autre encodage des caractères, comme par exemple Latin1 (ISO-8859-1). Même si vos documents indiquent le jeu de caractères utilisé, le paramètre donné par le serveur dans les en-têtes HTTP est prioritaire !

Pour corriger ce problème, il faudra éditer /etc/apache2/apache2.conf :

```
$ sudo gedit /etc/apache2/apache2.conf
```

Encodage par défaut en Latin1 (ISO-8859-1)

Cherchez la ligne suivante :

```
#AddDefaultCharset     ISO-8859-1
```

Décommentez-la en enlevant le # :

```
AddDefaultCharset      ISO-8859-1
```

Pour ceux qui ont la locale iso-8859-15 (sinon vous pouvez faire "sudo dpkg-reconfigure locales" pour l'ajouter) et qui désirent l'utiliser par défaut, ajoutez un 5 en fin de ligne :

```
AddDefaultCharset      ISO-8859-15
```

ainsi que la ligne suivante dans le paragraphe en-dessous :

```
AddCharset ISO-8859-15 .iso8859-15  .latin15 .fr
```

Il ne vous reste plus qu'à mettre "fr" en première position dans la ligne LanguagePriority (juste au-dessus), et à demander à apache de relire sa configuration :

```
$ sudo /etc/init.d/apache2 reload
```

Aucun encodage par défaut

Il est également possible de s'affranchir de tout encodage par défaut, de la manière suivante :

Cherchez la directive *AddDefaultCharset* :

```
AddDefaultCharset      ISO-8859-1
```

Remplacez l'attribut par la valeur *Off* :

```
AddDefaultCharset      Off
```

Là encore, on demandera à Apache de relire sa configuration :

```
$ sudo /etc/init.d/apache2 reload
```

Maintenant, les en-têtes HTTP ne contiendront plus d'indication d'encodage des caractères. Attention : il faudra alors que chaque page indique l'encodage utilisé, car s'en remettre à la détection automatique par les navigateurs peut s'avérer assez aléatoire !

Configuration de Apache2

Le fichier de configuration principal de Apache2 est **/etc/apache2/apache2.conf**. Dans le cas de Apache version 1.3, ce fichier s'appelle **/etc/apache/httpd.conf**.

Par défaut, Apache utilise deux fichiers de logs :

- **/var/log/apache2/access.log** : contient les logs de connexion au serveur Web
- **/var/log/apache2/error.log** : contient les erreurs survenues

Dans les fichiers de configuration **/etc/apache2/apache2.conf** ou **/etc/apache2/conf.d/security**, on peut être amené à modifier :

```
# Pour que Apache ne donne pas son numéro de version et les modules chargés (via une page d'erreur)
ServerSignature    off

# Pour que Apache ne donne pas son numéro de version et les modules chargés (via le protocole HTTP)
ServerTokens       Prod

# Permet d'enregistrer les noms canoniques au lieu des adresses IP
# dans le fichier access.log
HostnameLookups On
```

Le fichier **/etc/apache2/ports.conf** contient la liste des ports sur lequel Apache écoute. On peut changer le port le par défaut (80), ou lui indiquer d'écouter sur d'autres ports (ex: 443 pour https) en rajoutant des lignes **Listen <port>**.

```
# cat /etc/apache2/ports.conf
Listen 80
```

Héberger plusieurs sites Internet

Apache permet de gérer plusieurs sites internet sur le même serveur, pour cela on va utiliser des sections **VirtualHost**. Pour chaque site Internet, on crée un nouveau fichier dans **/etc/apache2/sites-enabled** et on lui indique la section VirtualHost (voir ci-dessous). Par convention, on appelle ce fichier avec le nom du site Internet :

```
# cd /etc/apache2/sites-enabled
# vi www.mondomaine.fr

# Ne mettre cette ligne que dans UN seul fichier
NameVirtualHost 192.168.30.220

<VirtualHost 192.168.30.220>
      ServerName www.mondomaine.fr
      ServerAlias mondomaine.fr
      ServerAdmin webmaster@mondomaine.fr
      DocumentRoot /var/www/www.mondomaine.fr
      CustomLog /var/log/apache2/www.mondomaine.fr_access.log combined
      ErrorLog /var/log/apache2/www.mondomaine.fr_error.log
</VirtualHost>
```

Les différents sites vont être stockés dans le répertoire **/var/www**.On va créer un sous-répertoire par site :

```
# cd /var/www
# mkdir www.mondomaine.fr
```

Afin de tester, on crée une page HTML minimale :

```
# echo "<html><body>www.mondomaine.fr</body></html>" > /var/www/www.mondomaine.fr/index.html
```

Après toutes modifications des fichiers de configuration, on relance Apache2 avec la commande suivante :

```
# /etc/init.d/apache2 restart
```

On peut tester avec son navigateur en se connectant à l'URL suivante :

```
http://www.mondomaine.fr
```

Dans cet exemple, il faut bien entendu que **www.mondomaine.fr** pingue vers notre serveur. On peut simuler ceci en rajoutant une entrée dans le fichier **/etc/hosts** ou en modifiant le DNS.

Installer des modules supplémentaires

Apache 2 permet l'installation de nombreux modules pour offrir plus de services ou pour renforcer la sécurité.

Les modules sont dans le répertoire **/etc/apache2/mods-available** . C'est ici qu'il faut copier les modules additionnels que vous pourrez télécharger pour Apache.

Pour activer un module on peut créer manuellement un lien symbolique dans le répertoire mods-enable

Par exemple pour activer le mod **URL Rewriting**

```
# ln -s /etc/apache2/mods-available/rewrite.load   /etc/apache2/mods-enabled/rewrite.load
```

Une alternative consite à utiliser la commande a2enable / a2disable

```
# a2enable mod_rewrite
// Active le mod_rewrite dans apache2
```

```
# a2disable mod_rewrite
// Désactive le mod_rewrite dans apache2
```

NB : sur les dernières versions de apache2, ces deux commandes s'appellent respectivement **a2enmod** et **a2dismod**.

Protéger un répertoire avec un login / mot de passe

Pour protéger un répertoire avec un login et un mot de passe, on crée dans le répertoire à protéger un fichier **.htaccess** qui contient le code suivant :

```
# cd /var/www/www.mondomaine.fr
# vi .htaccess

# Type d'authentification
AuthType Basic

# Nom affiché dans la boite d'authentification
AuthName "Site personnel"

# Emplacement du fichier contenant les utilisateurs
AuthUserFile /etc/apache2/utilisateurs

# Emplacement du fichier contenant les groupes
AuthGroupFile /etc/apache2/groupes

# On autorise seulement les membres du groupe admin
Require group admin
```

On crée ensuite le fichier **/etc/apache2/utilisateurs** à l'aide de la commande **htpasswd** :

```
# cd /etc/apache2
# htpasswd -c utilisateurs alex
...
# htpasswd utilisateurs pierre
```

La première fois, on utilise l'option **-c** de la commande **htpasswd** pour créer le fichier.

On crée ensuite le fichier **/etc/apache2/groupes** :

```
# echo "admin: alex pierre" > /etc/apache2/groupes
```

Il faut également éditer le fichier **/etc/apache2/sites-enabled/000-default** afin d'activer la prise en charge du fichier **htaccess**. Ajouter, si cela n'existe pas encore :

```
<Directory /var/www/mondossierprotege/>
AllowOverride All
</Directory>
```

On relance le serveur web :

```
# /etc/init.d/apache2 restart
```

Outils pour générer des statistiques de connexion

Chaque fois qu'un visiteur vient sur le site Internet, Apache enregistre sa connexion dans les fichiers /var/log/apache2/*_access.log.

Voici deux logiciels libres qui analysent ces fichiers et qui permettent de générer des statistiques sur les connexions :

- Awstats (http://awstats.sourceforge.net/)
  - Exemple de statistiques Awstats (http://www.nltechno.com/awstats/awstats.pl?config=destailleur.fr)
- Webalizer (http://www.mrunix.net/webalizer/)
  - Exemple de statistiques Webalizer (http://www.mrunix.net/webalizer/sample/index.html)
- Piwik

La base de données MySQL

Introduction

MySQL est un serveur de bases de données relationnelles SQL (Open Source), fonctionnant sur le port 3306 en mode TCP.

Prérequis : Apache.

Moteurs de stockage

Par défaut, MySQL utilise le moteur de stockage **MyISAM**.

Toutefois, MySQL propose d'autres moteurs de stockage :

- **InnoDB** : support des transactions (compatible ACID), des clés étrangères et de l'intégrité référentielle
- **MERGE** ou **MRG_MYISAM** : permet de fusionner plusieurs tables de structures identiques
- **BDB** : Berkeley DB est un format de stockage très répandu (utilisé par exemple par OpenLDAP) et supporte les transactions (compatible ACID)
- **HEAP** ou **MEMORY** : ces tables sont stockées en mémoire vive (RAM), elles sont très rapides mais ne survivent pas à un redémarrage de MySQL
- **CSV** : les données sont stockées dans des fichiers au format CSV (**C**omma **S**eparated **V**alues)
- **ARCHIVE** : stocke des informations en utilisant le minimum de place disque (perte de la notion d'index)
- **BLACKHOLE** : les données sont tout simplement envoyées vers /dev/null et donc perdues. Moteur utilisé pour faire des tests

Types de données

Chaque information stockée dans une table est définie par un type décrivant la nature de l'information enregistrée.

Les nombres

- TINYINT
- SMALLINT
- MEDIUMINT
- INT
- BIGINT
- FLOAT
- DOUBLE
- DECIMAL

Les chaînes de caractères

- CHAR
- VARCHAR
- BINARY
- VARBINARY
- BLOB
- TEXT
- ENUM
- SET

Les dates et heures

- DATE
- TIME
- DATETIME
- TIMESTAMP
- YEAR

Installation

Pour installer mysql sous Debian, on tape la commande suivante :

```
# apt-get install mysql-server mysql-client
```

Fichier de configuration

Le fichier de configuration est **/etc/mysql/my.cnf**.

Dans ce fichier de configuration on trouve plusieurs sections :

```
# Section du client Mysql
[client]
port            = 3306
socket          = /var/run/mysqld/mysqld.sock

# Section de mysqld_safe
[mysqld_safe]
socket          = /var/run/mysqld/mysqld.sock
nice            = 0


# Section du serveur Mysql
[mysqld]
# Utilisateur qui lance le daemon
user            = mysql

# Fichier qui contient le PID du processus
pid-file        = /var/run/mysqld/mysqld.pid

# Fichier socket qui permet une communication locale avec Mysqld (plus performant que de passer par le port 3306)
socket          = /var/run/mysqld/mysqld.sock

# Port sur lequel écoute Mysql
port            = 3306

# Répertoire de base de Mysql
basedir         = /usr

# Répertoire contenant les bases de données
datadir         = /var/lib/mysql

# Répertoire temporaire
tmpdir          = /tmp

# Permet de personnaliser le langage
language        = /usr/share/mysql/english

# Évite le blocage externe
skip-external-locking

# Adresse IP sur lequel écoute Mysql
bind-address            = 127.0.0.1

# Taille des caches Mysql : permet d'optimiser les performances de Mysql
key_buffer              = 16M
max_allowed_packet      = 16M
thread_stack            = 128K
thread_cache_size       = 8
#max_connections        = 100
#table_cache            = 64
#thread_concurrency     = 10
query_cache_limit       = 1M

# Emplacement du fichier de Log
log            = /var/log/mysql/mysql.log

# On peut logger les requêtes lentes
#log_slow_queries       = /var/log/mysql/mysql-slow.log
#long_query_time = 2
#log-queries-not-using-indexes

# Permet d'exécuter Mysql dans une cage chroot
# chroot = /var/lib/mysql/

# Options pour définir les certificats Mysql
# ssl-ca=/etc/mysql/cacert.pem
# ssl-cert=/etc/mysql/server-cert.pem
# ssl-key=/etc/mysql/server-key.pem


# Section pour mysqldump
[mysqldump]
```

```
quick
quote-names
max_allowed_packet      = 16M


# Section pour ?
[mysql]
#no-auto-rehash # faster start of mysql but no tab completition


# Section pour le format ISAM
[isamchk]
key_buffer              = 16M
```

Chaque fois que l'on modifie ce fichier, il faut relancer mysql :

```
# /etc/init.d/mysql restart
```

Les bases de données sont stockées dans le répertoire **/var/lib/mysql/**. Il faut donc sauvegarder ce répertoire.

Pour se connecter à Mysql à partir du shell, on utilise le client Mysql :

```
# mysql -h <host> -u <login> -p <password>
```

Commandes SQL d'importation / exportation

LOAD DATA INFILE

La commande SQL **LOAD DATA INFILE** permet d'importer des données dans une table SQL à partir d'un fichier texte.

Par exemple, nous avons la table suivante :

```
mysql> DESC contacts;
+--------+-------------+------+-----+---------+----------------+
| Field  | Type        | Null | Key | Default | Extra          |
+--------+-------------+------+-----+---------+----------------+
| id     | int(11)     | NO   | PRI | NULL    | auto_increment |
| prenom | varchar(50) | YES  |     | NULL    |                |
| nom    | varchar(50) | YES  |     | NULL    |                |
+--------+-------------+------+-----+---------+----------------+
3 rows in set (0.00 sec)
```

Nous voulons insérer le fichier suivant :

```
# cat /var/lib/mysql/dbalex/data.txt
Anne;Aconda
Clara;Sheller
Sophie;Garo
Pauline;Machin
```

Pour importer le fichier dans la table contact, nous tapons la commande suivante :

```
mysql> LOAD DATA INFILE 'data.txt' INTO TABLE contacts
       FIELDS TERMINATED BY ";" (prenom,nom);
Query OK, 4 rows affected (0.00 sec)
Records: 4  Deleted: 0  Skipped: 0  Warnings: 0
```

Vérification :

```
mysql> SELECT * FROM contacts;
+----+---------+---------+
| id | prenom  | nom     |
+----+---------+---------+
| 18 | Anne    | Aconda  |
| 19 | Clara   | Sheller |
| 20 | Sophie  | Garo    |
| 21 | Pauline | Machin  |
+----+---------+---------+
```

```
4 rows in set (0.02 sec)
```

Pour connaître l'ensemble des options disponibles de la commande **LOAD DATA INFILE**, voir la documentation Mysql : Doc Mysql / LOAD DATA INFILE (http://dev.mysql.com/doc/refman/5.0/fr/load-data.html)

SELECT INTO OUTFILE

La commande SQL **SELECT INTO OUTFILE** permet d'exporter des données de la base de données dans un fichier texte.

```
mysql> SELECT prenom,nom FROM contacts
       INTO OUTFILE '/tmp/result.txt'
       FIELDS TERMINATED BY ';'
       LINES TERMINATED BY '\n';
Query OK, 4 rows affected (0.02 sec)
```

Dans le cas présent, les champs **"prenom"** et **"nom"** ont été sélectionnés à partir de la table **"contacts"** afin de pouvoir être exportés dans le fichier **"result.txt"** situé dans le répertoire **"/tmp"**

Attention, il faut penser à préciser les **clauses "FIELDS"** et **"LINES"** destinées à traiter correctement les délimitations des champs et des lignes lors de l'export !

Remarque : C'est l'utilisateur **mysql** qui crée ce fichier, il doit donc avoir les droits nécessaires :

```
# ls -l /tmp/result.txt
-rw-rw-rw- 1 mysql mysql 53 2008-11-27 09:26 /tmp/result.txt
```

Les fichiers de données

Les fichiers contenant les données sont stockés dans **/var/lib/mysql/**.

/var/lib/mysql/ est donc le répertoire important à sauvegarder.

Chaque base de données dispose de son propre répertoire **/var/lib/mysql/<nomdelabase>/**.

Le fichier **db.opt** contient la configuration de la base de données (encodage des caractères ...).

Le fichier <nomdetable>.**FRM** décrit la structure de la table.

Dans le cas du format **MyISAM**, chaque table est enregistrée dans deux fichiers :

- <nomdetable>.**MYD** : contient les données
- <nomdetable>.**MYI** : contient les index

Dans le cas du format **Innodb**, toutes les tables sont stockées dans un ou plusieurs seuls fichiers : **ibdata1**, **ibdata2** ...

Les fichiers journaux

Les fichiers journaux sont contenus dans le répertoire **/var/log/mysql/**.

Il y a 3 fichiers importants :

- **mysql.log** : contient les authentifications des utilisateurs et les requêtes SQL
- **mysql.err** : contient les erreurs. Sous Debian, ce fichier n'existe pas ou est vide car les erreurs Mysql sont envoyées à **syslog** (Debian improvement).
- **mysql-slow.log** : contient les requêtes lentes

Pour activer les fichiers log, il faut décommenter les deux lignes dans **/etc/mysql/my.cnf** :

```
log = /var/log/mysql/mysql.log
log_slow_queries = /var/log/mysql/mysql-slow.log
```

Puis relancer mysql :

```
/etc/init.d/mysql restart
```

Les documentations

Package Debian mysql-doc-5.0

Debian propose un package contenant les documentations Mysql. Il faut pour y accéder installer le package **mysql-doc-5.0** :

```
# apt-get install mysql-doc-5.0
```

La documentation sur mysql est accessible dans le répertoire **/usr/share/doc/mysql-doc-5.0**.

Dans ce répertoire, on trouve des pages HTML qui contiennent toute la documentation de mysql.

Documentations en ligne

Sur le site de mysql

- Toutes les documentations (http://dev.mysql.com/doc/)

- Le manuel de référence en français (http://dev.mysql.com/doc/refman/5.0/fr/index.html)

Listes de diffusion

Toutes les listes de diffusion (http://lists.mysql.com)

IRC

Canal **#mysql** sur Freenode (http://www.freenode.net)

Modifications des privilèges

Il y a quatre techniques pour modifier les privilèges :

- modifier directement les tables **user** et **db** de la base de données **mysql** (ne pas oublier de faire un **FLUSH PRIVILEGES;** après)
- utiliser les instructions SQL **GRANT** et **REVOKE**

Exemple : enlever tous les droits à l'utilisateur **alex** sur la base de données **dbalex** :

```
mysql> REVOKE ALL PRIVILEGES ON dbalex.* FROM alex;
```

Exemple : donner tous les droits à l'utilisateur **alex** sur la base de données **dbalex** :

```
mysql> GRANT ALL PRIVILEGES ON dbalex.* to alex;
```

on peut également spécifier les droits à donner :

```
mysql> GRANT select,insert,update,delete ON dbalex.* to alex;
```

- utiliser la commande shell **mysql_setpermission** (cf ci-dessous)
- utiliser un outil graphique de type PhpMyAdmin

Les commandes d'administration

Utile : pour ne pas avoir à spécifier le mot de passe du root chaque fois que l'on invoque une commande **mysql***, il suffit de créer un fichier **.my.cnf** dans son répertoire de travail et qui contient le mot de passe à utiliser. Attention aux droits d'accès à ce fichier !

```
# echo -e "[client]\npassword=root" > ~/.my.cnf && chmod 600 ~/.my.cnf
```

```
# ls -l /root/.my.cnf
-rw------- 1 root root 23 déc  4 15:53 /root/.my.cnf
```

```
# cat /root/.my.cnf
[client]
password=root
```

mysql

La commande **mysql** est le client Mysql en ligne de commande. Il permet de se connecter à Mysql et de saisir des commandes SQL.

Voici les options les plus courantes :

Options de base (communes à la plupart des commandes mysql) :

- -h (--host=) : définit **l'hôte** hébergeant la base de données
- -D (--database=) : définit **la base** sur laquelle l'utilisateur va se connecter.
- -u (--user=) : précise **le nom d'utilisateur Mysql** sous lequel l'utilisateur se connecte.
- -P (--port=) : détermine le **port à utiliser** pour la connexion
- -p (--password=) : demande la saisie du **mot de passe** *(obligatoire si l'utilisateur a été défini pour se connecter avec un mot de passe)*

Exemple :

```
$ mysql --database=dbalex -u paul -p
PASSWORD:
```

Dans cet exemple la requête demande de se connecter **par mot de passe** à la base **dbalex** en tant que **paul**.

Autres options intéressantes :

- -H (--html) : permet de produire des **sorties en HTML**
- -X (--xml) : permet de produire des **sorties en XML**

Pour aller plus loin :

```
# mysql --help
```

Donne la liste et un bref descriptif de toutes les options à utiliser avec la commande mysql

mysqldump

La commande **mysqldump** permet d'exporter des données d'une base de données.

Son fonctionnement est particulièrement intéressant car **elle génère les commandes SQL permettant de re-créer la base de données sur un autre serveur**.

Pour exporter la base de donnée « myBase », on utilise la commande suivante :

```
# mysqldump -u root -p myBase > myBase_backup.sql
```

Ceci fera l'export dans un fichier « myBase_backup.sql ».

Pour importer une base de données sauvegardée via **mysqldump**, on utilise la commande cliente **mysql** et une redirection en entrée :

```
# mysql -u root -p myBase < myBase_backup.sql
```

L'option **--compatible** permet de spécifier à mysqldump le format à utiliser pour être compatible avec les bases de données existantes. Exemple :

```
# mysqldump --compatible=oracle -u root -p myBase > myBase_backup.sql
```

Cette option peut prendre les valeurs suivantes : ansi, mysql323, mysql40, postgresql, oracle, mssql, db2, maxdb, no_key_options, no_table_options, or no_field_options

En utilisant **mysqldump** et **ssh**, on peut dupliquer une base de données sur une machien distante :

```
# mysqldump testdb -p<mot de passe local> | ssh pc211 'echo "create database dbalex;" | mysql -p<mot de passe distant> ; cat
- | mysql -p<mot de passe distant> dbalex'
```

mysqlimport

...

```
# mysqlimport -p --fields-terminated-by="\;"
              --lines-terminated-by="\n"
              --columns=prenom,nom
              dbalex /var/lib/mysql/dbalex/contacts.txt
Enter password:
dbalex.contacts: Records: 4  Deleted: 0  Skipped: 0  Warnings: 4
#
```

Attention : le nom du fichier (sans extension) doit être le même que la table.

mysqladmin

La commande **mysqladmin** permet de passer des commandes à Mysql.

```
$ mysqladmin [OPTIONS] commande [options de la commande]
```

OPTIONS :

```
-h host
-u user
-p password
```

Commandes :

- **create** : crée une base de données
- **drop** : supprime une base de données
- **flush-privileges** : recharge les tables de droits
- **flush-hosts** : met à jour les informations concernant les hôtes
- **flush-logs** : met à jour les informations concernant les journaux
- **flush-status** : efface les variables status
- **flush-tables** : met à jour les informations concernant les tables
- **flush-threads** : met à jour les informations concernant les threads
- **password** : change le mot de passe
- **old-password** : change le mot de passe en utilisant l'ancien algorithme de chiffrement
- **ping** : teste si mysql fonctionne
- **reload** : recharge la configuration
- **refresh** : vide les caches
- **shutdown** : arrête mysql
- **status** : connaitre des informations sur l'état du serveur
- **extended-status** : connaitre des informations détaillées sur l'état du serveur
- **proc** ou **processlist** : connaitre les utilisateurs connectés
- **debug** : Passe Mysql en mode debug
- **kill** : Permet d'arrêter des threads. On indique le numéro de connexion à terminer obtenu avec **proc** ou **processlist**
- **start-slave** : Démarre la réplication sur le serveur de réplication esclave
- **stop-slave** : Arrête la réplication sur le serveur de réplication esclave
- **variables** : Affiche les variables internes Mysql
- **version** : Affiche le numéro de version de Mysql
- ...

Utile : j'ai perdu le mot de passe du compte **root** mysql, comment le changer :

1. On arrête Mysql :

```
# /etc/init.d/mysql stop
```

2. On relance le daemon **mysqld** avec l'option **--skip-grant-tables**

```
# mysqld --skip-grant-tables
```

3. Depuis une autre fenêtre, on se connecte à Mysql

```
# mysql mysql
```

4. On met à jour la table **user** :

```
mysql> update user set password=PASSWORD('root') where user='root';
Query OK, 0 rows affected (0.00 sec)
Rows matched: 3  Changed: 0  Warnings: 0
```

```
mysql> flush privileges;
Query OK, 0 rows affected (0.00 sec)
```

5. On arrête proprement le daemon **mysqld** :

```
# killall mysqld
```

6. On relance Mysql :

```
# /etc/init.d/mysql start
```

mysqlcheck

La commande **mysqlcheck** permet de vérifier l'intégrité d'une base de données. On indique la (les) base(s) de données à vérifier (et optionnellement les tables à vérifier), et la commande donne l'état : OK ou corrompue.

Options courantes :

- **-r** ou **--repair** : permet de corriger presque tout, sauf les problèmes de doublons pour les clés uniques.
- **-a** ou **--analyse** : permet d'analyser les tables indiquées.
- **-o** ou **--optimize** : permet d'optimiser les tables indiquées.

Exemple : vérifier l'intégrité de toutes les tables de la base de données **testdb** :

```
# mysqlcheck testdb
testdb.client                                      OK
testdb.client2                                     OK
```

Vérifier uniquement la table **client** :

```
# mysqlcheck testdb client
testdb.client                                      OK
```

Vérifier les tables **user** et **db** de la base de données **mysql** :

```
# mysqlcheck mysql user db
mysql.user                                         OK
mysql.db                                           OK
```

Remarque : **mysqlcheck** ne fonctionne qu'en utilisant des instructions SQL (CHECK TABLE, REPAIR TABLE, ANALYZE TABLE, et OPTIMIZE TABLE), il a donc besoin que le serveur Mysql soit en train de fonctionner.

myisamchk

La commande **myisamchk** permet la restauration d'une table ou plusieurs tables endommagées à la suite de crash répétés du démon mysqld.

Attention : **myisamchk** intervient directement sur les fichiers, il se passe de **mysqld** contrairement à **mysqlcheck** !

Quelques précautions d'usage: : Si vous utilisez myisamchk pour réparer ou optimiser les tables, vous devez toujours vous assurer que mysqld n'utilise pas cette table (ce qui s'applique aussi si vous utilisez --skip-external-locking). Si vous n'éteignez pas le serveur mysqld , vous devez au moins utiliser mysqladmin flush-tables avant de

lancer myisamchk.

Attention : **Vos tables peuvent être corrompues si le serveur mysqld et myisamchk travaillent dans une même table simultanément.**

Vérifier la cohérence d'une table:

On indique à **myisamchk** les tables à vérifier en désignant les fichiers index **MYI** :

```
# myisamchk /chemin/bases/mysql/nom_de_la_bd/nom_de_table.MYI
```

On peut également spécifier toutes les tables comme ceci :

```
# myisamchk /chemin/bases/mysql/nom_de_la_bd/*.MYI
```

Voire carrément vérifier toutes les tables de toutes les bases :

```
# myisamchk /chemin/bases/mysql/*/*.MYI
```

La commande affiche un rapport d'analyse, et si tout est OK, ne signale pas d'erreur :

```
# myisamchk /var/lib/mysql/ampache/album.MYI

Checking MyISAM file: /var/lib/mysql/ampache/album.MYI
Data records:     542   Deleted blocks:       0
- check file-size
- check record delete-chain
- check key delete-chain
- check index reference
- check data record references index: 1
- check data record references index: 2
- check data record references index: 3
- check data record references index: 4
- check record links
```

Cette commande trouvera 99.99% de toutes les erreurs. Elle ne peut toutefois détecter les erreurs qui impliquent uniquement le fichier de données (ce qui est très inhabituel).

Si vous voulez uniquement vérifier une table sans que la commande produise un affichage, il faut utiliser l'option **-s** (ou **--silent**).

mysql_setpermission

La commande **mysql_setpermission** permet de définir les permissions des utilisateurs Mysql de manière intéractive.

Ecrit en Perl, ce script a besoin des modules **DBI** et **DBD::mysql** pour fonctionner.

```
# mysql_setpermission -p
Option p is ambiguous (password, port)
Password for user  to connect to MySQL:
######################################################################
## Welcome to the permission setter 1.4 for MySQL.
## made by Luuk de Boer
######################################################################
What would you like to do:
 1. Set password for an existing user.
 2. Create a database + user privilege for that database
    and host combination (user can only do SELECT)
 3. Create/append user privilege for an existing database
    and host combination (user can only do SELECT)
 4. Create/append broader user privileges for an existing
    database and host combination
    (user can do SELECT,INSERT,UPDATE,DELETE)
 5. Create/append quite extended user privileges for an
    existing database and host combination (user can do
    SELECT,INSERT,UPDATE,DELETE,CREATE,DROP,INDEX,
    LOCK TABLES,CREATE TEMPORARY TABLES)
 6. Create/append full privileges for an existing database
    and host combination (user has FULL privilege)
 7. Remove all privileges for for an existing database and
    host combination.
    (user will have all permission fields set to N)
 0. exit this program
```

```
Make your choice [1,2,3,4,5,6,7,0]:
```

- **1** : Permet de modifier le mot de passe d'un utilisateur éxistant

- **2** : Permet d'ajouter/modifier le droit SELECT d'un utilisateur sur une table en créant un utilisateur et une table :

```
# Make your choice [1,2,3,4,5,6,7,0]: 2


# Which database would you like to add: dbuser //On donne le nom de la base de donnée à créer.
# The new database dbuser will be created

# What username is to be created: user //On donne le nom de l'utilisateur à créer.
# Username = user

# We now need to know from what host(s) the user will connect.
# Keep in mind that % means 'from any host' ...
# The host please: % //On donne l'host
#
```

L'utilisateur **user** aura le droit **SELECT** sur la base de donnée **dbuser**

- **3** : Même fonction que la sélection 2, sauf que l'on donne le droit SELECT sur une table existante.

- **4** : Même fonction que la sélection 2, pour les droits SELECT,INSERT,UPDATE,DELETE.

- **5** : Même fonction que la sélection 2, pour les droits SELECT,INSERT,UPDATE,DELETE,CREATE,DROP,INDEX,LOCK TABLES,CREATE TEMPORARY TABLES

- **6** : Même fonction que la sélection 2, pour tous les droits.

- **7** : Met les droits par défaut (N) pour un utilisateur existant.

- **0** : sortir du programme

Les différentes options de mysql_setpermission :

- **--help** (ou **-h**) : permet d'afficher l'aide
- **--host** (ou **-h**) : se connecte au serveur Mysql donné
- **--password** (ou **-p**) : pour qu'il demande le mot de passe à la connexion
- **--port** (ou **-P**) : donne le numéro de port si différent de celui par défaut
- **--user** (ou **-u**) : se connecte au serveur avec le nom de compte donné

mysqlhotcopy

La commande **mysqlhotcopy** permet de copier une base de données **à chaud**, c'est à dire sans arrêter le serveur Mysql.

Pour cela, la commande bloque les tables afin qu'il n'y ai pas de modification des tables durant la copie.

Les tables sont ensuite débloquées.

Exemple d'utilisation :

```
# mysqlhotcopy -p root testdb /tmp
Locked 2 tables in 0 seconds.
Flushed tables (`testdb`.`client`, `testdb`.`client2`) in 0 seconds.
Copying 8 files...
```

```
Copying indices for 0 files...
Unlocked tables.
mysqlhotcopy copied 2 tables (8 files) in 0 seconds (1 seconds overall).
```

Vérification :

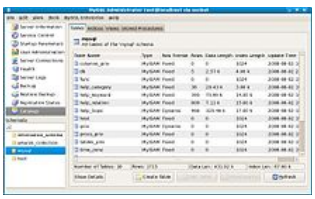
```
# ls -l /tmp/testdb/
total 44
-rw-rw---- 1 mysql mysql 8584 déc  2 15:08 client2.frm
-rw-rw---- 1 mysql mysql    0 déc  2 15:08 client2.MYD
-rw-rw---- 1 mysql mysql 1024 déc  4 16:23 client2.MYI
-rw-rw---- 1 mysql mysql 8618 déc  2 15:50 client.frm
-rw-rw---- 1 mysql mysql 3090 déc  4 14:01 client.MYD
-rw-rw---- 1 mysql mysql 2048 déc  4 16:23 client.MYI
-rw-rw---- 1 mysql mysql   65 déc  2 14:58 db.opt
```

Autres programmes utiles

MySQL Workbench

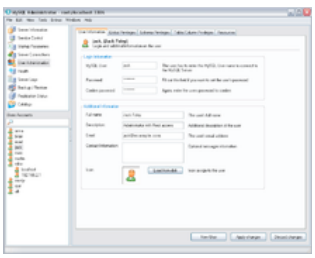
Anciennement nommé *MySQL Administrator*, *MySQL Workbench* est aujourd'hui maintenu par Oracle.

Accueil de MySQL Workbench



Accueil de MySQL Administrator

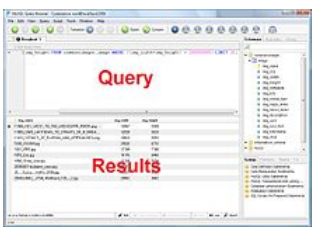
Configuration de MySQL Workbench



Configuration de MySQL Administrator

MySQL Query Browser

MySQL Query Browser n'est plus développé depuis 2012[1].

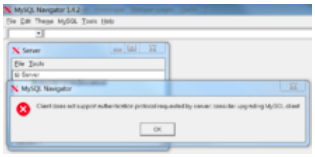


MySQL Control Center

mysqlcc est l'ancien logiciel d'administration de MySQL, il s'est arrêté en 2013[2], et donc n'est pas 100 % compatible avec la version 5 de MySQL. Il est désormais remplacé par *MySQL Workbench*.

MySQL Navigator

MySQL Navigator n'est plus développé depuis 2013[3].



Références

1. http://sqlbrowser.www2.pl/?act=download
2. http://sourceforge.net/projects/mysqlcc/?source=navbar
3. http://sourceforge.net/projects/mysqlnavigator/?source=typ_redirect

Le serveur de mails Postfix

Le serveur de mail Postfix

Postfix est un serveur de mail écrit dans l'idée de remplacer Sendmail, serveur de mail historique de moins en moins utilisé car sa configuration est très complexe et qu'il contient des failles de sécurité.

Postfix effectue 99 % des fonctions de Sendmail, les 1% restant ne sont que purement applicable dans une minorité de cas. De plus, il allie performance et facilité d'installation et de paramétrage.

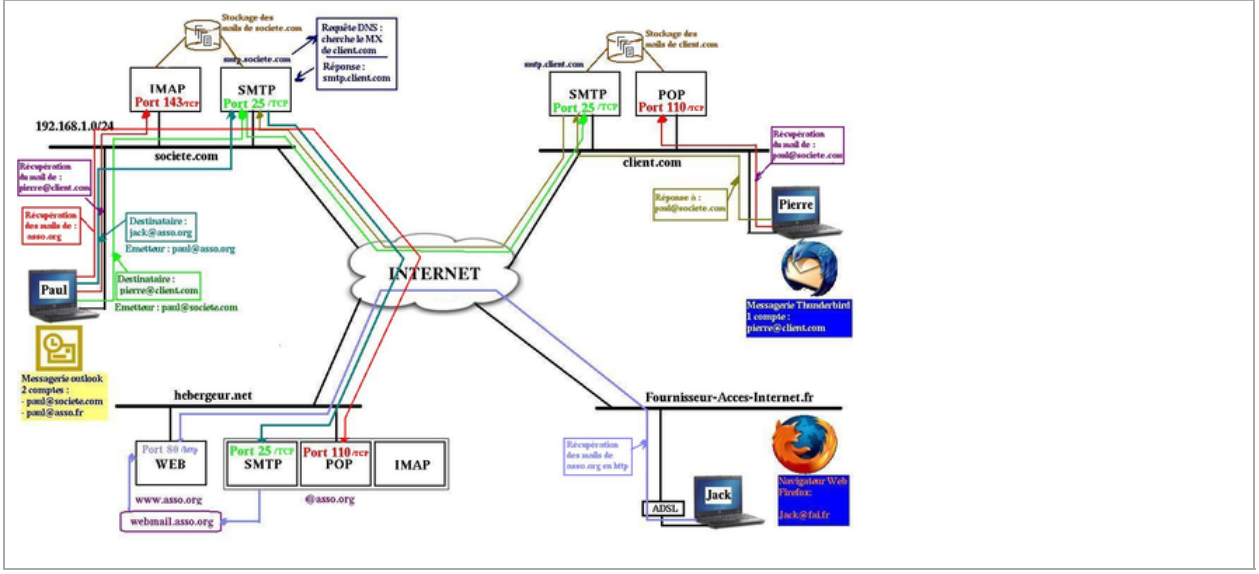
Il existe de nombreux autres serveurs de mails, parmi les plus connus : Qmail, Exim, Qpsmtpd, Courier...

Quelques définitions

- **DNS** : Domain Name System (port 53), pour convertir les adresses URL en adresses IP.
- **SMTP** : Simple Mail Transfer Protocol (port 25), pour l'envoi et l'acheminement du courrier.
- **POP** : Post Office Protocol (port 110), pour la réception du courrier.
- **IMAP** : Internet Message Access Protocol (port 143), sert à la réception du courrier. À la différence de POP, les emails restent stockés sur le serveur IMAP.
- **MTA** : Mail Transfer Agent. Il s'agit des serveurs de mails SMTP.
- **MDA** : Mail Delivery Agent. Il s'agit du serveur qui dépose les messages dans chaque boîte aux lettres (POP ou IMAP).
- **MUA** : Mail User Agent. Il s'agit des logiciels clients de messagerie, ex : Mozilla Thunderbird, Microsoft Outlook, KMail, Eudora, ou Evolution.

Comment ça marche ?

Schéma du cheminement des mails entre les correspondants (trois cas illustrés en fonction des moyens mis en œuvre).



Le schéma précédent nous propose trois cas de figure :

Cas numéro 1

- Nous avons deux sociétés : societe.com et client.com
- Paul (paul@societe.com) veut envoyer un mail à Pierre (pierre@client.com)
- Le mail part en direction du serveur SMTP de sa société (smtp.societe.com)
- Celui-ci est accepté par le serveur de mail smtp.societe.com car il vient de son propre réseau
- smtp.societe.com fait une demande au serveur DNS pour connaitre le serveur de mail qui gère les courriers de @client.com : client.com MX ? (Mail eXchanger)
- Le serveur DNS répond : smtp.client.com
- Le serveur de mail smtp.societe.com contacte donc le serveur SMTP smtp.client.com
- Lorsque le client de messagerie de Pierre veut consulter ces mails, il va demander à POP si il y a quelque chose pour lui et télécharge le mail de Paul
- Pierre a donc bien reçu le mail de Paul

- Pierre répond à Paul de la même manière
- Le mail suit le cheminement inverse : smtp.client.com demande au serveur DNS le serveur de messagerie de societe.com. Celui-ci lui répond : smtp.societe.com
- Le mail est acheminé vers le serveur SMTP smtp.societe.com
- Il est stocké jusqu'à la demande de Paul de lire ces mails. Contrairement à Pierre, le client de messagerie de Paul interroge le serveur IMAP imap.societe.com. La différence est que le mail de Pierre est lu directement depuis le serveur IMAP et reste stocké sur ce dernier

Cas numéro 2

- Notre ami Paul fait partie d'une association (asso.org) et dispose d'un email sur ce domaine : paul@asso.org
- L'association dispose d'un hébergement chez hebergeur.net qui lui fournit différents services serveur SMTP, POP, IMAP et accès webmail
- Paul utilise son email sur asso.org (paul@asso.org) pour envoyer un email à son ami Jack qui dispose aussi d'un email sur asso.org : jack@asso.org
- Ce mail est acheminé via son réseau vers son serveur SMTP. Dans son client Outlook, Paul a paramétré plusieurs comptes
- Le serveur de sa société (smtp.societe.com) accepte d'envoyer le mail sur Internet, même si l'email émetteur et le destinataire utilisent des noms de domaine différents, car il vient de son propre réseau
- Ce mail est donc envoyé au serveur SMTP asso.org : smtp.hebergeur.net
- Paul peut donc s'il le souhaite aller consulter ses mails sur asso.org via le serveur POP ou IMAP qui va rapatrier directement les mails paul@asso.org

Cas numéro 3

- Jack utilise une connexion à Internet fournie par un fournisseur d'accès à Internet : fai.fr. Pour lire son email sur asso.org (jack@asso.org), Jack utilise son navigateur Web et l'accès Webmail fournit par fai.fr
- Il se connecte donc sur l'adresse webmail.asso.org et peut ainsi visualiser ses mails. Aucun email ne sera rapatrié sur son PC.
- Il peut également répondre à tous ses mails via un Webmail.

Installation de Postfix

Installer en premier Postfix, si un autre serveur de mail est déjà présent, il le supprimera :

```
# apt-get install postfix
```

Pour l'instant, nous laissons toutes les options par défaut, nous allons le configurer par la suite.

Une fois l'installation terminée, nous allons configurer Postfix.

```
sudo adduser postfix sasl[1]
sudo dpkg-reconfigure postfix
```

Configuration de Postfix

> **Attention !**
>
> Pour que cela fonctionne sur Internet, il faut que le nom local du serveur (hostname) soit le même que celui du réseau (défini dans les DNS). Dans cet exemple, il s'appellera *mail.mondomaine.fr*. De plus, il faut commenter la ligne ci-dessous `mydestination` qui bride les envois vers l'extérieur.

Le fichier de configuration principal de Postfix est `/etc/postfix/main.cf`. Copier le suivant en remplaçant `mondomaine.fr` par le vrai domaine du serveur :

```
mv /etc/postfix/main.cf /etc/postfix/main.oldcf  # Archivage
> /etc/postfix/main.cf                           # Blanchiment
vim /etc/postfix/main.cf                         # Accès pour coller la configuration ci-dessous
```

```
# Message affiché à la connexion
smtpd_banner = $myhostname ESMTP

# Ne pas utiliser le service biff qui sert à la notification
# des nouveaux mails
biff = no

# On ne rajoute pas le domaine car c'est le boulot du client mail
append_dot_mydomain = no

# Permet d'envoyer un email a l'émetteur si son mail n'est pas
```

```
# parti au bout d'un certain temps
#delay_warning_time = 4h

# Emplacement des alias Unix
# Attention, à chaque modification de /etc/aliases, il faut taper
# la commande newaliases pour qu'il régénère le fichier /etc/aliases.db
alias_maps = hash:/etc/aliases
alias_database = hash:/etc/aliases

# Le domaine d'origine
myorigin = /etc/mailname

# Le nom de machine FQDN (doit être en accord avec le DNS)
mydomain = mondomaine.fr
myhostname = mail.$mydomain

# Liste des domaines principaux dont on accepte le courrier
mydestination = $mydomain, $myhostname, localhost, localhost.$mydomain

# Permet d'indiquer un autre serveur SMTP pour l'envoi des emails
# Utilisé si on ne peut pas envoyer directement les emails
relayhost =

# Indique les réseaux dont on accepte d'acheminer les emails
#mynetworks = 127.0.0.0/8, 192.168.30.0/24

# Le dépôt des emails dans les boites aux lettres est confié
# à procmail (cas du format mbox)
mailbox_command = procmail -a "$EXTENSION"

# Taille des boites aux lettres. 0 : pas de limite
mailbox_size_limit = 0

# Séparateur utilisé pour les usernames + adresses
recipient_delimiter = +

# Écoute sur toutes les interfaces réseaux
inet_interfaces = all

# On force postfix à utiliser le protocole IPv4
inet_protocols = ipv4
```

On peut visualiser la totalité des options possibles de Postfix et leurs valeurs par défaut via la commande **postconf** :

```
# postconf | more
2bounce_notice_recipient = postmaster
access_map_reject_code = 554
address_verify_default_transport = $default_transport
address_verify_local_transport = $local_transport
address_verify_map =
address_verify_negative_cache = yes
address_verify_negative_expire_time = 3d
...
```

En français ça donne : *délivre en local les emails vers* `mydestination` *avec* `mailbox_command`*, sinon délivre-les sur Internet mais en refusant ceux qui ne viennent pas de* `mynetworks` *ou* `relayhost` *(s'ils sont vides on accepte rien, s'ils sont commentés on accepte tout, même de relayer des spams).*

Une fois `main.cf` sauvegardé pour la première fois, on vérifie que le fichier **/etc/mailname** contient le nom de domaine de notre réseau :

```
# cat /etc/mailname
mondomaine.fr
```

Chaque fois que l'on modifie le fichier **/etc/postfix/main.cf**, il faut relancer Postfix :

```
# /etc/init.d/postfix restart
```

Pour chaque mail envoyé ou reçu, Postfix enregistre les informations d'acheminement dans le fichier **/var/log/mail.log**. Il est donc très utile de surveiller ce fichier durant les tests de fonctionnement.

Pour vérifier que le serveur de mail fonctionne, on envoi un mail à un utilisateur et on regarde le fichier **/var/log/mail.log**.

```
# mail alex
Subject: essai
essai
```

```
.
Cc:
CTRL+D
```

Dans un autre terminal :

```
# tail -f /var/log/mail.log
...
Feb 13 13:10:40 mail postfix/pickup[13353]: B782A1FD6F: uid=0 from=<root>
Feb 13 13:10:40 mail postfix/cleanup[13675]: B782A1FD6F: message-id=<20080213121040.B782A1FD6F@mail.mondomaine.fr>
Feb 13 13:10:40 mail postfix/qmgr[13354]: B782A1FD6F: from=<root@mail.mondomaine.fr>, size=355, nrcpt=1 (queue active)
Feb 13 13:10:40 mail postfix/local[13677]: B782A1FD6F: to=<alex@mail.mondomaine.fr>, orig_to=<alex>, relay=local,
delay=0.08, delays=0.04/0.02/0/0.02, dsn=2.0.0, status=sent (delivered to command: procmail -a "$EXTENSION")
Feb 13 13:10:40 mail postfix/qmgr[13354]: B782A1FD6F: removed
...
```

Ajouter une boite aux lettres

Pour ajouter une boite aux lettres, il existe trois solutions :

1. Créer un compte Unix.
2. Créer une adresse virtuelle.
3. Créer un alias.

Les voici en lignes de commandes, mais elles sont aussi réalisables au moyen d'un webmail.

Nouveau compte système

Pour des raisons de sécurité, on positionne le shell de l'utilisateur à **/bin/false** (ainsi il ne pourra pas se logger sur le serveur). Le mot de passe saisi sera celui de sa boite aux lettres. On rajoute l'utilisateur dans le groupe **mail** afin qu'il puisse écrire le fichier <login>.lock dans le répertoire **/var/mail**.

```
# adduser --shell /bin/false --ingroup mail pierre
```

Adresse virtuelle

Activer la fonction dans :

```
vim /etc/postfix/main.cf
```

avec :

```
virtual_alias_maps = hash:/etc/postfix/virtual
```

Ensuite il faut définir les adresses emails dans `/etc/postfix/virtual`, sous forme de paires clé-valeur[2] :

```
postmaster@mail.mondomaine.fr        root
```

Puis créer la base de données à partir du fichier :

```
postmap /etc/postfix/virtual
postfix reload
```

Alias

Pour éditer la liste des adresses emails alias de celles d'une boite :

```
vim /etc/aliases
```

Par exemple en y ajoutant les plus fréquents :

```
MAILER-DAEMON:  root
postmaster:     root
abuse:          root
```

Ainsi, quelqu'un qui voudra reporter un problème à abuse@mondomaine.fr écrira dans la boite de root, ce qui sera visible par une notification à chaque fois que ce dernier

se connectera en SSH.

Puis la commande `newaliases` crée alors `/etc/aliases.db` à partir de cette liste.

MDA : filtres de courrier électronique

Par défaut les emails des boites créées ci-dessus s'empilent tous dans un fichier de `/var/mail/`. Afin de les router vers des boites mails séparées, on utilise un ou plusieurs[3] logiciels Mail Delivery Agent.

- Pour les transferts locaux (vers une boite mail personnelle du serveur), ils utilisent le protocole LMTP.
- Pour les transferts distants (vers un client de messagerie), il faut choisir entre le protocole POP3 (qui déplace les emails vers le client) et IMAP (qui laisse une copie des emails sur le serveur).

Voici plusieurs exemples de MDA :

procmail

Pour installer procmail[4][5] :

```
 $ sudo apt-get install procmail
 $ vim ~/.forward
"|IFS=' '&&exec /usr/bin/procmail -f-||exit 75 #user"

 $ vim ~/.procmailrc
PATH=/usr/bin:/usr/local/bin
MAILDIR=$HOME/Maildir

 $ maildirmake ~/Maildir
```

- En remplaçant *user* par l'utilisateur du processus.
- Sur Ubuntu il s'installe dans `/usr/bin/procmail`, mais d'autres distributions nécessitent d'adapter ce chemin ci-dessus.

Explications :

1. Le fichier utilisateur `.forward` permet d'insérer le filtrage `procmail` dans le parcours des emails à destination de cet utilisateur.
2. Le fichier `.procmailrc` configure le filtrage pour l'utilisateur concerné.
3. `maildirmake` crée l'arborescence nécessaire à l'accueil des emails :
   - `new` (nouveau) : boite de réception.
   - `cur` (courante) : boite courante.
   - `tmp` (temporaire) : mails en transit.

Pour voir s'il fonctionne, envoyer un email à l'utilisateur configuré ci-dessus, puis regarder s'il est arrivé :

```
ls -alh /home/user/Maildir/new/
```

En cas de problème :

```
tail /var/log/mail.log
```

fetchmail

Pour installer fetchmail :

```
apt-get install fetchmail
```

Il se configure avec :

```
vim ~/.fetchmailrc
```

On peut par exemple lui dire de tout transférer vers procmail en y ajoutant :

```
mda "/usr/bin/procmail -Y -d %T"
```

Ou bien lui faire récupérer directement des boites utilisateurs distantes avec :

```
poll example.com protocol pop3 username "user1" password "password1"
```

> **Attention !**
>
> La ligne `mailbox_command` du fichier `/etc/postfix/main.cf` doit pointer vers procmail, qui reroute ensuite vers fetchmail.

fetchmail peut récupérer les emails de fournisseurs externes, tels que Gmail, Yahoo ou Outlook.com, par cron. C'est un mail retrieval agent (MRA).

Dovecot

En plus d'un MDA intégré, Dovecot intègre un serveur POP ou IMAP.

Tout comme pour fetchmail, il peut fonctionner en aval de procmail. Toutefois, il est possible de remplacer procmail par Dovecot dans `/etc/postfix/main.cf`[6] :

```
mailbox_command = /usr/local/libexec/dovecot/dovecot-lda -f "$SENDER" -a "$RECIPIENT"
```

pour Ubuntu :

```
mailbox_command = /usr/lib/dovecot/dovecot-lda -f "$SENDER" -a "$RECIPIENT"
```

Ensuite, l'installation de Dovecot dépend du protocole souhaité dans le MUA qui viendra relever le courrier :

Serveur POP

```
apt-get install dovecot-pop3d
```

> **Remarque :** il existe de nombreux autres serveurs POP : courier-pop, teapop, ipopd, qpopper, solid-pop3d...

Serveur IMAP

```
apt-get install dovecot-imapd
```

> **Remarque :** il existe aussi plusieurs autres serveurs IMAP, par exemple courier-imap.

IMAP nécessite le format de stockage **Maildir**. Nous allons donc tout d'abord configurer Postfix pour lui indiquer d'utiliser le format **Maildir** au lieu du traditionnel **mbox**.

On modifie le fichier `/etc/postfix/main.cf`. On commente l'option **mailbox_command** (qui appelle procmail) et on rajoute l'option **home_mailbox** :

```
vim /etc/postfix/main.cf
# Commenter la ligne :
#mailbox_command = procmail -a "$EXTENSION"
# Puis pour utiliser le format Maildir, ajouter :
home_mailbox = Maildir/
```

On relance postfix :

```
# /etc/init.d/postfix restart
```

Pour chaque utilisateur existant, il faut ensuite créer le répertoire *Maildir*. On fait ceci à l'aide de la commande *maildirmake* fournie par *dovecot-imapd* ou *courrier-imap* :

```
$ cd      # pour retourner dans mon home directory
$ maildirmake Maildir
$ ls -l Maildir
drwx------ 2 alex grpalex 4096 2008-02-13 13:58 cur
drwx------ 2 alex grpalex 4096 2008-02-13 13:58 new
drwx------ 2 alex grpalex 4096 2008-02-13 14:11 tmp
```

Pour automatiser la création du répertoire **Maildir** pour les nouveaux utilisateurs, on exécute la commande **maildirmake** dans le répertoire **/etc/postmaster**. Ainsi tous les nouveaux utilisateurs auront automatiquement le répertoire **Maildir** dans leur home directory :

```
# cd /etc/postmaster
# maildirmake Maildir
```

Webmails

> **Attention !**
>
> Par défaut les webmails cherchent les courriels dans le dossier définit dans le serveur POP ou IMAP, et qui peut être différent de là où ils sont distribués.

Par exemple pour la configuration ci-dessus avec Dovecot, `/etc/dovecot/conf.d/10-mail.conf` doit contenir `mail_location = maildir:/home/%u/Maildir`.

SquirrelMail

SquirrelMail est un webmail très simple à installer, qui fonctionne sans SGBD. Outre les envois et réceptions d'emails, il permet d'archiver en créant ou supprimant des dossiers.

Si n'est déjà le cas, on installe le serveur Web Apache :

```
# apt-get install apache2
```

Pour installer SquirrelMail sous Debian, on tape la commande suivante :

```
# apt-get install squirrelmail
```

On peut configurer Squirrelmail en utilisant le programme **squirrelmail-configure**. Ce fichier permet configurer le fichier **/etc/squirrelmail/config.php**. A noter que l'on peut modifier directement ce fichier sans passer par **squirrelmail-configure**.

SquirrelMail est fourni avec la section à rajouter à Apache pour l'activer. On va donc dans le répertoire de configuration de Apache pour lui indiquer d'utiliser ce fichier :

```
# ln -s /etc/squirrelmail/apache.conf /etc/apache2/conf.d/squirrelmail.conf
```

On relance Apache :

```
# /etc/init.d/apache2 restart
```

On peut tester si SquirrelMail fonctionne avec son navigateur en se connectant à l'URL suivante :

```
http://localhost/squirrelmail/
```

Postfixadmin

Postfixadmin est un webmail qui fonctionne avec MySQL.

```
apt-get install postfixadmin
```

Roundcube

Entrer un compte mail déjà configuré pour recevoir des emails

Roundcube est un webmail qui fonctionne avec MySQL, PostgreSQL, SQLite ou MSSQL. Il est très ergonomique (glisser-déposer, effacement en masse avec Maj + clic puis Suppr), plus graphique mais plus lent et moins stable que SquirrelMail.

```
apt-get install roundcube roundcube-mysql
php5enmod mcrypt
/etc/init.d/apache2 restart
```

Et voilà : http://localhost/roundcube

Nouveau mail

Configuration pour des envois distants

Pour que les emails ne soient pas directement écartés ou classés dans les spams par les serveurs de messagerie des destinataires, il faut impérativement montrer patte-blanche de plusieurs façons :

1. Règle de reverse DNS.
2. Règle DNS SPF (1 ou 2[7]).
3. Règle DNS DMARC.
4. Signature DKIM dans chaque email[8] correspondant à l'enregistrement DNS TXT dédié.
5. Absence de blacklistage sur l'adresse IP publique du SMTP[9].
6. Optimisation du ratio texte / image à 60 / 40 %[10].

> **Attention !**
>
> Un serveur de messagerie qui accepte d'envoyer des emails pour le compte de n'importe qui se retrouvera rapidement blacklisté par des organismes antispams, dont les listes sont utilisées par de nombreux serveurs MX. Et cela peut coûter de l'argent pour se déblacklister. Pour s'en prémunir, il est donc très fortement recommandé de laisser le paramètre `relayhost =` non commenté.

Sécurisation TLS

Tout d'abord, il faut générer une clé de cryptage asymétrique, puis indiquer son chemin dans `main.cf` :

```
# Paramètres liés au chiffrement TLS
smtpd_tls_cert_file=/etc/ssl/certs/ssl-cert-snakeoil.pem
smtpd_tls_key_file=/etc/ssl/private/ssl-cert-snakeoil.key
smtpd_use_tls=yes
smtpd_tls_session_cache_database = btree:${queue_directory}/smtpd_scache
smtp_tls_session_cache_database = btree:${queue_directory}/smtp_scache
# See /usr/share/doc/postfix/TLS_README.gz in the postfix-doc package for
# information on enabling SSL in the smtp client.
```

En cas d'erreur, on peut vérifier sa mise en place avec

```
openssl s_client -debug -starttls smtp -crlf -connect localhost:25
```

Pour plus d'information sur les certificats .pem, voir Apache/HTTPS.

Problèmes connus

421 Server Busy Error

Trouver le goulet d'étranglement avec `qshape`.

451 4.3.0 Temporary lookup failure

Sinon, recopier le `main.cf` ci-dessus.

454 4.7.1 Relay access denied / relaying denied

- Commenter le bridage par adresses réseaux dans `/etc/postfix/main.cf` :

```
mynetworks =
```

- Sinon, vérifier que le domaine du destinataire figure bien dans `/etc/postfix/main.cf` :

```
mydestination =
```

- Sinon, dans `/etc/postfix/main.cf`, le paramètre *virtual_mailbox_domains* est vide[11].
- Sinon, ajouter ou modifier les lignes suivantes à `/etc/postfix/main.cf` :

```
smtpd_recipient_restrictions = permit_mynetworks permit_sasl_authenticated reject_unauth_destination
```

- Sinon, recopier le `main.cf` ci-dessus.

> **Attention !**
>
> Lors des tests, ne pas commenter `relayhost =` pendant plus d'une heure sous peine de devenir spammeur à son insu.

501 5.1.7 Bad sender address syntax

Modifier l'adresse de l'expéditeur (ex : dans les options Squirrel).

550 relay not permitted / Sender verify failed

Vérifier le reverse DNS.

550 unknown recipient / 550 5.1.1: Recipient address rejected: User unknown in local recipient table

Le domaine ou sa boite mail n'est pas installé sur le MX.

S'il s'agit bien d'un utilisateur local, créer un alias d'une boite existante.

Sinon, si le serveur MX distant fonctionne par ailleurs, commenter dans `/etc/postfix/main.cf` :

```
mydestination =
```

Et relancer postfix.

Connection closed by foreign host / ou aucune commande ne répond après la connexion au SMTP

Si le serveur s'arrête immédiatement après son lancement, certaines erreurs sont visibles dans les logs. Sinon, recopier le `main.cf` ci-dessus.

dsn=4.4.1, status=deferred

Si les emails fonctionnent en local, mais pas depuis l'extérieur (avec la même adresse d'expéditeur), et qu'ils ne sont pas visibles dans

```
tail -30 /var/log/mail.log
```

Retester en ouvrant le port SMTP du pare-feu :

```
iptables -A INPUT -i eth0 -p tcp --dport 25 -j ACCEPT
```

Ou sinon en ouvrant tous les ports :

```
#!/bin/sh
echo "Flushing iptables rules..."
sleep 1
iptables -F
iptables -X
iptables -t nat -F
iptables -t nat -X
iptables -t mangle -F
iptables -t mangle -X
iptables -P INPUT ACCEPT
iptables -P FORWARD ACCEPT
iptables -P OUTPUT ACCEPT
```

dsn=5.4.6, status=bounced (mail for mail.mondomaine.fr loops back to myself)

- Le paramètre *mydestination* n'est pas bien définit.
- Le domaine est inactif dans la table MySQL[12]

Erreurs Dovecot

Error: Invalid settings in userdb

Survient quand on se connecte par exemple au webmail en root. Cette protection incite à utiliser un autre compte, la solution la plus simples est de définir un alias dans pour root dans /etc/aliases.

Si cela survient alors qu'un alias existait déjà, il suffit de relancer `newaliases` pour réparer.

Error: stat(/home/postmaster/Maildir/tmp) failed: Permission denied

```
chmod -R 777 /home/postmaster/Maildir/
```

Erreurs fetchmail

Les symptômes suivants sont liés à la même erreur fetchmail :

- POP3 : Erreur de login ou d'identification inconnue / erreur socket durant la réception
- IMAP : Connexion refusée / Échec de l'autorisation
- SMTP : The recipient server did not accept our requests to connect. Socket error

Pour déboguer, utiliser la commande :

```
fetchmail -v
```

Sinon, bien vérifier que l'utilisation et le mot de passe sont entre guillemets dans :

```
vim ~/.fetchmailrc
```

Et le chemin vers le programme dans :

```
vim ~/.forward
```

Les retours de mails sont visibles dans :

```
ls -alh /var/lib/fetchmail/Maildir/new
```

ou :

```
ls -alh /home/user/Maildir/new
```

Erreurs procmail

Mails perdus / delivered to command: procmail -a "$EXTENSION" / delivered to command: IFS=' '&&exec /usr/bin /procmail -f-||exit 75 #user

Si les logs montrent que les mails sont bien envoyés, mais restent introuvables par ailleurs, vérifier la boite système via :

```
procmail -v
```

Si cela commence par */var/mail/*, les emails perdus sont probablement tous dans le fichiers :

```
cat /var/mail/nobody
```

Pour les router vers les boites des utilisateurs, revoir la configuration des `.forward` et `.procmailrc` ci-dessus, jusqu'à voir les nouveaux dans :

```
ls -alh /home/user/Maildir/new
```

Unable to connect to remote host: Connection refused

```
/etc/init.d/postfix start
```

ou si les logs donnent :

```
NO [AUTHENTICATIONFAILED] Authentication failed.
```

alors réinstaller.

unknown key version / dkim=temperror (no key for signature)

Lorsqu'on envoie par webmail, il faut qu'un logiciel insère automatiquement la signature DKIM dans chaque courrier sortant. C'est pourquoi il faut installer OpenDKIM[13].

Lorsqu'il redémarre, le syslog peut indiquer des erreurs de permissions sur la clé qu'il faut corriger :

```
/etc/init.d/opendkim restart
tail /var/log/syslog
chown opendkim /etc/ssl/private/dkim.key
chmod 700 /etc/ssl/private/dkim.key
```

warning: cannot get RSA private key from file: nomdedomaine.fr.key disabling TLS support

Ceci apparait dans mail.log quand on chiffre la clef privée avec un mot de passe. Il ne faut donc pas en mettre.

warning: connect #1 to subsystem private/proxymap: Connection refused

Il manque la ligne dans master.cf :

```
proxymap  unix  -       -       n       -       -       proxymap
```

Références

1. http://doc.ubuntu-fr.org/postfix
2. https://www.digitalocean.com/community/tutorials/how-to-install-and-setup-postfix-on-ubuntu-14-04
3. http://www.troubleshooters.com/lpm/201202/images/dovecot_setup.png
4. *(fr)* Utilisation simple de procmail (http://www.linux-france.org/article/memo/procmail/node3.html)
5. *(en)* A Quick, Practical Procmail Guide (http://www.galtham.org/procmail.html)
6. http://wiki.dovecot.org/LDA/Postfix
7. http://www.clickz.com/clickz/column/1695095/hotmail-delivery-tips-sender-id-spf
8. https://www.isalo.org/wiki.debian-fr/Amavisd-new_et_DKIM
9. http://whatismyipaddress.com/blacklist-check
10. http://themediaonline.co.za/2011/07/the-five-rules-of-designing-email/
11. http://postfix.traduc.org/
12. http://unix.stackexchange.com/questions/128630/postfix-email-bounced-mail-for-domain-loops-back-to-myself
13. https://sourceforge.net/projects/opendkim/files/

Les annuaires LDAP

Configuration

Pour Debian Squeeze le fichier slapd.conf se situe dans **/usr/share/doc/slapd/examples/slapd.conf**. Il faut donc changer son emplacement avec **cp /usr/share/doc/slapd /examples/slapd.conf /etc/ldap/**. Puis modifier le fichier **/etc/default/slapd** et changer la ligne 'SLAPD_CONF=/etc/ldap/slapd.conf'.

La configuration de OpenLDAP se situe dans le fichier **/etc/ldap/slapd.conf** :

```
# more /etc/ldap/slapd.conf

# Les schémas à inclure à l'annuaire
include         /etc/ldap/schema/core.schema
include         /etc/ldap/schema/cosine.schema
include         /etc/ldap/schema/nis.schema
include         /etc/ldap/schema/inetorgperson.schema

# le fichier contenant le PID du processus
pidfile         /var/run/slapd/slapd.pid
# Le fichier contenant les arguments
argsfile        /var/run/slapd/slapd.args
# La facilité utilisée avec syslog
loglevel        none

# Emplacement des modules
modulepath     /usr/lib/ldap

# Chargment de modules permettant d'étendre les fonctionnalités de OpenLDAP
moduleload     back_hdb

# Limites concernant les recherches
sizelimit 500
tool-threads 1

#################################
# Définition du premier annuaire #
#################################
# On utilise un stockage basé sur le format de fichier '''HDB'''
backend         hdb
database        hdb

# On définit la ''racine'' de l'arbre (on parle de ''suffixe'' ou de ''basedn'')
suffix          "dc=domaine,dc=fr"

# Répertoire stockant les données de l'annuaire, dans notre cas, des fichiers au format '''HDB'''
directory       "/var/lib/ldap"

# L'administrateur de l'annuaire
rootdn          "cn=admin,dc=domaine,dc=fr"
rootpw          motdepasse        # en clair ou généré via la commande '''slappasswd'''

# Des paramêtres d'optimisation du format HDB
dbconfig set_cachesize 0 2097152 0
dbconfig set_lk_max_objects 1500
dbconfig set_lk_max_locks 1500
dbconfig set_lk_max_lockers 1500

# Les attributs à indexer en vue d'améliorer drastiquement les recherches
index           objectClass eq

# à commenter ...
lastmod         on
checkpoint      512 30

# contrôle d'accès aux attributs sensibles
access to attrs=userPassword,shadowLastChange
        by dn="cn=admin,dc=domaine,dc=fr" write
        by anonymous auth
        by self write
        by * none

# Annuaire avec accès anonyme en lecture seule
access to dn.base="" by * read
access to *
        by dn="cn=admin,dc=domaine,dc=fr" write
        by * read
```

Après chaque modification de ce fichier, on relance OpenLDAP avec la commande suivante :

```
# /etc/init.d/slapd restart
```

Création de l'annuaire

On crée le fichier LDIF qui va initialiser le *sommet* de l'arbre :

```
# cat cree_arbre.ldif
dn: dc=domaine,dc=fr
objectClass: top
objectClass: dcObject
objectClass: organization
dc: domaine
o: domaine
```

On injecte le fichier LDIF dans l'annuaire via la commande **ldapadd** :

```
# ldapadd -x -h localhost -W -D "cn=admin,dc=domaine,dc=fr" -f cree_arbre.ldif
Enter LDAP Password:
adding new entry "dc=domaine,dc=fr"
```

Pour créer les autres entrées, on peut désormais utiliser **PhpLdapAdmin**.

Les commandes d'administration OpenLDAP

Les programmes complémentaires

Les outils en ligne de commande

NB : les commandes suivantes sont fournies par le paquet Debian **ldap-utils**. Elles disposent des options génériques suivantes :

- **-h <adresse IP ou nom>** : permet d'indiquer le serveur hébergeant l'annuaire LDAP
- **-x** : on utilise l'authentification simple et non l'authentification TLS/SSL.
- **-b <basedn>** : permet d'indiquer l'annuaire à consulter (le basedn ou suffix)
- **-D <dn d'un objet>** : permet de s'authentifier avec le DN indiqué
- **-W** : demande la saisie du mot de passe du DN précédemment indiqué
- **-f <fichier LDIF>** : permet d'indiquer un fichier LDIF

ldapadd

La commande **ldapadd** permet d'ajouter des entrées dans l'annuaire LDAP.

Prenons par exemple le fichier LDIF suivant qui crée le suffixe (le basedn) de notre annuaire :

```
# cat cree_arbre.ldif
dn: dc=domaine,dc=fr
objectClass: top
objectClass: dcObject
objectClass: organization
dc: domaine
o: domaine
```

Pour injecter ce fichier LDIF dans notre annuaire :

```
# ldapadd -x -h localhost -D "cn=admin,dc=domaine,dc=fr" -f cree_arbre.ldif -W
```

Si tout se passe bien, vous devez voir des lignes :

```
adding new entry "dc=domaine,dc=fr"
```

ldapsearch

La commande **ldapsearch** permet d'interroger l'annuaire LDAP.

Affiche toutes les entrées de l'annuaire :

```
# ldapsearch -x -h localhost -b "dc=domaine,dc=fr"
```

Affiche les attributs de l'utilisateur *bob* via l'utilisation d'un filtre :

```
# ldapsearch -LLL -x -h localhost -b "dc=domaine,dc=fr" "(cn=bob)"
```

Affiche uniquement les attributs **uid** et **homeDirectory** de l'utilisateur *bob* :

```
# ldapsearch -LLL -x -h localhost -b "dc=domaine,dc=fr" "(cn=bob)" uid homeDirectory
```

ldapdelete

La commande **ldapdelete** permet de supprimer une entrée de l'annuaire LDAP. On indique à cette commande le(s) dn(s) à supprimer.

Supprime l'utilisateur bob :

```
# ldapdelete -x -h localhost -D "cn=admin,dc=domaine,dc=fr" -W "cn=bob,ou=People,dc=domaine,dc=fr"
```

ldapmodify

La commande **ldapmodify** permet de modifier une entrée de l'annuaire LDAP.

ldapmodrdn

La commande **ldapmodrdn** permet de renommer une entrée de l'annuaire LDAP. Cette opération revient à changer le **dn** (distinguished name) d'une entrée.

ldapcompare

La commande **ldapcompare** permet de comparer des entrées de l'annuaire LDAP.

ldappasswd

La commande **ldappasswd** permet de changer le mot de passe d'une entrée de l'annuaire LDAP.

ldapwhoami

La commande **ldapwhoami** permet de connaître avec quelle identité on est connecté à l'annuaire LDAP, c'est l'équivalent de la commande Unix *whoami*.

slapadd est utilisé pour ajouter des entrées spécifiées dans le format LDAP Directory Interchange Format (LDIF) dans une base de données slapd.

slapcat est utilisé pour générer une sortie LDIF LDAP basé sur le contenu d'une base de données slapd.

slapd est un serveur LDAP autonome.

slapindex est utilisé pour regénérer les index de slapd suivant le contenu actuel d'une base de données.

slurpd est un serveur réplicat autoname pour LDAP.

PhpLdapAdmin

PhpLdapAdmin est un logiciel qui permet d'administrer un annuaire LDAP via une interface Web.

Installation :

```
# apt-get install phpldapadmin
```

Configuration :

PhpLdapAdmin s'est configuré avec l'annuaire définit par défaut dans OpenLDAP. Si on a changé ce dernier, il faut modifier les deux lignes suivantes dans la configuration de PhpLdapAdmin :

```
# vi /etc/phpldapadmin/config.php
...
$ldapservers->SetValue($i,'server','base',array('dc=domaine,dc=fr'));
$ldapservers->SetValue($i,'login','dn','cn=admin,dc=domaine,dc=fr');
...
```

À chaque modification de ce fichier, il est préférable de redémarrer apache :

```
# /etc/init.d/apache2 restart
```

Accès à PhpLdapAdmin : http://localhost/phpldapadmin/

La réplication LDAP

Configuration du provider LDAP

On modifie le fichier /etc/ldap/slapd.conf du provider LDAP pour indiquer l'annuaire à répliquer, on rajoute les lignes en **gras** dans la configuration de l'annuaire et on modifie les attributs à indexer :

```
# vi /etc/ldap/slapd.conf
      ...
      # On indique à OpenLDAP de charger le module
      # permettant la réplication
      moduleload syncprov.la
      ...
      database bdb
      suffix dc=domaine,dc=fr
      rootdn dc=domaine,dc=fr
      directory /var/ldap/db
      index objectclass,entryCSN,entryUUID eq
      overlay syncprov
      syncprov-checkpoint 100 10
      syncprov-sessionlog 100
```

Configuration du consumer LDAP

On modifie **/etc/ldap/slapd.conf** du second OpenLDAP (le **consumer**) pour lui indiquer l'annuaire à répliquer et l'adresse du premier OpenLDAP (le **provider**).

```
# vi /etc/ldap/slapd.conf
...
database hdb
suffix dc=domaine,dc=fr
rootdn dc=domaine,dc=fr
directory /var/lib/ldap2
index objectclass,entryCSN,entryUUID eq

syncrepl rid=123
        provider=ldap://<adresse IP ou nom du provider LDAP>:389
        type=refreshOnly
        interval=00:00:00:10
        searchbase="dc=domaine,dc=fr"
        filter="(objectClass=*)"
        scope=sub
        attrs="*"
        schemachecking=off
        bindmethod=simple
        binddn="cn=admin,dc=domaine,dc=fr"
        credentials=admin
```

Il faut ensuite créer le répertoire du nouvel annuaire et le donner à l'utilisateur et au groupe *openldap* :

```
# mkdir /var/lib/ldap2
```

```
# chown openldap.openldap /var/lib/ldap2
```

Il ne reste plus qu'à relancer OpenLDAP :

```
# /etc/init.d/slapd restart
```

Validation de la réplication

L'objectif est de constater la réplication des objets de l'annuaire.

Procédure n°1 : validation de l'ajout de données sur le réplicat

Coté Provider

- on crée un fichier LDIF contenant l'utilisateur à rajouter

```
# cat bob.ldif
dn: cn=bob,ou=People,dc=domaine,dc=fr
objectClass: inetOrgPerson
objectClass: posixAccount
objectClass: top
givenName: Eponge
sn: Eponge
cn: bob
uid: bob
userPassword: {MD5}F7WfMlfPOGcecKZLEkmjTA==
uidNumber: 1001
gidNumber: 1001
homeDirectory: /home/bob
loginShell: /bin/sh
mail: bob@domaine.fr
```

- on injecte ce fichier LDIF dans notre annuaire

```
# ldapadd -x -h localhost -D "cn=admin,dc=domaine,dc=fr" -f bob.ldif -W
```

- on vérifie que l'utilisateur a bien été crée

```
# ldapsearch -LLL -x -h localhost -b "dc=domaine,dc=fr" "(cn=bob)"
```

Coté Consumer

- on attend quelques secondes et on constate que l'utilisateur a bien été créé sur le réplicat

```
# ldapsearch -LLL -x -h localhost -b "dc=domaine,dc=fr" "(cn=bob)"
```

Procédure n°2 : validation de la suppression de données sur le réplicat

Coté Provider

- on supprime l'utilisateur de notre annuaire

```
# ldapdelete -x -h localhost -D "cn=admin,dc=domaine,dc=fr" -W "cn=bob,ou=People,dc=domaine,dc=fr"
```

- on vérifie que l'utilisateur a bien été supprimé

```
# ldapsearch -LLL -x -h localhost -b "dc=domaine,dc=fr" "(cn=bob)"
```

Coté Consumer

- on attend quelques secondes et on constate que l'utilisateur a bien été supprimé sur le réplicat

```
# ldapsearch -LLL -x -h localhost -b "dc=domaine,dc=fr" "(cn=bob)"
```

L'outil d'administration Webmin

Installation de Webmin

Webmin est un logiciel qui permet de configurer un serveur Unix / Linux via une interface Web[1].

Webmin est disponible sur de nombreuses plateformes[2] : Windows, Redhat, Fedora, CentOS, SuSE, Mandrake, Debian, et Ubuntu[3].

Il est supporté par Debian[4], mais on peut aussi télécharger sur leur site un fichier .deb.

On installe les dépendances de Webmin, puis on télécharge l'archive de Webmin et on l'installe en root :

```
apt-get -f install libnet-ssleay-perl libauthen-pam-perl libio-pty-perl libmd5-perl
wget http://prdownloads.sourceforge.net/webadmin/webmin_1.780_all.deb
dpkg -i webmin_1.780_all.deb

# anciennement :
apt-get install libnet-ssleay-perl libauthen-pam-perl libio-pty-perl libmd5-perl
wget http://prdownloads.sourceforge.net/webadmin/webmin_1.400_all.deb
dpkg -i webmin_1.400_all.deb
```

Quelques secondes après l'interface web doit être accessible (ex : https://localhost:10000/).

Configuration de Webmin

Changement du mot de passe

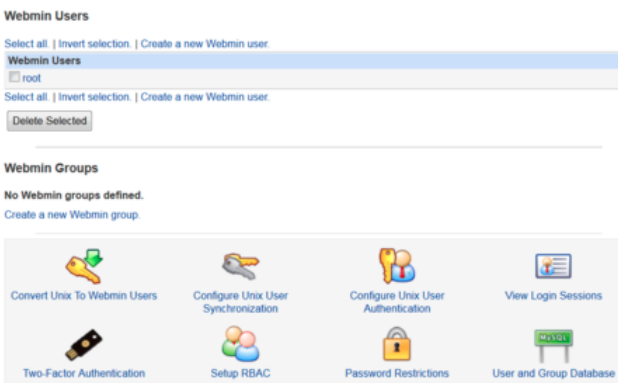
```
# /usr/libexec/webmin/changepass.pl /etc/webmin root nouveau_pass
```

Cela changera le mot de passe de root dans Webmin et ne modifie en aucun cas le mot de passe root du système. Pensez à supprimer ensuite la ligne de votre historique (.bash_history par exemple) afin que le mot de passe n'y reste pas enregistré en clair.

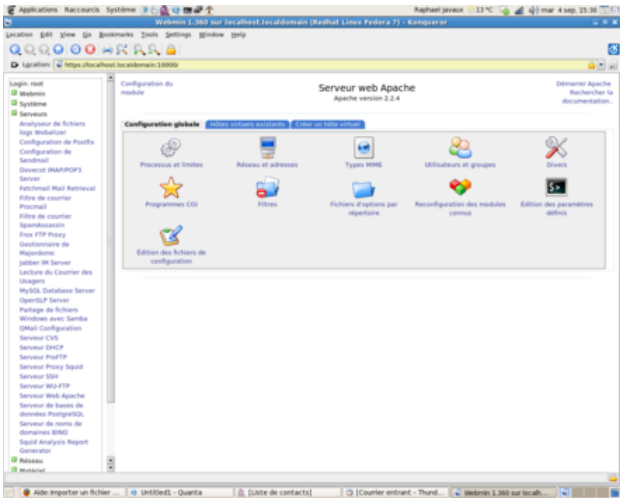
Accueil de Webmin

Gestion des utilisateurs de Webmin

Webmin permet aux utilisateurs Unix de se connecter au portail Webmin, et à certaines bases de données.



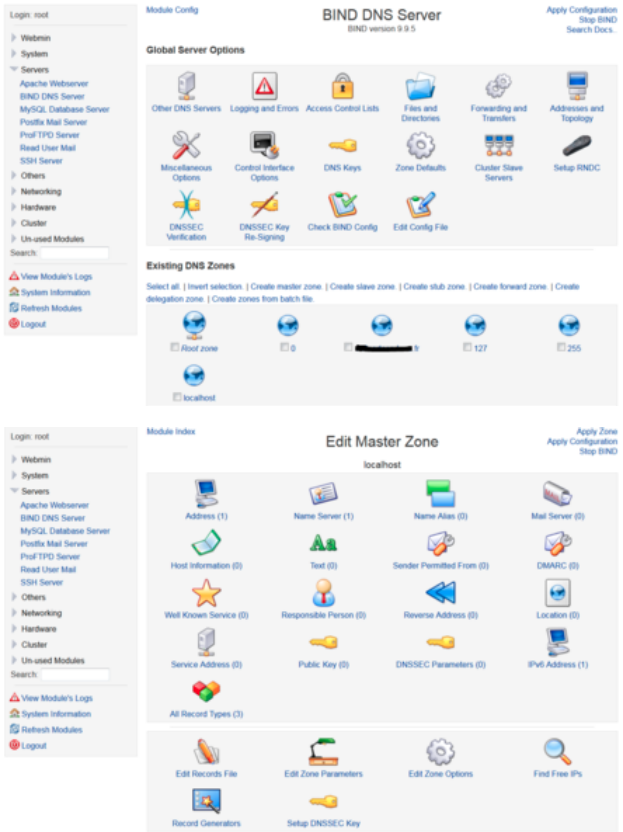
Gestion des serveurs HTTP

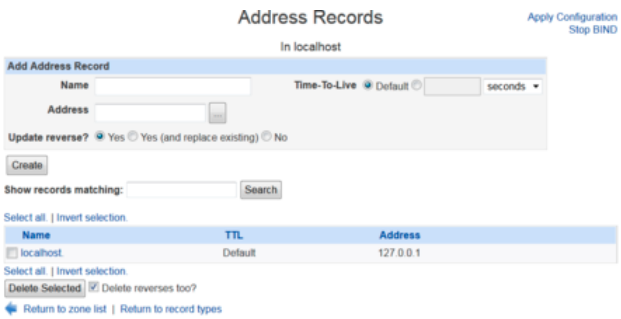


Gestion des serveurs BDD

Il est possible de créer des bases MySQL, des tables et d'exécuter du code SQL via l'interface. Elle se révèle être moins développée que phpMyAdmin.

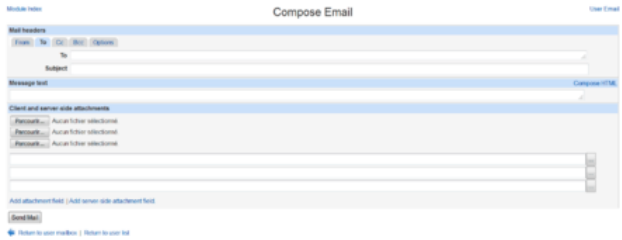
Gestion des serveurs DNS





Gestion des serveurs Mail

Il est possible d'utiliser les boites emails des utilisateurs depuis l'interface Webmin, pour envoyer et recevoir (du moins théoriquement via `send_mail.cgi`) :



Leurs logs étant :

```
tail /var/webmin/webmin.log
```

Si les envois ne partent pas, tester en shell :

```
cd /usr/share/webmin/mailboxes
./send_mail.cgi
```

S'il y a une erreur sur les variables d'environnements :

```
printenv
export WEBMIN_CONFIG="/etc/webmin"
export PERLLIB="/usr/lib/x86_64-linux-gnu/perl5"
```

Gestion des sauvegardes

Des backups peuvent être planifiés ou effectués en live travers l'interface :

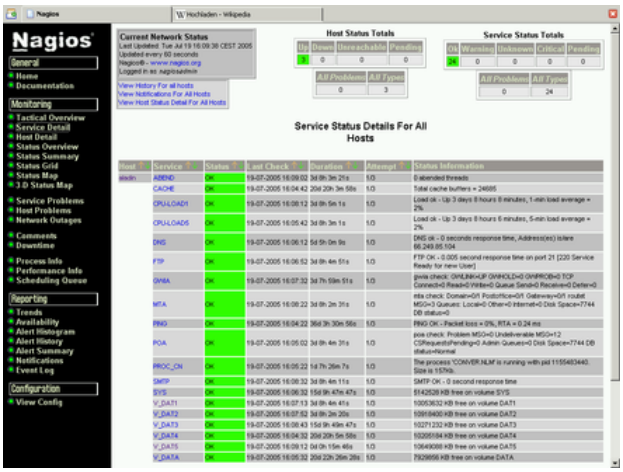
Références

1. http://www.webmin.com
2. http://www.webmin.com/download.html
3. http://doc.ubuntu-fr.org/webmin
4. http://www.debianadmin.com/install-webmin-on-debian-7-6-wheezy.html

La supervision

Nagios

Nagios s'installe assez facilement pour surveiller, archiver et représenter les périodes où les machines sont en ligne, leurs services sont lancés et leurs ressources disponibles[1] :



Il peut ainsi prévenir automatiquement par email, chaque contact associé à une machine quand l'état d'un service change (ex : processeur saturé, disque dur plein à 90 %, page web inaccessible...).

Installation du serveur

Le paquet Ubuntu s'arrête à la version 3 :

```
apt-get install nagios3
```

Sinon, pour être sûr d'obtenir la dernière version, il est possible de récupérer le tarball en renseignant son nom sur le site officiel[2], puis de l'installer à partir de l'étape 2 en root :

```
wget https://assets.nagios.com/downloads/nagioscore/releases/nagios-4.0.8.tar.gz

useradd nagios
groupadd nagcmd
usermod -a -G nagcmd nagios

tar xzf nagios-4.0.8.tar.gz
cd nagios-4.0.8
./configure --with-command-group=nagcmd
make all
make install
make install-init
make install-config
make install-commandmode

/etc/init.d/nagios start
```

Enfin, pour ajouter un service à monitorer, il suffit de modifier les fichiers .cfg.

Interface graphique

Le portail web (ex : http://127.0.0.1/nagios/) a pour prérequis Apache. Il s'installe comme suit :

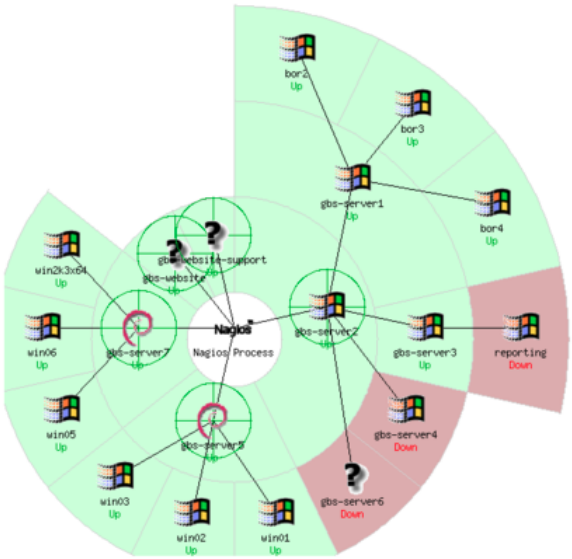
```
make install-webconf
htpasswd -c /usr/local/nagios/etc/htpasswd.users nagiosadmin
/etc/init.d/apache2 reload
```

Par défaut il est protégé par le mot de passe du compte *nagiosadmin*.

Par ailleurs, la page d'accueil par défaut ne montrant pas les services monitorés, il peut être rentable d'utiliser `http://127.0.0.1/nagios/?corewindow=`

`/nagios/cgi-bin/status.cgi?host=all` à la place.

Le menu *Map* permet un affichage sous forme de carte :



Fichiers de configuration .cfg

Pour configurer ce que Nagios doit surveiller, on doit éditer les fichiers de configuration .cfg importés depuis `/usr/local/nagios/etc/nagios.cfg` (généralement tout ce qui est dans /usr/local/nagios/etc/objects/).

Que la surveillance soit opérationnelle, ces fichiers doivent comporter les quatre objets suivants[3] (par défaut le classement est d'un type d'objet par fichier mais un .cfg peut tous les contenir) :

```
define command {
        command_name
        command_line
}

define service {
        use
        host_name
        service_description
        check_command
}

define host {
        use
        host_name
        alias
        address
        check_command
}

define contact {
        name
}
```

L'objet commande pointe sur le fichier exécutable du système d'exploitation qui sera exécuté par Nagios (en Shell, Perl, Java, Python...). Ex :

```
define command {
  command_name    check_sftp
  command_line    java -jar $USER1$/check_sftp.jar -port=22 -t=10 -temp=/tmp/ -host=$ARG1$ -user=$ARG2$ -pwd=$ARG3$
-warning=$ARG4$ -critical=$ARG5$
}
```

Plusieurs objets services peuvent ensuite appeler une même commande avec différents arguments (ex : SFTP du serveur 1, du serveur 2...).

Nagios 3

```
cd /etc/nagios3/conf.d
```

```
vim localhost_nagios2.cfg   # (ajouter la machine)
vim hostgroups_nagios2.cfg  # (définir l'emplacement de la machine sur le site Nagios)
vim services_nagios2.cfg    # (préciser ce qu'il faut surveiller sur la machine, ex : HTTP)
/etc/init.d/nagios3 restart
```

Nagios 4

```
cd /usr/local/nagios/etc/objects
vim localhost.cfg
/etc/init.d/nagios restart
```

> **Attention !**
>
> Avant de mettre à jour Nagios ou Linux, bien sauvegarder la configuration de `/usr/local/nagios/etc/` (ou `/etc/nagios3/`) selon la version.

Plugins

Les plugins permettent de monitorer plus de services, ils sont téléchargeables depuis https://www.nagios.org/download/plugins :

```
tar xzf nagios-plugins-2.0.3.tar.gz
cd nagios-plugins-2.0.3
./configure --with-nagios-user=nagios --with-nagios-group=nagios
make
make install
```

Addons

Les addons ajoutent des fonctionnalités supplémentaires, comme des représentations graphiques : https://www.nagios.org/download/addons/.

Surveillance SFTP

- En shell (https://exchange.nagios.org/directory/Tutorials/Other-Tutorials-And-HOWTOs/Check-SFTP-Availability-With-check_sftp_avail-In-Nagios/details).
- En Java (https://exchange.nagios.org/directory/Plugins/Network-Protocols/SFTP/check_sftp/details).
- En Perl (https://exchange.nagios.org/directory/Plugins/Network-Protocols/FTP/check_ftp_rw-%28w-2FSFTP-support%29/details).

Test de formulaire HTML

- https://exchange.nagios.org/directory/Plugins/Websites,-Forms-and-Transactions/Check-form/details

Installation d'un client

Il suffit d'installer http://www.nsclient.org/ en administrateur sur la machine à surveiller. Parfois le pare-feu doit être ouvert sur le port 5666 ou 12489.

Sur Ubuntu : `apt-get install xinetd`.

Problèmes connus

- Les erreurs du serveur peuvent être obtenues avec `tail /usr/local/nagios/var/nagios.log`.
- Celles du client Windows dans `C:\Program Files\NSClient++\nsclient.log` (si configuré dans le nsclient.ini).

connect to address 127.0.0.1 and port 12489: Connexion refusée

Si le pare-feu est déjà ouvert à Nagios ou au port 12489, et si `C:\Program Files\NSClient++\nsc.ini` autorise déjà l'IP du serveur Nagios, et que le processus nscp.exe est bien lancé, c'est peut être qu'avec Nagios 4 il faut utiliser `C:\Program Files\NSClient++\nsclient.ini`.

Normalement ensuite un `telnet 127.0.0.1 12489` depuis le client fonctionne.

Erreurs sur `/usr/local/nagios/var/spool/checkresults`

Ex :

```
Error in configuration file '/usr/local/nagios/etc/nagios.cfg' - Line 452 (Check result path '/usr/local/nagios/var/spool
```

```
/checkresults' is not a valid directory)
Error: Unable to write to check_result_path ('/usr/local/nagios/var/spool/checkresults') - Permission denied
```

En effet, `/usr/local/nagios/var/spool/checkresults` est nécessaire au lancement du processus Nagios 4, et peut être créé manuellement à cet effet :

```
mkdir /usr/local/nagios/var/spool/
mkdir /usr/local/nagios/var/spool/checkresults/
chown nagios /usr/local/nagios/var/spool/checkresults
chgrp nagios /usr/local/nagios/var/spool/checkresults
```

Error: Could not open command file '/usr/local/nagios/var/rw/nagios.cmd' for update!

Si depuis le portail web on ne peut pas planifier de maintenance à cause de cette erreur, alors que les permissions du fichier semblaient correctes, il faut modifier le fichier suivant[4] :

```
vim /etc/init.d/nagios
```

Rechercher la ligne démarrant par "chown $NagiosUser:$NagiosGroup $NagiosRunFile", puis ajouter en dessous :

```
sleep 10
chmod 666 /usr/local/nagios/var/rw/nagios.cmd
```

```
/etc/init.d/nagios restart
```

Internal Server Error

Si la page d'accueil fonctionne mais pas les vues monitoring, c'est certainement que le CGI n'arrive pas à s'exécuter.

Parfois l'erreur est notée plus clairement : `You don't have permission to access /cgi-bin/nagios3/status.cgi on this server.`

It appears as though you do not have permission to view information for any of the services you requested

Dans l'interface web, passer "use_authentication=1" à 0 dans cgi.cfg.

Kernel panix

Peut se produire si la RAM est insuffisante (ex : < 512 Mo sur Ubuntu 16.04).

Network Unreachable

Si le serveur Nagios ping une machine qui fonctionne, mais qu'elle y apparait comme injoignable, c'est à cause de la différence entre le ping IPv4 et le ping6. Il faut juste modifier le *check-host-alive* de *command.cfg* en ajoutant "-4" à la fin :

```
define command{
        command_name    check-host-alive
        command_line    $USER1$/check_ping -H $HOSTADDRESS$ -w 3000.0,80% -c 5000.0,100% -4
}
```

NSClient - ERROR: Could not get data for 5 perhaps we don't collect data this far back?

Un reboot du service ne change rien, il faut redémarrer l'OS.

NSClient - ERROR: Could not get value

Idem que ci-dessus.

NSClient - ERROR: Failed to get PDH valuee

Idem que ci-dessus.

NSClient - ERROR: Invalid password

Sur le client Windows, modifier dans `C:\Program Files\NSClient++\NSC.ini`, la ligne `password=`, afin qu'il corresponde à celui définit sur le serveur, dans `/usr/local/nagios/etc/objects/resource.cfg` à la ligne `$USER4$=`. Ou vice-versa.

Sinon réinstaller le client, sa version n'est peut-être plus à jour.

Si la commande suivante fonctionne depuis le serveur, c'est qu'il faut compléter `commands.cfg` :

```
/usr/local/nagios/libexec/check_nt -H Mon_IP_Cliente -v USEDDISKSPACE -p 12489 -l c -s Mon_Mot_De_Passe
```

Status UNKNOWN, Status Information Utilisation:

La connexion au client Nagios fonctionne, mais le statut est flou : il faut réinstaller et reconfigurer Nagios client (problème de version avec le serveur incompatible).

Warning: Host 'xxx' has no default contacts or contactgroups defined!

Survient dans les logs au lancement de Nagios pour avertir qu'en cas d'alerte sur la machine mentionnée, personne ne sera prévenu.

Pour y remédier, vérifier la ligne *contact_groups* dans *template.cfg* :

```
define host{
 name                    xxx
 contact_groups          admins
 ...
}
```

Références

1. https://library.nagios.com/library/products/nagioscore/manuals/
2. https://www.nagios.org/download/core/
3. http://doc.ubuntu-fr.org/nagios
4. http://alexnogard.com/error-could-not-open-command-file-usrlocalnagiosvarrwnagios-cmd-for-update/

- http://djibril.developpez.com/tutoriels/linux/nagios-pour-debutant/

Voir aussi

- en:System Monitoring with Xymon

Installation d'un service en mode chroot

La commande chroot permet de faire fonctionner des applications Linux dans un environnement protégé. En fonctionnant dans un "root directory" virtuel, les applications sont alors cloisonnées dans un espace qui leur est propre, évitant ainsi les éventuels méfaits commis à l'encontre du système en cas d'infraction.

chroot est un outil GNU faisant partie de GNU Core Utilities, plus précisément des *shellutils*.

Protection avec iptables

Iptables est un outil permettant de paramétrer Netfilter le filtre de paquet intégré à Linux.

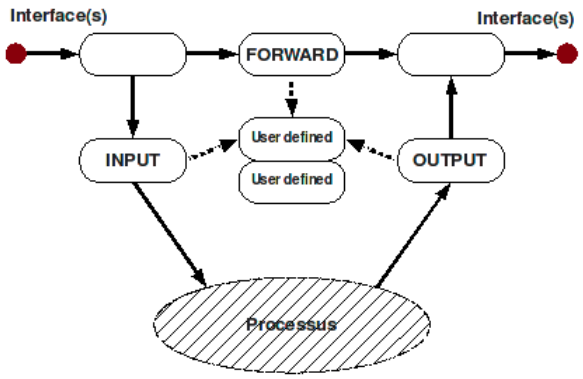
Principe des tables

Netfilter utilise une liste de règles inscrites dans trois tables : INPUT, OUTPUT, FORWARD.

Le paquet arrivant est voué à:

- soit être filtré suivant les règle de la table **INPUT** pour être utilisé par un processus de la machine
- soit être transféré à une autre machine en respectant les règle de la table **FORWARD**
- soit quitter la machine dans les condition de la table **OUTPUT**

Le but est donc de modifier les tables en ajoutant et supprimant des règles dans les différentes tables. pour cela nous utilisons la commande *iptables*.



Syntaxe globale

```
#iptables <une action sur une table de netfilter> <la table en question> <condition(s) d'application> -j <action sur le paquet>
```

```
#exemple
iptables  -A  INPUT  --p icmp  -j  ACCEPT
#on ajoute à la table INPUT la règle ACCEPT pour les paquet icmp (ping) entrant,
#bien entendu il faut aussi accepter les paquet sortants. D'où :
iptables  -A  OUTPUT  --p icmp  -j  ACCEPT
```

Les actions sur les tables

Le principe

Par défaut, iptables est composé de 3 catégories appelées "chaînes" (chain) :

- INPUT : Chaîne où l'on règle le trafic entrant.
- FORWARD : Chaîne où l'on règle le trafic qui sera redirigé.
- OUTPUT : Chaîne où l'on règle le trafic sortant.

A chaque chaîne est attribuée une "politique" (policy), les deux principales sont :

- ACCEPT : Politique d'acceptation. Par défaut, tout ce qui n'est pas dans la chaîne est accepté.
- DROP : Politique de refus. Par défaut, tout ce qui n'est pas dans la chaîne sera rejeté. (utile pour filtrer les connections entrantes)

Un exemple concret. Par défaut, si vous n'avez jamais touché à votre iptables, voici ce que vous devez avoir :

```
# iptables -L
Chain INPUT (policy ACCEPT)
target     prot opt source               destination

Chain FORWARD (policy ACCEPT)
target     prot opt source               destination

Chain OUTPUT (policy ACCEPT)
target     prot opt source               destination
```

On constate ici que rien n'est sécurisé : tout entre (Chain INPUT, policy ACCEPT), tout sort (Chain OUTPUT, policy ACCEPT).

Réglage simple

Tout d'abord, si votre iptables ne ressemble pas à l'exemple ci-dessus, ou tout simplement si vous souhaitez réinitialiser vos réglages, tapez ces commandes :

```
# iptables -F
(Efface toutes les règles définies par l'utilisateur)
# iptables -X
(Efface toutes les chaînes définies par l'utilisateur)
```

Les commandes de base :

Les règles sont lues dans un ordre précis. De la première vers la dernière. De cette manière, dans l'hypothèse où la première règle interdit une connexion alors que la suivante l'autorise, la connexion sera interdite.

- APPEND : Ajouter une règle dans une chaîne. La règle ainsi créée s'ajoute après la dernière règle.

```
# iptables -A INPUT -m state --state ESTABLISHED,RELATED -j ACCEPT
(Ajoute une règle dans la chaîne INPUT. L'exemple sera détaillé plus loin.)
```

- INSERT : Insère une règle dans un chaîne. La règle ainsi créée s'ajoute avant la première règle.

```
# iptables -I INPUT -m state --state ESTABLISHED,RELATED -j ACCEPT
(Insère une règle au début de la chaîne INPUT. L'exemple sera détaillé plus loin.)
```

- LIST : Liste les règles existant dans toutes les chaînes.

```
# iptables -L
Chain INPUT (policy DROP)
target     prot opt source               destination
ACCEPT     all  --  anywhere             anywhere            state RELATED,ESTABLISHED
ACCEPT     tcp  --  anywhere             anywhere            tcp dpt:ssh
ACCEPT     tcp  --  anywhere             anywhere            tcp dpt:www
ACCEPT     icmp --  anywhere             anywhere
ACCEPT     tcp  --  anywhere             anywhere            tcp dpt:pop3s

Chain FORWARD (policy ACCEPT)
target     prot opt source               destination

Chain OUTPUT (policy ACCEPT)
target     prot opt source               destination
```

- DELETE : Supprime une règle d'une chaîne.

```
# iptables -D INPUT 2
(Supprime la règle n°2 de la chaîne INPUT.)
```

- POLICY : Configure la politique d'une chaîne.

```
# iptables -P INPUT DROP
(Passe la politique de la chaîne INPUT en DROP. Toutes les connections seront rejetées par défaut.)
```

Créer une configuration de base :

Nous allons autoriser une connexion déjà établie (ESTABLISHED) à recevoir du trafic.

```
# iptables -A INPUT -m state --state ESTABLISHED,RELATED -j ACCEPT
```

Autorisons maintenant le trafic sur deux ports utilisés par des services très répandus / utilisés. Bien évidemment à vous de faire le tri en fonction de ce que vous utilisez ou non.

- Trafic SSH :

```
# iptables -A INPUT -p tcp -i eth0 --dport ssh -j ACCEPT
```

- Trafic Web :

```
# iptables -A INPUT -p tcp -i eth0 --dport 80 -j ACCEPT
```

- Requêtes Ping :

```
# iptables -A INPUT -p icmp -j ACCEPT
```

- Trafic Local :

```
# iptables -I INPUT 2 -i lo -j ACCEPT
Nous insérons cette règle en deuxième position afin qu'elle prime sur les autres.
Comme il a déjà été dit plus haut, les règles sont parcourues de haut en bas.
```

Voici ce que vous devriez avoir pour l'instant si vous listez les filtres (l'option -v permet de voir sur quelle interface la règle est appliquée) :

```
# iptables -L -v
 Chain INPUT (policy ACCEPT 155 packets, 21332 bytes)
  pkts bytes target     prot opt in     out     source               destination
    20  1562 ACCEPT     all  --  any    any     anywhere             anywhere            state RELATED,ESTABLISHED
     0     0 ACCEPT     all  --  lo     any     anywhere             anywhere
     0     0 ACCEPT     tcp  --  eth0   any     anywhere             anywhere            tcp dpt:ssh
     0     0 ACCEPT     tcp  --  eth0   any     anywhere             anywhere            tcp dpt:http
     0     0 ACCEPT     icmp --  any    any     anywhere             anywhere

 Chain FORWARD (policy ACCEPT 0 packets, 0 bytes)
  pkts bytes target     prot opt in     out     source               destination

 Chain OUTPUT (policy ACCEPT 343 packets, 59235 bytes)
  pkts bytes target     prot opt in     out     source               destination
```

Comme vous pouvez le constater, la politique est une politique d'acceptation (policy ACCEPT). Donc tous les paquets entrent sans restriction. En résumé, jusque là, nos modifications n'ont servies à rien. Nous allons donc devoir modifier la politique en politique de refus (policy DROP).

```
# iptables -P INPUT DROP
```

La modification est effective immédiatement. A partir de maintenant, seuls les paquets tcp entrants sur les ports 80, 22 ainsi que les requêtes ICMP sont acceptés.

Ces modifications ne seront pas prises en compte au redémarrage. Il faut pour cela créer un fichier script contenant toutes les commandes précédentes.

```
#!/bin/sh
iptables -A INPUT -m state --state ESTABLISHED,RELATED -j ACCEPT
iptables -A INPUT -p tcp -i eth0 --dport ssh -j ACCEPT
iptables -A INPUT -p tcp -i eth0 --dport 80 -j ACCEPT
iptables -A INPUT -p icmp -j ACCEPT
iptables -I INPUT 2 -i lo -j ACCEPT
iptables -P INPUT DROP
```

Créez ce fichier dans le répertoire */etc/init.d/* et n'oubliez pas de le rendre exécutable (`chmod 700 nom_du_fichier`). Enfin, utilisez la commande update-rc.d afin que ce script soit exécuté au démarrage.

```
 # update-rc.d nom_du_fichier defaults
```

Logs

Par défaut, les journaux d'iptables ne renseignent pas les actions effectuées par le par-feu. Pour les obtenir, il faut définir un niveau de log, par exemple[1] :

```
iptables -t filter -A FORWARD -p all -j LOG --log-level debug
iptables -t filter -A INPUT -p all -j LOG --log-level debug
iptables -t filter -A OUTPUT -p all -j LOG --log-level debug
```

Ensuite les opérations sont visibles dans le syslog.

Conditions

Adresses IP et ports en destination et en source

Pour sécuriser un serveur FTP, en n'autorisant que son adresse IP et interdisant toutes les autres à se connecter au Linux sur le port 21 :

```
iptables -A INPUT -i eth0 -p tcp --dport 21 -s 144.76.38.140 -j ACCEPT
iptables -A INPUT -i eth0 -p tcp --dport 21 -j DROP
```

Limites

Pour se prémunir des attaques DDoS on peut limiter le trafic à un certain nombre de connexions par minute sur un port :

```
iptables -A INPUT -p tcp --dport 80 -m state --state NEW -m limit --limit 500/minute --limit-burst 2000 -s 144.76.38.140 -j
ACCEPT
```

Protocoles


Cette section est vide, pas assez détaillée ou incomplète.

États


Cette section est vide, pas assez détaillée ou incomplète.

Suivi

conntrack


Cette section est vide, pas assez détaillée ou incomplète.

Les actions sur les paquets


Cette section est vide, pas assez détaillée ou incomplète.

Références

1. http://www.inetdoc.net/guides/iptables-tutorial/logtarget.html

Médiagraphie

A. S. Tanenbaum, « Modern Operating Systems », Prentice Hall, 951 p., 2001.

S. Pierre, « Introduction aux ordinateurs : organisation, exploitation et programmation », Télé-université, 253 p., 1996.

Ellen Siever et al., « Linux in a nutshell : a desktop quick reference », O'Reilly and Associates, 2000.

Jeffrey Dean, « LPI Linux Certification in a Nutshell: a desktop quick reference », O'Reilly and Associates, 2000.

Renaldo Garcia, "Linux – Un système d'exploitation Unix », Éditions Vermette, 2000.

Auteurs

L'écriture de ce livre a été initié par Stéphane Gill. Les deux premiers chapitres sont basés sur les notes de cours "Système d'exploitation" de Stéphane Gill utilisées au collège Ahuntsic.

Les chapitres suivants ont été écrits par Alexandre GUY et les étudiants de la formation **Administrateur Systèmes & Réseaux (http://perso.univ-perp.fr/mceljai/Admisys/)** de l'Université de Perpignan Via Domitia (UPVD) (http://www.univ-perp.fr).

Merci également aux autres wikinautes qui ont participés à l'élaboration de ce wikibook en le complétant et corrigeant.
