HEH.be
Sciences
et technologies

Avenue V. Maistriau 8a
B-7000 Mons
+32 (0)65 33 81 54
scitech-mons@heh.be

WWW.HEH.BE

UE : Linux

- **AA :** Administration Linux - théorie
  Antoine Malaise

Bachelier en Informatique

Orientation réseaux & télécommunications

Antoine Malaise
antoine.malaise@heh.be

WALLONIE-BRUXELLES ENSEIGNEMENT
pôle hainuyer

Bachelier en informatique et systèmes, finalité réseaux et télécommunications

Linux

Titulaire du cours : Malaise Antoine
Cours co-écrit par : Mandoux Denis

Table des matières

Module 1 : Introduction à GNU/Linux

1. Les logiciels libres

Les logiciels libres c'est quoi ?

De manière générale, un logiciel libre peut se définir comme étant un logiciel fourni avec les autorisations pour quiconque :

- d'utiliser le logiciel,
- de copier le logiciel,
- d'étudier le logiciel (ce qui nécessite un accès au code source du logiciel),
- de modifier le logiciel (ce qui nécessite un accès au code source du logiciel)
- de distribuer le logiciel, soit dans sa forme originale, soit avec des modifications.

Les logiciels libres sont-ils gratuits ?

Un logiciel libre n'est pas nécessairement gratuit, même si dans la pratique la plupart d'entre eux se trouvent "gratuitement" sur Internet. De plus, il y a toujours des frais "cachés" qui font que probablement rien n'est totalement gratuit. Ainsi, pour télécharger un logiciel sur Internet, il faudra nécessairement disposer d'une connexion à Internet (qui n'est pas gratuite), d'un ordinateur qui consomme du courant, d'un CD pour y graver votre logiciel, …

Au niveau d'une entreprise il y aura probablement d'autres frais qui entreront en ligne de compte : frais de migration vers un logiciel libre, de maintenance, de formation du personnel, etc.

Libre donc pas de licences ?

Au contraire, il existe une multitude de licences, même dans le monde des logiciels libres, certaines étant plus restrictives que d'autres. La plus connue est sans doute la licence GPL[1] (General Public Licence) qui assure que le logiciel est libre et restera libre (d'utilisation, copie, distribution et modification), tout en protégeant les droits des développeurs. Cette licence stipule que l'auteur du logiciel demeure toujours le propriétaire du code, mais n'impose aucune restriction à l'utilisateur quant aux modifications qu'il pourrait y apporter. Les utilisateurs peuvent à leur guise, modifier le code source et redistribuer cette nouvelle version, **à la condition** qu'elle soit elle-même sous licence GPL.

La licence a évolué vers la GPL version 2 et GPL version 3 (depuis 2007) qui ne seront pas abordées dans ce cours. Pour plus de détails, vous pouvez vous référez aux sites suivants :

- Version originale de la GPL3 : http://www.gnu.org/licenses/gpl.html
- Traduction non officielle de la GPL3 : http://www.rodage.org/gpl-3.0.fr.html
- Traduction non officielle de la GPL2 : http://www.linux-france.org/article/these/gpl.html

[1] C'est la plus connue, mais un logiciel libre n'est pas nécessairement sous licence GPL.

Et l'Open Source ?

Un logiciel « open source » est un logiciel qui répond à une série de critères[1] dont le principal est la disponibilité du code source qui rend possible l'étude du logiciel. Pour plus de détails sur les licences "open source", vous pouvez consultez le site http://www.opensource.org/

2. Qu'est-ce que Linux ?

2.1. Bref historique

Au milieu des années 1980, Richard Stallman fonde la FSF (Free Software Foundation, Fondation pour les logiciels libres) dont l'objectif était de développer des logiciels libres de droits, contrairement aux logiciels propriétaires. Ainsi, des dizaines de logiciels libres ont été développés dans le cadre du projet GNU[2] (le shell BASH, Ghostscript, GNU C compiler (la bibliothèque glibc), etc.

En 1991, un étudiant finlandais, Linus Torvalds, a développé un noyau qu'il baptisa Linux et mis sous licence GPL. Ceci encouragea l'intégration de Linux et du système GNU en ce que l'on appelle le système GNU/Linux[3] (qui fit sa première apparition en 1993). L. Torvalds distribua son système sur Internet en invitant quiconque à l'aider à l'améliorer. Ainsi, des développeurs du monde entier ont aidé au développement de Linux. Par abus de langage, on appelle aujourd'hui Linux l'ensemble du système GNU/Linux ainsi que les applications libres qui l'accompagnent.

2.2. Le système d'exploitation GNU/Linux

Linux est le noyau d'un système d'exploitation (de type UNIX, on parle dès lors de "UNIX like"), entièrement libre et ouvert.

Un système d'exploitation est une couche d'instructions logicielles permettant de faire le lien entre le matériel (le hardware : les disques, la mémoire, etc.) et les programmes d'application (le software : traitement de texte, tableur, navigateur Internet, etc.). Le cœur d'un système d'exploitation est constitué par un noyau fournissant les fonctions de base du système :

- le chargement et l'exécution des processus ainsi que le partage du processeur entre les processus ;
- la gestion des entrées/sorties ;
- la mise à disposition d'une interface permettant aux utilisateurs d'exécuter des programmes et accéder au système de fichiers.

Hormis ces fonctions de base, les systèmes d'exploitation peuvent aussi fournir d'autres services utiles à l'exploitation d'un ordinateur :

- une structure pour stocker les informations sur le disque dur : le système de fichiers,
- le support du réseau (TCPIP, pare-feu,…),
- des pilotes (drivers) pour le matériel,
- …

[1] Les licences libres correspondent généralement à ces critères

[2] GNU est un sigle GNU's not Unix dont l'objectif était de produire un système d'exploitation complètement libre.

[3] Dans le reste du cours, nous utiliserons indifféremment Linux ou GNU/Linux pour faire référence à un système utilisant un noyau Linux.

Linux, un système libre

Le système GNU/Linux est placé sous licence GPL, il est donc libre et chacun a le droit de l'utiliser comme bon lui semble, d'en comprendre le fonctionnement voire de modifier le code afin de répondre à ses propres besoins. Attention de ne pas confondre un logiciel libre à un freeware[1] ou un shareware[2] dont le code source n'est pas fourni.

A qui est destiné Linux ?

Tout le monde peut utiliser Linux, depuis les particuliers jusqu'aux développeurs d'applications, fournisseurs de connectivités réseau, programmeurs système, etc. Ainsi, Linux est utilisé dans les entreprises, les hôpitaux, les écoles, les administrations, etc. Des systèmes Linux ont, par exemple, été utilisés dans des stations de recherche de l'Antarctique, dans l'espace (NASA) ainsi que pour la réalisation d'effets spéciaux pour le cinéma, pour ne citer que quelques exemples.

2.3. Les distributions Linux

Une distribution Linux est constituée d'un système d'exploitation à base d'un noyau Linux et d'un ensemble d'applications. Celles-ci peuvent être des logiciels libres mais aussi propriétaires.

Il n'y a pas une distribution standard unique, chaque distribution possède ses propres spécificités : sa propre documentation, sa propre procédure d'installation, ses propres outils d'administration, etc. Les applications disponibles varient également d'une distribution à l'autre.

Parmi les distributions « généralistes », on citera : RedHat, fédora, Suse, Debian, Ubuntu, Mandriva, Slackware ou encore Gentoo. Cependant, il existe probablement plus de 300 distributions différentes. Vous en trouverez une série d'autres listée sur ce site : http://fr.wikipedia.org/wiki/Liste_des_distributions_Linux

Les distributions peuvent être très différentes les unes des autres, et en dehors des distributions généralistes, certaines se spécialisent dans un domaine particulier. Ainsi il existe des distributions :

- Minimalistes : certaines ne nécessitent que quelques mégaoctets d'espace disque pour s'installer et peuvent tenir sur une disquette.
- Qui permettent de transformer un ordinateur en Media Center (GeeXboX).
- Dédiées à la création multimédia (AGNULA/Demudi).
- Destinées à « transformer » un ordinateur en pare-feu (IPCop).
- Dites "LiveCD[3]" telles que Knoppix (basée sur Debian) ou Slax (basée sur slackware). Pour plus de choix, consultez ce site : http://fr.wikipedia.org/wiki/Liste_des_LiveCD

Voici deux sites qui fournissent une brève description de quelques distributions :
http://www.linux-france.org/article/choix-distri/
http://www.zdnet.fr/actualites/informatique/0,39040745,39307078,00.htm

---

[1] Un freeware est un logiciel gratuit dont l'auteur se réserve la propriété et dont le code source n'est pas modifiable.
[2] Un shareware est un logiciel qui est distribué gratuitement mais n'est utilisable que pendant une période déterminée. Au terme de cette période, l'utilisateur doit acheter le logiciel pour continuer à l'utiliser.
[3] Un liveCD est un CD contenant un système d'exploitation exécutable sans installation préalable.

3. Caractéristiques techniques de linux

3.1. Système de base

- Des architectures 32 bits et 64 bits sont disponibles.
- Système multi-tâches : Linux est capable d'exécuter plusieurs programmes simultanément.
- Systèmes multi-utilisateurs : plusieurs utilisateurs peuvent être connectés sur la même machine et travailler simultanément sur le système. Chaque utilisateur dispose d'un environnement propre.
- Système multi plates-formes[1] : Linux peut fonctionner sur différents processeurs (Intel, AMD, Sun SPARC, …). Il peut également être installé en "multiboot[2]".
- Mémoire protégée : grâce à cette protection, un programme ne peut à lui seul compromettre le fonctionnement de l'ensemble du système.
- Gestion de volumes logiques : les volumes logiques permettent de créer des "partitions virtuelles" (les volumes) qui peuvent recouvrer plusieurs disques physiques et partitions. Il est alors possible d'ajouter, modifier ou retirer une partition d'un volume à la volée.
- Support RAID[3] : le noyau de Linux inclut de base le support des systèmes RAID. RAID est une méthode qui permet de distribuer l'information sur plusieurs disques, pour augmenter la vitesse ou la redondance des informations.
- Reconfiguration dynamique : il est très rarement nécessaire de redémarrer un serveur Linux après une modification de la configuration d'un logiciel ou du système lui-même. Il suffit généralement de redémarrer le service en question.
- Système de fichiers : Linux prend en charge de nombreux systèmes de fichiers : ext2fs, ext3fs, ext4, xfs, vfat, reiserfs, "ISO 9660" (CDROM), …
- …

3.2. Extensions par rapport au système de base

- Grappe de serveurs (Clustering) : les grappes (clusters) ou fermes (farms) d'ordinateurs Linux permettent d'offrir des fonctionnalités de supercalculateur (exemple : Beowulf, Oscar, Mosix).
- Balancement de charge (load balancing[4]) : permet de répartir un ou plusieurs services sur plusieurs ordinateurs ou serveurs afin de ne pas surcharger une machine.
- Tolérance aux pannes : des services de tolérances aux pannes utilisant un serveur redondant peuvent être mis en place. Un tel système détecte automatiquement que le serveur principal ne répond plus et peux, par exemple, basculer la charge vers le serveur redondant (projet "Linux high-availability" http://www.linux-ha.org/).
- …

[1] Il y a néanmoins des exceptions comme la distribution Yellow Dog qui ne fonctionne que sur les machines à base de processeurs PowerPC (Macintosh, PlayStation 3)
[2] Le multiboot désigne la possibilité de démarrer plusieurs systèmes d'exploitation sur un même ordinateur
[3] Redundant Array of Inexpensive Disks
[4] On parle également de "link balancing" lorsqu'il s'agit de s'assurer la répartition de la connectivité.

4. Installation d'un système GNU/Linux

L'installation complète d'une distribution Centos sera décrite au laboratoire.

5. Premiers pas sous Linux

5.1. Le mode graphique et le mode texte

Le mode texte (aussi appelé mode console ou mode commande) est un mode d'affichage tel que les informations apparaissent à l'écran uniquement sous forme de caractères. Le principal organe de communication est le clavier, la souris est rarement utilisée. Les possibilités d'affichage offertes sont très limitées en comparaison au mode graphique, dès lors pourquoi s'y intéresser ?

D'abord, il est universellement disponible sur toutes les cartes graphiques et il est le moins gourmande en ressources pour le système. Il est donc idéal si l'on veut que certains programmes bénéficient du maximum de la puissance d'une machine. Ensuite, le mode texte permet de se familiariser avec l'utilisation des commandes ce qui facilite l'apprentissage de la programmation de scripts[1]. De plus, le mode texte est souvent incontournable si l'on doit réparer un système endommagé : l'utilisateur exécute un mode « rescue » dans lequel il ne dispose que d'une interface texte pour dépanner le système. Enfin, la plupart des logiciels graphiques sont en fait des interfaces ergonomique (appelée « front end ») permettant d'exécuter des logiciels en mode texte. Le mode graphique présente potentiellement plus de failles pour la sécurité. D'un point de vue purement sécuritaire, il serait donc préférable de n'utiliser que le mode texte sur une machine serveur.

Beaucoup plus exigeant, le mode graphique offre aussi beaucoup plus de possibilités. Les interfaces graphiques offrent un confort visuel appréciable ainsi qu'une multitude de manières d'afficher les informations (images, dessins, graphiques) permettant d'adapter l'affichage au type d'information. Le mode graphique est idéal pour la convivialité, donc incontournable pour une station de travail dont l'objectif est de faciliter le travail de l'utilisateur.

Lors des séances de laboratoire, nous travaillerons principalement en mode texte afin de s'y familiariser. En effet, en plus des raisons données précédemment :

- Les outils graphiques n'offriront pas de difficultés de navigation et d'emploi aux utilisateurs. La connaissance des fichiers et des outils qui se cachent derrière ces fenêtres graphiques est le plus important pour une appréhension professionnelle de Linux.
- Même si l'on ne travaille pas sur une machine serveur, il est utile d'apprendre à accomplir des tâches depuis l'invite de commande car cette technique permet parfois de gagner du temps.
- Enfin, la plupart des commandes sont les mêmes sur toutes les distributions Linux. Leur connaissance permet donc de passer sans trop de difficultés d'une distribution à l'autre. Le mode graphique nécessiterait, pour sa part, d'apprendre les outils graphiques mis à disposition par chaque distribution.

[1] Un script est une suite de commandes écrites dans un langage interprété ne nécessitant pas de compilation avant d'être exécuté. Un script est généralement destiné à automatiser certaines tâches (d'administration par exemple)

5.2. Connexion au système

5.2.1. Première connexion après l'installation

De base, l'accès à un système Linux se fait par une procédure d'identification, en fournissant un nom d'utilisateur (login) et une procédure d'authentification, en fournissant un mot de passe. De cette manière, Linux dispose d'une protection de base, dans le sens où seules les personnes autorisées (appelées les utilisateurs du système) peuvent l'utiliser. L'ensemble de cette procédure (nom et mot de passe) est appelé *procédure de login*, ou simplement *login*. Bien que cela ne soit pas très français, on dit aussi d'un utilisateur qu'il se « logue » sur le système.

Pendant l'installation, au moins un compte utilisateur a été défini : le compte *root* appelé aussi super-utilisateur (C'est l'équivalent d'un administrateur, il possède tous les droits et permissions d'accès sur le système,). C'est avec son nom (qui est root par défaut) et son mot de passe qu'il faudra se loguer si aucun autre compte utilisateur n'a été défini lors de l'installation. Cependant, comme tout compte administrateur, il ne faut pas l'utiliser comme un compte personnel, le compte root ne doit être utilisé que pour réaliser des fonctions d'administration. Donc, si cela n'a pas été fait durant l'installation, la première chose à faire est de se créer un compte personnel en utilisant la commande **useradd**. On entre alors le mot de passe deux fois (la deuxième fois pour confirmer). Le compte personnel créé dispose (généralement automatiquement) d'un répertoire personnel à son nom dans le répertoire /home.

Remarque : lors de la frappe du mot de passe, aucun caractère ne s'affiche, pas même le ●●● que l'on a l'habitude de voir sous Windows. En effet, cela permet de connaître le nombre de caractères du mot de passe et donc faciliter la tâche des « pirates ».

Une fois logué, le système attend vos ordres sous la forme d'une ligne d'invite de commande que l'on appelle également *prompt*. Celui-ci peut présenter différentes formes selon sa configuration, par exemple :

```
[root@PC12    home]#
```

Dans celui-ci, les informations affichées sont (de gauche à droite) :

- le nom de l'utilisateur (ici root);
- le nom de la machine sur laquelle cet utilisateur est connecté (ici PC12);
- le répertoire courant (ici /home).

Par défaut, si aucun nom n'est donné lors de l'installation du système, l'invite aura l'allure suivante :

```
[localhost@localdomain   nom_du_répertoire]$
```

Le caractère # est utilisé pour préciser que l'utilisateur est connecté avec le compte root alors que pour un simple utilisateur le caractère est $.

5.2.2. Connexions suivantes

Après s'être connecté via la procédure de login, l'utilisateur dispose d'un accès total à son répertoire personnel ainsi qu'un accès limité à certains autres répertoires du système. Seul le super-utilisateur (root) dispose d'un accès total à tous les répertoires du système.

5.2.3. Le shell

Le shell est un logiciel fournissant une interface permettant à un utilisateur d'interagir avec le système. On distingue deux catégories de shells :

1. Les shells en mode texte, appelé CLI (Command Line Interface). Ils sont des interpréteurs de lignes de commandes. Ils s'exécutent lorsque vous vous loguez et sont capable d'interpréter (et donc exécuter) les commandes que vous entrez au clavier. Le shell CLI installé par défaut sous de nombreuses distributions Linux s'appelle BASH (Bourne Again Shell), mais il existe de nombreux autres shells pouvant fonctionner sous Linux : csh, tcsh, sash,… Sous Windows, on trouve les shells suivants : command.com, cmd.exe, Windows Powershell.
2. Les shells graphiques, appelé GUI (Graphical User Interface) qui fournissent une interface graphique plus intuitive pour l'utilisateur. Sous Linux, cette même fonctionnalité est assurée par des logiciels « gestionnaires de bureau » ou « environnement de bureau » comme GNOME ou KDE.

5.2.4. Notion de console et terminal

Une console est constituée d'un ensemble écran-clavier-souris directement branchés sur l'ordinateur. Le terminal est constitué des mêmes éléments écran-clavier-souris mais accède à l'ordinateur par l'intermédiaire d'un réseau.

A de rares exceptions près, un ordinateur n'offre qu'une seule console. Cependant, Linux dispose de consoles dites virtuelles qui offrent la possibilité de « faire comme si » l'ordinateur était équipé de plusieurs écrans, claviers et souris. Bien sur, seule l'une d'entre elles peut être utilisée à la fois car il n'y a en réalité qu'un seul clavier raccordé directement à l'ordinateur. Néanmoins, ces consoles virtuelles peuvent être très pratiques :

- On utilise une console pour saisir et exécuter des commandes, et une ou plusieurs autres consoles pour visualiser diverses documentations détaillant l'utilisation de ces commandes.
- On utilise une console pour saisir et exécuter des commandes d'administration (par exemple la création d'un utilisateur) et une autre pour vérifier que les modifications ont bien été effectuées (se connecter au système avec l'utilisateur créé dans la première console).

Les consoles purement en mode texte sont désignées par le terme « tty » (qui vient du mot « télétype »). Par défaut, ces consoles sont au nombre de six mais ce nombre peut être adapté en modifiant le contenu du fichier /etc/inittab. Le passage d'une console virtuelle à l'autre peut se faire à n'importe quel moment en pressant simultanément les touches [CTRL], [Alt] et [Fn], (où n est le numéro de la console virtuelle que vous souhaitez obtenir). La console graphique est accédée pressant simultanément les touches [CTRL], [Alt] et [F7]

Il est possible de disposer d'une ou plusieurs console(s) texte(s) dans le mode graphique[1], elles sont alors désignées par le terme « pts/n » (où n représente le numéro de la console virtuelle).

[1] De la même manière que l'on ouvre une « fenêtre dos » sous Windows.

5.2.5. Quitter le système

Théoriquement Linux peut fonctionner vingt-quatre heures sur vingt-quatre pour une durée indéfinie. Néanmoins, il sera parfois nécessaire d'arrêter le système : en cas de maintenance ou défaillance du matériel, de modification du noyau ou simplement l'arrêt d'une station de travail en fin de journée.

De la même manière que pour d'autres systèmes d'exploitation (Windows, MacOS, …), Linux doit être arrêté proprement. Le système doit, en effet, être averti de l'arrêt afin de s'assurer que les fichiers ouverts sont fermés proprement, que tout ce qui doit être sauvegardé l'a bien été, ainsi que prévenir les éventuels utilisateurs ou machines connectés.

Comme l'expliquent Welsh M., Dalheimer K. et Kaufman L. (2000), Linux cache les accès disque en mémoire, c'est à dire que les écritures sont mises en mémoire tant qu'il n'est pas nécessaire de les réaliser. L'avantage étant l'amélioration des performances lorsque de multiples lectures des mêmes blocs sont effectuées car celles-ci se font directement depuis la mémoire, beaucoup plus rapide que les disques. Par contre, si la machine est arrêtée brusquement, les données en mémoire ne sont pas écrites sur le disque et il y a un risque non négligeable de perdre des données et/ou d'endommager les systèmes de fichiers. Ainsi, le programme /sbin/update écrit sur le disque, à intervalles réguliers, le contenu du cache disque afin d'éviter de trop gros dégâts en cas d'arrêt brutal du système.

L'arrêt immédiat du système se fait par l'une des deux commandes, suivantes

```
[root]$ shutdown –h now
[root]$ halt
```

Si l'on désire redémarrer la machine juste après cet arrêt, l'une des deux commandes suivantes peut être utilisée :

```
[root]$ shutdown –r now
[root]$ reboot
```

Sur un serveur, seul l'utilisateur root doit être autorisé à utiliser ces commandes. Le redémarrage peut également se faire via la combinaison de touche [CTRL], [ALT], [DEL] qu'il est possible de désactiver dans le fichier /etc/inittab.

5.3. Les commandes de bases

5.3.1. Syntaxe

Au fur et à mesure des séances de laboratoires, vous apprendrez le rôle de nombreuses commandes, ainsi qu'à les utiliser. Bien que les différentes commandes réalisent des fonctions très diverses, elles possèdent toutes une syntaxe qui suit les règles suivantes :

- Le premier mot de la ligne (après le prompt) est le nom de la commande.
- Le shell tient compte de la casse, c'est à dire qu'il fait la différence entre les majuscules et minuscules. On dit qu'il est « case sensitive »
- Les commandes peuvent être utilisées avec des arguments ou des options (les options sont généralement précédées du signe moins mais ce n'est pas toujours le cas)
- Les différents mots, options, arguments ou chiffres doivent être séparés par un espace.

Exemple de commandes

```
[root]# ls
```

Ici, "ls" est le nom la commande, elle permet de lister le contenu d'un répertoire.

```
[root]# Ls
```

Cette commande est différente de la précédente. En effet, ls et Ls sont deux commandes différentes puisque leur casse est différente.

```
[root]# ls –a
```

Ici, -a est une option de la commande ls

5.3.2. Les pages d'aide

Un des objectifs du cours théorique et du laboratoire est de vous permettre d'être autonome par rapport à l'utilisation d'un système de type Linux. Ainsi, c'est vous qui devrez chercher la formulation correcte des différentes commandes à utiliser afin de réaliser une tâche précise. Pour y parvenir, vous devrez apprendre à trouver les informations concernant l'utilisation des commandes car il est impossible de connaître toutes les commandes et leurs options.

Pour faciliter l'utilisation de ces commandes, il existe un système d'aide en ligne intégré à Linux : les pages de manuel[1]. Ces pages ne sont pas toujours très facile à lire, leur compréhension exige parfois une connaissance approfondie de Linux (ainsi que de l'anglais si une traduction n'existe pas). Ces pages d'aide peuvent être consultées en utilisant la commande **man** suivit du nom de la commande dont vous souhaitez afficher l'aide.

Exemple

Vous souhaitez afficher le contenu d'un répertoire, et en particulier les fichiers cachés (le nom de ceux-ci commence par un point). Vous savez que la commande ls permet de lister le contenu d'un répertoire mais les fichiers cachés ne sont pas repris dans le listing. Il faut donc utiliser un argument afin de préciser à la commande ls qu'elle doit également afficher les fichiers cachés, mais lequel ?

Pour le savoir, il faut utiliser la commande man. Il vous suffit de taper :

```
[user]# man ls
```

En parcourant la page d'aide qui s'affiche, vous verrez que l'argument à utiliser est "a". Dès lors la commande à utiliser sera :

```
[user]# ls –a
```

Une version française de nombreuses pages de manuel peuvent être obtenue ici : ftp://ftp.lip6.fr/pub/linux/french/docs/

Structure d'une page d'aide (man page)

En général, la documentation est structurée de la manière suivante.

NAME
Ce paragraphe renseigne le nom de la commande et sa fonction y est brièvement décrite.

SYNOPSIS
Le synopsis reprend toutes les possibilités syntaxiques de saisie (les options, les arguments,…) liées à la commande. Généralement, les termes placés entre crochet sont optionnels.

[1] Appelées aussi « man pages », elles sont généralement installées par défaut, mais peuvent l'être à postériori. Elles existent souvent en plusieurs langues, il est ainsi possible de les utiliser dans sa propre langue.

DESCRIPTION
La description consiste en une explication courte et condensée des conséquences de la commande.

FILES
Ce paragraphe renseigne les fichiers qui seront modifiés par l'exécution de la commande ou nécessaires au moment de la saisie.

SEE ALSO
Ce paragraphe liste d'autres commandes qui sont en relation avec la commande dont l'aide est affichée.

DIAGNOSTICS
Ce paragraphe fourni des explications sur les messages d'erreurs susceptibles d'être affichés par la commande lorsque celle-ci est utilisée dans des situations précises.

BUGS
L'utilisation, dans des circonstances particulières, de certaines commandes génère des erreurs. Les erreurs connues sont répertoriées dans ce paragraphes ce qui permet d'éviter d'utiliser la commande à mauvais escient.

EXAMPLE
Ce paragraphe présente quelques exemples d'utilisation de la commande et de ses arguments. Ces exemples sont malheureusement souvent trop peu nombreux voire inexistant.

5.3.3. Autres commandes d'aide

D'autres commandes d'aide existent, nous les utiliseront lors des séances de laboratoire. Il s'agit de **apropos**, **info** et **whatis**.

Module 2 : L'administrateur

1. Rôles de l'administrateur

L'administration d'un système d'exploitation et d'un réseau sont des tâches complexes. Dès lors, il est indispensable que les administrateurs respectent certaines règles et travaillent avec discipline et méthodologie. Un aperçu des tâches courantes que devra réaliser un administrateur est repris ici[1].

La gestion des besoins, du budget et des priorités

Les entreprises ont des besoins informatiques spécifiques qui sont fonction de leur domaine d'activité. Certaines de ces fonctions peuvent éventuellement être externalisée (serveur Web, serveur mail,…) mais ce n'est pas possible pour les applications stratégiques ni les applications « métiers ». Les divers éléments du réseau (ordinateurs, serveurs, câbles, switches, …) doivent donc permettre le fonctionnement de ces applications de manière optimale mais aussi permettre l'installation d'autres applications en fonction des évolutions.

L'administrateur doit donc établir un cahier des charges dans lequel sont renseignés les besoins en services, applications et matériels. Les budgets étant toujours insuffisants, il faut déterminer, au départ du cahier des charges, quels sont les achats prioritaires parmi les nombreux besoins : le système de câblage, les éléments d'interconnexions, les serveurs, les systèmes de sauvegardes, les logiciels métiers, les logiciels de supervision, etc. Une fois les priorités établies, l'administrateur doit rédiger les appels d'offres et comparer les devis afin de réaliser un choix judicieux : la solution est-elle sécurisée ? Evolutive ? Tolérante aux pannes ? Pérenne ? Etc.

La gestion des ordinateurs et des périphériques

L'administrateur doit pouvoir gérer les machines (et leurs composants : disque dur, carte graphique,…) ainsi que leurs périphériques (scanner, imprimante, clavier, …)

- Installer les systèmes d'exploitation, paramétrer le démarrage et l'arrêt ;
- Gérer les disques (initialisation, partitionnement, remplacement,...) ;
- Ajouter ou enlever un périphérique ;
- Planifier le vieillissement du matériel et son remplacement ;
- Ajouter (ou supprimer) un pilote de périphérique ("driver") au noyau ou l'installer comme un module[2].

La gestion des performances des systèmes

L'administrateur doit surveiller (voire améliorer) les performances des serveurs et des stations de travail, notamment :

- Paramétrer les ressources de manière à garder un système parfaitement fonctionnel. (répartition des ressources, …).
- Surveiller les ressources (processeur, mémoire, bande passante, entrées/sorties et espace disque) afin de prévoir et donc réagir avant un éventuel manque de ressources.

---

[1] Cet aperçu est inspiré de Berlat A., Bouchaudy J-F., Goubet G., 2003, pp 1-9 et 1-10

[2] Plutôt que d'être compilée dans le noyau, une fonction peut être disponible sous forme de module et être chargée, selon les besoins, pendant le fonctionnement du système (donc dynamiquement). Le chargement des modules peut être automatique, par exemple lorsque le système détecte un nouveau périphérique.

La gestion des utilisateurs

L'administrateur doit gérer les comptes des utilisateurs. Il devra notamment :

- Créer, modifier et supprimer les comptes utilisateurs sur les systèmes dont il a la charge.
- Etre capable de modifier l'environnement de travail des utilisateurs, changer leur mot de passe, gérer leurs droits d'accès, etc.
- L'administrateur doit également informer et "éduquer" les utilisateurs pour qu'ils utilisent correctement les outils informatiques mis à leur disposition.

La gestion des fichiers et des disques

L'administrateur doit gérer les fichiers et systèmes de fichiers présents sur les disques Il devra notamment :

- Mettre en place et gérer les systèmes de fichiers (création, configuration des permissions d'accès, cryptage,…).
- Veiller à l'intégrité[1] des systèmes de fichiers et donc des données.
- Gérer l'arborescence des fichiers (organisation et contrôle des accès).
- Surveiller l'espace disque : contrôler le taux d'occupation des disques, mettre en place des quotas.

La gestion des services

L'administrateur doit savoir configurer et utiliser les services adéquats pour répondre aux besoins de l'entreprise tels que les services de base fournis par le système Linux (gestion des tâches (cron et at[2]), service d'impression, …) ou d'autres services spécifiques.

La gestion des problèmes

L'administrateur doit connaître ses machines (partitionnement, services actifs, …) et son réseau (architectures, protocoles, …) de manière à pouvoir intervenir rapidement et efficacement en cas de problème. Cela nécessite la mise en place d'outils aidant au diagnostique des problèmes et générant des alarmes face aux pannes qui peuvent survenir.

Il peut être utile de préparer des fiches préformatées que les utilisateurs peuvent compléter, à l'attention du service informatique, avec le détail des problèmes qu'ils rencontrent.

La gestion des sauvegardes et du stockage des données

L'organisation des sauvegardes est TRES TRES TRES TRES IMPORTANTE, c'est un point crucial pour l'administrateur. Celui-ci doit absolument être capable de récupérer toutes les données importantes dans le délai le plus bref possible.

[1] Des données sont intègres lorsqu'elles n'ont subi aucune altération (volontaire ou involontaire) et sont toujours utilisable.
[2] Cron et at sont des services permettant de gérer le moment auquel un processus doit s'exécuter.

La gestion du réseau

Gestion des éléments constitutifs du réseau

L'administrateur doit être capable de :

- Choisir, installer et paramétrer les éléments adéquats (stations de travail, switches, routeurs, points d'accès, antennes, câbles, …).
- Paramétrer le démarrage et l'arrêt de tous les systèmes.
- Il devra également automatiser le processus de démarrage des nouveaux services et produits sur les machines clientes et serveurs.

Sélectionner la bonne architecture réseau

L'administrateur doit mettre en place ou modifier l'architecture du réseau :

- choisir la topologie du réseau,
- choisir les protocoles réseau ,
- mettre en place la redondance,
- organiser le routage et le filtrage.

Surveiller le réseau

L'administrateur doit mettre en place des outils de surveillance afin de vérifier que les performances du réseau ne se dégradent pas, notamment suite à l'utilisation de nouvelles applications ou à des modifications de configuration.

La gestion de la sécurité

L'administrateur doit veiller à la sécurité, en prenant en compte trois axes :

- Les accès doivent être contrôlés afin que seuls les utilisateurs autorisés accèdent aux ressources auxquelles ils ont droits (contrôle des accès, des connexions, non répudiation, …).
- Les données doivent toujours être intègres et accessibles aux seuls utilisateurs autorisés.
- Les services doivent toujours être intègres et accessibles aux seuls utilisateurs autorisés.

Cela implique, entre autre, que l'administrateur devra :

- Garder les systèmes à jour en appliquant les correctifs de sécurité.
- Mettre en place des outils de sécurisation tels que pare-feu, anti-virus, systèmes de détection d'intrusion, cryptage des données, …
- Former le personnel.

Méthodologies de l'administrateur

Ces quelques aspects de la fonction d'administrateur système/réseau, montrent que la mission de celui-ci est complexe et variée. Aussi, l'administrateur informatique doit-il travailler de manière rigoureuse et méthodique. Voici quelques éléments de méthodologie pour améliorer l'efficacité et la qualité de son travail[1].

Documenter

L'administrateur doit suffisamment documenter ses actions car le fait de les consigner lui permet de connaître exactement l'état de ses machines et de son réseau. Cela facilitera la réalisation de son travail journalier, mais aussi ses différents dépannages. Il pourra, par exemple, faire le lien entre le jour où il a opéré une modification et le jour où tel service ne fonctionne plus correctement. D'autre part, l'administrateur ne travaille pas toujours seul et doit donc laisser une trace de ses actes aux autres administrateurs afin que ceux-ci trouvent rapidement les informations dont ils ont besoins.

Ainsi, l'administrateur doit tenir un journal de bord. Il s'agit d'un document qui doit être daté et dans lequel sont consignées les opérations importantes d'exploitation telles que

- l'installation du système (partitionnement, systèmes de fichiers,…),
- la configuration réseau du système,
- l'ajout de périphériques,
- l'installation de nouveaux logiciels ou services,
- la mise à jour du système,
- la sauvegarde complète du système et des données,
- les événements anormaux (lenteur inhabituelle du système, messages d'erreurs, ...),
- les fichiers de configuration,
- il peut également être utile de consigner les fichiers journaux ("logs").

Il peut être pratique de disposer d'un journal de bord par machine et d'un autre concernant le réseau (plan du réseau, topologie, adressage,…). Tenir ce journal sous forme de fichier électronique le rend plus facile à manipuler/modifier, mais l'idéal est de disposer d'une copie papier à proximité des systèmes concernés (salle serveur ou local d'administration).

L'administrateur doit aussi veiller à ce que la documentation du matériel – ou plutôt une copie de la documentation (manuel d'installation, guide de l'utilisateur,…) – soit accessible à proximité des éléments correspondant dans le réseau.
Enfin, il faut réaliser un repérage adéquat sur les appareils (étiquette, code couleur des câbles,…) afin de les repérer et les identifier facilement.
De manière générale il faut aussi documenter les programmes, scripts et fichiers de configuration en y insérant des lignes de commentaires.

Sauvegarder

Le fait de savoir qu'il faut sauvegarder ne suffit pas. La gestion des sauvegardes comprend un ensemble d'éléments et de choix parmi lesquels on peut citer le choix des types de sauvegarde, du logiciel à utiliser, la fréquence des sauvegardes, le type de support de stockage, la liste des personnes chargées de réaliser les sauvegardes, etc. Pour ne rien laisser au hasard, il faut

[1] Ces éléments sont inspirés de Berlat A., Bouchaudy J-F., Goubet G., 2003, pp 1-11 et 1-12 et Bouchaudy J-F., 2007, pp 1-1 à 1-10

mettre en place un plan de sauvegarde et même mieux, un plan de recouvrement après sinistre[1].

Une sauvegarde qui n'a pas été testée n'a que peu de valeur. Dès lors, l'idéal est de disposer d'un système de test (identique au système en production) via lequel il est possible de tester les restaurations des sauvegardes effectuées. De plus, la machine de test peut être utilisée comme système de secours. A défaut d'une machine spécifique dédiée aux tests, on se tournera vers un système de virtualisation[2] ou de cluster.

Automatiser

L'automatisation des différentes tâches d'administration (à l'aide de scripts, par exemple) permet de gagner un temps précieux en « laissant » la machine travailler pour l'homme. En pratique, lorsqu'une procédure doit être utilisée plus d'une fois, il est souvent intéressant de l'automatiser. A titre d'exemple, il est possible d'automatiser :

- l'installation des systèmes d'exploitation,
- l'installation des mises à jour,
- la création de comptes utilisateur,
- les sauvegardes,
- etc.

De plus, l'automatisation réduit le risque d'erreur encouru lors de la réalisation manuelle des tâches (faute de "frappe", oubli, …)

Agir de manière réversible

Chaque fois que l'administrateur effectue une tâche d'administration (installation de périphérique, de logiciels, de drivers, modifications de configurations,…), un problème peut survenir. Or, quel que soit le problème, l'activité doit continuer ou, au pire, reprendre dans les plus brefs délais. Ainsi, il faudra, à chaque fois que cela est possible, agir de manière réversible afin de savoir revenir rapidement à un système fonctionnel. Dans ce contexte, on se rend compte de l'importance de la journalisation de ses opérations, ainsi que des sauvegardes.

Etre proactif

L'administrateur doit agir de manière proactive, c'est-à-dire qu'il doit constamment anticiper les problèmes qui ne manqueront pas de survenir. Un exemple simple consiste à installer un anti-virus sur un ordinateur. Cette action permet d'éviter que le système ne soit infecté par de nombreux virus qui risquent de le rendre non opérationnel.

Voici quelques autres exemples d'événements à prévoir :

- Si l'administrateur tombe malade, qui peut le remplacer ?
- Si le switch qui interconnecte tous les utilisateurs tombe en panne, dispose-t-on d'un appareil de rechange? En d'autres termes, le réseau est-il tolérant aux pannes?
- Si la salle serveur brûle, le local technique est inondé, le bâtiment s'effondre suite à un tremblement de terre,… comment redémarrer l'exploitation au plus tôt ? Cette question

[1] Il s'agit d'un plan qui permet de traiter une crise (l'arrêt d'une production, …) de manière organisée et ainsi, d'assurer le maintien des activités critiques de l'entreprise ou, à tout le moins, de les reprendre rapidement.
[2] Technique permettant de faire fonctionner simultanément, sur une seule machine, plusieurs systèmes d'exploitation et/ou applications, comme s'ils fonctionnaient sur des machines distinctes

peut faire sourire, mais sachez que les « grandes » sociétés (notamment bancaires) disposent de locaux et bâtiments dédoublés.

- …

Autres qualités de l'administrateur

Savoir communiquer

La communication est essentielle car un informaticien travaille rarement seul, il est en contact avec tous les acteurs de l'entreprise.

- il doit se montrer persuasif afin de disposer des budgets nécessaires à sa tâche,
- il doit discuter avec les fournisseurs de matériel, les prestataires de services, …,
- il doit discuter avec les utilisateurs finaux afin de connaître leurs besoins,
- il doit comprendre les problèmes que leurs soumettent les utilisateurs,
- il doit former les utilisateurs à l'utilisation des outils informatiques et les informer des contraintes liées à ces outils,
- etc.

Avoir une bonne connaissance du marché

L'administrateur doit se tenir au courant de l'évolution du marché car celui-ci peut influer sur les choix de gestion et d'évolution d'un parc informatique. Il est difficile de réaliser des choix judicieux ou mettre en place des solutions si l'administrateur n'est pas au courant que ces solutions existent.

Connaître ses limites

Pour de petites entreprises, il est possible que l'administrateur s'occupe de « tout ». Cependant, lorsque le système devient plus complexe ou plus important, il sera nécessaire de répartir l'administration entre plusieurs personnes. Il peut s'agir d'une aide ponctuelle de quelques employés, ou bien d'une charge de travail nécessitant plusieurs administrateurs.

Dans tous les cas, il faut être conscient de ses limites. D'une part, personne ne peut tout connaître sur tout, d'autre part, personne ne peut réaliser un travail de quatre heures en une heure. Ainsi, que ce soit par manque de temps ou manque de connaissance sur un sujet précis, il vaut mieux faire appel à un aide ou un expert plutôt que de mettre en place une solution insatisfaisante.

2. Avec quels outils administrer un système Linux ?

Il existe plusieurs manières d'administrer un système Linux. Chaque administrateur pourra donc utiliser celle qui lui convient le mieux, voire une combinaison des méthodes disponibles.

Les commandes

Cette méthode nécessite de connaître les commandes d'administration et leur « maniement ». Elle permet d'automatiser facilement l'exécution des commandes, ce qui la rend populaire chez les administrateurs expérimentés. De plus, l'utilisation des commandes offre généralement plus de possibilités que celles fournies par une interface graphique.

Les fichiers de configuration

Cette méthode consiste à éditer les fichiers de configuration concernés (fichiers de type texte) afin de les modifier directement. Cette méthode nécessite de bien connaître la structure des

fichiers et les liens éventuels avec d'autres fichiers, sans quoi des incohérences empêcheront le système de fonctionner correctement. Les fichiers de configuration sont faciles à copier d'une machine à l'autre et permettent d'y insérer des commentaires facilitant la documentation du système.

Les scripts

Comme nous l'avons énoncé précédemment, l'écriture d'un script simplifie le travail de l'administrateur et sécurisent le fonctionnement du système en automatisant certaines tâches. Un script permet l'édition des fichiers d'administration ainsi que la mise en place des contrôles préalables à l'exécution d'une commande.

Les outils spécifiques

Beaucoup de distributions Linux fournissent des outils intégrés qui permettent de réaliser les principales tâches d'administration. Ces outils peuvent être des commandes, des outils en mode texte ou en mode graphique (voire dans les deux modes) avec des menus déroulants et des boîtes de dialogue. Ces outils sont plus faciles à prendre en main car ils évitent souvent de devoir mémoriser la syntaxe des commandes ou la structure des fichiers de configuration.

Ci-dessous, le tableau 1 reprenant quelques outils d'administration:

Outil	**Description**
Webmin	Interface graphique permettant d'administrer un serveur Linux via un navigateur web.
linuxconf	Linuxconf est un programme frontal qui écrit ou met à jour les fichiers de configuration du répertoire /etc. Il existe en mode texte et en mode graphique.
mc (midnight commander)	Outil, en mode texte, de gestion de fichiers et répertoires.
YAST	Yast est un outil spécifique de la distribution SuSe qui couvre « toutes » les tâches d'administration.
SWAT	SWAT est un outil d'administration du serveur de fichiers samba via une interface web (Samba Web Administration Tool)
Plesk	Plesk est un outil d'administration de serveurs

Tableau 1 : exemples d'outils d'administration

Module 3 : Gestion des Fichiers

1. Introduction

Qu'est-ce qu'un fichier ?

Fondamentalement, un fichier est une suite de bits qui constituent un ensemble cohérent (en principe ;-) d'informations. Le fichier est l'élément fondamental d'un système d'exploitation, les données qu'il contient peuvent représenter le code d'un programme, la configuration d'un logiciel, … Les fichiers sont aussi utilisés pour stocker les documents des utilisateurs. Ils sont identifiables par un nom, qui doit respecter certaines contraintes (nombre maximum de caractères, caractères interdit, etc).

Avec Linux, tout est fichier !

Sous Linux, tous les éléments du système sont manipulés par des fichiers : les répertoires, le clavier, les disques, … Cette uniformisation permet de simplifier la manipulation du système ainsi que sa programmation.

Bien sur, utiliser un répertoire ou un disque ne correspond pas aux mêmes opérations pour le système d'exploitation. Il existe donc différents types de fichiers dont nous parlerons un peu plus loin.

2. L'arborescence des fichiers

La multitude de fichiers nécessaires au fonctionnement du système (pilotes, fichiers de configurations,…) ainsi que les fichiers personnels des utilisateurs (documents, sons, vidéos,…) seraient ingérables sans un classement structuré.

Ainsi, les répertoires permettent d'organiser les fichiers en les regroupant par catégorie. Pour un classement plus fin, ces répertoires peuvent eux-mêmes contenir des sous-répertoires, et ainsi de suite. Cette structure hiérarchique ressemble à un arbre, d'où son nom d'arborescence. Pour assurer la compatibilité entre les différents systèmes Linux une norme a été établie pour l'arborescence des fichiers : la FHS (File Hierarchy Standard).

A défaut de mémoriser l'ensemble de l'arbre d'un système Linux, l'administrateur doit connaître les noms, les contenus et les emplacements des principaux répertoires.

2.1. Hiérarchie de base

Sur son site, Pillou J.-F. commente les répertoires principaux de la FHS dont nous reprenons ici un large extrait[1] :

/		la racine, elle contient les répertoires principaux
/bin		Contient les exécutables essentiels au système, employés par tous les utilisateurs.
/boot		Contient les fichiers de chargement du noyau, dont le chargeur d'amorce.
/dev		Contient les points d'entrée des périphériques.
/etc		Contient les fichiers de configuration nécessaires à l'administation du système (fichiers *passwd, group, inittab, ld.so.conf, lilo.conf, ...*).
	/etc/X11	Contient les fichiers spécifiques à la configuration de X (contient **XF86Config** par exemple)
/home		Contient les répertoires personnels des utilisateurs. Dans la mesure où les répertoires situés sous */home* sont destinés à accueillir les fichiers des utilisateurs du système, il est conseillé de dédier une partition spécifique au répertoire */home* afin de limiter les dégâts en cas de saturation de l'espace disque.
/lib		Contient les bibliothèques standards partagées entre les différentes applications du système.
/mnt		Permet d'accueillir les points de montage des partitions temporaires (cd-rom, disquette, ...)[2].
/proc		Regroupe un ensemble de fichiers virtuels permettant d'obtenir des informations sur le système ou les processus en cours d'exécution.
/root		Répertoire personnel de l'administrateur root. Le répertoire personnel de l'administrateur est situé à part des autres répertoires personnels car il se trouve sur la partition racine, afin de pouvoir être chargé au démarrage, avant le montage de la partition */home*.
/sbin		Contient les exécutables essentiels du système (par exemple la commande **useradd**).
/tmp		Contient les fichiers temporaires
/usr		Hiérarchie secondaire
	/usr/X11R6	Ce répertoire est réservé au système X version 11 release 6
	/usr/X386	Utilisé avant par X version 5, c'est un lien symbolique vers /usr/X11R6
	/usr/bin	Contient la majorité des fichiers binaires et commandes utilisateurs
	/usr/include	Contient les fichiers d'en-tête pour les programmes C et C++
	/usr/lib	Contient la plupart des bibliothèques partagées du système
	/usr/local	Contient les données relatives aux programmes installés sur la machine locale par le root
	/usr/sbin	Contient les fichiers binaires non essentiels au système et réservés à l'administrateur système
	/usr/share	Réservé aux données non dépendantes de l'architecture
	/usr/src	Contient des fichiers de code source
/var		Contient des données versatiles telles que les fichiers de bases de données, les fichiers journaux (logs), les fichiers du spouleur d'impression ou bien les mails en attente.

[1] Pillou Jean-François, "La hiérarchie des fichiers sous Unix", [en ligne] http://www.commentcamarche.net/unix/unix-fichiers.php3 . Contrat de licence Creative commons V2.0

[2] Le répertoire /media est apparu dans la FHS 2.3. On utilise alors, par exemple, /mnt pour les systèmes de fichiers temporaires et /media pour les systèmes de fichiers amovibles.

2.2. Chemins et répertoires

2.2.1. Répertoire personnel et répertoire courant

Le répertoire personnel (home directory) d'un utilisateur est le seul répertoire du système dans lequel il est entièrement maître de créer, supprimer ou modifier des fichiers. Ce répertoire contient toutes les données et tous les paramétrages personnels d'un compte utilisateur.

Par défaut :

- le répertoire personnel porte le même nom que le compte utilisateur,
- chaque utilisateur abouti dans son répertoire personnel chaque fois qu'il se connecte,
- tous les répertoires personnels des utilisateurs se trouvent dans /home sauf celui du super utilisateur qui est /root.

Le répertoire courant est le répertoire dans lequel « se trouve » l'utilisateur. La commande **cd** (change directory), permet de modifier le répertoire courant et donc de se déplacer dans l'arborescence. La commande **pwd** permet d'afficher le répertoire courant.

2.2.2. Nom de fichier et nom de chemin

Dans une arborescence, plusieurs fichiers peuvent porter le même nom. On peut, en effet, trouver un fichier nommé lettre.txt ou compte.xls dans le répertoire personnel de plusieurs utilisateurs. Dès lors, pour identifier individuellement les fichiers Linux utilise les chemins d'accès, dont le nom du fichier n'est qu'une des composantes.

Noms de fichiers

Les noms de fichiers sont limités à 255 caractères et ne doivent en principe comporter que des caractères alphanumériques, ainsi que les symboles - _ + %. L'utilisation de tout autre caractère est déconseillée, notamment les caractères ? * / qui ont une signification particulière pour le shell.

Les fichiers (ou répertoires) dont le nom commence par un point sont des fichiers cachés, c'est-à-dire qu'ils ne sont pas listés par défaut par la commande ls.

Noms de chemins

Pour identifier sans ambiguïté chaque fichier de l'arborescence, Linux tient compte du nom du fichier mais aussi de sa position dans l'arborescence. Ainsi, les mêmes noms de fichiers peuvent être utilisés dans deux répertoires différents. Le nom complet du fichier est donc complété par l'indication de son chemin d'accès appelé le *nom de chemin* (path name en anglais)

Le répertoire racine contenant tous les autres répertoires est symbolisé par un slash (/) et appelé répertoire racine (root[1]). Le nom de chemin contient tous les noms des sous-répertoires, en partant de la racine, permettant d'arriver au fichier. Tous ces sous-répertoires sont séparés par un slash qui joue également le rôle de signe de séparation entre les sous-répertoires du chemin d'accès. On parle également de chemin absolu, car le point de départ du nom de chemin commence à la racine de l'arborescence.

[1] Malheureusement, la racine de l'arborescence porte le même nom que le compte du super-utilisateur, ce qui ne manque pas de provoquer des confusions.

Nom de chemin relatif

Le système d'exploitation mémorise le répertoire courant dans lequel se trouve chaque utilisateur. De cette manière, il n'est pas nécessaire d'indiquer l'ensemble du chemin d'accès au fichier, mais uniquement le chemin à partir de la position courante dans l'arborescence.

Par exemple, pour accéder au répertoire /usr/bin vous devriez normalement entrer la commande :

```
$ cd  /usr/bin
```

C'est à dire entrer le chemin complet menant au répertoire bin. Par contre, si vous vous trouver dans le répertoire /usr, vous pouvez atteindre le répertoire /usr/bin en utilisant le chemin relatif, c'est à dire depuis la position où vous êtres. Dès lors, la commande sera :

```
$ cd  bin
```

Attention : dans ce cas, il ne faut pas placer le signe / devant bin, sinon le chemin sera interprété comme étant un chemin absolu dont le point de départ est la racine de l'arborescence. Auquel cas le système recherchera un répertoire nommé bin juste sous la racine (/).

Nom de chemin relatif à l'utilisateur

Ce nom de chemin prend comme point de départ le répertoire personnel de l'utilisateur symbolisé par le caractère « tilde » (~). Par exemple, si vous êtes dans le répertoire /usr et souhaitez accéder à un répertoire nommé *test* se trouvant dans votre répertoire personnel, il vous est possible de le faire via la commande suivante :

```
[user1@localhost   usr]# cd  ~/test
```

2.3. Gestion de l'arborescence

De nombreuses commandes permettent de gérer au mieux une arborescence. Nous parcourons dans cette section les plus fondamentales, d'autres commandes seront introduites lors des séances de laboratoire.

2.3.1. Opérations de base sur les répertoires

Pour organiser et ordonner la multitude de fichiers à gérer, vous utiliserez nécessairement des répertoires. Vous devez ainsi créer des répertoires et sous-répertoires pour classer vos différents fichiers :

- les fichiers texte et les fichiers de documentation,
- les codes source des programmes,
- les copies de sauvegarde,
- les scripts,
- etc.

Créer un répertoire

Pour créer un répertoire, il est possible d'utiliser la commande *mkdir* (make directory). Chaque répertoire nouvellement créé contient dès le départ deux éléments standard :

- un élément qui fait référence au répertoire de niveau supérieur (il a pour nom "**..**") ;
- un élément qui a pour nom « point » " **.** " et fait référence au répertoire courant.

Ces deux éléments sont en fait des répertoires cachés. Pour les lister, vous devez utiliser l'option -a de la commande ls.

Supprimer un répertoire vide

Il est possible de supprimer un répertoire avec la commande *rmdir* (remove directory) uniquement si le répertoire ne contient aucun de fichier. Dans les autres cas, il faudra utiliser la commande *rm* (remove). Il n'est pas possible de supprimer le répertoire courant puisque celui-ci est actif.

2.3.2. Opérations de bases sur les fichiers

Créer un fichier

Pour créer un fichier, vous pouvez utiliser un éditeur de texte tel que *vi*. Cependant, pour créer rapidement un fichier dans le cadre d'un test, par exemple, la méthode la plus simple consiste à utiliser la commande *touch*. Par exemple, la commande suivante crée un fichier vide nommé essai dans le répertoire courant.

```
[user]# touch essai
```

La commande touch a en réalité un autre rôle : elle permet de modifier l'horodatage d'un fichier.

Copier un fichier

La commande *cp* (copy) permet d'effectuer ces copies de fichiers ou de répertoires. Dans l'exemple suivant, le fichier *Fichier1* est copié sous le nom *Fichier2* dans le répertoire courant.

```
[user]# cp Fichier1 Fichier2
```

Dans ce deuxième exemple, le fichier *Fichier1* est copié dans le répertoire *directory1* et conserve son nom original.

```
[user]# cp Fichier1 directory1
```

L'exemple suivant montre comment copier le fichier *Fichier1* dans le répertoire courant.

```
[user]# cp /tmp/Fichier1 .
```

Par défaut, si le fichier cible existe déjà, le système demande si vous souhaitez supprimer le fichier existant. Restez cependant prudent en matière de copie de fichiers (et de suppression de fichiers) car en mode texte, il n'y a pas de corbeille pour récupérer les fichiers effacés ou écrasés.

Renommer ou déplacer un fichier

La commande *mv* (move) permet de déplacer un fichier d'un répertoire vers un autre. Elle est aussi utilisée pour renommer un fichier. La syntaxe de cette commande est la même que celle de la commande cp.

L'exemple suivant permet de renommer Fichier1 en Fichier2. Après l'opération, il n'existe plus de Fichier1 dans le répertoire.

```
[user]# mv Fichier1 Fichier2
```

L'exemple suivant permet de déplacer le fichier *Fichier1* du répertoire /tmp vers le répertoire cible /home/user en conservant le nom du fichier.

```
[user]# mv /tmp/Fichier1 /home/user/
```

Restez prudent avec l'utilisation de cette commande, sous peine de perdre des fichiers.

Supprimer un fichier

C'est la commande *rm* (remove) qui permet de supprimer des fichiers. Elle respecte la syntaxe suivante :

```
[user]# rm [Options] Fichierl [Fichier2 ...]
```

Dans la plupart des distributions, la commande rm utilise l'option –i qui demande une confirmation de suppression de fichiers. Cependant ce n'est pas toujours le cas, dès lors, prudence.

Afficher le contenu d'un fichier

Outre les éditeurs de texte, les commandes *cat, head* et *tail* permettent d'afficher tout ou partie du contenu d'un fichier à l'écran. Ces commandes ne peuvent lire que des fichiers « texte » et les problèmes peuvent apparaitre si le fichier n'a pas ce format. Pour éviter de telles erreurs, la commande *file* peut être utilisée pour déterminer le type de contenu d'un fichier. L'exemple suivant nous indique le type de données contenues dans les fichiers */etc/fstab* et */bin/df* .

```
# file /etc/fstab /bin/df
/etc/fstab: ASCII text          → fichier texte codé en ASCII
/bin/df: ELF 32-bit LSB executable, Intel 80386, version 1 (SYSV), … → fichier exécutable
```

2.4. Types de fichiers

Types de fichiers

Sous Linux, on distingue les types de fichiers suivants :

- Les fichiers normaux (ordinary files) symbole : - (appelé dash)
- Les répertoires (directories) symbole : d
- Les liens (link) symbole : l (la lettre L)
- Les fichiers spéciaux (special files) symboles : b, c, p ou s

Les fichiers dits normaux contiennent soit du texte tel que les phrases d'une lettre, le code sources de programmes ou de scripts (on parle alors de fichier texte) soit un programme exécutable (on parle alors de fichiers binaires). Il n'est pas possible de savoir au premier abord quel est le contenu d'un fichier normal, il faudra donc utiliser la commande file pour obtenir ces informations.

Les répertoires sont des fichiers particuliers qui permettent d'organiser l'espace du disque en y « plaçant » les fichiers.

Les liens permettent d'affecter plusieurs noms à un fichier, de manière à pouvoir accéder à ce fichier depuis différents endroits de l'arborescence.

Enfin, les fichiers spéciaux représentent notamment des interfaces avec les périphériques (lecteur CD, disque dur,…) du système.

Extensions de fichier

Une extension est le nom donné à la suite de caractères, précédée d'un point, qui termine[1] le nom d'un fichier. Le système Linux ne gère pas d'extension, néanmoins certains programmes

[1] Cependant, un fichier ne doit pas nécessairement avoir une extension.

nécessites de respecter la convention de nommage liée à l'extension du fichier. Voici une courte liste d'extensions utilisées dans les systèmes Linux.

Extension	Fichier
.c	Fichier source du langage C
.bz2	Fichier compressé par la commande bzip2
.h	Fichier d'en-tête du langage C
.gz	Fichier compressé au format ZIP par la commande gzip
.png	Format ouvert d'images numériques
.rpm	Paquetage de la distribution RedHat
.sh	Script shell
.tar	Archive de type TAR (Tape ARchive)
.tgz	Fichier TAR compressé par gzip
.txt	Fichier texte

2.5. Les liens

Les liens sont des fichiers particuliers permettant d'affecter plusieurs noms à un fichier, de manière à pouvoir accéder au contenu de ce fichier original depuis différents endroits de l'arborescence. Le lien se distingue de la copie de fichiers par le fait que le fichier initial n'est pas recopié plusieurs fois sur le support de stockage. Lorsque l'on crée un lien vers un fichier, on dit que le lien « pointe » vers le fichier en question. Cela signifie qu'en éditant le lien, on a accès au contenu du fichier.

Il existe deux types de liens, les liens durs (hard links) et les liens symboliques (symbolic links). Ils sont outs deux créés à l'aide de la commande *ln*.

Les liens durs (hard links)

Avec les liens durs, toutes les modifications effectuées sur un des fichiers (le fichier initial ou un des liens durs liés à ce fichier) seront répercutées sur le ou les autre(s). Les données du fichier, physiquement présentes sur le disque, ne seront inaccessibles que lorsque tous les fichiers et liens durs qui leur sont liés auront été effacés. La commande ls -l indique le nombre de liens que comporte un fichier : c'est le chiffre venant juste après les permissions.

Alors que nous identifions les fichiers par leur nom, le système identifie chaque fichier par un identificateur unique qui s'appelle le numéro d'i-nœud (inode en anglais). Lorsqu'un lien dur pointe vers un fichier, il a le même i-nœud que ce fichier[1].

Dans l'exemple suivant, nous créons le lien /home/fichier2 pointant vers /tmp/fichier1 :

```
[user]# ln /tmp /fichier1 /home/fichier2
```

Pour connaître l'i-nœud de fichier1 :

```
[user]# ls –i /tmp/fichier1
```

Liens symboliques

Les liens durs ont des possibilités limitées. Par exemple, il n'est pas possible de donner plusieurs noms à des répertoires. Il n'est pas non plus possible de créer un lien dur entre deux fichiers situés sur des systèmes de fichiers différents.

[1] Nous reviendrons sur cette notion "d'inode" dans le chapitre consacré aux systèmes de fichiers.

Les liens symboliques fonctionnent différemment, ils n'utilisent pas l'i-nœud du fichier. De cette manière, ils ne présentent pas les mêmes limitations que les liens durs. Par contre, dès que le fichier initial vers lequel pointe le lien symbolique est supprimé, le lien n'est plus utilisable : il est dit « cassé ».

Pour créer un lien symbolique, il suffit de passer l'option « -s » à la commande *ln* :

```
# ln -s   /tmp/fichier1  /home/fichier2
```

Cette commande crée un lien symbolique /home/fichier2 qui permet d'accéder au contenu du fichier /tmp/fichier1 mais qui possède un i-noeud différent de celui-ci.

La commande « ls –l » indique qu'il s'agit d'un lien symbolique en affichant la lettre l comme premier caractère. En fin de ligne, une flèche indique vers quel fichier ce lien fait référence. Les permissions de /home/fichier2 seront les mêmes que pour /root/fichier1, même si la commande ls -l affiche lrwxrwxrwx pour /home/fichier2. La taille d'un lien symbolique correspond à la longueur du chemin auquel il fait référence.

2.6. Les droits d'accès

2.6.1. Propriétaire et droits d'accès aux fichiers

Linux étant un système multi-utilisateur, les accès aux fichiers doivent être protégés : les fichiers personnels d'un utilisateur ne doivent pas être modifiés par d'autres utilisateurs, les fichiers de configuration du système ne doivent pas être modifiés par n'importe qui. Pour sécuriser ces accès, Linux dispose d'un système de permissions offrant des droits d'accès en lecture, écriture et exécution qu'il est possible de paramétrer selon ses besoins.

Linux utilise trois catégories d'utilisateurs, auxquels il est possible d'octroyer ces droits d'accès :

1. Le propriétaire du fichier (symbole "u" pour user)
2. Les membres du groupe auquel est associé le fichier (symbole "g" pour group)
3. Tous les utilisateurs qui ne sont repris ni en 1) ni en 2) (symbole "o" pour others)

A chaque fichier sont associés un propriétaire et un groupe. Par défaut, le propriétaire est le compte utilisateur qui a créé le fichier, le groupe est le groupe par défaut (UPG[1]) du propriétaire.

Les droits d'accès sont de trois types :

1. Accès en lecture : symbole "r" (pour Read), indique que le fichier peut être lu ;
2. Accès en écriture : symbole "w" (pour Write), indique que le fichier peut être modifié ;
3. Accès en exécution : symbole "x" (pour eXecute), indique que le fichier peut être exécuté. Attention, ce n'est pas parce qu'un fichier présente le droit en exécution qu'il est possible de l'exécuter. En effet, le droit en exécution peut être positionné sur n'importe quel fichier, or, pour que celui-ci soit réellement exécuté, le fichier doit nécessairement être un programme (un binaire exécutable ou un script).

Lorsqu'un droit n'est pas accordé, la lettre r, w ou x est remplacée par un tiret "–". La commande ls –l permet d'afficher les droits concernant les fichiers.

```
[user]#  ls –l nom_fichier
```

[1] User Private Group : cf point 2.2 page 34

```
d rwx r-x r--  2  root  grpadmin  16  sep 12 12:34  nom_fichier
```

Comment interpréter le résultat de la commande ls –l ?

Le premier champ indique le type de fichier. Dans notre exemple, il s'agit d'un répertoire symbolisé par la lettre "d".

Le deuxième champ indique les droits d'accès du propriétaire sur ce fichier (rwx : r=readable, w=writable, x=executable). Ce fichier sera donc accessible en lecture, écriture et exécution pour le propriétaire, *root* dans notre exemple.

Le troisième champ indique les droits (r-x) des utilisateurs membre du groupe auquel est associé le fichier (dans l'exemple, il s'agit du groupe *grpadmin*). Ceux-ci auront donc un accès en lecture et exécution sur le fichier mais ne pourrons en aucun cas le modifier car il ne possède pas le droit en écriture (- au lieu de w).

Le quatrième champ indique les droits d'accès (r--) de tous les autres utilisateurs, on parle généralement du *reste du monde*. Dans notre exemple, ils n'auront qu'un accès en lecture au fichier.

Le cinquième champ est un chiffre qui indique le nombre de liens qui pointent vers ce fichier. S'il s'agit d'un répertoire, il indique le nombre de sous-répertoires.

Le sixième champ indique le nom (root) du propriétaire du fichier.

Le septième champ indique le nom du groupe (grpadmin) associé au fichier.

On trouve ensuite la taille du fichier en octets (16), la date de sa dernière modification (sep 12 12:34) et enfin, le nom du fichier (nom_fichier).

Remarques importantes

Les droits d'accès peuvent avoir une signification différente suivant le type de fichier auquel ils se réfèrent. Ainsi, dans le cas d'un répertoire, elles ont les significations suivantes :

- le droit de lecture autorise à consulter le contenu d'un répertoire,
- le droit d'écriture autorise à ajouter ou à supprimer des fichiers dans le répertoire,
- le droit d'exécution autorise à se déplacer dans ce répertoire.

Sauf cas rarissimes, les droits de lecture et d'exécution des répertoires vont de pair. Il est rare de vouloir autoriser l'accès à un répertoire pour interdire à l'utilisateur de lister son contenu.

Par défaut, si vous autorisez les utilisateurs à ajouter des fichiers dans un répertoire, vous leur donnez par la même occasion le droit de supprimer n'importe quel fichier du répertoire. En effet, autoriser l'accès en modification sur le répertoire, rend possible le fait de modifier le contenu du répertoire. Il est toutefois possible de partager un répertoire sans qu'aucun utilisateur ne puisse supprimer les fichiers des autres en utilisant le *sticky bit*[1].

2.6.2. Modifier les droits d'accès à un fichier

Lorsqu'un fichier est créé, le système lui attribue automatiquement des droits d'accès par défaut. Ceux-ci dépendent du type de fichier ainsi que de l'utilisateur qui le crée. Ils conviennent dans la plupart des cas, mais pas toujours. Par exemple, si vous créez un script, vous devez attribuer vous-même le droit d'accès d'exécution au fichier pour qu'il puisse s'exécuter. N'oubliez pas que les permissions sont là pour protéger les fichiers. Réfléchissez

[1] cf 2.7 les droits d'accès étendus

bien aux conséquences lorsque vous les modifiez. Deux méthodes permettent de modifier les droits d'accès en mode texte.

A. La méthode symbolique

La commande **chmod** permet de changer les droits d'accès à un fichier. Pour ce faire, il est nécessaire de préciser au moins deux points :

a) A quelle(s) catégorie(s) d'utilisateurs est appliqué le changement :
1) symbole "u" pour le propriétaire du fichier,
2) symbole "g" pour les membres du groupe auquel appartient le fichier,
3) symbole "o" pour tous les utilisateurs qui ne sont repris ni en 1) ni en 2),
4) symbole "a" pour toutes les catégories. Par défaut, si aucune catégorie (u, g, o ou a) n'est précisée, le changement s'applique à toutes les catégories.

b) Quelle modification doit être effectuée :
1) le symbole "+" ajoute un droit,
2) le symbole "-" retire un droit,
3) le symbole "=" fait en sorte qu'il s'agisse de l'unique autorisation.

Par exemple, la commande suivante rend exécutable le fichier *prog1*

```
[user]# chmod +x prog1
```

La commande suivante rends exécutable le fichier *prog1* pour le propriétaire et le groupe

```
[user]# chmod ug+x fichier
```

N'oubliez pas de vérifier les modifications effectuées à l'aide de la commande ls –l.

B. La méthode absolue

La méthode absolue consiste à spécifier de manière binaire si un droit doit être appliqué ou non : 0 = pas d'accès ; 1 = accès autorisé. Comme il existe trois séries de trois droits à spécifier, on utilise une méthode octale, plus facile à manipuler.

Droits d'accès activé (1) ou non (0)			Chiffre octal correspondant	Signification
r	w	x		
0	0	0	0	- - - : Aucun accès autorisé
0	0	1	1	- - x : accès en exécution
0	1	0	2	- w - : accès en écriture
0	1	1	3	- w x : accès en écriture et exécution
1	0	0	4	r - - : accès en lecture
1	0	1	5	r – x : accès en lecture et exécution
1	1	0	6	r w - : accès en lecture et écriture
1	1	1	7	r w x : accès en lecture, écriture et exécution

Comme il est nécessaire de spécifier les accès pour toutes les catégories d'utilisateurs (u, g, o), on aura, au final, un nombre composé de trois chiffres. Par exemple, rwx r-x r-- équivaut aux chiffres 7 (rwx), 5 (r-x) et 4 (r--). Il suffira d'accoler ces trois chiffres et la commande à utiliser sera :

```
[user]$ chmod 754 fichier
```

Les paramètres d'autorisations 666 ou 777 autorisent n'importe quel utilisateur à lire et modifier un fichier ou un répertoire. Il est évidemment déconseillé de mettre en place une telle configuration à moins que ce ne soit réfléchi et voulu.

La commande umask

Nous l'avons déjà signalé, le système attribue des droits par défaut à chaque fichier nouvellement créé. Ceux-ci se paramètrent avec la commande *umask*.

Exemple de configuration grâce à la notation symbolique :

```
[root]#  umask u=rxw,g=rx,o=r
```

Exemple de configuration grâce à la méthode absolue :

```
[root]#  umask 022
```

Attention, la valeur du "umask" correspond à une valeur de masquage. Les droits de tout nouveau fichier seront masqués par ce nombre. Ainsi, dans le cas d'un fichier exécutable ou d'un répertoire, les droits seront fixés en réalisant l'opération AND NOT entre les deux valeurs binaires. Dans le cas d'un répertoire : 777 AND NOT 022 = 755 (-rwxr-xr-x). Dans le cas d'un fichier, les droits seront 666 AND NOT 022 = 664 (-rw-rw-r--).

2.7. Droits d'accès étendus

2.7.1. Le droit d'endossement

Le droit d'endossement peut s'appliquer à un fichier exécutable ou à un répertoire. Son comportement est cependant différent avec l'un ou avec l'autre.

Les droits d'endossement permettent d'obtenir les droits du propriétaire ou du groupe associé à un fichier exécutable quand celui-ci est exécuté par un autre utilisateur. Ainsi, lorsqu'une commande est exécutée, elle endosse l'identité du propriétaire ou du groupe associé plutôt que celle de l'utilisateur qui en a demandé l'exécution.

Lorsque le droit d'endossement s'applique au groupe associé à un répertoire, les fichiers qui sont créés dans ce répertoire, sont automatiquement associés au groupe du répertoire et non au groupe de l'utilisateur qui les crée (Les fichiers « héritent » du groupe du répertoire).

Pour définir ces droits d'endossement, il faut utiliser l'option u+s ou g+s (on parle également de Set UID bit ou Set GID bit) ou bien ajouter les valeurs 4 pour l'utilisateur ou 2 pour le groupe en préfixe aux droits d'accès usuels (r, w, x).

Les deux exemples suivants permettent de positionner respectivement le SUID bit et le SGID bit sur un fichier exécutable :

```
# chmod  4755  prog1.sh
# chmod  2755  prog2.sh
```

Un exemple concret, lié à l'utilisation du droit d'endossement, concerne la commande *passwd* (permettant de modifier le mot de passe d'un utilisateur).
Si l'on observe les droits de la commande /usr/bin/passwd, on obtient :

```
-  rws  - -x - -x        1       root     root     13476          aoû     7        2001    passwd
```

On voit le bit, normalement positionné à "x" pour le propriétaire, remplacé par un "s". Le droit d'endossement est donc "activé". A la base, un utilisateur ne peut pas modifier son mot de passe car l'opération nécessite la modification de fichiers de configuration accessibles

uniquement au compte root. En utilisant cette technique, un utilisateur quelconque ne pourra toujours pas éditer et modifier le fichier contenant les mots de passe. Par contre, en utilisant la commande /usr/bin/passwd, celle-ci accède au fichier /etc/passwd avec les droits de root et peut dès lors le modifier. Cependant, l'utilisateur ne pourra évidemment modifier que son propre mot de passe (et rien d'autre dans le fichier) car l'accès au fichier ne se fait que via la commande passwd.

Risques potentiels

L'utilisation du droit d'endossement avec un fichier exécutable présente des risques du point de vue de la sécurité car elle « donne » le droit d'accès à une commande et pas à une catégorie d'utilisateurs. Cette technique ne doit donc pas être généralisée, surtout quand elle permet l'exécution en tant qu'utilisateur root.

2.7.2. Le sticky bit

De nouveau, ce droit d'accès, appelé « sticky bit » ou bit "t", peut être appliqué sur un fichier exécutable ou sur un répertoire.

Dans le cas d'un fichier exécutable, celui-ci restera en mémoire même après la fin de son exécution. Cela lui permet d'être relancé plus rapidement en cas d'utilisation fréquente.

Dans le cas d'un répertoire, il signifie qu'un fichier situé dans ce répertoire ne pourra être modifié que par son propriétaire. Par exemple, tout utilisateur peut copier des fichiers dans le répertoire /tmp (droits rwx pour tout le monde). Pour que ces fichiers ne puissent pas être effacés par un autre utilisateur que celui qui les a créés, il faut positionner le Sticky Bit avec une des commandes suivantes :

```
chmod o+t /tmp
chmod 1777 /tmp
```

Module 4 : Gestion des utilisateurs et des groupes

1. Introduction

Il est nécessaire de disposer d'un compte utilisateur pour se connecter à une station Linux. Ce compte, identifié par un nom (le login) et authentifié (par exemple via un mot de passe), présente plusieurs avantages, tant pour l'utilisateur que pour l'administrateur:

- Chaque utilisateur dispose de son propre répertoire de travail (le home directory) dans lequel il peut stocker ses fichiers personnels mais aussi ses fichiers de configurations. L'utilisateur dispose ainsi d'une relative « intimité » (relative car l'administrateur à accès à tous les fichiers) ;
- L'identification de l'utilisateur permet de personnaliser son environnement de travail ;
- Le système garde une trace des connexions/déconnexions, ce qui permet de connaître l'auteur d'éventuels événements indésirables, tels qu'une tentative d'attaque du système.

Le contrôle des comptes utilisateurs est au cœur de l'administration de système et une bonne gestion de ceux-ci est indispensable au bon fonctionnement.

2. Notions d'utilisateurs et de groupes

2.1. Les utilisateurs

Un compte utilisateur peut être utilisé par des utilisateurs physiques (vous et moi) qui se connectent au système via un login et un mot de passe. On parle alors de « compte physique ». Il existe également des comptes dit « système » ou « logique », utilisés uniquement par des applications spécifiques (par exemple pour démarrer des applications avec des permissions liées à ce compte système). En parcourant le fichier /etc/passwd, vous constaterez que le système, dès son installation, a créé une série de « comptes système » et un seul compte physique : root.

Chaque utilisateur physique dispose d'un répertoire personnel dans /home. Ce répertoire contient un ensemble de fichiers de configuration tels que .bashrc ou .kderc. Ceux-ci sont lus à chaque connexion d'un utilisateur et permettent de personnaliser l'environnement de celui-ci. De tels fichiers préconfigurés sont copiés automatiquement dans le « home directory » de chaque utilisateur nouvellement créé au départ du répertoire /etc/skel.

2.2. Les groupes

Un groupe est un regroupement d'utilisateurs qui partagent les mêmes permissions sur un ensemble de fichiers (exécutables ou non) et répertoires. Les utilisateurs membre d'un groupe donné disposent de toutes les permissions des fichiers et répertoires associés à ce groupe.

Chaque utilisateur doit faire partie au moins d'un groupe : son groupe primaire. Un compte utilisateur peut être membre de plusieurs autres groupes, ceux-ci sont appelés ses groupes secondaires.

Groupes propres à l'utilisateur

Dans les distributions de type RedHat, Linux utilise un système de groupes propres à l'utilisateur appelé UPG ("User Private Group") dont le but est de faciliter la gestion des groupes d'utilisateurs. Un UPG est créé chaque fois qu'un nouvel utilisateur est ajouté au système. Par défaut, les UPG portent le même nom que le compte utilisateur pour lequel ils ont été créés et seul cet utilisateur est membre de l'UPG.

2.3. UID- GID

En interne, le système Linux travaille avec des nombres, les noms sont seulement utilisés pour faciliter leurs manipulations par des "humains". Ainsi, chaque utilisateur se voit attribuer un numéro d'identification unique : le User IDentifier (UID). Il en est de même pour les groupes, qui sont identifiés par leur Group-IDentifier (GID).

La bonne gestion des utilisateurs, des groupes et des permissions associées aux fichiers font partie des tâches de base mais importantes qu'un administrateur système doit effectuer.

2.4. Outils de gestion des utilisateurs et des groupes

Comme souvent sous Linux, une multitude de possibilités existent : commandes, outils graphiques, scripts,… La méthode la plus simple pour gérer les utilisateurs et les groupes consiste à utiliser une application graphique, cependant celle-ci ne permet pas toujours de réaliser toutes les opérations souhaitées. Dès lors, le recours à la ligne de commande peut vous aider à créer vos propres scripts de gestion.

Voici quelques commandes liées à la gestion des comptes

Commandes	Rôle
useradd, usermod et userdel	permettent d'ajouter, de supprimer et modifier des comptes d'utilisateurs
groupadd, groupmod et groupdel	permettent d'ajouter, de supprimer et modifier des groupes d'utilisateurs
pwck, grpck	permettent de vérifier le mot de passe, le groupe et les fichiers masqués associés
Passwd	permet de modifier le mot de passe d'un utilisateur
Id	permet de connaitre l'UID d'un utilisateur ainsi que les groupes dont l'utilisateur est membre.

3. Utilisateurs et sécurité

La première règle à respecter en matière de sécurité consiste à utiliser le compte *root* uniquement pour réaliser des tâches administratives. Un compte utilisateur « normal », doit être préféré pour le travail quotidien. En effet, root dispose de tous les droits sur le système, de telle sorte que la moindre erreur commise sous ce compte peut endommager les données, mais également l'ensemble du système. De la même manière, un virus ou un programme « bogué » exécuté avec les permissions associées au compte root a potentiellement accès à l'ensemble du système, pouvant ainsi causer des dommages sérieux. Ne pas travailler sous le compte root restreint à un seul utilisateur les dégâts qui pourraient être occasionnés.

Ainsi, il est indispensable que chaque utilisateur dispose d'un compte ayant des droits d'accès limités. Le compte root ne doit donc être réservé qu'aux tâches d'administration, et toute opération réalisée sous cette identité doit être contrôlée deux fois avant d'être effectivement exécutée.

3.1. Mécanismes d'authentification des utilisateurs

Les différents utilisateurs (locaux ou distants) d'un système (client ou serveur) n'ont pas tous les mêmes droits d'accès sur ces systèmes. Il convient donc de s'assurer que chaque utilisateur est bien celui qu'il prétend être, afin de donner à chacun les privilèges auxquels il a droit.

Ainsi, les accès aux systèmes et à leurs services reposent sur deux opérations essentielles : l'identification et l'authentification. L'opération d'identification consiste à préciser qui est l'utilisateur (compte utilisateur) afin de déterminer quelles sont ses privilèges. L'opération d'authentification consiste à prouver que l'utilisateur est bien celui qu'il prétend être. La plupart du temps, un système doit refuser ses services aux utilisateurs inconnus ou qui n'ont pas passé avec succès la phase d'authentification. Cependant, il existe des situations où l'authentification importe peu car l'objectif est de fournir un service à tout le monde. Une solution consiste à utiliser un compte spécifique (compte « invité » par exemple) via lequel tous les utilisateurs inconnus pourront se connecter.

De base, l'authentification des utilisateurs se fait par mot de passe, bien que d'autres mécanismes soient possibles (carte magnétique, biométrie,…). Classiquement, ces mécanismes se déroulent de la manière suivante :

- le système demande à l'utilisateur son login,
- il demande ensuite son mot de passe,
- il vérifie la validité du couple "login / mot de passe",
- si l'utilisateur dispose bien d'un compte sur le système et s'est correctement authentifié, le système lui ouvre une session en fixant son environnement et ses préférences personnelles, puis lui donne accès au système via un shell.

Pour mener à bien l'authentification, les informations relatives aux utilisateurs doivent être stockées quelque part. Par défaut, elles se trouvent sur le système dans les fichiers de configuration */etc/passwd* et */etc/shadow*[1]. Cette technique fonctionne très bien pour un usage local, mais n'est pas pratique pour la gestion d'utilisateurs itinérants. Dans ce genre de configuration, on utilisera plutôt un service réseau pour récupérer les informations concernant les utilisateurs à partir d'un serveur d'authentification centralisé (LDAP par exemple).

3.2. Former les utilisateurs

Dans un système informatique, c'est souvent l'utilisateur qui est le maillon faible. Celui-ci peut intentionnellement chercher ou créer des failles de sécurité (tentative d'accès à des services auxquels il n'a pas droit ou à des données confidentielles pour les revendre). Cependant, l'utilisateur va le plus souvent affaiblir la sécurité du système sans le vouloir, simplement par méconnaissance ou naïveté (en divulguant son login et son mot de passe, en installant des programmes non vérifiés, etc).

Un utilisateur à qui l'on interdit de réaliser une action qui lui semble normale (télécharger des vidéos, exécuter des programmes issu d'Internet,....) à tendance à critiquer les responsables informatiques et cherche parfois à contourner cette limitation. En revanche, l'utilisateur qui a compris les risques engendrés par ses actions peut généralement s'en passer ou faire appel à un responsable afin de trouver une solution adaptée. Une des tâches de l'administrateur sera donc d'informer et de former les utilisateurs aux bases de la sécurité informatique.

[1] Le fichier /etc/passwd propose des permissions en lecture à tout le monde, de sorte que les informations utilisées pour l'authentification pouvaient être utilisées pour essayer de « casser » les mots de passe. A présent, les informations sont stockées dans le fichier /etc/shadow qui n'est lisible que par l'utilisateur root.

3.3. Les mots de passe

Les mots de passe constituent un point faible dans une stratégie de sécurité. D'une part, ils sont souvent négligés par les utilisateurs : noté sur des post-it, confié à un collègue, … (d'où l'importance de former les utilisateurs). D'autre part, ils peuvent être « crackés » parfois très facilement.

« Lors de ces RIAM 2007, Cyber-Ark avait dévoilé une étude démontrant que : 51% des administrateurs reconnaissent noter leurs mots de passe sur un Post-It, 8% avouent avoir laissé les mots de passe par défaut (éditeurs), 21% craignent de ne pouvoir subir avec succès un audit de sécurité et 27% ont connaissance d'anciens collègues dont les droits sont toujours valides. »[1]

Les mots de passe ne doivent donc pas être stockés en clair sur un système, mais sous une forme cryptée. Pour ce faire, Linux utilise le système *shadow password* (mots de passe masqués). Ce système permet de placer les mots de passe cryptés dans */etc/shadow*, lisible seulement par root, et autorise la gestion d'informations sur l'expiration de mots de passe.

L'authentification par mot de passe n'est sans doute pas la solution la plus sécurisée, mais elle est facile à mettre en œuvre et les utilisateurs y sont habitués, raison pour laquelle elle est utilisée par défaut. Pour améliorer la sécurité sur les systèmes critiques, une bonne solution consiste à utiliser des systèmes d'authentification plus sûrs, voire plusieurs en série (carte magnétique, carte à puce, biométrie). Le tout étant de garder un compromis entre facilité d'usage (et de gestion) et efficacité de la sécurité.

Complexité d'un mot de passe

Les utilisateurs (et malheureusement certains administrateurs) n'ont pas envie de retenir des mots de passe compliqués, ils vont donc chercher à utiliser des séquences simples (« 12345 », « azerty », « 123abc »…) ou qui ont une signification les rendant faciles à retenir (leur acronyme, le nom du chien, la date de mariage, …). Certain penseront à inverser deux lettres ou remplacer une lettre par un caractère spécial ($ pour s, @ pour a) mais de tels mots de passe restent trop simpliste pour être efficace. En effet, des programmes spécialisés permettent de réaliser une attaque par dictionnaire. Ceux-ci tentent toutes les valeurs d'une liste de mots de passe (le dictionnaire), ainsi qu'une série de variantes. Tous les mots de passe simples seront trouvés en quelques secondes.

En ce qui concerne les mots de passe un peu plus complexes, mais basés sur des informations personnelles à l'utilisateur, un logiciel ne sera peut-être même pas nécessaire. Le nom du chien ou la date du mariage peuvent facilement s'acquérir autour d'une tasse de café pendant une pause. Cette technique de récupération d'informations personnelles est l'une des bases du social engineering[2].

Confidentialité d'un mot de passe

L'utilisateur doit bien comprendre que son mot de passe est strictement personnel, et qu'il ne doit être communiqué à PERSONNE (y compris à l'administrateur). N'hésitez pas à insister lourdement sur ce point, cela vous permettra d'éviter les mots de passe inscrit sur des post-it ou les utilisateurs donnant leur mot de passe au premier qui le demande.

---

[1] Information Presse - Nanterre, le 21 novembre 2007, [en ligne] www.atheos.fr/presse/CPAtheos-CyberArk.pdf

[2] Le social engineering est une technique qui a pour but d'extirper des informations à des personnes, sans qu'elles ne s'en rendent compte, en exploitant leur confiance, leur ignorance ou crédulité.

Module 5 : Gestion des systèmes de fichiers

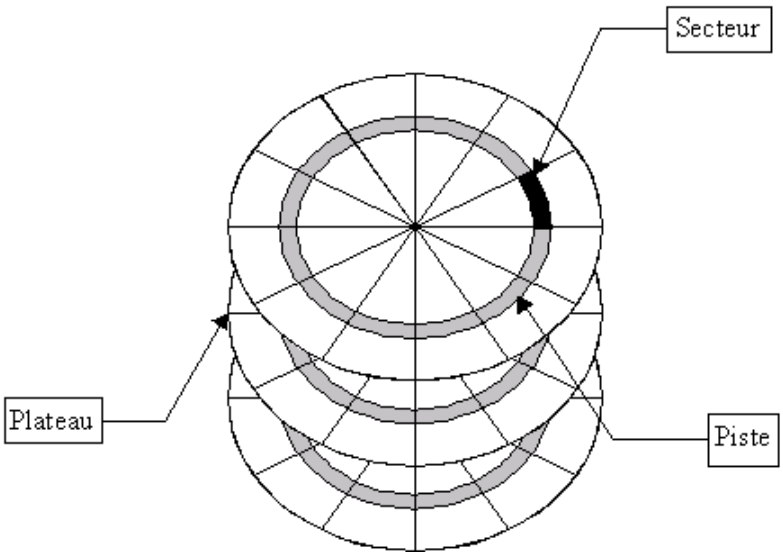
1. Qu'est-ce qu'un système de fichiers ?

Le système de fichiers (ou File System ou FS), c'est la manière dont le système d'exploitation structure les données sur un support de stockage (disque dur, par exemple). Cette structure est nécessaire afin de retrouver et accéder rapidement à un fichier. En effet, imaginez que vous utilisez un container dans lequel vous déposez toutes vos notes de cours, factures, lettres, publicités, etc. Après quelque temps de fonctionnement il sera très difficile de mettre la main sur un document précis. Par contre, si le container est structuré, par exemple, en contenant des armoires à tiroir avec des étiquettes permettant de trier vos papiers en fonction des différents types de données, des années, ..., il sera plus facile de retrouver ce que vous cherchez. Ne confondez pas le système de fichiers avec l'arborescence. Celle-ci est une représentation logique (une vue hiérarchique sous forme arborescente) de l'ensemble des fichiers et répertoires présents sur le système.

Comme tout système d'exploitation, Linux dispose de son propre système de fichiers : EXT2FS (Extended 2 File System) nommé aussi simplement ext2. En comparaison, Windows utilise quant à lui un système de fichiers appelé NTFS (ou encore les anciens FAT, FAT32). Il existe des dizaines de FS, chacun ayant ses propres caractéristiques : FFS (Fast File System), HPFS (High Performance FileSystem), ReiserFS, JFS (Journaled File System), NFS (Network File System)… La description d'un système de fichiers dépend du support utilisé, un FS réseau tel que NFS ne fonctionne pas comme ext2. Nous nous contenterons, ici, d'aborder l'étude du FS ext2 utilisé sur un disque dur.

1.1. Structure d'un disque dur

Comme le montre la figure 1, un disque dur est composé de pistes circulaires concentriques découpées en arcs de cercles appelés secteurs. Ceux-ci sont repérés par un couple (n° piste/n° secteur) et sont formés d'un ensemble d'octets (128, 256, 512,… octets).



Figue 1 : pistes et secteurs d'un disque dur

Source :
http://docs.mandragor.org/files/Operating_systems/Linux/Guide_d_installation_et_de_configuration_de_Linux_fr/images/hard_drive.png

Un disque dur étant un périphérique de stockage de type bloc, la modification d'un octet ou d'un bit n'est pas faisable, tous les échanges avec le disque (lecture, écriture) se font par blocs. Un bloc peut correspondre à un, deux ou plusieurs secteurs, mais jamais moins d'un secteur, le secteur étant indivisible.

Vu leur structure, les systèmes de fichiers tel ext2 considèrent le disque dur comme une succession de blocs (ayant chacun un numéro : leur adresse physique) dont ils rangeront les adresses dans une table afin de les retrouver facilement. Les fichiers, suivant leur taille, sont stockés dans un ou plusieurs blocs, de préférence, contigus. Si les blocs contenant le fichier ne sont pas contigus, le fichier sera fragmenté, c'est-à-dire réparti un peut partout sur le disque. Trop de fragmentation peut ralentir le système et les accès aux données, c'est pourquoi il existe des logiciels permettant de défragmenter le FS.

1.2. Structure du système de fichiers ext2

Un système de fichiers ext2 se compose de plusieurs éléments qui ont été rassemblés en *groupes de blocs* dans le but de permettre un accès plus efficace aux données. Cette structure est représentée par le tableau 2.

	Bloc d'amorçage
Groupe de blocs 1	Superbloc
	Liste de description des groupes de blocs
	carte de blocs
	carte d'inodes
	Table des inodes
	Blocs de données
Groupe de blocs 2	Superbloc
	Liste de description des groupes de blocs
	carte de blocs
	carte d'inodes
	Table des inodes
	Blocs de données
⋮	Superbloc
	Liste de description des groupes de blocs
	…

Tableau 2 : Structure du système de fichiers ext2

1.2.1. Le bloc d'amorçage

Le bloc d'amorçage est le premier bloc (bloc n°0) du système de fichiers. Il contient un programme pour lancer et initialiser le système. Sa taille est de 1024 octets en ext2. Après ce bloc, se trouve une suite de groupe de blocs (constitué de 8192 blocs) contenant chacun :

- un superbloc,
- une liste de description des groupes de blocs,
- une carte de blocs,
- une carte d'inodes,
- une table des inodes,
- des blocs de données.

1.2.2. Le super bloc

Le super bloc contient des informations cruciales pour le fonctionnement du FS car elles définissent la structure et l'état ce celui-ci et permettent la gestion de toutes les informations qui y sont contenues. Voici quelques informations contenues dans le super bloc :

- la taille totale du système de fichiers (en blocs et en inodes),
- le nombre de blocs libres,
- le nombre de blocs réservés aux inodes,
- la taille d'un bloc de données,
- l'heure de la dernière modification effectuée,
- l'heure de la dernière vérification du système de fichiers.

Le FS comporte plusieurs copies du super bloc, permettant d'en trouver un intact si l'un d'eux est défectueux.

1.2.3. Liste de description des groupes de blocs

Derrière chaque super bloc figure une liste de description des groupes de blocs. Celle-ci permet de connaître rapidement l'adresse des groupes de blocs et des inodes, ce qui facilite l'accès aux blocs de données. Elle répertorie également le nombre de blocs ou d'inodes libres.

1.2.4. Les cartes de blocs et d'inodes

Ces cartes (map en anglais) répertorient l'occupation des blocs et des inodes, permettant au système de connaître rapidement les zones libres. Pour savoir si un bloc ou un inode est utilisé ou non, on lui affecte un bit : positionné à 1, le bloc ou l'inode est utilisé ; positionné à 0 le bloc ou l'inode est libre.

1.2.5. La table des inodes

Inodes

Un inode (Information node, nœud d'information) est un enregistrement de taille fixe qui contient « tous » (tous, sauf le nom du fichier) les éléments décrivant un fichier :

- le type de fichier (ordinaire, répertoire, lien,...),
- la taille du fichier,
- le propriétaire et le groupe associé au fichier,
- les permissions d'accès associées au fichier,
- les dates de création, modification et accès (ctime, mtime, atime),
- des pointeurs vers des blocs de données (où sont contenus réellement les fichiers),
- le nombre de liens (liens durs),
- ...

Lorsque l'on crée un fichier dans un répertoire, les informations sur ce fichier ne sont pas inscrites dans le répertoire mais dans un inode non affecté. Seuls le numéro de cet inode et le nom du fichier sont inscrits dans répertoire. Le répertoire est en réalité un fichier particulier qui contient des liens, c'est à dire un couple « nom de fichier + numéro d'inode ». C'est donc le répertoire qui permet de retrouver un fichier à partir de son nom.

La commande ls –i permet de connaître le numéro d'inode d'un fichier.

La commande df –i permet de connaître le nombre d'inodes libres et occupés d'un système de fichiers

Table des inodes

Tous les inodes sont regroupés dans un tableau spécial du système de fichiers : la table des inodes. Il existe une table des inodes par groupe de blocs et sa taille est un paramètre important du système de fichiers car elle limite le nombre de fichiers que l'on peut y créer. En effet, d'une part, le nombre d'inodes est limité par la taille de la table, d'autre part il est nécessaire de disposer d'un inode libre pour y stocker les informations du fichier à créer. Il est nécessaire de tenir compte de ce point lorsque l'on met en place un système de quotas d'espace disque.

La taille de cette table est spécifiée par l'utilisateur lors de l'installation. Certains inodes sont réservés, par exemple, le premier inode de la table sert à mémoriser les "bad blocs" du système de fichiers, le deuxième correspond au répertoire racine du système.

Plus cette table est grande, plus le nombre de fichiers que l'on pourra créer est grand. En contrepartie, il faudra plus de temps pour parcourir la table à la recherche d'un fichier, et une trop grande table risque de ralentir les accès. Il faut donc trouver un compromis entre la taille de cette table et le nombre de fichiers que l'on est susceptible de créer. Par conséquent, il est courant de définir un taux d'inodes par bloc.

Remarques

Un fichier n'est supprimé que lorsque son nombre de liens tombe à zéro. De plus, lorsque l'on supprime un fichier, l'inode qui le caractérise est libéré et le nom du fichier est supprimé du répertoire dans lequel il se trouvait. D'ailleurs, la commande *rm* utilise l'appel système unlink qui ne fait que détruire un lien. Le nombre de liens matériel (hard link) inscrit dans l'inode est décrémenté à chaque suppression d'un lien et l'inode est libéré lors de la suppression du dernier lien qui le référence. C'est la raison pour laquelle l'effacement d'un fichier est plus rapide que son écriture : en réalité, les données sont toujours présentes sur le disque, on a simplement supprimé le lien qui permettait d'accéder aux blocs contenant les données concernées.

1.2.6. Les blocs de données

C'est dans ces blocs de données que sont stockés les fichiers. Ceux-ci sont repérés grâce aux pointeurs qu'il y a dans l'inode. Certains de ces pointeurs sont directs, mais il y en a aussi des indirects, doublement indirects... Quand les pointeurs directs sont épuisés et que le fichier grossit, on alloue parmi les blocs de données du disque un bloc qui va servir à ranger de nouveaux pointeurs vers d'autres blocs. Le premier pointeur indirect de l'inode pointera vers le bloc ainsi alloué.

1.3. Création des systèmes de fichiers

Le système de fichiers ext2 travaille avec des blocs de taille fixe, de telle sorte qu'il est impossible d'allouer seulement une partie d'un bloc. L'allocation de l'espace disque se fait par multiple de la taille d'un bloc. L'allocation et la libération d'espace disque est donc très simple, ce qui la rend rapide. Malheureusement, il existe une perte d'espace disque chaque fois que l'espace occupé par un fichier ne correspond pas à un nombre entier de blocs. En effet, dans l'ensemble des blocs occupés par le fichier, le « dernier » bloc ne sera que partiellement rempli.

La taille d'un bloc (multiple de la taille d'un secteur) et le nombre d'inodes du système de fichiers sont paramétrés lors de la création du système de fichiers. La taille des secteurs d'un disque dur étant généralement de 512 octets, la taille d'un bloc sera un multiple de 512 octets.

Des blocs de 1024 octets (2 secteurs) avec 4096 octets par inode sont des valeurs couramment utilisées.

La création d'un FS en ext2, se fait à l'aide de la commandes *mkfs* (ou *mke2fs*) dont un exemple d'utilisation est donné ci-dessous :

```
# mke2fs  -b  taille  fichier
```

où *taille* représente la taille d'un bloc en octets et *fichier* est le nom du fichier spécial de périphérique représentant la partition sur laquelle le système de fichiers doit être créé. Si la taille n'est pas précisée, *mke2fs* prend des valeurs par défaut appropriées pour cette partition

1.4. Gestion des systèmes de fichiers

L'utilisateur moyen ne voit que l'arborescence des fichiers, la répartition de l'espace physique sur le disque et la gestion des systèmes de fichiers est du ressort de l'administrateur. Celui-ci a la possibilité, lors de la mise en place d'un système de fichiers, de laisser les paramètres par défaut ou de choisir lui-même la configuration des systèmes de fichiers.

1.4.1. Pourquoi gérer soi-même les systèmes de fichiers ?

La méthode la plus simple consiste à ne s'occuper de rien, et espérer que la configuration par défaut soit adaptée à notre utilisation. Cependant, plusieurs avantages montrent que l'administrateur a tout intérêt à gérer lui-même les systèmes de fichiers :

- Disposer de plusieurs systèmes de fichiers (partitions) augmente la sécurité car si l'un d'eux est endommagé, les dégâts sont limités à ce système de fichiers. Il est également possible d'utiliser des systèmes de fichiers cryptés pour y stocker des données confidentielles.
- Les performances sont améliorées car le morcellement des fichiers (fragmentation) est réduit à l'espace occupé par le système de fichiers (et non à l'ensemble du disque).
- La répartition de l'espace disque sur plusieurs systèmes de fichiers permet de cloisonner l'espace pour les utilisateurs (et/ou les applications). Ceux-ci n'auront la possibilité de créer des fichiers que dans les répertoires de leurs propres systèmes de fichiers, ce qui permet de mieux contrôler l'espace disque utilisé.
- La gestion des backups « physiques » est facilitée car il est possible de définir des systèmes de fichiers de même taille que les unités d'archivages.

1.4.2. Accéder à un nouveau système de fichiers

Chaque système de fichiers est indépendant des autres et, est, en principe, mis en place au moment de l'installation de Linux. Si par la suite vous souhaitez brancher un nouveau disque dur sur votre système, un nouveau système de fichiers devra y être créé (ou plusieurs suivant le nombre de partitions, chaque système de fichiers donnant lieu à une partition séparée sur le nouveau disque dur).

Nous avons vu que l'arborescence d'un système Linux est unique. Dès lors, tous les répertoires, partitions et périphériques se trouvent nécessairement quelque part « sous » la racine de cette arborescence unique. Ainsi, Linux doit permettre de rassembler plusieurs systèmes de fichiers dans une même arborescence car pour que ce système de fichiers soit

accessible à l'utilisateur, il doit obligatoirement être « intégré » dans le premier système de fichiers (le root system, le système racine)[1].

L'action d'attacher un nouveau système de fichiers à un répertoire d'un système de fichiers actif[2] est appelée « montage ». Le répertoire en question est appelé *point de montage*, c'est à partir de lui qu'il est possible d'accéder au nouveau système de fichiers. Pour autant qu'ils soient reconnus par le système Linux, de nombreux types de système de fichiers peuvent ainsi être montés : peu importe leur type (vfat, ext2, swap, iso ..), leur support (partitions d'un disque dur, périphériques amovibles (cd, disquettes, clé USB, ...), ou leur localisation (supports accessibles par le réseau : partages NFS, Samba,…). Linux autorise aussi le montage d'un système de fichiers sur un système de fichiers différent. Ainsi, il sera possible de monter une partition NTFS sur un FS ext2 (par exemple) et ainsi accéder de manière transparente aux données de cette partition.

1.4.3. Comment monter un système de fichiers

Pour monter les différents systèmes de fichiers dans une même arborescence, l'administrateur Linux dispose de plusieurs outils. Il peut utiliser le fichier /etc/fstab, qui permet de monter automatiquement des FS au démarrage de l'ordinateur ou lorsqu'un périphérique est détecté. Pour réaliser un montage manuellement ou par l'intermédiaire d'un script, il peut utiliser la commande *mount*. Celle-ci attend trois paramètres:

1. Le type de système de fichiers (ext2, NFS ? NTFS,…)
2. Le nom du fichier de périphérique par lequel Linux accède au système de fichiers (par exemple /dev/hda3).
3. Le nom du répertoire devant servir de point de passage vers le nouveau système de fichiers (le point de montage).

Voici un exemple de montage permettant d'accéder au système de fichiers présent sur un CD-ROM.

```
# mount  –t  iso9660  /dev/cdrom  /media/cdrom
```

Les commandes mount/umount[3] exécutent le montage/démontage des systèmes de fichiers en suivant les directives contenues dans le fichier /etc/fstab. Si celui-ci contient les informations sur la manière de monter le CD-ROM, l'utilisation de la commande mount en est simplifiée car il n'est plus nécessaire que de déterminer le point de montage :

```
# mount  /media/cdrom                # préconfiguré dans /etc/fstab
```

Le fichier /etc/fstab est donc très pratique pour préconfigurer certains montages.

- Il évite de devoir monter manuellement les systèmes de fichiers à chaque démarrage ;
- Il évite de devoir, lors de chaque montage/démontage, taper manuellement tous les renseignements ;
- Il permet d'automatiser le montage/démontage de certains systèmes de fichiers. L'option *supermount* permet, par exemple, de monter automatiquement un CD-ROM lorsque celui-ci est inséré dans le lecteur.

La commande mount appelée sans aucun paramètre affiche tous les FS intégrés à l'arborescence.

---

[1] Ou bien dans un autre système de fichiers intégré au système de fichiers racine.
[2] Un système de fichier actif est soit le système de fichiers racine, soit un système de fichiers déjà intégré (donc monté) au système de fichiers racine.
[3] La commande umount permet de démonter un système de fichiers. cf point 1.4.4.

Attention

Si le répertoire /media/cdrom n'est pas vide au moment du montage (parce que vous y avez créé ou copié des fichiers) les fichiers qu'il contient ne sont plus visibles et donc plus accessibles. Cependant, ils ne sont pas perdus, ils réapparaissent dès que le système de fichiers du CD-ROM est démonté. Cela est du au fait que vous utilisez, lors du montage, le répertoire /media/cdrom comme point de montage pour accéder à un système de fichiers.

Remarque : Linux mémorise les paramètres du montage des systèmes de fichiers dans le fichier /etc/mtab

1.4.4. Démonter un système de fichiers

Dès qu'un système de fichiers n'est plus utilisé, il est possible de le détacher (le démonter, en anglais unmount). Pour cela, vous disposez de la commande *umount* qui prend les mêmes paramètres que ceux utilisé pour réaliser le montage.

Par exemple

```
# umount  /media/cdrom
# umount  –t  iso9660  /dev/cdrom  /media/cdrom
```

Pour démonter un système de fichiers, l'administrateur doit s'assurer qu'aucun fichier de ce système n'est en cours d'utilisation (ni par un utilisateur, ni par un processus). Si des utilisateurs travaillent encore avec le système de fichiers, un message d'erreur est retourné indiquant que le système de fichiers en question est encore actif (ou occupé, en anglais busy). Dans ce cas, deux réactions sont possibles : soit attendre patiemment que les utilisateurs quittent d'eux-mêmes le système de fichiers, soit les y inviter expressément.

1.4.5. Commandes de gestion d'un système de fichiers

fsck

La commande fsck permet de contrôler un FS et est capable de réparer certaines erreurs si celui-ci est corrompu. Les systèmes de fichiers contrôlés doivent être démontés car fsck ne supporte pas les accès (écriture ou lecture) sur le disque pendant leur vérification.

Par exemple, si le super bloc est corrompu, on utilise la commande

```
# fsck  -b  valeur
```

où *valeur* désigne le numéro du bloc qui contient la copie du super bloc à utiliser.

Le répertoire lost + found
C'est dans ce répertoire que sont placées les données récupérées par fsck. Par exemple, lorsqu'elle rencontre un fichier perdu (qui possède un inode mais plus de lien), elle crée un lien pour ce fichier dans le répertoire lost + found.

debugfs

La commande debugfs permet de déboguer un FS en autorisant l'administrateur à lire ou modifier n'importe quel bloc du système de fichiers. Son utilisation nécessite une parfaite connaissance du FS, dès lors son utilisation sort du cadre de ce cours.

lsof

La commande lsof permet de lister les fichiers ouverts et de connaître les applications qui y accèdent. Cette commande peut s'avérer utile pour démonter un FS en cours d'utilisation car elle permet de savoir quelle application il faut arrêter pour démonter le FS.

Par exemple, les commandes suivantes permettent de lister les applications accédant à /dev/hda2 et des les tuer (arrêter).

```
# lsof  /dev/hda2
# kill  $(lsof –t /dev/hda2)
```

tune2fs

La commande tune2fs permet de modifier les paramètres d'un système de fichiers de type ext2. Par exemple, il est possible de définir des vérifications automatiques du FS :

- Soit en indiquant le nombre de montage maximum autorisé du FS entre deux vérifications (options –c) ;
- Soit en indiquant le nombre de jours maximum autorisés entre deux vérifications (options –i).

dumpe2fs

La commande dumpe2fs affiche les informations sur le superbloc et les groupes de blocs. Il est ainsi possible de savoir où sont placé les copies du superbloc, ce qui permet de savoir quelle valeur indiquer dans la commande « **fsck -b valeur »**.

badbloks

La commande bad bloks recherche les blocs physiques endommagés d'une partition.

1.5. Types de systèmes de fichiers

1.5.1. Système de fichiers journalisé

Caractéristiques des systèmes de fichiers traditionnels

Pour accélérer les opérations d'entrée/sortie dans les systèmes de fichiers, les données manipulées par les utilisateurs sont placées temporairement dans un cache au lieu d'être directement écrites sur le disque. Or, en cas d'arrêt brusque du système (coupure de courant), les données présentes dans le cache qui n'ont pas été écrites sur le disque sont perdues. Cela engendre une vérification du système de fichiers lors du redémarrage afin de détecter les éventuelles incohérences du système et si possible les corriger.

Ainsi, toute la partition doit être contrôlée au niveau des inodes, des blocs de données et du contenu des répertoires. Dès lors, restaurer l'intégrité d'un disque de grande taille (Téraoctet) risque de prendre beaucoup de temps, rendant le système indisponible.

Amélioration du cache

Sur les systèmes de fichiers dits « journalisés », les données sont toujours mises en cache mais la liste des données mises en cache est enregistrée dans un fichier journal sur le disque. Après une coupure de courant, la vérification sera plus rapide car elle ne portera que sur les données listées dans le journal. Les données qui n'étaient pas en cours d'utilisation, donc qui n'étaient pas dans le cache, n'ont pas de raison d'être contrôlée.

Travailler avec des systèmes de fichiers journalisés induit une diminution des performances, mais celle-ci est compensée par leurs avantages.

1.5.2. Exemples de systèmes de fichiers

Il existe de nombreux systèmes de fichiers, chacun ayant des qualités et des défauts. C'est à l'administrateur de choisir et configurer le FS, même si parfois le choix est imposé (c'est le cas pour les CD-ROM et la RAM, par exemple).

Exemples

- FAT (FAT12 ; FAT16,T32) : système de fichiers qui permet le montage de systèmes de fichiers MS-DOS dans l'arborescence Linux.
- NTFS (New Technology FileSystem) : système de fichiers de MS-Windows, plus performant et sécurisé que que FAT.
- exFAT : évolution de la FAT destinée à la remplacer sur les supports amovibles notamment.
- ext2 : le standard des systèmes de fichiers sous Linux.
- ext3 : il s'agit d'un ext2 auquel est ajouté un système de journalisation.
- HFS (Hierarchical File System) : développé par Apple Inc. pour le système d'exploitation Mac OS.
- HFS+ : successeur de HFS.
- HPFS (High Performance File System) : le système de fichiers natif d'OS/2.
- JFS (Journaled File System) : système de fichiers journalisé disponible notamment sur les plates-formes AIX, Linux et OS/2. La version 2 (JFS2) permet notamment une allocation dynamique des inodes.
- ReiserFS : système de fichiers journalisé optimisé pour le traitement de très nombreux fichiers de petites tailles.
- NFS (Network File System) : protocole (on parle aussi de système de fichiers en réseau) développé par Sun Microsystems qui permet de partager des données principalement entre systèmes UNIX (et avec Windows depuis la version 4)
- CIFS (Common Internet File System) : anciennement Server Message Block (SMB), protocole réseau permettant le partager des fichiers (mais aussi des imprimantes, …) entre différents ordinateurs d'un réseau.
- Système de fichiers pour CDROM : ISO9660, Joliet, Rock Ridge, CD-i, UDF
- /proc : système de fichiers particulier qui permet d'accéder à la mémoire du noyau.
- RAMFS : système de fichiers utilisé pour créer des disques en mémoire vive.
- …

Module 6 : Archivages, sauvegardes et compressions

1. Archivage et compression

La compression consiste à réduire la taille d'un ensemble d'informations numériques (fichiers, programmes,…) de manière à leur permettre d'occuper moins de place. Cette technique est largement utilisée afin d'économiser de la place mais aussi pour télécharger plus rapidement de grands fichiers depuis Internet. Sous Linux, il existe de nombreux instruments de compression, dont les plus utilisés sont probablement *gzip* et *bzip2*. Vous pouvez également utiliser *zip* pour éviter les problèmes de compatibilité lors de l'échange de fichiers vers des systèmes non Linux.

Par convention, les fichiers comprimés se voient attribuer une extension en rapport avec le format de compression utilisé (.gz, .bz2, .zip,..). La commande gzip crée un fichier comprimé se terminant automatiquement par .gz ; *gunzip* extrait les fichiers comprimés et efface l'extension .gz.

L'archivage[1] est une technique permettant de placer plusieurs fichiers sous forme compactée ou non « à l'intérieur » d'un seul fichier (appelé fichier archive ou archive). La manipulation et le transfert de ces fichiers est facilitée car il suffit de manipuler le seul fichier archive. Il existe plusieurs formats d'archives, mais les plus connus sont les formats .zip et .rar. Sous Linux, c'est la commande *tar* qui est utilisée pour archiver (et/ou compresser).

Par exemple, pour compresser un fichier, entrez la commande suivante à l'invite du shell :

```
$ gzip  filename.txt
$ ls
filename.gz
```

Le fichier sera compressé et renommé en filename.ext.gz.

Pour décompresser un fichier comprimé avec gzip, tapez :

```
$ gunzip  filename.txt.gz
```

Le filename.txt.gz est effacé et remplacé par le fichier décompressé filename.txt.ext.

Pour compresser un fichier à l'aide de zip :

```
$ zip  -r  filename.zip  files
```

Dans cet exemple, *filename* représente le fichier que vous créez, et *files* représente les fichiers que vous voulez placer dans le nouveau fichier.

Pour extraire le contenu d'un fichier zip :

```
$ unzip filename.zip
```

Vous pouvez compresser plusieurs fichiers en même temps avec zip ou gzip. Enumérez les fichiers en les séparant par un espace.

```
$ gzip  filename.gz  file1  file2  file3  /user/work/school
```

La commande ci-dessus compresse les fichiers file1, file2, file3, et le contenu du répertoire /user/work/school pour les placer dans filename.gz.

[1] Le terme archivage est aussi utilisé dans le « sens » de sauvegarde. Cf point 2

2. Sauvegardes

La sauvegarde des données est une des tâches les plus importantes dévolue à l'administrateur système. Celui-ci doit mettre en place une stratégie de sauvegarde adaptée et vérifier régulièrement que les fichiers sont bien sauvegardés et que les médias sont encore valables. Les sauvegardes vont de la simple copie de fichiers jusqu'au clonage[1] complet d'un système.

Linux propose en standard plusieurs commandes de gestion de sauvegarde :

- La commande **tar** (Tape ARchive) est utilisée pour sauvegarder des fichiers ou des arborescences (d'un utilisateur ou d'une application). A l'origine, ces sauvegardes s'effectuaient sur des bandes magnétiques (Tape) ;
- La commande **cpio** (copy in/out) est utilisée pour effectuer des transferts de données mais peut servir de programme de sauvegarde ;
- La commande **dd** (disk dump) réalise des copies physiques, elle peut être utilisée pour sauvegarder des disques et des systèmes de fichiers ;
- Les commandes **dump** et **restore** sont utilisées pour sauvegarder des systèmes de fichiers non montés.

Cependant, ces programmes ne constituent pas des solutions complètes de sauvegarde. Pour cela, il faudra se tourner vers des logiciels spécialisés ou développer ses propres outils sur base de ces commandes.

Il y a généralement deux types de données à « sauvegarder » : des données qui seront très rarement consultées et d'autres auxquelles il faut pouvoir accéder rapidement. Comme l'explique Kire Terzievski (Technology & Business magazine), ZDNet Australie, c'est là que se fait la distinction entre sauvegarde et archivage. La sauvegarde consiste simplement à créer une copie strictement identique des données et la stocker sur un support. L'archivage, par contre, doit répondre à un besoin d'accès permanent aux données.

Ainsi, l'archivage implique des moyens permettant de conserver les documents d'une manière fiable et de permettre leur restitution fidèle, même après une longue période (plusieurs années, voire plusieurs dizaines d'années). On parle également d'archivage légal (ou probant) qui implique que des précautions supplémentaires permettent à coup sûr d'authentifier le document et son auteur afin qu'il possède une valeur de preuve.

2.1. Les scripts

Les scripts permettent à l'administrateur de réaliser des sauvegardes et des restaurations adaptés à ses applications. L'exécution de ces scripts peut être automatisée de manière à réaliser les sauvegardes à heure fixe, lorsque le système ou le réseau est peu chargé ou encore lorsque les systèmes de fichiers sont démontés.

2.2. Les outils de sauvegarde du commerce

Lorsque les sauvegardes sont plus complexes à réaliser, notamment si le volume de données est très important, si elles sont distribuées sur un réseau composé de systèmes hétérogènes (Linux, Windows, HP-UX, SOLARlS, ...), l'administrateur devra probablement utiliser un logiciel de sauvegarde du commerce.

[1] C'est le terme généralement employé pour désigner une duplication complète. Le terme ghoster en référence au logiciel Ghost est aussi largement utilisé.

Exemples d'outils fonctionnant sous Linux : ARCserve (CA), Networker (Legato), Time Navigator (Quadratec), ADSM (IBM). On citera également le logiciel Arkeia[1] qui est disponible gratuitement dans une version limitée à un serveur et deux clients.

Au niveau des logiciels libres, on citera Mondo Rescue et Amanda. Le logiciel Partimage est quand à lui un équivalent à Ghost.

2.3. Plan de sauvegarde

La sauvegarde nécessite de l'administrateur du système beaucoup de discipline et une grande rigueur. Il ne suffit pas de prendre le premier utilitaire, ni de réaliser une simple copie de fichiers sur un autre support. Sans être exhaustive, la liste de questions ci-dessous donne une idée des nombreuses questions à se poser pour construire un plan de sauvegarde.

- Que faut-il sauvegarder ?
- Avec quelle fréquence ?
- Combien de temps conservera-t-on les sauvegardes, en combien d'exemplaires ?
- A quel endroit seront stockées les sauvegardes et l'historique des sauvegardes ?
- Quels sont les besoins, en capacité, du support de sauvegarde ?
- Combien de temps, au plus, doit durer la sauvegarde ?
- Combien de temps prévoit-on pour restaurer un fichier, un système de fichiers, est-ce raisonnable ?
- La sauvegarde doit-elle être automatique ou manuelle ?
- Quelle est la méthode de sauvegarde la plus appropriée ?
- Quel est le support le plus approprié ?
  * les performances,
  * la fiabilité ,
  * la pérénité,
  * la portabilité,
  * le prix des supports,
  * le prix des unités de lecture/écriture,
  * la compatibilité,
  * la robustesse,
  * l'encombrement (comprend le lieu de stockage).

Remarques

Une sauvegarde n'a que peu de valeur si elle n'a jamais été restaurée. Lorsque c'est possible, veillez donc toujours à réaliser des « tests de restauration ». La consultation d'une archive, de type tar notamment, n'est pas une action inutile, elle répond à deux besoins :

1. Vérifier que l'archive est complète et que son contenu correspond à ce qui est attendu ;
2. Vérifier, lors de la restauration, que les noms de fichiers s'intégreront bien dans l'arborescence où l'on se trouve.

Il faut avoir en mémoire qu'un fichier stocké sous un certain format sur un certain système n'est pas irrémédiablement lié à ce système. Malheureusement, une mauvaise habitude des systèmes propriétaires est de ne pas prendre en compte les formats utilisés par leurs concurrents. C'est un tort qui a fait le succès des systèmes ouverts.

[1] cf http://www.arkeia.com

2.4. La commande tar

La commande tar permet de placer plusieurs fichiers, le contenu d'un ou plusieurs répertoires ou encore d'une arborescence de fichiers dans un seul fichier. Il est également possible d'exclure certains répertoires ou fichier de l'arborescence que l'on sauvegarde ou encore compresser le fichier archive. Cette commande est très utilisée pour réaliser des archives compressées, c'est notamment le cas pour les fichiers que l'on télécharge à partir de serveur de fichiers sur Internet (format .tar.gz).

La syntaxe de la commande est : $ tar [options] cible source

Où *source* désigne un ensemble de fichiers ou toute une arborescence.
cible désigne le nom que portera le fichier archive.

ATTENTION

- L'ordre des arguments est cible puis source. Intervertir ces deux arguments peut provoquer l'écrasement des données sources ;
- Il est recommandé d'indiquer un chemin absolu pour les sources. Celui-ci sera conservé dans l'archive et permettra ensuite un désarchivage correctement positionné (sinon il y a installation conformément au chemin relatif conservé, ce qui nécessiterait un exact positionnement dans le système de fichiers) ;
- Lors de la création d'une archive, il faut veiller à choisir un nom significatif (avec une extension adéquate) afin de la retrouver facilement.

La syntaxe de la commande tar (c'est aussi le cas pour d'autres commandes) accepte plusieurs formes de paramètres, de façon à conserver une compatibilité avec les autres systèmes de type UNIX. Ainsi, la création d'une sauvegarde peut-être demandée par :

```
$ tar  c ...
$ tar  -c
$ tar  –create
```

Exemples d'utilisations courantes de la commande tar

Pour sauvegarder un fichier

```
$ tar  cvzf  fichier.tar.gz  fichiers  /repertoire
```

Cette commande effectue une sauvegarde de tous les fichiers du répertoire dans le fichier "fichier.tar.gz" en compressant les données.

Détail : l'option c crée une nouvelle sauvegarde
l'option v active mode bavard
l'option z utilise gzip compresser l'archive
l'option f indique que la chaîne suivante de la ligne de commande est le nom du fichier à créer ou le périphérique à utiliser.

Pour extraire le contenu d'un fichier tar

```
$ tar  -xvf  nom_du_fichier.tar
```

Cette commande n'élimine pas le fichier .tar, mais elle place des copies du contenu du fichier .tar dans le répertoire courant. Si les fichiers à restituer ne sont pas précisés, tar restitue la totalité de l'archive.

Pour afficher la liste du contenu d'un fichier tar

```
$ tar  -tvf  nom_du_fichier.tar
```

On remarque dans l'archive, que le premier caractère "." ou "/" indiquant les chemins d'accès aux fichiers à disparu. Il convient donc de se déplacer dans le répertoire qui sert de racine pour la restitution avant d'exécuter la commande tar.

Pour décompresser une archive

Pour décompresser une archive tar comprimée avec gzip, il suffit d'utiliser la commande gunzip.

```
$ gunzip  nom_du_fichier.tar.gz
$ ls
nom_du_fichier.tar
```

3. Stockage de données informatiques

Parmi les différentes technologies de stockage, trois grandes familles se disputent le marché du stockage informatique :

- le stockage sur bandes magnétiques,
- le stockage sur disque,
- le stockage optique.

Chacune ayant des avantages et inconvénients, l'administrateur aura sans doute intérêt à utiliser une combinaison de celles-ci pour mettre en place une solution de stockage efficace.

3.1. Les types de supports de stockage

3.1.1. Bande magnétique

Comme le fait remarquer Lionel Lumbroso (2002), le stockage sur bandes magnétiques a été et reste un support de choix pour la sauvegarde et l'archivage des données principalement car elles sont peu chères, facilement transportable et offrent de grande capacités de stockage, avec la possibilité de mettre en place des baies pouvant contenir plusieurs dizaines de bandes magnétiques.

Le stockage sur bande est fiable, mais comme le souligne Paul Dujancourt (2006), « *les utilisateurs ne vérifient pas si leurs sauvegardes sur bandes sont valides. Il y a un réel problème d'éducation aux techniques de sauvegarde, aussi bien chez les PME que chez les grands comptes* »[1]

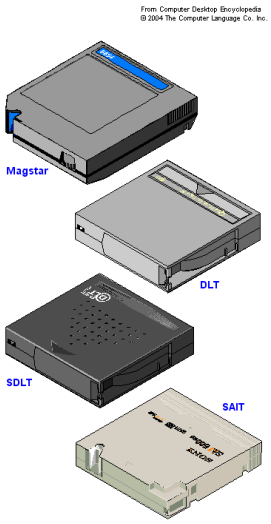
Principales technologies de stockage sur bandes

DLT (Digital Linear Tape) et SDLT (Super Digital Linear Tape)

A titre d'exemple, le lecteur DLT-S4 accepte des cartouches d'une capacité de 800Go (1,6To avec une compression 2 :1), et offre des taux de transfert de 120Mo/s (toujours avec une compression 2 :1).

[1] Propos repris d'un article de Eddye Dibar, 2006, « Gare au stockage sur bande ! », reseaux-telecoms.net, [en ligne], http://www.reseaux-telecoms.net/actualites/lire-gare-au-stockage-sur-bande-14183.html

Figure 1 : Exemple de cartouche DLT et SDLT



Source : http://img.tfd.com/cde/SINGHUB.GIF

Figure 2 : Exemple de lecteur

Source : http://www.sqsdatastorage.co.uk/khxc/media/gbu0/cat/SDLT320_BlkINT_QUA_Web.jpg

LTO (Linear Tape-Open)

Un des problèmes du stockage sur bande a été le manque de standards, chaque fournisseur développant sa propre technologie incompatible avec les autres. A ce titre, l'avantage de la technologie LTO est d'être open-source.

Différents formats de cartouches ont été développés, allant de LTO-1 ayant une capacité de 100 Go et un débit de 20 Mo/s à LTO-4 proposant une capacité de 800 Go et un débit de 120 Mo/s. Le format LTO-5 propose une capacité de 1,5 To pour un taux de transfert de 140Mo/s (en natif).

Figure 3 : Exemple « d'autoloader » LTO (10 cartouches)

Source : http://www.environmentalcontrolinc.com/BowRevisedFolder/Assets/DLTa-II.jpg

Figure 4 :Exemple « d'autoloader » LTO

Source : http://content.etilize.com/Large/1010663900.jpg

Figure 5 : Exemple de cartouches LTO

Source : http://www.materiel.net/live/2702.jpg

Figure 6 : Exemple « d'autoloader » LTO

Source : http://www.ultriumlto-tape.com/assets/images/LTO3tt_left_web.jpg

Le site suivant propose un générateur d'étiquettes à codes barre pour bibliothèques de sauvegarde LTO Ultrium : http://tapelabels.librelogiciel.com/

Autres technologies sur bande

SLR (Scalable Linear Recording), AIT (Advanced Intelligent Tape), DAT (Digital Audio Tape).

3.1.2. Disque dur

Yves DROTHIER (2005), fourni un ensemble de caractéristiques, reprises ci-dessous, concernant les disques en tant que support de stockage.

- Le disque supporte les systèmes de protections RAID capables d'assurer une restauration rapide voire quasi-automatique d'un disque en panne.
- Les disques autorisent une lecture partagée des données, plus rapide que les lectures séquentielles imposées par les librairies de stockage sur bandes. Les restaurations sont donc plus rapides.
- Les disques offrent des temps d'accès[1], de lecture et d'écriture très performants. Les temps d'accès y sont de quelques millisecondes contre quelques secondes (voire dizaine de secondes) pour accéder aux informations placées sur une bande magnétique. Avec de tels temps d'accès, les disques sont bien adaptés à la sauvegarde incrémentale[2] ce qui permet d'optimiser la capacité de stockage en minimisant le volume de données archivées. Pour caractériser les disques, on utilise plutôt le temps de transfert qui correspond au temps mis par les données pour être transférées entre le disque dur et l'ordinateur par le biais de son interface.

[1] Le temps d'accès correspond au temps moyen nécessaire pour placer la tête de lecture sur un secteur quelconque du disque.
[2] La sauvegarde incrémentale « éparpille » les données sur le support, ce qui multiplie les temps d'accès

Figure 7 : Exemple de caractéristiques d'un disque dur :

WD VelociRaptor 3,5 pouces

L'évolution de la technologie SATA 10 000 tr/mn.

N° de modèle	Interface	Vitesse de transfert	tr/min	Capacité	Moy. Recherche	Tampon
WD3000GLFS	sata	3 Gbit/s	10 000	300 Go	4,2 ms	16 Mo

Source : http://www.wdc.com/fr/products/productcatalog.asp#jump1

Au niveau des inconvénients des disques, on citera :

- leur prix,
- les disques se transportent moins aisément, le stockage hors site est difficile,
- les utilisateurs n'ont pas la certitude que l'archivage de leurs fichiers a abouti,
- moins bonne la durée de vie (10 ans en moyenne) contre 100 ans pour les médias optiques ou 30 ans pour les bandes magnétiques. Attention que l'on ne parle ici que de la durée de vie, au niveau de la fiabilité, les médias optiques reste moins performant que les deux autres supports.

Solid State Drive (SSD)

Ce type de lecteur, encore cher actuellement, est pressenti pour remplacer les disques dur. Basé sur de la mémoire flash[1] ou DRAM, il ne contient pas de pièces mobiles ce qui le rend plus fiable et bien plus rapides en termes de temps d'accès. Les vitesses des SSD SLC[2] dépassent les 130 Mo/s en lecture et arrivent presque à 100 Mo/s en écriture

Figure 8 : Exemple de disque SSD

Source : http://www.presence-pc.com/image/Memoright-MR255-32Go-pers,0101-111200-0-2-3-1-jpg-.html

[1] La mémoire flash présente aussi l'avantage de consommer très peu d'énergie, cependant ce n'est apparemment pas le cas des SSD. Cf le test de Patrick Schmid et Achim Roos (2008), « presence-pc.com », *SSD : la grande duperie*, [en ligne] http://www.presence-pc.com/tests/SSD-autonomie-22793/18/

[2] SLC : Single Level Cell

3.1.3. Supports optiques

Ces supports (dont il existe de nombreuses déclinaisons CD-R, CD-RW, DVD- RAM, DVD-R, DVD-RW, DVD+R, blue-ray,…) sont principalement constitués d'un disque de matière synthétique, recouvert sur une face d'une ou plusieurs couches sensibles. C'est sur celles-ci que les données sont gravées et lues par un procédé optique (laser). Il existe également des disques magnéto-optique (MO) qui emploient une combinaison des technologies optiques et magnétiques.

Sur ces supports, l'archivage des données est relativement lent, par contre la restauration de fichiers est généralement rapide. Les médias optiques peuvent être facilement stockés hors site et la réalisation de copies est simple et rapide. La capacité des disques se situe entre 650Mo (CD-R) et 50Go (blue-ray) selon le type de disque.

La durée de vie du média est très grande. En tout cas les prévisions, car aucun disque vieux de 100 ans n'a pu être testé…

Comme pour les bandes, des « bibliothèques » (juke-box) permettent d'automatiser le chargement des disques.

Figure 9 : Exemple de juke-box

Source : http://images.ciao.com/ifr/images/products/normal/279/product-711279.jpg

Interface: SCSI
Stockage amovible : disque magnéto-optique
Capacité de support amovible: 32
Capacité totale de stockage: 291.2 Go, 291 Go
Temps d'échange du support (moyenne): 6.5 sec
Temps moyen de positionnement/d'accès: 25 ms

Figure 10 : Exemple de juke-box

Source : http://images.ciao.com/ifr/images/products/normal/508/product-413508.jpg

Interface: SCSI
Stockage optique : CD-ROM
Vitesse de lecture: 24x
Capacité du chargeur: 6

3.1.4. SAN et NAS

Voir les compléments fournis au cours

3.1.5. Autres supports

Disquette

Ce sont par exemple les disquettes. Si, si, elles existent encore ;-). On ne les utilise pas comme support de stockage car elles ont des capacités trop limitées et sont trop peu fiable. Par contre, elles peuvent notamment rendre service pour charger les drivers de disques SCSI ou le setup de certains serveur.

Caractéristiques : Lent, faible durée de vie, consommable bon marché, capacité faible, rechargeable, très portable, peu fiable.

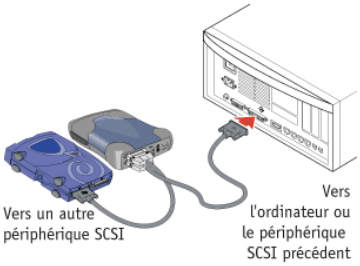
Zip, JAZ et REV

Le ZIP et le JAZ sont des supports amovibles composés d'un disque souple placé dans un boîtier rigide. Ils ressemblent un peu à de grosses disquettes (750Mo pour le ZIP, 2Go pour le JAZ) avec des performances et une fiabilité légèrement supérieures. De nouveau, ces technologies ne sont pas indiquées pour le stockage de données à moyen et long terme, mais plutôt pour le déplacement géographique de celles- ci. Le format REV, plus récent, suit la même lignée.

Figure 11 : Exemple de cartouche JAZ

Source : http://img.zdnet.com/techDirectory/JAZ.GIF

Figure 12 : Exemple de lecteur JAZ



Source : http://www.iomega.com/europe/support/francais/manuals/jaz2e/images/man_jaz2e_scsi_id_mac.gif

Electrique amovible

La mémoire flash est une mémoire réinscriptible et non volatile (les données sont conservées en mémoire lorsque l'alimentation électrique est coupée). Ce type de mémoire présente une capacité limitée (de l'ordre de 20Go au maximum en 2008), une vitesse d'écriture entre10 et 50Mo/s, une faible consommation et une bonne résistance aux chocs car il n'y a pas d'éléments mécaniques (le transport et le stockage en est facilité).

Ce type de mémoire est généralement vendu dans un format allant du timbre poste (voire moins pour les carte au format micro) à la carte de crédit et est largement utilisé comme support de stockage pour appareils « mobiles » tels que les assistants personnels (PDA), les GSM, les appareils photos numériques, etc. Ces cartes mémoires couramment appelées Compact Flash, Smart Media, Secure Digital ou encore Memory Stick peuvent être insérée directement dans les appareils si ceux-ci sont pourvus du lecteur approprié ou utiliser un adaptateur USB.

Figure 13 : exemple de carte mémory stick

Source : http://www.materiel.net/live/30793.jpg

Figure 14 : exemple de lecteur de carte

Source : http://img.alibaba.com/photo/10931174/Storium_usb_Compact_Flash_Card_Reader.jpg

Figure 15 : exemple de carte microSD

Source : http://hardwarezone.co.th/img/data/nnews/2006/5484/Image/MicroSD_w_adapter_1GB.jpg

Figure 16 : exemple de lecteur de carte

Source : http://www.ubergizmo.com/photos/2007/8/digix-super-multi-reader.jpg

Module 7 : Installer des programmes

1. Linux en tant que plate-forme d'applications

Bien que l'on trouve presque tous les types d'applications pour Linux (traitements de texte, tableurs, programmes graphiques, multimédia, etc) de nombreuses applications courantes ne tournent pas correctement sous Linux[1] ou ne s'exécutent pas du tout (comme par exemple la dernière version de Microsoft Office). Dans ces situations, il est possible de faire tourner un autre système d'exploitation au sein de Linux afin de pouvoir accéder à ces applications (bien sur, il faudra vérifier licences correspondantes et s'en acquitter).

Il est également possible de convertir les fichiers de ces applications pour qu'ils puissent être utilisés par un programme similaire dans Linux, cependant la conversion n'est pas toujours parfaite (par exemple au niveau de la mise en page des documents textes). Cela dit, la conversion de fichiers entre deux versions d'un même logicielle est-elle parfaite ? Il faut aussi préciser que de nombreuses erreurs de conversion sont dues à une mauvaise utilisation du logiciel et pas à une incapacité du programme.

Il existe beaucoup plus d'applications commerciales éprouvées pour les systèmes d'exploitation Windows qu'il n'en existe pour Linux, surtout au niveau des stations de travail. Le marché des utilisateurs des systèmes Windows est énorme, de nombreuses sociétés de logiciel ne développent des produits que pour ce marché.

Cependant, il y a certains avantages à utiliser des systèmes libres, en particulier dans un contexte scolaire.

- Il existe une énorme communauté de développement qui travaille sur des applications sous licence GPL pour répondre aux besoins et développer des correctifs. Ceux-ci sont souvent corrigés rapidement.
- Ces logiciels sont libres d'utilisation, de copie, de modification, de distribution et d'accès au code source. De plus, la plupart sont gratuits, ce qui les rend accessibles à tous les utilisateurs, quelque soient leurs moyens. Attention cependant que, dans une utilisation professionnelle, le coût des logiciels libres ne peut pas être considéré comme nul. En effet, si l'achat des logiciels et systèmes d'exploitation peut s'avérer gratuit, d'autres coûts seront à prendre en compte :
    - La migration des applications va nécessiter beaucoup de travail et de temps (donc de l'argent). Les services réseaux seront peut-être inaccessibles un certain laps de temps (downtime) ce qui peut aussi avoir un coût important ;
    - La formation des personnes devant apprendre à utiliser de nouveaux logiciels ;
    - Le contrat de maintenance.

Bien que plus récents sur le marché, les logiciels libres sont de plus en plus adoptés par les entreprises et les administrations. Par exemple, le lien suivant donne une liste non exhaustive de sociétés utilisant la suite Openoffice : http://fr.openoffice.org/about-information.html

[1] Il en va de même pour d'autres systèmes d'exploitation, sur lesquels les programmes fonctionnant sous Linux ne « tournent » pas.

2. Où trouver des applications fonctionnant sous Linux ?

La plupart des applications Linux sont disponibles et téléchargeables sur Internet, elles peuvent également y être commandées. Par contre, il y a peu de chances d'en trouver dans un magasin d'informatique.

- Voici quelques sites Internet où vous trouverez des applications Linux. Tous les logiciels listés sur ces sites ne sont pas sous licence GPL et/ou gratuits, à vous de vérifier. Certains programmes listés fonctionnent aussi sous Windows ou MAC.
- http://linux.scola.ac-paris.fr/default.asp?id=110&mnu=110
- http://www.framasoft.net/
- http://www.linuxapps.com/
- http://www.rpmfind.net

3. Formats des sources

La plupart des programmes sont fournis d'une des manières suivantes :

1. Soit sous forme d'un paquetage, c'est-à-dire un ensemble de fichiers (généralement précompilé et constituant une application) regroupés à l'intérieur d'un seul fichier. Pour les distributions RedHat, Fedora ou encore Mandriva, on parle de paquetage RPM (Redhat Package Manager). Il s'agit d'un fichier portant l'extension .rpm permettant une installation automatique à l'aide de la commande rpm ou d'un utilitaire graphique.
2. Soit sous forme d'une archive. Il s'agit de fichiers, généralement compressés, et archivés à l'intérieur d'un seul fichier. La plupart du temps ce fichier comporte l'une des extensions .tar.gz, .tgz, .gz ou .tar.bz2 (on parle aussi de tarball). L'archive contient généralement les sources du logiciels et il sera généralement nécessaire de compiler le programme (après décompression de l'archive) afin de rendre son exécution possible. L'utilisation d'un compilateur et des libraires ad hoc est donc indispensable. Il est possible de se trouver face aux mêmes « problèmes » de dépendances que lors de l'installation manuelle de paquetages durant l'installation de Linux.

3.1. Les paquetages RPM

Les fichiers portant l'extension .rpm contiennent une version pré-compilée d'un programme. L'outil de gestion de paquetages appelé RedHat Package Manager (commande **rpm**) permet d'installer ou désinstaller de tels programmes. Par défaut, les fichiers s'installent automatiquement dans les répertoires prédéfinis, à moins d'utiliser les options de la commande pour redéfinir le répertoire d'installation.

3.1.1. Structure du nom d'un paquetage RPM

Le nom d'un paquetage fourni généralement une série d'informations sur le programme qu'il contient. Prenons comme exemple le paquetage nommé "packagename-1.2.3.i386.rpm". Ce nom de fichier comporte les informations suivantes (cf point 5 pour plus de détails):

1) Le nom du paquetage : packagename
2) La version de l'application : 1.2.3
3) Le i386 indique que le package contient des binaires prévus pour être exécutés sur des ordinateurs d'architecture de type Intel 386.
4) L'extension .rpm indique qu'il s'agit d'un paquetage de type RPM.

3.1.2. Installer/désinstaller un rpm à l'aide de la ligne de commande

Voici quelques-unes des commandes rpm les plus utilisées :

Installer un package

Pour installer le paquetage "packagename-1.2.3.i386.rpm" il suffira d'entrer

```
[root]# rpm -ivh packagename-1.2.3.i386.rpm
```

En même temps que l'option -i, vous pouvez utiliser les options suivantes pour être informé durant l'installation:

- L'option -v affiche des informations de débogage au cours de l'installation. Cette option est très pratique pour voir ce qui se passe au cours de la procédure d'installation (les informations peuvent être longues, mais vous pouvez les rediriger sur une des commandes more ou less). Au plus il y a de « v », au plus il y aura d'informations affichées (v, vv ou vvv. Il y en a rarement plus de 3).
- L'option -h fournit un affichage plus facilement lisible pour un « Humain ».

Avant d'installer un paquetage, le programme rpm contrôle les éventuelles dépendances ainsi que l'existence de fichiers plus récents. Ainsi, il est possible qu'une erreur de dépendances de s'affiche indiquant le paquetage manquant. Si les dépendances ne sont pas résolues automatiquement, votre package risque de ne pas fonctionner. Pour les installez manuellement, consultez le site http://rpmfind.com ou le CD-ROM de la distribution afin de vous les procurer.

Supprimer un paquetage

L'option -e permet de désinstaller un paquetage. Avant de supprimer un paquetage, rpm contrôle les dépendances, vous n'avez donc pas à vous en occuper.

```
[root]# rpm -e packagename
```

Attention, c'est bien le nom du package sans aucune précision : ni numéro de version, ni extension qu'il faut indiquer

De même que pour l'installation, si vous décidez de supprimer un paquetage, il est préférable de faire appel à l'option -vv. Ceci permet de voir les fichiers qui sont supprimés et éventuellement d'en garder une trace. Il est aussi possible d'exécutez la désinstallation en mode test (--test) avant de le réaliser réellement. Le mode test vous montre tout ce qui se passerait au cours de la désinstallation sans réellement l'exécuter (de nouveau, ajoutez -vv pour voir les détails).

```
[root]# rpm -evv --test packagename
```

Mettre à jour des paquetages

RPM est capable de mettre à jour automatiquement des paquetages, sans désinstaller une version précédente avant d'installer la nouvelle. Pour mettre à jour un paquetage (ce qui nécessite évidemment qu'une version précédente du paquetage a déjà été installée) il faut utiliser l'option –U.

```
[root]# rpm -Uvh packagename-2.2.4.i386.rpm
```

Interroger des paquetages

RPM est capable d'interroger un paquetage afin d'en connaître le contenu, ou pour savoir à quel paquetage appartient un fichier. Il existe plusieurs commandes visant à interroger un ou plusieurs packages selon divers critères. On notera :

```
[root]# rpm -qa
```

Pour connaître des détails sur un paquetage particulier

```
[root]# rpm -q packagename.rpm
```

Remarque

Il existe d'autres utilitaires très pratiques (en mode texte ou graphique) pour gérer les paquetages. Yum sous Redhat, Yast sous Suse, urpmi sous Mandriva, …

3.2. Installer à partir d'archives

La solution la plus simple pour installer un programme consiste à utiliser le paquetage précompilé (binaire ou paquetage) pour la distribution. Cependant, celle-ci n'est pas toujours disponible et même si elle l'est, le paquetage ne sera sans doute pas parfaitement adapté à vos besoins ou à votre matériel. Ainsi, si vous voulez adapter un paquetage, il est toujours possible de l'installer à partir du code source. Le nommage des fichiers sources suit le même principe que celui des paquetages. Voici le détail des différentes parties généralement rencontrées dans un nom d'archive.

<Nom>-<Version>.<src ou bin>.<(Type de binaire)>.<Type d'archive>.<type de compression>

On distingue :

1) Le nom qui identifie l'application.
2) Le numéro de version : plus le numéro est grand, plus la version est récente.
3) Un mot clé (tel que bin ou src) indiquant si le fichier contient un binaire (prêt à être exécuté) ou le code source.
4) Un champ comprenant des informations complémentaires pour décrire le contenu de l'archive.
5) Le type d'archive : tar est le plus souvent utilisé (extension .tar).
6) Le type de compression : gzip est le plus souvent utilisé (extension .gz). On rencontre de plus en plus l'extension bz2, format de compression plus récent et plus performant.

Voici quelques exemples

1) packagename-1.2.3.src.tar.gz
2) packagename-1.2.3.bin.SPARC.tar.gz
3) packagename-1.2.3.bin.i586.tar.gz

Dans ces exemples, tous les fichiers qui composent le paquetage ont été archivés en utilisant la commande tar (extension .tar) et comprimés en utilisant la commande gzip (extention .gz). Dans l'exemple un, l'indication *src* indique que l'archive contient les sources du programme. A la ligne deux, .bin.SPARC indique qu'il s'agit d'un fichier binaire destiné à une station SPARC[1]. Dans le dernier exemple, le .bin.i586 indique que le paquetage contient des binaires prévus pour être exécutés sur des ordinateurs d'architecture Intel 586.

[1] SPARC (Scalable Processor ARChitecture) désigne une architecture de microprocesseur construit sur un modèle RISC (Reduced Instruction-Set Computer).

3.2.1. Compiler un programme

Lorsque l'archive fourni le code source d'un programme, il est nécessaire procéder à la compilation, c'est-à-dire la conversion du code source en un programme exécutable. Bien que la compilation se fasse souvent de la même façon pour la majorité des applications, les instructions exactes d'installation varient d'une archive à l'autre. Il est donc nécessaire de lire les fichiers INSTALL, SOURCES, README, ou autre afin de connaître exactement les commandes et les ressources nécessaires à la compilation du programme. Pour installer à partir de codes sources, il faudra notamment disposer d'un compilateur et de librairies (ils sont généralement précisés dans les fichiers INSTALL). Ceux-ci se trouvent dans les paquetages dits de « développement ». Il est tentant d'installer tous les paquetages de développement lors de l'installation du système pour ne pas devoir systématiquement rechercher quels sont les composants qui manquent lors d'une compilation. Cette approche est pratique pour le développement, mais est à proscrire sur les serveurs où il est fortement déconseillé d'installer des programmes qui ne sont pas indispensable car :

- ceux-ci consomment des ressources,
- ils sont des risques potentiels pour la stabilité et la sécurité du système (plus vous installez de programmes, plus vous avez de chance que l'un d'eux présente une faille).

La solution consiste alors à réaliser une compilation croisée : il s'agit de compiler un programme sur une machine différente de la machine cible mais en paramétrant la compilation pour la machine cible.

Méthode générale d'installation

Cette méthode est une méthode générale, en pratique il faut toujours lire les fichiers relatifs au logiciel concerné (fichiers INSTALL, SOURCES, README, …)

Etape 1 : exécuter la commande « ./configure »

Dans un premier temps, il faut exécuter la commande ./configure dans le répertoire de décompression de l'archive. Cette commande est un script permettant de paramétrer (créer) le fichier contenant les informations indispensables à la compilation (fichier Makefile). Elle vérifie également les librairies et outils présents sur la machine. S'il en manque, le programme le signalera et ne générera pas le fichier Makefile (ce dernier est cependant nécessaire pour passer aux étapes suivantes).

```
# ./configure
```

Pour obtenir de l'aide sur les options de la commande configure :

```
# ./configure --help
```

Pour installer l'application dans un répertoire spécifique :

```
# ./configure --prefix="repertoire"
```

Etape 2 : Compiler le programme

Dans un second temps il faut compiler le programme, grâce à la commande *make* (à effectuer dans le répertoire où se situe le fichier makefile).

```
# make
```

Cette commande peut prendre plusieurs minutes selon la puissance de la machine et le logiciel à compiler. En cas de problème, la compilation s'arrêtera en affichant un message d'erreur. Si tout se passe normalement de nombreuses lignes vont être affichées, puis le programme reviendra à la ligne de commande (prompt).

Etape 3 : Installer le programme

Il faut pour cela exécuter le programme make avec l'option *install*.

```
# make  install
```

Après une compilation (réussie ou non), il est préférable de nettoyer les fichiers ayant été créés automatiquement lors de la compilation. Cela se fait grâce à l'option *clean*

```
# make  clean
```

Une fois ces étapes terminées, le programme devrait être exécutable et se trouver dans le répertoire spécifié lors de l'installation, ou dans /usr/local/nom_du_programme.

Remarque

Il arrive parfois que vos librairies ne se trouvent pas là où le programme « configure » les cherche. Vérifiez que vous avez bien la version citée par le fichier « configure » et regardez où le programme tente de trouver votre librairie. Créez alors des liens symboliques (ln -s) depuis cet endroit.

Etape 4 : désinstaller le programme

Pour désinstaller un logiciel, référez vous aux fichiers INSTALL ou README. S'il n'y a pas d'exécutable prévu pour la désinstallation (tous les programmeurs ne le prévoient pas), vous pouvez essayer la commande make unsinstall.

Module 8 : Gestion des processus

1. Programme et processus

Sous Linux, il faut distinguer le programme du processus au sein duquel il s'exécute, les deux expressions ne sont pas synonymes.

Variable d'environnement

Ces variables stockent certains paramètres définissant le contexte d'exécution des commandes et des programmes. Un exemple de variable d'environnement est « LANG=fr_FR » qui détermine la langue. Dans la pratique, l'administrateur est amené à ajouter, pour le bon déroulement de logiciels applicatifs, la définition de nouvelles variables d'environnement. Pour cela, il dispose de plusieurs fichiers dont

- Le fichier /etc/profile pour définir les variables valables pour tous les utilisateurs ;
- Les fichiers ~/.bash_profile ou ~/.profile pour définir les variables valables uniquement pour les utilisateurs concernés.

Processus et environnement de processus

Un processus désigne un programme chargé en mémoire et en cours d'exécution. Ainsi, à chaque fois que vous exécutez un programme, un processus est créé. En effet, la machine ne peut exécuter les commandes (en langage machine) du programme que lorsque celui-ci est en mémoire. Pour que le programme en mémoire s'exécute correctement, le système d'exploitation doit lui fournir de nombreuses informations (un numéro d'identification, une priorité, …). Un processus se compose donc d'un programme chargé en mémoire et d'un environnement de processus[1]. Un programme ne peut pas être exécuté sans les informations fournies par le système d'exploitation. Dès lors, programme et environnement de processus sont liés par nécessité.
Chaque processus travaille dans un espace d'adressage virtuel propre et distinct de sorte qu'aucun processus ne peut en gêner un autre sciemment ou inconsciemment.

2. Gestion des processus

Dès qu'un programme (une commande) est démarré, le noyau Linux crée un (ou plusieurs) processus qui exécute les instructions de ce programme. Ce processus peut transiter par différents états selon qu'il s'exécute (état « actif »), attend que le noyau lui alloue le processeur (état « prêt ») ou qu'un événement se produise (état « en attente »). Pour chaque processus en cours d'exécution, le système d'exploitation stocke un certain nombre d'informations notamment dans la *table des processus*. La description des éléments de cette table, présentée ci-après, est inspirée de l'ouvrage de Wielsch M, Prahm J., Esser H.-G., *La bible Linux*.

- le numéro qui identifie le processus (Process IDentification ou PID),
- le numéro de processus parent (Parent Process Identification ou PPID),
- le numéro de l'utilisateur qui a demandé l'exécution (User id ou UID),
- le numéro de groupe (Group ID ou GID),
- la durée de traitement utilisée par le processus (temps CPU),

[1] L'environnement de processus est l'ensemble des variables d'un programme en cours d'exécution, ainsi que les ressources qui lui sont attribuées.

- la priorité du processus,
- une référence au répertoire de travail courant,
- une liste des fichiers ouverts.

Numéro de processus (PID)

Linux est un système multitâche, il autorise donc plusieurs programmes à être exécutés simultanément. Tous les processus associés à ces programmes doivent pouvoir être différenciés d'une autre manière que par leur nom, car un même programme peut être démarré plusieurs fois. C'est pourquoi le système leur affecte un numéro spécifique de processus : le PID. A un instant donné, une valeur de PID est unique dans le système et cela garantit qu'un processus est identifié sans ambiguïté par son PID. Un PID est associé à un processus lors de son exécution et « désassocié » lors de son arrêt, de sorte qu'après un redémarrage, le PID d'un processus ne sera plus le même que précédemment. Ce numéro est utilisé par l'administrateur pour transmettre des signaux[1] à un processus, par exemple pour le stopper s'il ne répond plus.

Processus parent

Chaque processus peut lui-même créer des processus[2]. Ces nouveaux processus sont alors appelés processus enfants (*child process* en anglais*).* Processus parent et enfant sont liés[3], il faut donc que les enfants connaissent leur origine. Pour ce faire, le système leur communique le numéro d'identification de leur processus parent (PPID). Tous les processus sont ainsi associés à un numéro de processus parent, puisqu'ils sont tous créés par un autre processus.

Il y a une exception à cette règle : le tout premier processus n'a pas pu être généré par un autre processus. Un pseudo-processus, portant généralement le numéro 0, est créé au moment du démarrage du système afin de générer le premier processus appelé *init* (portant le numéro 1). Ce processus est particulier car il est responsable, directement ou indirectement, de tous les processus en cours dans le système.

Numéro d'utilisateur et de groupe (UID et GID)

De même que pour les utilisateurs, les processus ne doivent pas avoir un accès complet au système. Il est donc nécessaire de gérer des droits d'accès, mais au niveau des processus. Dans cette optique, chaque processus reçoit, à sa « naissance », un numéro d'utilisateur et un numéro de groupe qui vont permettre au système de vérifier si un processus possède les permissions nécessaires pour, par exemple, accéder à certains fichiers. La plupart du temps, les numéros d'utilisateurs et de groupes sont hérités[4] par le processus enfant de son processus parent.

Durée de traitement et priorité

Sur un système, on dispose généralement de plus de processus que de processeurs. L'accès au(x) processeur(s) doit donc être réglementé de manière à ce que celui-ci ne soit pas monopolisé par l'un ou l'autre processus. Il est évidemment inadmissible qu'un processus monopolise le processeur. Aussi, Linux gère le multitâche de manière *préemptive.* Le temps

[1] Cf point 5 : les signaux

[2] C'est d'ailleurs indispensable, car nous ne savons pas créer de processus, nous demandons au système de le faire pour nous.

[3] Par exemple, si l'on exécute la commande « updatedb » dans un shell, le shell est le processus parent et updatedb le processus enfant. Si l'on arrête le shell, ses processus enfants vont se terminer également.

[4] Cf point 4 : la notion d'héritage.

de processeur total disponible est divisé en petites plages (*time slice)* et un processus ne peut utiliser le processeur que le temps défini par le système. La durée de l'accès, ainsi que l'ordre, sont déterminés par le planificateur de processus *(process scheduler)* qui se base sur un système de priorité. Chaque processus est doté d'une priorité (un nombre entre -20 et 20) et celui ayant la plus haute priorité (chiffre le plus bas) sera traité en premier. Pour chaque plage de temps de traitement où le processus n'a pas accès au processeur, sa priorité est augmentée. A l'inverse, lorsqu'il est en cours de traitement, sa priorité baisse. De cette manière, le processus qui a la plus haute priorité à un moment donné ne la garde pas indéfiniment et les processus sont traités à tour de rôle. Lors de son exécution, la priorité de départ d'un processus est fixée par le système, mais les utilisateurs ont la possibilité de la modifier notamment avec la commande *nice*.

Répertoire courant de processus

Certaines commandes doivent connaître le répertoire courant pour s'exécuter correctement. C'est le cas de la commande ls utilisée sans option (elle affiche le contenu du répertoire courant). De même, pour permettre à Linux de créer ou utiliser des chemins relatifs, les processus doivent contenir l'indication du répertoire courant.

Table des fichiers ouverts

Pour permettre aux processus d'accéder en lecture et en écriture à des fichiers, il leur faut les références des fichiers ouverts en mémoire de travail C'est pourquoi une table des fichiers ouverts est créée pour chaque processus.

3. Exécution d'un processus

Sous Linux, il existe quatre manières d'exécuter un processus.

- En avant-plan (foreground) : c'est le mode normal d'exécution d'une commande dans une session de travail. Exécutée dans le shell, l'affichage est « monopolisé » par la commande, le prompt n'est plus disponible pour entrer d'autres commandes.
- A l'arrière-plan ou en tâche de fond (background). Une commande exécutée en arrière plan permet de garder la main pour lancer d'autres commandes depuis le terminal. Un processus d'arrière-plan est insensible au caractère de contrôle qui met habituellement fin au processus d'avant-plan (CTRL+C).
- En mode détaché : un processus détaché n'a plus de terminal de contrôle, ce qui signifie, par exemple, que le processus peut continuer de s'exécuter même si tous les shell (interfaces texte et graphiques) sont arrêtés. En mode détaché, le caractère "?" apparait dans la colonne TTY produite par la commande *ps* qui liste les processus.
- En tant que démons. Les démons (daemon) désignent communément les processus détachés associés aux services du système Linux, souvent en mode client-serveur.

4. Héritage

Quand un processus crée un processus fils, ce dernier hérite d'un ensemble d'éléments de son père parmi lesquels on trouve :

- le répertoire courant,
- les variables d'environnement,
- le « umask » qui définir les droits par défaut des fichiers créés par le processus,
- le « ulimit » qui fixe la taille du plus gros fichier que le processus peut créer.

5. Les signaux

Un signal est symbolisé par un chiffre qui a une signification bien précise pour les processus. Les signaux constituent le seul moyen d'agir sur l'exécution d'un processus détaché. Ces signaux peuvent être envoyés avec la commande *kill* (kill -l pour lister les signaux possibles).

Exemples de signaux

Le signal 15 (SIGTERM) est conventionnellement utilisé pour dire à un démon de se terminer. C'est d'ailleurs le signal que la commande shutdown envoie aux processus actifs.

```
# kill  -15  nomduprocessus
```

Le signal 9 (SIGKILL) tue le processus sans qu'il puisse intercepter le signal et exécuter une quelconque action (comme enregistrer les fichiers ouverts) avant de se terminer. Le signal 9 ne doit, pour cette raison, être utilisé qu'en dernier recours.

```
# kill  -9  nomduprocessus
```

6. Traitement en tâche de fond

Lors de la saisie d'une commande, le shell interprète celle-ci et un nouveau processus est créé[1] pour exécuter le programme. Ce processus ne traite que la commande et fonctionne comme processus enfant du shell. Le shell se met alors en attente de la fin du processus, et n'est plus disponible pour l'utilisateur ; le prompt ne s'affichera de nouveau que lorsque le traitement de cette commande sera terminé. Le shell et le processus sont en quelque sorte synchronisés par l'attente.

Pour lancer plusieurs processus sans attendre la fin du précédent, il faut les envoyer en tâches de fond. Il suffit pour cela d'ajouter à la fin de la commande le caractère & (*ampersand)*. Dans ce cas, le shell et ses processus enfant fonctionnent en parallèle, sans aucune coordination entre eux : le prompt est disponible tout de suite après avoir démarré chaque commande. Cependant, bien que les processus fonctionnant en tâche de fond travaillent indépendamment du shell actif, il y a toujours le lien de parenté « parent-enfant » entre le shell et les processus. Cette parenté fait qu'à la fin du shell, tous les processus en tâche de fond sont automatiquement obligés de s'arrêter. Si vous souhaitez que les processus continuent en tâche de fond après la fin du shell, il faut le signifier en leur envoyant un signal de type nohup.

Conditions pour envoyer un processus en arrière plan

Pour envoyer un programme en tâche de fond il faut respecter certaines contraintes

1. Le processus envoyé en tâche de fond ne doit pas attendre de saisie au clavier pour continuer son exécution. En effet, le système d'exploitation n'a aucun moyen de déterminer à quel processus en tâche de fond sont destinées ces saisies.
2. Le processus envoyé en tâche de fond ne doit pas retourner ses résultats à l'écran (sortie standard). En effet, les sorties de ce processus risquent d'entrer en conflit avec celles du processus en avant plan ou du shell et l'écran deviendrait difficilement lisible voire inutilisable.

[1] Si la commande tapée n'est pas une commande interne du shell.

3. Le processus parent (généralement le shell) des processus envoyés en tâche de fond ne doit pas être arrêté au risque de stopper également tous ses enfants.

Redirection

Ainsi, il est problématique d'envoyer en arrière plan une commande qui utilise, comme sortie, la sortie standard (l'écran). Or, c'est le cas de nombreuses commandes, ce qui réduirait fortement le nombre de commandes qu'il est possible d'utiliser en tâche de fond. Pour éviter cela, il est possible de rediriger la sortie de la commande ailleurs que vers la sortie standard (dans un fichier par exemple). Prenons l'exemple de la commande suivante

```
# ls  -lR  /
```

Celle-ci monopolise l'écran, le rendant inutilisable un certain temps.

En utilisant une redirection au moyen du caractère > , le résultat de la commande est redirigé vers une autre sortie (dans l'exemple ci-dessous, vers un fichier nommé listing.txt)

```
# ls  -lR  /  > listing.txt
```

Cependant, il subsiste un problème. La commande pose toujours problème alors que nous avons redirigé sa sortie standard. L'exemple suivant met en évidence ce problème :

```
# ls  -lR  /bidule  > listing.txt
```

En plus de la sortie standard, il existe une sortie d'erreur qui est utilisée pour afficher les messages d'erreur. Dans notre exemple, un tel message sera affiché si le répertoire /bidule n'existe pas. Ainsi, il ne faut pas oublier de rediriger cette sortie.

Les canaux d'entrée et de sortie sont symbolisés par des chiffres :

- Le canal d'entrée standard (clavier) est représenté par le chiffre 0
- Le canal de sortie standard (écran) est représenté par le chiffre 1
- Le canal d'erreur standard est représenté par le chiffre 2

Nous allons maintenant pouvoir rediriger toutes les sorties :

```
# ls  -lR  /bidule  > listing.txt     2 >  /dev/null
```

Avec l'indication 2 > /dev/null, la redirection du canal d'erreur standard s'effectue vers le « néant ». Ce fichier de périphérique /dev/null est utilisé ici comme poubelle, tout ce qui y est écrit est perdu sans possibilité de le récupérer. Il est aussi connu sous le terme de fichier vide ou fichier nul.

Exemple d'utilisation

La commande suivante, envoyée en arrière plan, recherche tous les fichiers dont le nom commence par *install* et liste les résultats dans le fichier *liste*. Les éventuels messages d'erreur sont envoyés dans le fichier nul.

```
[root]# find /  -name "install*" > liste  2> /dev/null &
```

Avant de revenir au prompt du shell, l'écran affiche un numéro de tâche (ou job) entre crochets ainsi que le numéro (PID) du processus lancé.

```
[root]# find /  -name "install*" > liste  2> /dev/null &
[1] 439
```

Le numéro de tâche qui identifie le processus en arrière-plan peut être utilisé avec certaines commandes servant à la gestion des processus (kill, bg, …).

Liste des processus en tâche de fond

La commande **jobs** sans option liste les processus que l'utilisateur a envoyés s'exécuter à l'arrière-plan en indiquant leur numéro de tâche, leur état et le texte de la commande.

```
# jobs  -l
[1]-  Done                    find / -name "install*" >liste 2>/dev/null
```

L'état des processus qui « tournent » à l'arrière-plan est renseigné par le mot clé qui suit le numéro du job. Les statuts les plus courant sont :

- Done : statut d'un processus lorsqu'il s'est terminé normalement ;
- Running : statut d'un processus en cours de fonctionnement ;
- Stopped : statut d'un processus qui est suspendu temporairement ;
- Terminated : statut d'un processus qui a été obligé d'arrêter son exécution à la suite d'un signal.

Basculer un processus entre l'avant et l'arrière plan

Les commandes **bg** (background) et **fg** (foreground) permettent de faire passer des processus respectivement vers l'arrière-plan et vers l'avant plan. Pour utiliser la commande bg, un processus en cours qui s'exécute au premier plan doit être suspendu (et non pas arrêté). Par défaut, il est possible de suspendre un processus via la combinaison de touches CTRL+Z. La combinaison exacte est indiquée par la commande **stty -a** derrière la mention *susp*. La commande **bg** attend comme paramètre un numéro de tâche précédé d'un signe de pourcentage (%).

```
# bg %1
```

De la même manière, la commande fg attend comme paramètre un numéro de tâche précédé d'un signe de pourcentage.

```
# fg %1
```

Suspendre puis reprendre un processus en arrière-plan

Il est également possible de suspendre ou reprendre l'exécution d'un processus à l'aide de la commande kill et des options STOP ou CONT.

Le processus d'arrière-plan avec le numéro de tâche 1 peut être suspendu de cette manière :

```
# kill  -STOP  %1
```

puis repris avec la commande :

```
# kill -CONT %1
```

Parfois, vous aurez besoin d'attendre les résultats d'un processus lancé en tâche de fond pour continuer votre travail. Vous devrez donc surveiller la liste des processus en cours (commandes ps ou top), pour voir si le processus concerné est encore en activité. Une autre solution fait appel à la commande *wait* qui ne rend la main que lorsque le ou les processus en tâche de fond spécifiés sont terminés.

7. Priorité d'un processus

Chaque processus se voit affecter un numéro qui correspond à une *priorité*. Lorsqu'un programme est démarré, le processus engendré a une priorité maximale. Plus le processus occupe de temps d'exécution au niveau du processeur, plus ce numéro de priorité baisse. A l'inverse, moins il en occupe plus son numéro de priorité augmente.
Les commandes *nice* et *renice* permettent de diminuer ou augmenter la priorité d'un processus, ce qui peut être utile si vous souhaitez qu'un processus soit traité le plus rapidement possible.

Exemple

```
$ nice  -n  20  updatedb
```

La valeur spécifiée indique le facteur de modification de la priorité. Seul l'administrateur système (compte root) peut également augmenter la priorité avec un nombre négatif.

```
# nice  -n  -15  updatedb
```

La commande renice permet de changer le facteur de priorité en cours d'exécution de la commande, en spécifiant le nouveau facteur de priorité et le numéro de processus.

```
# renice 10  733
733: old priority 19, new priority 10
```

Module 9 : Gestion des ressources

1. Introduction

Voir diapositives

2. Ordonnancement de travaux

2.1. La commande crontab

La commande crontab est un planificateur de tâches, elle contient une liste (sous forme d'un fichier texte) de commandes qui doivent être exécutées à intervalle régulier. Elle permet ainsi d'automatiser une série de tâches comme la sauvegarde journalière de documents ou le nettoyage de fichiers journaux. Ces tâches peuvent être de simples commandes, des scripts ou des programmes complexes.

Le service crond

Il est chargé de faire exécuter, par le système, toutes tâches définies et planifiées dans la table crontab. Il génère des comptes rendus après leur exécution.

Par défaut, le démon crond est lancé au démarrage et contrôle toutes les minutes les fichiers présents dans le répertoire /var/spool/cron, /etc/cron.d ainsi que le fichier /etc/crontab, pour voir si des tâches doivent être exécutées. Il n'est donc pas nécessaire de redémarrer le démon si un fichier crontab est modifié. Chaque action de crond ajoute une ligne de message dans le fichier /var/log/cron, il faut donc veiller à ce que la taille de ce fichier ne grandisse pas indéfiniment.

La commande /usr/bin/crontab

C'est cette commande (crontab –e) qui doit être utilisée pour définir des tâches périodiques, et non le fichier de configuration /etc/crontab. Les utilisateurs autorisés ou non à l'utiliser peuvent être listés dans le fichier /etc/cron.allow (utilisateurs autorisés) ou /etc/cron.deny (utilisateurs non autorisés). Toutes les tâches crontab définies par l'utilisateur sont stockées dans le répertoire /var/spool/cron/ avec comme nom, le nom de l'utilisateur. Ce fichier utilise le même format que /etc/crontab.

Format du fichier de configuration : /etc/crontab

De base, le fichier de configuration principal de cron, contient les lignes suivantes :

```
SHELL=/bin/bash
PATH=/sbin:/bin:/usr/sbin:/usr/bin
MAILTO=root
# run-parts
01 * * * * root run-parts /etc/cron.hourly
02 4 * * * root run-parts /etc/cron.daily
22 4 * * 0 root run-parts /etc/cron.weekly
42 4 1 * * root run-parts /etc/cron.monthly
```

Les quatre premières sont des variables servant à configurer l'environnement dans lequel les tâches cron sont exécutées.

- La valeur de la variable SHELL indique au système quel environnement shell il doit utiliser (dans cet exemple, le shell bash) ;
- La variable PATH définit les chemins d'accès via lesquels cron trouve les exécutables des commandes à démarrer ;
- La variable MAILTO indique à quel compte utilisateur le compte rendu des tâches doit être envoyé (l'envoie se fait par mail). Si la variable MAILTO est une chaîne vide (MAILTO=""), aucun e-mail ne sera envoyé.

Chaque ligne du fichier destinée à la définition de la tâche, contient six champs. Les cinq premiers, séparés l'un de l'autre par un espace, déterminent les moments d'exécution de la tâche décrite au sixième champ. La signification de ces champs ainsi que les valeurs qu'ils peuvent prendre sont décrites ci-dessous.

1. Minute (0 à 59) ;
2. Heure (0 à 23) ;
3. Jour du mois (1 à 31) ;
4. Mois (1 à 12) ;
5. Jour de la semaine (0 à 6 : 0 = Dimanche, 1 = Lundi, etc).

Mis à part les valeurs citées ci-dessus, chaque champ temporel peut contenir différents symboles :

- Une liste de valeurs valides, séparées par des virgules (1,3,5 dans le champ mois correspond à janvier, mars, mai) ;
- Un intervalle valide composé de deux chiffres ou nombres séparés par un tiret (1-5 dans le champ jour correspond à exécuter la tâche tous les jours du lundi au vendredi)
- Le caractère * signifie toutes les valeurs possibles du champ (* dans le champ minute signifie toutes les minutes)
- Une combinaison de symbole : */5 (dans le champ minutes : tous les 5 minutes), 0-23/3 (dans le champ heures : toutes les 3 heures)

Le sixième champ reprend la commande ou le script à exécuter. Dans le fichier /etc/crontab, la commande est remplacée par des scripts run-parts utilisés pour l'exécution de scripts placés dans les répertoires /etc/cron.hourly, /etc/cron.daily, /etc/cron.weekly et /etc/cron.monthly (toutes les heures, tous les jours, toutes les semaines ou tous les mois, respectivement).

Exemples

La commande doit être exécutée tous les lundis à deux heures du matin

```
0 2 * * 1 commande
```

La commande doit être exécutée chaque premier et quinze de chaque mois à minuit

```
0 0 1,15 * * commande
```

La commande doit être exécutée tous les matins du lundi au vendredi à 7 h 30

```
30 7 * * 1-5 commande
```

La commande doit être exécutée tous les quarts d'heure de 15 à 19h du lundi au vendredi seulement en 1ère quinzaine du troisième trimestre

```
0,15,30,45 15-18 1-15 7-9 1-5 commande
```

2.2. Anacron

L'exécution des tâches programmées nécessite que la machine soit allumée au moment prévu pour son exécution, ce qui est probable dans le cas d'un serveur. Néanmoins, même les serveurs et à fortiori les stations de travail, ne restent pas sous tension sans arrêt (crash, maintenance, coupure de l'alimentation électrique,…). Dès lors, il y a un risque que la tâche ne soit jamais réalisée. Pour éviter cela, le service *anacron* se charge de vérifier et, le cas échéant, exécuter ces tâches dès la remise en route de la machine

A la différence de crond, anacron n'est pas un démon, ce qui signifie qu'il ne fonctionne pas en permanence sur le système. Il doit être démarré spécifiquement et s'arrête une fois ses tâches terminées. Pour automatiser son exécution, il pourra être placé dans un script de démarrage.

2.3. La commande at

Cette commande permet définir la date d'exécution d'une tâche donnée. Contrairement à crontab qui gère les tâches périodiques, avec *at* la tâche ne s'effectue qu'une seule fois. Pour cela, le démon *atd* doit être actif, c'est lui qui vérifie chaque minute si une tâche doit être effectuée. Comme pour crontab, les tâches sont envoyées en arrière plan pour leur exécution, il y a donc lieu de prévoir des redirections dans le cas où les tâches utilises les entrées/sorties standards.

La commande *atq* permet de visualiser la liste des travaux en attente.

Syntaxe de la commande at

at Heure [date] [incrément]

Il existe de très nombreuses manières de spécifier la date à laquelle une tâche doit s'effectuer. Quelques exemples sont fournis ici à titre d'exemple, pour plus de détails il faut se référer à la man page. L'heure doit être spécifiée en premier, la date est optionnelle.

Exemples d'argument et formats possibles pour indiquer l'heure et la date :

- L'heure au format HH:MM. Par exemple, 04:00 indique 4h00. Si l'heure programmée est déjà passée, le processus est exécuté le lendemain à la même heure ;
- La date ;
- Le format « mois jour année » : par exemple, May 15 2008 indique le 15mai 2008 ;
- Le format MMJJAA, MM/JJ/AA ou MM.JJ.AA : par exemple, 051508 pour le 15 mai 2008 ;
- Les mots clés suivants :
  - midnight : indique 00h00,
  - noon : indique 12h00,
  - teatime : peut être utilisé pour 16h00.
- L'argument now +temps : le temps est exprimé en minutes (minutes), heures (hours), jours (days) ou semaines (weeks). Par exemple, now +5 days indique que la commande sera exécutée à l'heure actuelle mais dans cinq jours.

Après avoir validé (avec la touche [ENTER]) la commande at, le prompt est modifié en **at>**. C'est à ce moment qu'il faut définir la ou les commandes à exécuter. Une fois la commande définie, pressez la touche [ENTER] pour afficher une ligne vide et tapez CTRL+D pour quitter.

Exemples de commande at

```
# at  13:10  [ENTER]
at>  echo  'hello'
at>  CTRL+D (le système  affiche alors <EOT> : End Of Text)
job 1 at 2003-03-19 09:17
```

Dans cet exemple, comme il n'est pas précisé de destination pour l'affichage du mot hello, il sera envoyé dans la boite mail de l'utilisateur (/var/spool/mail) qui a invoqué la commande.

3. Quotas d'espace disque

Les quotas s'appliquent sur un système de fichiers et sont de deux types :

- Les quotas qui définissent le nombre maximal de fichiers (en nombre d'inodes) qu'il est possible de créer ;
- Les quotas qui définissent la taille maximale (en nombre de blocs) qui peut être occupée.

Les quotas peuvent être fixés pour un utilisateur ou pour un groupe d'utilisateurs. Leur but est d'obliger les utilisateurs à mieux gérer leur espace et les empêcher de saturer une partition.

Plusieurs possibilités s'offrent à l'administrateur lorsqu'un utilisateur (ou un groupe) dépasse la taille maximale fixée par le quota :

- Soit l'utilisateur est simplement prévenu qu'il dépasse son quota et l'administrateur lui fait confiance pour qu'il « nettoie » ses fichiers ;
- Soit l'utilisateur est prévenu qu'il est proche de son quota maximum lorsqu'il atteint une certaine limite (limite douce). Il peut cependant continuer de créer de nouveaux fichiers jusqu'à atteindre son quota (limite dure).

Les quotas sont très utilisés, notamment sur les serveurs de messageries : les fournisseurs d'accès à Internet « offrent » un espace de stockage limités pour le stockage des mails de leurs abonnés.

Activation des quotas sur un système de fichiers

Pour activer les quotas, il faut monter les systèmes de fichiers avec l'option *usrquota* et/ou *grpquota* selon qu'ils s'appliquent à un utilisateur, à un groupe d'utilisateur ou au deux. Cela peut être réalisé manuellement avec la commande mount ou au démarrage en configurant le fichier /etc/fstab.

Par exemple, supposons que le point de montage de la partition /dev/hda3 est /home. Pour configurer le fichier /etc/fstab, on ajoute l'option usrquota (et/ou grpquota) sur la ligne qui configure le montage de /home :

Exemple de configuration de la ligne correspondante dans le fichier /etcfstab

```
/dev/hda3   /home   ext2    defaults,usrquota,grpquota    1    2
```

Les fichiers contenant les définitions des quotas (quota.user et quota.group) sont situés dans la racine de chaque système de fichiers concerné. Ils peuvent être créés automatiquement grâce à la commande quotacheck. Cette commande vérifie la présence et la cohérence des informations de gestion des quotas.

```
# /usr/sbin/quotacheck -avug
```

L'activation des quotas est ensuite lancée par la commande quotaon. Pour les activer automatiquement lors du démarrage du système, il faut ajouter dans un fichier d'initialisation (situé généralement dans /etc/rc.d). Activation des quotas :

```
# /usr/sbin/quotaon -avug
```

Attribution d'un quota à un utilisateur

La commande *edquota* lance un éditeur de texte permettant d'affecter ou modifier un quota à un utilisateur ou à un groupe d'utilisateurs. Deux limites peuvent être fixées :

- La limite "douce" (soft) : lorsque cette limite est atteinte ou dépassée, un message d'avertissement est affiché lors de chaque nouvelle allocation de fichier ou de bloc ;
- La limite "dure" (hard): lorsque cette limite est atteinte (elle ne peut être dépassée), il est impossible à l'utilisateur de créer de nouveaux fichiers ou d'allouer de nouveaux blocs.

La limite douce se transforme en limite dure quand elle a été atteinte ou dépassée depuis un temps prédéfini appelé « la période de grâce » (grace period).

Statistiques sur les quotas

Tout utilisateur peut s'informer de l'état des quotas qui lui sont attribués en utilisant la commande *quota*. L'administrateur peut, en plus, utiliser la commande *repquota* pour obtenir une liste des quotas associés à un ou plusieurs systèmes de fichiers.

Bibliographie

Cours

FUCKS P., S*écurité firewalls*, Financial Art, 2008.

Ouvrages

BERLAT A., BOUCHAUDY J.-F., GOUBET G., *Linux administration*, Eyrolles, 2003.

BOUCHAUDY J-F, *TCP/IP sous Linux*", Eyrolles, Paris 2003

FRISCH Æ., *Les bases de l'administration système*, O'Reilly, 2002.

HENRY J., *Outils d'administration Linux en entreprise*, Vuibert, Paris, 2005.

PÉPING J., *Solutions de stockage*, Eyrolles, 1999.

PURDY G.N., *Les iptables Linux. Précis & concis*, O'Reilly, 2004.

Red Hat Inc, *Red Hat Enterprise Linux 3: Guide d'administration système*, 2003.

Red Hat Inc, *Red Hat Enterprise Linux 3: guide de référence*, 2003 .

ROBBINS A, BEEBE N.H.F., *Introduction aux scripts shell*, 3ème Ed., O'Reilly, 2005.

WELSH M., DALHEIMER K.ET KAUFMAN L., *Le système Linux*, O'Reilly, 2000.

WIELCH M., PRAHM J., ESSER H.-G., *La bible Linux*, MicroEdition, Paris, 2000.

ZIEGLER R.L., *Linux sécurié*, Campus Press, Paris, 2000.

Sources électroniques

AVI ALKALAY, ET ALII, *Gestion précise de l'heure et de la date*, septembre 2005, http://www.ibiblio.org/pub/Linux/docs/HOWTO/translations/fr/text/TimePrecision-HOWTO.txt

CALECA C., *SNMP*, septembre 2003, [en ligne], http://christian.caleca.free.fr/snmp/install_snmp1.htm

CENTOS.ORG, *Managing Software with yum*, [en ligne], 2005, http://www.centos.org/docs/4/html/yum/CISCO SYSTEM, INC., *Simple Network Management Protocol (SNMP)*, 1992-2010, [en ligne], http://www.cisco.com/en/US/docs/internetworking/technology/handbook/SNMP.html

COOPER M., *Guide avancé d'écriture des scripts Bash*, mai 2008, [en ligne], http://abs.traduc.org/abs-fr/

FONTAINE S., CALON A., PERAIRE S., *SNMP*, Décembre 2007, [en ligne], http://www.frameip.com/snmp/

GELDENHUYS M., *Les codes de statut HTTP*, juin 2002, [en ligne], http://www.apachefrance.com/Articles/7/page2.html

LEWIS AJ, *LVM HOWTO*, 2002-2006, [en ligne], http://tldp.org/HOWTO/LVM-HOWTO/

KADIONIK P., *L'administration de réseau. Howto mise en œuvre de Net-SNMP*, s.d., [en ligne], http://uuu.enseirb-matmeca.fr/~kadionik/embedded/snmp/net-snmp.html

NETCRAFT LTD, *Web Server Survey*, 2009, http://news.netcraft.com/

PUBLIC DOMAIN, *NTP: The Network Time Protocol*, octobre 2009, [en ligne], http://www.ntp.org/index.html

RED HAT, INC., *Red Hat Enterprise Linux 4. Introduction à l'administration système*, 2005, [EN LIGNE], http://web.mit.edu/rhel-doc/4/RH-DOCS/pdf/rhel-isa-fr.pdf

RED HAT, INC., *Red Hat Enterprise Linux 4. Guide de référence*, 2005, [EN LIGNE], http://web.mit.edu/rhel-doc/4/RH-DOCS/pdf/rhel-rg-fr.pdf

RENARDIAS V., *Remise des pendules à l'heure avec NTP*, 2000, Diamond Editions/Linux magazine France.

SAMBA TEAM, *Samba. Opening windows to a wider world*, s.d., [en ligne], www.samba.org

THE APACHE SOFTWARE FOUNDATION, *HTTP Server Project*, s.d., [en ligne], http://httpd.apache.org/

Télécommunications et réseaux 1

Support de cours

1

Gestion des ressources

Module 9

2

Introduction

- Pourquoi surveiller les ressources?
- Quelles sont les principales ressources à surveiller?
- Comment les surveiller?
- Comment résoudre un problème de performance du système?

3

Pourquoi surveiller les ressources?

- Leur mauvaise utilisation peut rendre un système très lent, voire inutilisable
  - Mauvais temps de réponse pour une tâche interactive.
  - Temps trop long pour l'exécution d'un programme
  - Tâche qui ne peut être exécutée par manque de ressources
- Réagir dans l'urgence
  - Tentative d'essayer de corriger le problème avant de l'avoir diagnostiqué
  - Un problème de performance ne se résout pas en 5 minutes en appliquant une liste de suggestions
- Surveiller avant les problèmes
  - Permet de connaitre l'utilisation des ressources dans un fonctionnement normal → servira de référence

4

Quelles sont les principales ressources à surveiller?

- Ressources principales
  - Le CPU
  - La mémoire
  - Les entrées/sorties
  - L'espace disque
- Avant les problèmes
  - Respecter la configuration demandée par les OS et les éditeurs de logiciels
  - Benchmark pour simuler le comportement de l'activité de l'application

5

Surveiller le CPU

- ps
  - Permet d'obtenir une vue globale de l'activité du système
- pstree
  - Affiche les processus sous forme d'une arborescence
  - Il permet d'avoir une vue des relations entre processus
  - Il permet de voir rapidement ce qui fonctionne sur le système
- top
  - Produit un affichage régulièrement mis à jour de l'état du système et des processus les plus actifs
  - Permet d'interagir avec les processus

6

Surveiller le CPU

- uptime
  - La commande uptime fournit une estimation de la charge du système
  - La charge moyenne doit être inférieure à 3
- time
  - Affiche le temps CPU consommé par un processus
    - Temps réel écoulé entre sa création et sa mort
    - Temps consommé pour exécuter les instructions en mode user, les instructions du processus, les calculs,…
    - Temps consommé pour exécuter les instructions en mode kernel (les primitives du noyau)
- ulimit
  - Permet de modifier la valeur courante d'une limitation de ressource
  - Généralement utilisé pour empêcher la création de fichiers core

7

Surveiller le CPU

- w
  - Affiche les utilisateurs connectés et ce qu'ils font
- vmstat
  - Affiche des statistiques sur les ressources du système, dont les processus, la mémoire et l'activité du CPU
    - Us : % de temps CPU inclus dans le temps utilisateur
    - Sy : % de temps CPU inclus dans le temps système
    - Id : temps d'inactivité
  - La première ligne est une moyenne à partir du démarrage, elle est à ignorer
- sar
  - Commande générale de surveillance des ressources
- iostat
  - Affiche l'utilisation du CPU et des disques et partitions

8

Le CPU

- Que faire en cas de niveau élevé d'utilisation de temps CPU?
  - Rien ?
    - Rien si cela se produit ponctuellement et s'il n'y a pas de problème
      - Cela signifie simplement que le système travaille
  - Agir ?
    - Si problème de performance + utilisation élevée de temps CPU
      → l'insuffisance de cycle CPU est sans doute un facteur qui contribue au problème

9

Le CPU

- Que faire lorsque des conflits de ressources CPU sont la source d'un goulot d'étranglement?
  - Favoriser certaines tâches en utilisant les priorités des processus
  - Réduire la consommation de CPU :
    - Déplacer une partie de la charge sur un autre système
    - Exécuter certaines tâches plus tard dans la journée
  - Modifier la méthode d'ordonnancement des processus (si le système le permet)

10

Le CPU

- Voir la priorité d'un processus
  - top ou ps
    - NI : priorité d'exécution demandée par le propriétaire du processus ou par root
    - PRI : priorité d'exécution courante calculée et mise à jour dynamiquement par l'OS
- Modifier la priorité d'exécution
  - nice, renice, top (+r)
- Différer l'exécution des tâches
  - at, batch
- Ordonnanceur de processus
  - sched_setscheduler
  - AIX : shedtune
  - Solaris : dispadmin

11

Gestion de la mémoire

- Mécanisme de pagination (swapping)
  - Répartition de la mémoire disponible entre processus lorsque ceux-ci demandent une quantité de mémoire qui excède la mémoire physique disponible
  - Swapping = transfert d'un processus sur le disque pour libérer de la mémoire
- Avantage
  - Il rend possible l'utilisation de mémoire virtuelle par le biais de laquelle la mémoire utilisée par un processus peut excéder la mémoire physique disponible
  - Cet état est rendu possible par le fait que les processus n'ont pas constamment besoin de toute la mémoire
- Inconvénient
  - Si la mémoire est insuffisante et la gestion des priorités mal étudiée, le système risque de passer beaucoup de temps à paginer plutôt qu'à exécuter les processus

12

Surveiller la mémoire

- vmstat
- top
- free
  - Affiche les quantités de mémoires (vive et swap) libres et utilisées
- Gestion de l'espace de pagination
  - En fonction du type de tâche
    - Mono-utilisateur : 1 à 2 fois la mémoire physique
    - Gros besoin de mémoire : 3 à 4 fois la mémoire physique
    - Le mieux étant de disposer de suffisamment de mémoire physique

13

Surveiller la mémoire

- Facteur influençant l'espace de pagination
  - Les tâches qui nécessitent beaucoup de mémoire
  - Les programmes très « gros »
  - Un nombre important de tâches s'exécutent simultanément
- Activation de l'espace de pagination
  - /etc/fstab
  - Commandes swapon ou swap

14

Les entrées-sorties

- Facteurs influençant les performances des entrées-sorties sur disque
  - La répartition des disques et des contrôleurs
    - Comparer le débit du contrôleur à ceux des disques qu'il contrôle
    - Se limiter à 80% du débit du contrôleur
  - La répartition des données sur les différents disques
    - Répartition équitable des entrées-sorties entre les disques et les contrôleurs
    - Par exemple, répartir les fichiers très utilisés sur deux ou plusieurs disques
  - Localisation des fichiers sur le disque physique
    - Fragmentation
    - L'accès séquentiel à de gros fichiers est plus efficace lorsque les fichiers sont contigus
    - L'accès aléatoire aux fichiers est plus efficace sur les secteurs du centre du disque (moins de déplacement des têtes de lecture)

15

Comment résoudre un problème de performance du système?

1. Poser le problème
   - Il faut décrire le problème de la manière la plus précise possible
     - Cette description augmentera les chances de trouver une solution
     - Aidez-vous en dessinant un schéma incluant tous les composants du système
2. Déterminer la cause
   - Déterminer le type de preuve que vous devez recueillir et les récolter
     - Quand et sous quelles conditions le problème survient-il?
     - Une modification du système a-t-elle été réalisée récemment?
     - Quelles ressources affectent les performances?
     - Qu'est-ce qui continue à fonctionner normalement?
     - Quelle métrique peut être utilisée pour mesurer les performances?
     - ...

16

Comment résoudre un problème de performance du système?

3. Formuler les objectifs à atteindre
   - Formuler les objectifs à atteindre facilite la réalisation des modifications
4. Concevoir les modifications adéquates
   - Déterminer les modifications
     - Sans doute la partie la plus compliquées
     - Procéder au réglage de l'ensemble du système car les performances sont le résultats de l'interaction de tous les composants du système
   - Généralement, le plus efficace est de procéder par des petites modifications successives

17

Comment résoudre un problème de performance du système?

5. Surveiller
   - La surveillance du système permet d'évaluer si les modifications ont apportés les améliorations attendues
     - Les effets des modifications doivent pouvoir être mesurées
6. Recommencer
   - Une modification du système entraînera de nouvelles interactions entre processus, il sera peut être nécessaire de revenir à la première étape et recommencer le réglage

- Remarque
  - Les ressources ne sont pas infinies → compromis sur la manière d'allouer et de partager les ressources
    - Fonction de la priorité et de l'importance relatives des différentes activités

18

Le noyau linux

Module 10

19

Introduction

- Rôle l'O.S.
  - Permettre l'utilisation d'un ordinateur et de ses périphériques via des logiciels
- Système d'exploitation
  - Un noyau (kernel)
  - Outils système

20

Introduction

- Noyau = linux
  - responsable des tâches de base de l'O.S.
  - gestion de la mémoire
  - gestion des processus pour les noyaux multi-tâches
  - gestion de systèmes de fichiers
  - gestion des entrées/sorties (avec des pilotes de periphériques)
  - interface avec l'espace utilisateur (avec des appels système)
  - services réseaux (TCP/IP, PPP, firewall, ...)
- Noyau + applications = système GNU/Linux

21

Version de noyau

- Noyau générique
  - Le noyau issu de l'installation est un noyau générique
- Nom du noyau
  - X.Y.Z-r (exemple : 2.4.12-2 ou 2.5.3-4)
    - Jusqu'au noyau 2.4, si le chiffre Y est un nombre pair, le noyau est stable
    - Depuis le noyau 2.6, le chiffre Y n'a plus de signification particulière
- Version du noyau (sept. 2011) : 3.0.4

22

Pourquoi recompiler le noyau?

1. Optimiser l'OS en fonction du matériel
   - Meilleure gestion des processus
   - Meilleure stabilité
   - Un temps de démarrage plus court
   - Construire un kernel qui est en parfaite adéquation avec le matériel
2. Améliorer la sécurité
   - Noyau monolithique (moins de modules)
   - Ajouter un correctif

23

Pourquoi recompiler le noyau?

3. Ajouter/supprimer des fonctionnalités
   - Supporter de nouveaux FS
   - Intégration du support de périphérique non présent dans le noyau générique
   - Transformer un PC en routeur
4. Meilleure gestion des ressources
   - Retrait des composants inutiles qui consomment des ressources
   - → Une utilisation plus faible de la mémoire
5. Un programme nécessite un noyau plus récent pour fonctionner

24

« Type » de noyau

- Noyau monolithique
  - Contient tous les pilotes
  - Ne prend pas en charge les modules
- Noyau modulaire
  - Trois options :
    - Yes : intégré au noyau
    - Module : compilé en tant que module
    - No : non compilé

25

Modules du noyau

- Noyau modulaire
  - Le noyau lancé à l'amorçage ne contient que les fonctions nécessaires au démarrage
  - Les autres fonctions peuvent être liées au noyau en tant que modules
- Avantages
  - Modules liés à la demande (consomment des ressources uniquement s'ils sont utilisés)
  - En cas de modification matérielle, le noyau ne doit pas être recompilé
  - L'utilisation de modules ne nécessite pas nécessairement un redémarrage de la machine
  - Les fabricants de matériel ne sont pas obligé de fournir leur code

26

Modules du noyau

- Version de module
  - Avant le noyau 2.6.15
    - La version du noyau et des modules devaient être identiques
    - Modules situés dans /lib/modules/*n°version*
  - Depuis le 2.6.15
    - Option du noyau *Module versioning* activée
    - Le module peut fonctionner avec des versions de noyau différentes
      - Le module enregistre des informations indiquant quels noyaux sont « compatibles »

27

Modules du noyau

- Modules utilisés au démarrage
  - Intégrés au noyau
  - Fournis au noyau sous la forme d'un disque virtuel (initrd) par le boot loader
- Modules de base
  - Chargés au démarrage par les scripts d'initialisation (modules de gestion de l'USB, …)
- Modules dynamique (USB,…)
  - Système udev
    - Charge les modules lorsqu'ils sont nécessaires

28

Modules du noyau

- Dépendances entre modules
  - Certains modules ne se chargent que si d'autres sont chargés
  - Les dépendances sont enregistrées le fichier /lib/modules/*n°noyau*/modules.dep
  - Mise à jour du fichier : depmod
    - Crée des fichiers *.map pour faire la correspondance entre composants matériels et modules nécessaires

29

Préparer un nouveau noyau

- Récupérer les sources
  - www.kernel.org
  - Attention : la plupart des distributions n'utilisent pas le noyau original
  - Où placer les sources?
    - /usr/src/ ?
    - En faire une copie dans son home directory
  - Décompacter les sources
    - tar -xvjf...

30

Préparer un nouveau noyau

- Outils adéquats
  - Il faut disposer des logiciels adéquats pour compiler (voir la documentation du noyau)
    - module-init-tools lors du passage au 2.6
    - Compilateur et librairies
- Faire l'inventaire de son matériel
  - lspci, /proc, outils graphique, …
    - Permet de savoir quelles options choisir lors de la configuration du nouveau noyau

31

Préparer un nouveau noyau

- Sauvegarder le noyau actuel
  - Renommer le noyau ou le copier (mv ou cp)
  - Sauvegarde de la configuration du noyau
    - # cd répertoire_contenant_les_sources
    - # make oldconfig
    - # cp .config .config.sav
- Sauvegarder les modules
  - Sauvegarde nécessaire dans le cas d'une compilation du noyau actuel
  - Copie du répertoire /lib/modules/N°_de_version_du_noyau
- Sauvegarde de la configuration de démarrage
  - lilo ou grub
  - mbr (dd)
  - Disquette de boot, system rescue,…

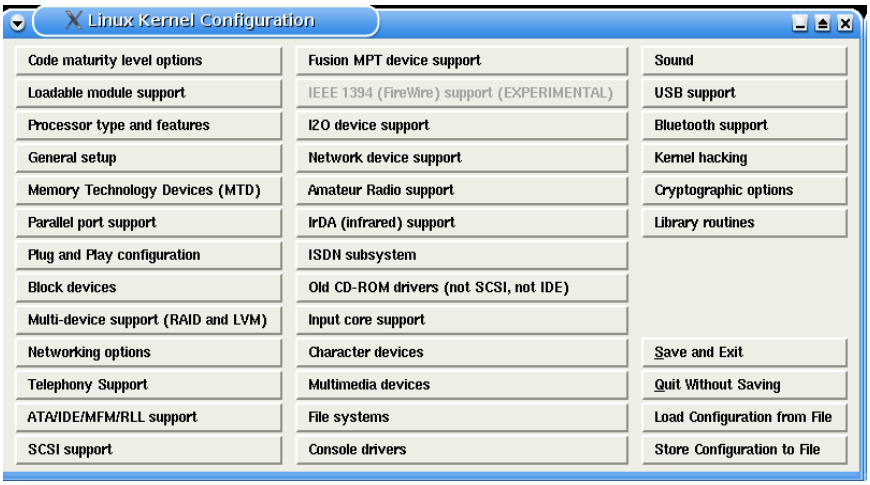
32

Configurer le nouveau noyau

- Configuration du noyau
  - Seule l'installation doit se faire sous le compte root
  - make config make xconfig make menuconfig
  - Noyau monolithique ou modulaire
  - Y = intégré, N = non pris en compte M= Module
- Les Makefiles
  - Ce sont des fichiers contenant les instructions de compilation
  - Un fichier makefile est créé à la fin de la configuration du noyau

33

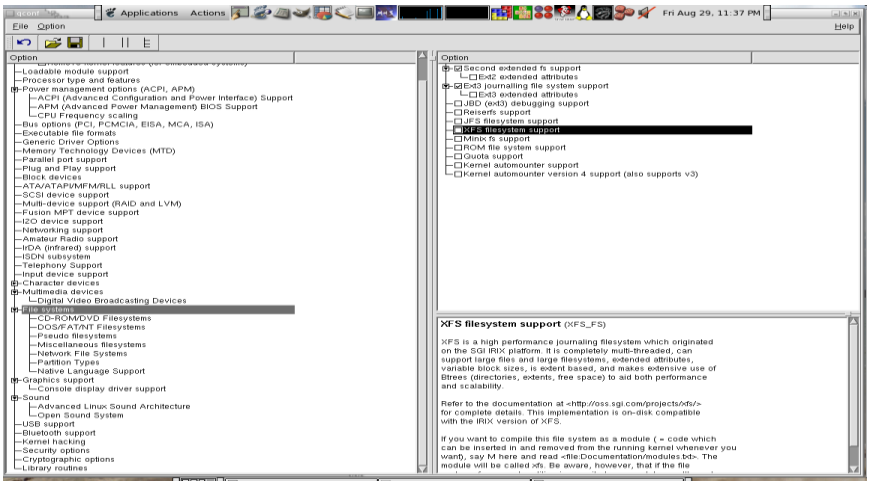
Exemple de make xconfig



Source : http://arnofear.free.fr/linux/mdksarge/kernel1.png

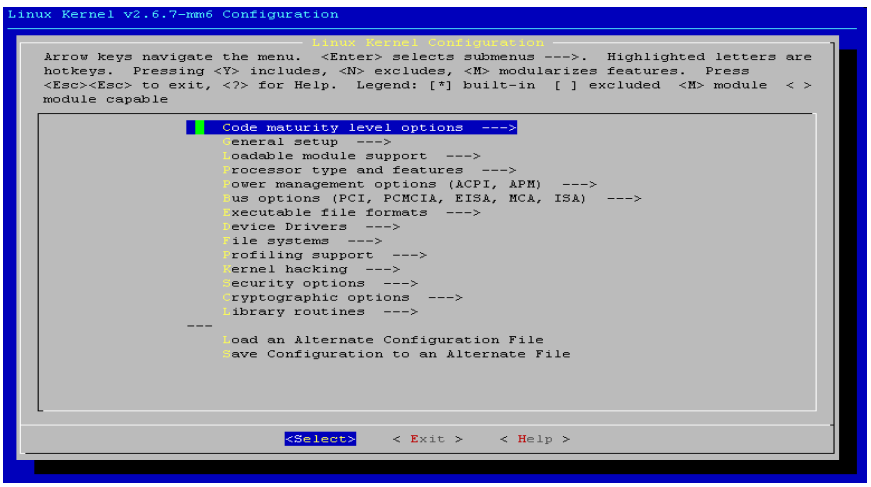
34

Exemple de make xconfig



Source http://kerneltrap.org/sup/799/xconfig.png

Exemple de Make menuconfig



Source : http://www.labo-linux.org/articles-fr/recompiler-son-noyau-linux/menuconfig.PNG

Comment compiler le noyau?

- Nettoyer et chercher les dépendances
  - make clean
  - make dep
- Compiler et installer le noyau
  - make bzImage
  - make install
  - make all
- Compiler et installer les modules
  - make modules
  - make modules_install
- Nettoyer après le travail
  - make clean permet d'effacer tous les objets créés depuis la dernière compilation
- Ajouter une image dans le loader
  - Placer le noyau (bzImage) dans le /boot
  - Éditer le fichier de configuration

37

Compilation croisée

- Compilation croisée
  - Consiste préparer un noyau sur une machine différente de la machine cible
  - Quel intérêt?
    - Machine cible est lente
    - Pas d'espace disque suffisant
    - Eviter d'installer des outils de développement sur la machine cible

38

LVM

Module 11
Logical Volume Manager

39

Les volumes logiques

- LVM (Logical Volume Manager)
  - Couche logicielle entre le système de fichiers et les partitions
  - Permet de gérer de manière souple les espaces de stockage
- Alternatives aux partitions physiques
  - Avantages
    - Flexibilité pour allouer de l'espace aux applications et aux utilisateurs
      - Une partition répartie sur plusieurs disques dur
    - Redimensionnement et déplacement des volumes de stockage à la demande et « à chaud »
    - Gestion par groupes d'utilisateurs (nommage)
    - Les instantanés (snapshots)
    - Fonctionnalités ≈RAID 0 et RAID1

40

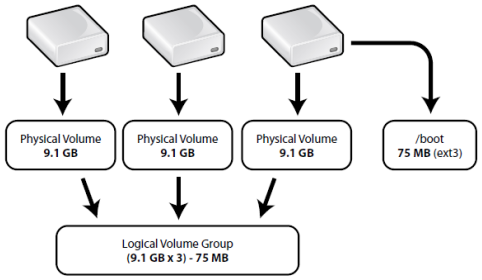
Les volumes logiques

- Inconvénients
  - Doivent être mis en place dès la création de la partition
  - S'occupent des partitions, pas du système de fichiers
  - Volume réparti sur plusieurs disques
    - Une erreur sur un disque rend les données de "tous" les disques inutilisables
  - Gestion/dépannage plus compliqué
  - Les boot loader ne parviennent pas à démarrer sur une partition LVM

41

Concepts

- Physical volume (PV)
  - Un disque physique ou une partition est associée à un volume physique
- Volume Group (VG)
  - Ensemble de volumes physiques
    - Possibilité de modifier sa composition (ajout/suppression de volumes physiques)
  - Cet ensemble est divisé en volumes logiques

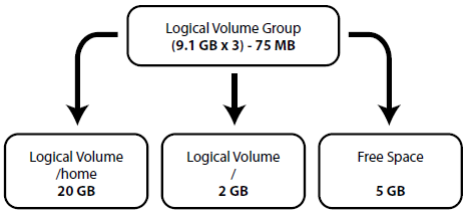


Source : RedHat Inc., System administration guide, chapitre 7

42

Concepts

- Logical volume (LV)
  - Equivalent logique d'une partition
  - Peut contenir un système de fichiers
  - Le LV « possède » son fichier dans /dev
  - Le LV peut être associé à un point de montage



Source : RedHat Inc., System administration guide, chapitre 7

43

Concept

- Physical Extent (PE)
  - Une unité d'allocation physique de données
  - Chaque PV est divisé en PE
  - Par défaut un PE = 4Mo
- Logical Extent (LE)
  - Une unité d'allocation logique de données
  - Chaque LV est divisé en LE
  - Les partitions logiques peuvent être étendues (agrandies) d'un ou plusieurs extents
  - La taille d'extents est la même pour tous les volumes logiques du groupe de volumes

44

Mise en place

1. Initialisation
   - Création des PV en fonction des ressources disponibles
   - Exemple : pvcreate /dev/hdb
2. Création d'un volume Group (VG)
   - vgcreate monVG /dev/hda1 /dev/hdb
3. Création d'un volume logique (LV)
   - lvcreate –L 150M –n storage monVG

45

Opérations sur les volumes

- Étendre un volume logique
  - lvextend –L +40M /dev/groupelvm/storage
  - e2fsadm
- Supprimer un volume logique
  - lvremove
- Réduire un volume logique
  - lvreduce
- Ajouter un volume physique à un groupe de volumes
  - vgextend
- Supprimer un volume physique d'un groupe de volumes
  - vgreduce

46

Remarques

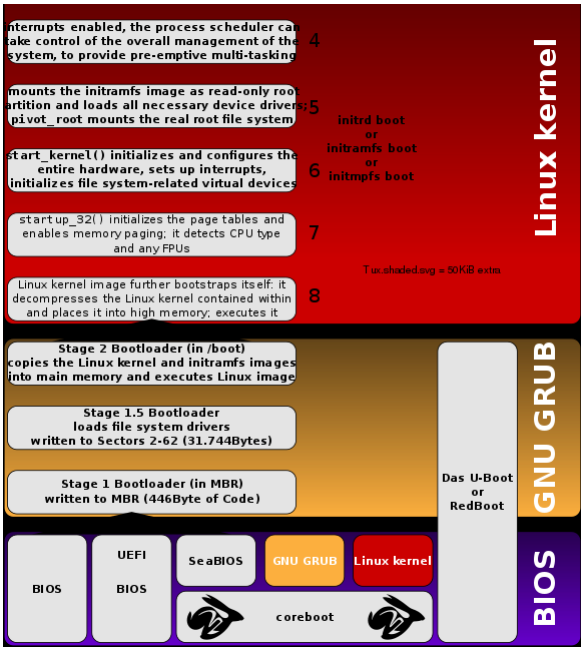
- LVM sur la racine du FS
  - Les procédures de maintenance et de réparation sont facilitées par l'absence de LVM
- Les boot loader ne gèrent pas encore les volumes LVM
  - Pas de /boot en LVM
- En production, un seul PV par disque réel
  - Facilité d'administration
  - Meilleures performances
- Plusieurs PV par disque réel
  - Migration d'un système existant vers LVM
  - Diviser un « gros » disque en plusieurs groupes de volumes

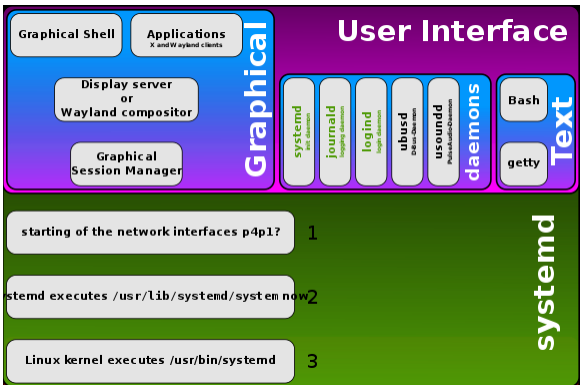
47

Démarrage et arrêt d'un système Linux et gestion des services

Module 12

48





Source :
https://commons.wikimedia.org/wiki/File:Linux_startup_process_wip.svg

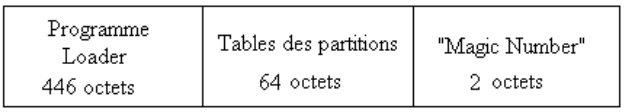
Procédure de démarrage

- Auto-amorçage
  - Une machine a besoin d'un O.S. pour fonctionner, mais elle doit aussi être capable de lancer son O.S. elle-même
- Procédure simplifiée
  1. Mise sous tension
  2. Exécution des instructions du programme de chargement du BIOS
     - Ces instructions sont stockées en mémoire ROM. Elles ont pour but de :
       - Vérifier la disponibilité des ressources : c'est le "Power On Self Test" (POST)
       - Localiser et exécuter le programme principal de démarrage (**boot loader**).
         - » Une partie de celui-ci est généralement situé dans le MBR (Master Boot Record) du disque dur

51

Organisation du MBR

Secteur de boot "MBR"



- Structure du MBR (512 octets)
  - 446 octets contiennent le programme de chargement (le loader)
  - 64 octets décrivent les partitions primaires (taille, localisation, type, statut). Avec ces 64 octets, le système est limité à 4 partitions primaires (les seules reconnues par le BIOS)
  - 2 octets qui représentent le "magic number". Il s'agit d'une valeur numérique qui est utilisée pour vérifier la signature du secteur

52

Procédure de démarrage

3. Le programme de démarrage principal (loader) se charge en mémoire
   - Exemples de boot loader : **Grub2**, Loadlin, …
   - Il donne la possibilité de choisir quel OS doit être démarré
   - Il peut être placé à différents endroits : disquette, MBR, serveur de fichiers,…
   - Il cherche sur la table des partitions une partition active, puis, si une telle partition existe, il charge le secteur de boot de cette partition.
     - Celui-ci contient :
       - » des données qui indiquent les propriétés de la partition
       - » Le reste du programme d'amorçage qui démarre le système d'exploitation
   - Il charge le reste du programme de démarrage (car seuls les 446 premiers octets sont contenus dans le MBR)
   - Il localise le noyau et le charge

53

Grub

- GNU GRUB (acronyme signifiant en anglais « GRand Unified Bootloader ») est un programme d'amorçage GNU qui gère la gestion du chargement des systèmes d'exploitation disponibles sur le système. Il permet à l'utilisateur de choisir quel système démarrer. Il intervient après allumage de l'ordinateur et avant le chargement du système d'exploitation.
- GRUB dans sa version 2 (entièrement réécrite) est un chargeur de démarrage libre au même titre que Das U-Boot ou Barebox pour du matériel embarqué.
- Ses nombreux avantages, son histoire et son fonctionnement sont décrits dans la page : http://doc.fedora-fr.org/wiki/GRUB2_:_Les_bases_pour_Fedora

54

Grub

- Fichiers Grub2 :
  - La configuration de GRUB2 est composé de trois principales dans des fichiers inclus :
    - /etc/default/grub - le fichier contenant les paramètres du menu de GRUB 2,
    - /etc/grub.d/ - le répertoire contenant les scripts de création du menu GRUB 2, permettant notamment de personnaliser le menu de démarrage,
    - /boot/grub2/grub.cfg - le fichier de configuration final de GRUB 2, non modifiable. (/boot/grub/grub.cfg sous Debian)
  - Ce dernier fichier est généré automatiquement par le programme grub2-mkconfig à partir des scripts /etc/default/grubet /etc/grub.d/ :

```
# cat /etc/default/grub
GRUB_TIMEOUT=5
GRUB_DISTRIBUTOR="$(sed 's, release .*$,,g' /etc/system-release)"
GRUB_DEFAULT=saved
GRUB_DISABLE_SUBMENU=true
GRUB_TERMINAL_OUTPUT="console"
GRUB_CMDLINE_LINUX="crashkernel=auto rd.lvm.lv=centos/root rd.lvm.lv=centos/swap rhgb quiet"
GRUB_DISABLE_RECOVERY="true"
```

55

Procédure de démarrage

4. Le noyau est chargé
   - Le noyau s'exécute et s'initialise : réservation de la mémoire, prise en compte de la zone d'échange (swap), chargement des pilotes des périphériques, montages des systèmes de fichiers, …
   - Le montage du système de fichiers racine (/) va permettre le lancement du premier processus : /sbin/init (dont le PID est 1)
   - Ou pour Systemd : /sbin/systemd

56

Gestionnaire de service

5. La suite de la procédure d'initialisation est donnée à un gestionnaire de service :
   - Il permet de les lancer, les arrêter, les relancer, d'en assurer un suivi, etc. C'est aussi un gestionnaire de session. Il permet l'utilisation du même matériel en passant d'une session (~loging) à une autre sans que cela pose de problème. Il permet encore de gérer les points de montage qu'ils soient automatiques ou non.

57

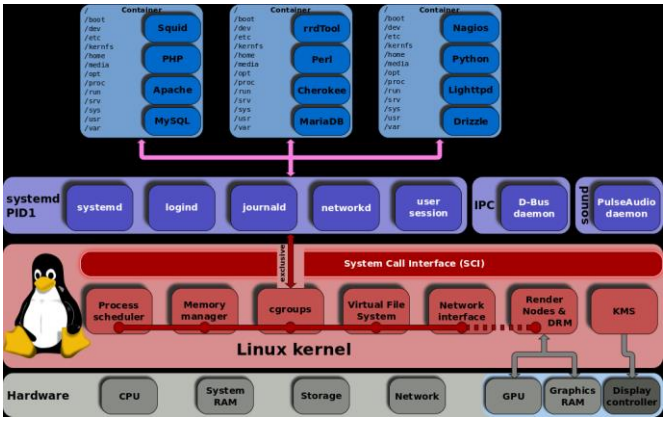
Init vs Systemd

- init (abréviation de "initialization") est le programme sous Unix qui lance ensuite toutes les autres tâches (sous forme de scripts). Il s'exécute comme un démon informatique.
- systemd est une alternative au démon init de System V. Il est spécifiquement conçu pour le noyau Linux. Il a pour but d'offrir un meilleur cadre pour la gestion des dépendances entre services, de permettre le chargement en parallèle des services au démarrage, et de réduire les appels aux scripts shell.
- systemd dispose d'une notion d'unité. Les services sont un type d'unité.

58

Systemd

- Systemd est le système d'initialisation installé par défaut avec les distributions Arch Linux, Centos 7, Debian 8 et à partir d'Ubuntu 15.04.



Source : https://commons.wikimedia.org/wiki/File:Linux_kernel_unified_hierarchy_cgroups_and_systemd.svg

59

Services

- Certains services sont lancés par défaut au démarrage du système. Par défaut, il s'agit d'un choix des concepteurs de la distribution. Ce choix, comprenant des services essentiels et d'autres qui le sont moins.
- Vous pouvez choisir de désactiver certains services qui ne sont pas essentiels. Pourquoi ? Parce que vous n'en avez pas besoin. Ou parce que vous n'en avez pas besoin souvent, voire très rarement ou même jamais.
- Vous pouvez aussi choisir d'arrêter momentanément des services. Par exemple, pour voir ce que ça fait (ou ne fait plus ...), avant de les désactiver complétement.
- Vous pouvez activer des services qui ne sont pas par défaut, mais dont, vous, vous avez besoin.
- Et enfin, vous pouvez lancer certain service uniquement au besoin.

60

Services

- Pourquoi me préoccuper des services ?
  - La sécurité : certains services sont à "l'écoute" de sollicitations externes. Toutes sollicitations externes peut venir d'une personne malveillante. Et donc, moins il y a de services de ce type en exécution, moins il y a de portes d'entrées potentielles pour ces personnes.
  - Optimiser le lancement du système : plus il y a de services qui sont lancés au démarrage, plus cela prend du temps pour que le système deviennent fonctionnel. Il s'agit d'ailleurs d'une des principales motivation de la naissance de systemd.
  - Optimiser le système : les concepteurs d'une distribution vont, généralement, essayer de faire en sorte que la distribution par défaut puisse convenir au plus grand nombre. Par exemple, pour cela, il faut être capable de gérer différents matériels. Ces matériels, peuvent être présents ou pas sur la machine de chaque utilisateur. Et le service associé, peut être utile ou pas. Cette utilité, vous êtes finalement le seul ou la seule à pouvoir la définir. Certain services sont donc présent pour répondre au besoin du plus grand nombre, mais ne vous sont pas forcément utile. Et au contraire, des services peuvent vous être utile, bien qu'ils ne sont pas lancés par défaut.

61

Init

- La suite de l'initialisation de Linux pour Init est basée sur des scripts d'initialisation
  - Init démarre certains scripts (et services) en fonction du niveau de fonctionnement demandé → /etc/inittab
    - L'administrateur modifie rarement /etc/inittab mais plutôt les scripts auxquels il fait appel
    - Tous les services et les outils de l'espace utilisateur sont chargés
    - init monte toutes les partitions répertoriées dans /etc/fstab
  - Enfin, l'utilisateur voit s'afficher un écran de connexion

62

Niveaux d'exécutions(run levels)

- Le run level, ou niveau de fonctionnement, est un chiffre ou une lettre utilisé par le processus init des systèmes de type Unix pour déterminer les fonctions activées du système.
- Un système Linux offre plusieurs niveaux de fonctionnement :
  - 0 : Arrêt du système
  - 1 : mode maintenance
  - 2 : mode multi-utilisateur en console (pas de nfs)
  - 3 : mode multi-utilisateur en console
  - 4 : A définir par l'utilisateur
  - 5 : mode multi-utilisateur avec interface graphique
  - 6 : Arrêt et redémarrage du système (reboot)
  - 7,8,9 : A définir par l'utilisateur

63

Les niveaux de fonctionnement

- Pour vérifier le niveau d'exécution :
  - *#runlevel*
- Pour se placer dans un des niveau d'exécution(x) :
  - *#init x*
  - *#telinit X*
- Au boot, il suffit d'entrer Linux suivit du niveau désiré :
  - Linux 5
  - /sbin/init 5 ou telinit 5

64

Les scripts de démarrage

- Les « run commands »
  - Les scripts qui exécutent les différentes opérations d'initialisation et de démarrage sont appelés Run Commands (préfixe rc)
- /etc/rc.d/rc.sysinit
  - C'est le premier scripts exécuté par le processus init
  - Il contient des commandes de contrôle et d'initialisation indispensables au fonctionnement de Linux
    - Définition de l'environnement d'exécution (configuration du clavier, de la police, niveau de log de la console, l'horloge, le nom de machine, …
    - Chargement des pilotes dynamiques (modules)
    - Vérification des systèmes de fichiers (fsck). Cette opération n'est pas réalisée systématiquement
    - Création du fichier /var/log/dmesg qui contient la configuration du système
    - …

65

Les scripts de démarrage

- /etc/rc.d/rc
  - Script responsable du démarrage et de l'arrêt des services lors d'un changement de niveau de fonctionnement
- /etc/rc.d/rc.modules
  - Script responsable du démarrage des modules
  - Par défaut, rc.modules n'existe pas, mais rc.sysinit vérifie sa présence et l'exécute s'il existe
- /etc/rc.d/rc.local
  - C'est le dernier script exécuté par init
  - L'ajout de commandes à la fin de ce script permet d'exécuter des tâches (démarrage de services spéciaux, initialisation de périphériques, …) sans devoir écrire de nouveaux scripts d'initialisation

66

Les scripts de démarrage

- /etc/rc.d/init.d/
  - Contient les fichiers exécutables vers lesquels pointent les liens de rc*.d/
- /etc/rc.d/rc*.d/
  - Chacun des répertoires rc*.d contient des liens pointant vers les scripts associés à un niveau de fonctionnement
  - Les liens ont un nom qui commence
    - par la lettre K (kill) qui signifie que ce service doit être arrêté pour ce niveau de fonctionnement,
    - par la lettre S (start) qui signifie que ce service doit être démarré pour ce niveau de fonctionnement,
    - dans les deux cas, le nombre qui suit la lettre permet de fixer l'ordre d'exécution des scripts
- La commande chkconfig
  - Automatise le lancement et l'arrêt des services (crond, …) en configurant ces scripts
  - Permet d'éviter de manipuler les liens symboliques

67

chkconfig

- Scripts et commentaires
  - Chaque service géré par chkconfig doit comporter au moins deux lignes de commentaires dans son script d'init.
    - 1ère ligne : indique dans quel(s) niveau(x) d'exécution le service doit être démarré et quels sont les priorités de démarrage et d'arrêt (entre 1 et 99)
    - 2ème ligne : contient la description du service

68

Systemd

- Les niveaux systemd :

Init	systemd	
0	runlevel0.target, poweroff.target	Arrêt
1,s	runlevel1.target, rescue.target	Mode utilisateur unique
3	runlevel3.target, multi-user.target	Mode multi non graphique
2,4	runlevel2.target, runlevel4.target, multi-user.target	Idem
5	runlevel5.target, graphical.target	Mode multi-user graphique
6	runlevel6.target, reboot.target	redémarrage
emerg	emergency.target	Shell d'urgence

69

Systemd

- Les outils graphiques :
  - system-config-services
  - systemadm

70

Systemd

- Les grands principes de systemd :
  - Le grand principe de systemd est de créer les canaux de communication entre les processus avant de lancer les processus des services.
  - La synchronisation effectuée par le système permet de gérer une partie des dépendances. Par exemple, si un service utilise un socket pour recevoir les requêtes à traiter, si le socket est créée par systemd les clients de ce services peuvent accéder au socket avant que le service soit effectivement lancé et donc y écrirent leurs requêtes sans être en erreur par absence du socket.
  - L'autre mécanisme de communication inter-processus est DBUS. DBUS utilisant des sockets ...

71

Systemd

- Notion d'unité :
  - systemd utilise une notion d'unité ("Unit"). Il y a différentes type d'unités : automount, path, mount, service, socket, target, timer, wants. Pour notre explication sur les services, nous retenons les unités de type socket et service.
  - Les unités de type socket permettent à systemd de connaitre les socket à créer.
  - Les unités de type service permettent à systemd de connaitre les services à lancer. Bien entendu, les socket sont traitées par systemd avant les services.

72

Systemd

- Emplacement des fichiers de configuration:
  - La configuration des services est effectuées dans le répertoire /lib/systemd/system avec des fichiers d'extensions ".service". Il s'agit de l'emplacement standard.
  - Il est possible d'utiliser le répertoire /etc/systemd/system pour effectuer des modifications sans risque quelles soient perdues suite à une mise à jour du système
  - Pour cela, il suffit de créer le fichier /etc/systemd/system/XXX.service avec le contenu du fichier /lib/systemd/system/xxx.service, puis de modifier à votre convenance ce nouveau fichier.

73

Systemd

- La commande #systemd-analyze blame, permet de lister le temps de démarrage des différents services. Il est ainsi possible de repérer les services très ou trop long à se lancer.
- Pour changer de niveau d'exécution :
  - #systemctl isolate CIBLE.target
  - Par exemple :
    - #systemctl isolate multi-user.target ou #systemctl isolate runlevel3.target, pour passer en mode multi-utilisateur non graphique.
    - #systemctl isolate graphical.target ou #systemctl isolate runlevel5.target, pour passer en mode multi-utilisateur graphique.

74

Systemd

- Changer le niveau / la cible d'exécution par défaut :
  - Il suffit de faire un lien sur le fichier de default.target :
    - #ln -sf /lib/systemd/system/multi-user.target /etc/systemd/system/default.target, pour passer en mode multi-utilisateur non graphique par défaut.
    - #ln -sf /lib/systemd/system/graphical.target /etc/systemd/system/default.target, pour passer en mode multi-utilisateur graphique par défaut.
- Lancer un service :
  - #systemctl start NOM.service
  - #systemctl stop NOM.service
  - != activer un service
- Lister les units :
  - #systemctl list-units

75

Systemd

- Activer un service :
  - #systemctl enable NOM.service
  - #systemctl disable NOM.service
- Vérifier l'état d'un service :
  - #systemctl is-enabled NOM.service
  - #systemctl status NOM.service
    - La ligne préfixée "Loaded" permet d'indiquer l'état de chargement du service:
      - Si le service est configuré, avec le mot clé "loaded", suivi du chemin du fichier de configuration.
      - Si le service est activé, avec le mot clé "enabled" en fin de ligne.
      - Si le service est déactivé, avec le mot clé "disabled" en fin de ligne.
      - Si le service n'est pas configuré ou s'il est en erreur, le mot clé "error" est présent, suivi de la raison.
    - Le ligne préfixée "Active" permet d'indiquer l'état d'exécution du service :
      - Si le service a été lancé avec la mot clé "active".
      - Si le service n'a pas été lancé avec le mot clé "inactive".
      - Une deuxième indication, donnée entre parenthèse, permet de préciser l'état du processus en cours pour le service. Avec "runnning" si ce processus est en cours d'exécution, "exited" si ce processus est terminé, et "dead" si ce processus est en erreur ou s'il n'a pas été lancé.

76

Systemd

- Créer son propre service :
  - On va construire un fichier montest.service, que l'on va mettre dans le répertoire de configuration systemd /etc/systemd/system/ : voir note complémentaire ou démo lors du cours !
- Journaux :
  - systemd possède son propre mécanisme de journalisation, appelé "The Journal".
  - #journalctl (voir man)
  - #journalctl /usr/sbin/dhcpcd
- Les fichier de configuration des units sont dans le dossier /lib/systemd/system

77

Les sessions

- Une fois le démarrage terminé, le système doit se rendre disponible aux utilisateurs
- getty
  - Cette commande affiche l'invite de connexion « login » sur les terminaux passif connecté par une liaison filaire
- Mingetty
  - Commande allégée de getty permettant à linux de gérer les consoles virtuelles (voir /etc/inittab)
  - 6 consoles virtuelles par défaut (ALT+F1 à F6)
  - pstree : dès que l'on se connecte mingetty s'arrête

78

Les sessions

- Login
  - Lorsque l'on fournit un nom à l'invite de connexion de mingetty, celui-ci exécute la commande login afin de valider/rejeter la demande de connexion
  - La connexion échoue si le fichier /etc/nologin existe
    - Sauf pour l'utilisateur root
    - Utile pour empêcher les connexions en cas de maintenance
- Shell
  - Une fois la connexion validée, le processus exécute la commande (généralement un shell) associée à l'utilisateur (voir /etc/passwd)
  - Le shell activé exécute les fichiers de configuration de l'utilisateur qui se connecte
  - Lorsque l'on quitte le shell, init exécute à nouveau mingetty

79

Fichiers associés à l'utilisateur

- Rôle
  - Ils permettent de définir un environnement de travail propre à chaque utilisateur
- /etc/profile
  - Fichier commun à tous les utilisateurs
  - Il permet de configurer le shell
  - Il définit les variables d'environnement globales et programmes de démarrage ($PATH, $LOGNAME, HISTSIZE…)
  - Ensuite il exécute les scripts placés dans /etc/profile.d/
    - Permet de paramétrer les couleurs de l'écran des consoles
    - Paramètres divers alias tel que ll=ls -l --color

80

Fichiers associés à l'utilisateur

- /etc/bashrc
  - Il contient les alias et fonctions systèmes comme le umask par défaut ou la variable PS1 qui définit le format du prompt par défaut
- /etc/skel/
  - Ce répertoire est utilisé par useradd pour s'assurer que tous les nouveaux utilisateurs commencent avec la même configuration de base
  - Les fichiers qu'il contient sont copiés dans le répertoire personnel de l'utilisateur

81

Fichiers de l'utilisateur

- ~/.bash_profile
  - Il contient les variables d'environnement et les programmes de lancement personnels. On peut aussi y personnaliser le prompt
- ~/.bashrc
  - Il contient les alias et fonctions personnels
- ~/.bash_logout
  - Il est lu par le shell quand un utilisateur se déconnecte du système
  - Il permet d'exécuter un processus à la fin d'une session (comme par exemple effacer l'écran)
- ~/.bash_history
  - Il contient l'historique des commandes exécutées

82

/etc/termcap

- /etc/termcap
  - Le fichier /etc/termcap est une base de données définissant les possibilités des différents terminaux et émulateurs de terminaux. Les programmes utilisent /etc/termcap pour accéder aux différentes caractéristiques des terminaux, comme la couleur et les graphismes
- /etc/terminfo
  - Alternative à termcap. Plus rapide car il existe un fichier par terminal.
  - /usr/share/terminfo/
- Principe :
  - Le programme qui a besoin d'utiliser un terminal doit trouver son nom dans la variable TERM afin de connaître les séquences de contrôle à utiliser.
  - C'est le programme lui-même qui choisi s'il va utiliser terminfo ou termcap

83

Journalisation du démarrage

- Journalisation
  - La plupart des systèmes Linux sauvegardent dans un journal certains ou la totalité des messages générés pendant le démarrage
  - L'enregistrement des événements systèmes est géré par deux programmes : *klogd* et *syslogd*
  - klogd est simplement à l'écoute des messages en provenance du noyau et les envoie à syslog
- Syslogd
  - /etc/syslog.conf répertorie dans quel fichier doit être enregistré chaque message (généralement /var/log/messages)
  - dmesg : affiche et contrôle le tampon des messages du noyau

84

Personnalisation

- Ajouter des commandes
  - Il est plus facile de les ajouter à la fin du processus de démarrage, mais ce n'est pas toujours possible
  - Il est préférable d'isoler les modifications des fichiers d'initialisation standards afin de
    - les tester plus facilement
    - diminuer les risques d'erreurs lors des mises à jour du système
  - En pratique, on crée de nouveaux scripts que l'on place dans le répertoire adéquat (/etc/rc.d/xxx)
- Modifier les scripts standards
  - Cette opération est déconseillée sauf s'il n'est pas possible de procéder autrement

85

Arrêt du système

- Arrêt
  - L'arrêt du système devra être réalisé dans certaines circonstances
    - pour des opérations de maintenance
    - pour établir des diagnostics
    - pour mettre en place un nouveau noyau
  - Serveur
    - Dans le cas d'un serveur, il est probable que plusieurs utilisateurs utilisent un service, il faudra donc veiller à avertir tous les utilisateurs de l'arrêt du système
  - Outils
    - Le système doit être arrêté en utilisant les outils adéquats sans quoi il y a des risques d'altération de fichiers et de perte de données
    - shutdown (-k = simulation)

86

Arrêt du système

- Procédure d'arrêt
  1. Run service <initscript> stop
     - Arrêt de tous les services
  2. Run kill -SIGTERM
     - Arrêt de tous les processus
  3. Pause pendant 5 secondes
  4. Run kill -SIGKILL
     - Les processus restant sont tués brutalement
- Sync
  - Pour garantir l'intégrité des disques, la commande sync termine le processus d'arrêt du système en forçant l'écriture des blocs modifiés et en mettant à jour le super-bloc
  - Sync ne doit pas être utilisé après un fsck manuel sinon les mauvais super-blocs stockés dans les zones tampon du noyau seront réécrits et annuleront les corrections de fsck

87

Commandes associées

- chkconfig
  - Permet d'automatiser le lancement et l'arrêt des services utilisateurs en configurant les scripts des répertoires /etc/rc.d
- ntsysv
  - Similaire à chkconfig mais sous une autre interface
- /etc/rc.d/init.d/crond start|stop
- who –r
  - Affiche le niveau de fonctionnement courant et la date à laquelle il a été lancé
- telinit
  - Permet de changer le niveau de fonctionnement
  - Il faut lui fournir en argument le n° du nouveau niveau de fonctionnement
- dmesg

88

En mode graphique

- Au niveau d'exécution 5,
  - /etc/inittab exécute un script appelé /etc/X11/prefdm
  - Le script prefdm exécute le gestionnaire d'affichage X préféré en fonction de ce qui est contenu dans le répertoire /etc/sysconfig/desktop
    - gdm si vous utilisez GNOME
    - kdm si vous utilisez KDE
  - Ensuite, l'écran de connexion est affiché

89

Mise en réseau sous Linux

Module 13

90

Identification des périphériques réseau

- Interfaces
  - Loopback interface (lo)
    - Interface réseau virtuelle utilisée pour faire communiquer la machine avec elle-même
      - Elle est nécessaire au fonctionnement de certaines fonctionnalités réseau internes
    - IP : 127.0.0.1
  - Ethernet (eth)
    - eth*x*
    - eth*x.y*
      - Aliasing d'interface : utiliser une carte réseau avec plusieurs adresses IP différentes. Ne prend pas en charge DHCP
  - Autres interfaces
    - ppp : point à point
    - wlan : Wi-Fi
    - tr : Token Ring
    - …

91

Configuration du réseau

- Configuration des interfaces
  - /sbin/ifconfig
    - Utilisée pour effectuer des opérations « volatiles » de configuration des interfaces
  - #nmtui
    - Commande permettant de configurer les interfaces réseaux
  - Exemples

    # ifconfig affiche les caractéristiques des interfaces actives

    # ifconfig eth0 [up |down] active/désactive une interface

    # ifconfig eth0 192.168.3.1 broadcast 192.168.3.255 netmask 255.255.255.0

    # ifconfig eth0:1 192.168.1.12 netmask 255.255.255.0
  - DHCP
    - /sbin/dhclient demande une adresse ip au dhcp

92

Configuration du réseau

- La commande ethtool
  - # ethtool -s eth1 speed 100 duplex full autoneg off
  - Via le fichier ifcg-eth

```
#
# File: /etc/sysconfig/network-scripts/ifcfg-eth0
#
DEVICE=eth0
IPADDR=192.168.1.100
NETMASK=255.255.255.0
BOOTPROTO=static
ONBOOT=yes
ETHTOOL_OPTS="speed 100 duplex full autoneg off"
```

93

Configuration du réseau

- La commande ip
  - # ip route
  - # ip address
  - # ip link

```
[root@demo ~]# ip address show
2: eth0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast qlen 1000
    link/ether 00:0c:29:8a:c9:fb brd ff:ff:ff:ff:ff:ff
    inet 192.168.226.128/24 brd 192.168.226.255 scope global eth0
    inet6 fe80::20c:29ff:fe8a:c9fb/64 scope link
       valid_lft forever preferred_lft forever
[root@demo ~]# ip route show
192.168.226.0/24 dev eth0  proto kernel  scope link  src 192.168.226.128
10.1.0.0/16 dev eth0  proto kernel  scope link  src 10.1.70.198
169.254.0.0/16 dev eth0  scope link
default via 192.168.226.2 dev eth0
```

Configuration du réseau

- Autres outils de configuration
  - Outils CentOS, webmin, netconfig, nmtui, …
- Le service réseau
  - Service network start|restart|load|reload
    - démarre/arrête une interface réseau selon les paramètres spécifiés dans /etc/sysconfig/network-scripts
    - Les modifications réalisées par ifconfig sont perdues au redémarrage réseau

95

Routage

- Tables de routage
  - Commandes ***route***
    - Manipule seulement la table de routage principale
  - Commande ***ip route***
    - Permet de manipuler aussi les autres tables (NAT, …)
- Exemples
  - Configuration d'une passerelle par défaut

  # route add default gw 10.1.254.254
  - Afficher la table de routage

  # route -n

```
Kernel IP routing table
Destination     Gateway         Genmask         Flags Metric Ref    Use Iface
10.1.0.0        0.0.0.0         255.255.0.0     U     0      0        0 eth0
169.254.0.0     0.0.0.0         255.255.0.0     U     0      0        0 eth0
0.0.0.0         10.1.254.254    0.0.0.0         UG    0      0        0 eth0
```

Les fichiers de configuration

- Fichier /etc/hosts
  - Assurer la résolution de noms
- Caractéristiques
  - Configuration statique
    - Contient une liste d'association nom de machine/adresse IP
    - A réaliser sur toutes les machines du réseau → Lourdeur administrative
  - /etc/host.conf
    - /etc/hosts est généralement interrogé en premier, il permet donc de bypasser la résolution de nom classique (qui interroge les serveurs DNS)

```
##
# Host Database
#
# localhost is used to configure the loopback interface
# when the system is booting.  Do not change this entry.
##
127.0.0.1       localhost
255.255.255.255 broadcasthost
::1             localhost
192.168.22.240  isabranch.carbonwind.net
```

Source : http://www.carbonwind.net/ISA/MacOSXVPNL2TP/MacOSXVPNL2TP1.htm

97

Les fichiers de configuration

- Commande hostname
  - Affiche ou change le nom de l'ordinateur
- Le répertoire /etc/sysconfig
  - Les fichiers s'y trouvant sont utilisés pour passer des informations de configuration à certains programmes qui démarrent
- /etc/sysconfig/network
  - Rôle
    - Permet de fixer le nom de la machine et quelques autres paramètres réseau
  - Exemple

HOSTNAME=localdomain.localhost → FQDN
NETWORKING=yes → le réseau doit être configuré
FORWARD_IPV4=yes → transfert de paquets
GATEWAY=10.1.254.254 → @ de passerelle

98

Les fichiers de configuration

- /etc/sysconfig/network-scripts/ifcfg-ethxxx
  - Le répertoire network-scripts
    - Contient les fichiers de configuration de toutes les cartes réseaux de la machine
    - Ces scripts sont lus au démarrage du service réseau
    - Ils sont utilisés par l'outil d'administration réseau (system-config-network) → Attention aux modifications manuelles
  - Exemples de fichier ifcfg-eth0

```
DEVICE=eth0
BOOTPROTO=none
ONBOOT=yes
NETWORK=10.0.1.0
NETMASK=255.255.255.0
IPADDR=10.0.1.27
```

```
DEVICE=eth0
BOOTPROTO=dhcp
ONBOOT=yes
```

99

Les fichiers de configuration

- /etc/resolv.conf
  - Rôle
    - Fichier utilisé pour la résolution de nom
    - Contient le nom de domaine, l'adresse du (ou des) serveur de nom à contacter
  - Exemple

```
# vim /etc/resolv.conf
search isims.be
nameserver 10.1.0.2
nameserver 8.8.8.8
```

100

Les fichiers de configuration

- /etc/nsswitch.conf
  - name server switch
    - Configuration des recherches relatives aux diverses bases de données
    - Liste les DB ainsi que les moyens d'obtenir l'information demandée

```
passwd:      files
shadow:      files
group:       files

#hosts:     db files nisplus nis dns
hosts:       files dns

# Example - obey only what nisplus tells us...
#services:   nisplus [NOTFOUND=return] files
```

Utilitaires de vérification

```
# ifconfig
eth0      Link encap:Ethernet  HWaddr 00:80:C8:F8:4A:51
          inet addr:192.168.99.35  Bcast:192.168.99.255  Mask:255.255.255.0
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:190312 errors:0 dropped:0 overruns:0 frame:0
          TX packets:86955 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:100
          RX bytes:30701229 (29.2 Mb)  TX bytes:7878951 (7.5 Mb)
          Interrupt:9 Base address:0x5000
```

1. Type d'encapsulation et adresse MAC
2. Adressage IP de l'interface
3. Etats + MTU + Metric
4. Statistiques
   - RX (paquets reçus), TX (transmis), erreurs, suppressions, engorgements, collisions

102

Utilitaires de vérification

- ping
  - Demande un ICMP ECHO_REQUEST de la part de la machine cible
  - Une absence de réponse n'est pas significative
  - Envoi ininterrompu par défaut
- traceroute
  - Permet de suivre le chemin d'un paquet de données IP
    - Utilisé pour résoudre des problèmes de routage au niveau d'un réseau
    - Utilisé pour déterminer la topologie d'un réseau
- tcptraceroute
  - Implémentation de traceroute qui utilise des paquets TCP au lieu de ICMP ou UDP

103

Utilitaires de vérification

- tcpdump
  - tcpdump scrute tout le trafic passant sur une interface donnée (sniffer)
  - Exemple

  # tcpdump -n -f -i eth0
    - -f affiche les IP
    - -n conversion d'adresse en nom (utilisé pour éviter les DNS lookup)

104

Utilitaires de vérification

- netstat
  - Affiche les connexions réseau, les tables de routage et des statistiques sur les interfaces
  - Exemples
    - netstat -nr #affiche les routes connues

```
[root@localhost ~]# netstat -nr
Table de routage IP du noyau
Destination     Passerelle      Genmask         Indic   MSS Fenêtre irtt Iface
192.168.226.0   0.0.0.0         255.255.255.0   U         0 0          0 eth0
169.254.0.0     0.0.0.0         255.255.0.0     U         0 0          0 eth0
0.0.0.0         192.168.226.2   0.0.0.0         UG        0 0          0 eth0
```

105

Utilitaires de vérification

- netstat -i #affiche les statistiques

```
[root@localhost ~]# netstat -i
Table d'interfaces noyau
Iface       MTU Met    RX-OK RX-ERR RX-DRP RX-OVR    TX-OK TX-ERR TX-DRP TX-OVR
Flg
eth0       1500   0       44      0      0      0       44      0      0      0
BMRU
lo        16436   0       14      0      0      0       14      0      0      0
LRU
```

- netstat -ta #affiche les connexions actives

```
tcp        0      0 servertr2tr1:ftp            servertr2tr1:59540          ESTABLISHE
tcp        0      0 servertr2tr1:59540          servertr2tr1:ftp            ESTABLISHE
```

106

Identification des périphériques réseau

- Lister les périphériques
  - Commandes lspci, lsusb ou lspcmcia
  - Exemple :

```
# lspci
00:09.0 Ethernet controller: Realtek Semiconductor Co., Ltd.
  RTL-8139/8139C/8139C+ (rev 10)
```

- Kudzu
  - Kudzu est un utilitaire qui détecte le matériel présent et le compare à la liste de /etc/sysconfig/hwconf afin de déterminer si un nouveau matériel a été ajouté au système

107

Identification des périphériques réseau

- Pilotes
  - Les pilotes (drivers) sont chargés dynamiquement
    - Généralement, ils sont chargés en mémoire sous forme de modules lors de l'initialisation du système d'exploitation
  - Vérifier que ces pilotes logiciels ont bien été chargés

```
# dmesg
dmesg | grep 8139
8139too Fast Ethernet driver 0.9.27
eth0: RealTek RTL8139 at 0xd090af00, …
eth0:  Identified 8139 chip type 'RTL-8100B/8139D'
# lsmod
8139too       28993      0
```

108

Identification des périphériques réseau

- Commande ethtool

```
[root@test ~]# ethtool -i eth0
driver: pcnet32
version: 1.32
firmware-version:
bus-info: 0000:02:00.0
```

109

Commandes diverses

- La commande host
  - Recherche la correspondance entre un nom de machine et son adresse IP (et vice versa)
- La commande arp
  - Permet de manipuler le cache arp
- Activer le transfert de paquet
  - echo “1” > /proc/sys/net/ipv4/ip_forward

110

NFS

Module 14
Network File System

111

NFS

- NFS
  - Network File System : Système de fichiers en réseau
- Objectif
  - Permettre à des hôtes distants de monter des systèmes de fichiers (répertoires, partitions) et les utiliser comme s'il s'agissait de systèmes de fichiers locaux
- Intérêt
  - Centraliser les systèmes de fichiers et leur administration
  - Facilité d'utilisation : le client n'a pas besoin de connaitre les détails réseau

112

Versions de NFS

- Version 1 et 2
  - Connexion sans statut entre client et serveur
  - Trafic réseau faible
  - Versions non sécurisée
- Version 3
  - Plus de fonctionnalités
    - taille variable d'indicateurs de fichier, rapports d'erreur avancés, …
  - Gestion de la sécurité élémentaire
  - Le système est sans état (stateless) et ne permet pas la reprise sur incidents

113

Versions de NFS

- Version 4 : Le code est réécrit
  - Une gestion totale de la sécurité
    - Négociation du niveau de sécurité entre le client et le serveur
    - Authentification forte (supporte Kerberos)
    - Les listes de contrôle d'accès (ACL)
  - Systèmes de maintenance simplifiés
    - Réplication possible d'un serveur A sur un serveur B
    - Migration : le serveur NFS est migré de la machine A vers la machine B de manière transparente pour le client
  - Reprise sur incidents
    - La gestion de la reprise sur incident est intégrée du côté client et du coté serveur
  - Compatibilité
    - NFSv4 supporte Unix et MS-Windows

114

Principe RPC

- Basé sur les RPC
  - Remote Procedure Call
    - Appels de procédure à distance
      - Mise œuvre client-serveur qui permet de faire coopérer des processus répartis sur une ou plusieurs machines
      - → faire exécuter une procédure à distance avec échange de données et renvoi de résultats
    - Chaque service est représenté par un numéro
      - indiqué dans le fichier /etc/rpc
  - Avantages
    - Permet de s'affranchir des couches sous-jacentes, à savoir de la programmation des sockets et du système d'exploitation

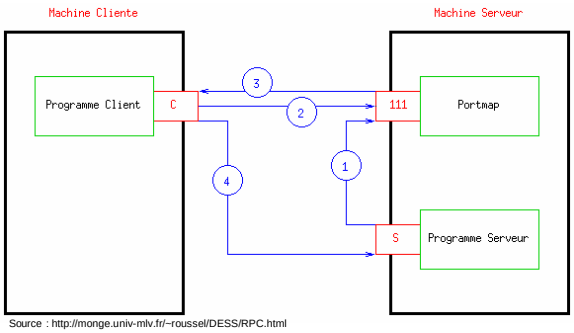
115

Principe RPC

- L'appel à une fonction consiste en l'envoi au serveur
  - du numéro de programme
    - 100003, pour NFS
  - du numéro de procédure à exécuter
    - 6 pour lire, 8 pour écrire,…
  - des arguments de la fonction
    - Regroupés dans une structure qui sera manipulée par le protocole XDR
- Portmap
  - Un client distant ne connaît pas
    - Les services RPC qui fonctionnent sur le serveur
    - Sur quel port ces services sont installés
  - Le démon portmap
    - Permet de mapper les demandes RPC aux bons services en faisant la correspondance entre le numéro de service RPC et celui du port TCP ou UDP associé.

116

Principe RPC



Source : http://monge.univ-mlv.fr/~roussel/DESS/RPC.html

1. Au démarrage, le programme serveur (NFS par exemple) s'enregistre auprès du serveur RPC (démon portmap ou rpcbind)
   - Il annonce le n° de port qu'il contrôle et les n° de programmes RPC qu'ils servent
2. Le client demande au serveur RPC le n° de port du service auquel il veut accéder
3. Réponse du serveur RPC qui communique le n° de port
4. Le client peut envoyer sa requête au serveur

Principe RPC

- Fichier /etc/rpc
  - Il contient la liste des numéros RPC des services (qui ne sont pas nécessairement opérationnel)

Nom	Référence	Alias
portmapper	100000	portmap sunrpc rpcbind
rstatd	100001	rstat rup perfmeter rstat_svc
rusersd	100002	rusers
nfs	100003	nfsprog
ypserv	100004	ypprog
mountd	100005	mount showmount

Principe RPC

- XDR (eXternal Data Representation)
  - Environnement hétérogène
    - Il faut envoyer des données et recevoir des résultats compréhensibles entre systèmes différents
    - Les données échangées sont donc représentées dans le format XDR indépendant des applications
  - Environnement vraiment hétérogène
    - XDR ne résout pas le problème lorsqu'il s'agit de faire collaborer des systèmes tout à fait différents comme Windows et Unix

119

Principe RPC

- Inconvénient des RPC
  - Bien qu'inactifs, tous les services RPC sont lancés au démarrage du système
    - → possible de les créer dynamiquement via (x)inetd lorsqu'un client requiert un service
  - Introduction difficile de nouvelles fonctions
    - Elle nécessite une programmation complexe de bas niveau donc délicate pour fonctionner entre systèmes différents
  - Manque de souplesse de XDR
    - Préférence aux langages de description textuel tel que XML
  - Socket
    - La programmation des sockets est indispensable pour les RPC
- → Nouveau protocole : XML-RPC

120

Référence au modèle OSI

- Couche 7 : Application → NFS
- Couche 6 : Présentation → XDR
- Couche 5 : Session → RPC
- Couche 4 : Transport → TCP/UDP
- Couche 3 : Réseau → IP
- Couche 1 et 2 : Ethernet et UTP/RJ45

121

UDP vs TCP

- UDP
  - Connexion plus rapide en LAN
    - Comme les RPC sont conçus pour fonctionner, à priori, dans un même réseau local, l'emploi d'UDP permet une communication plus rapide puisqu'il n'y a pas à établir de connexion
    - UDP → Connexion réseau sans état (stateless) entre le client et le serveur
  - Problèmes en dehors du LAN
    - UDP peut poser des problèmes lorsqu'on doit franchir des passerelles
- TCP
  - Il est nécessaire pour NFSv4
  - TCP est le protocole de transport par défaut pour NFS sous Red Hat Enterprise Linux

122

Les services NFS

- nfs
  - Service qui lance les processus RPC appropriés pour répondre aux requêtes pour les systèmes de fichiers NFS partagés
- nfslock
  - Un service facultatif qui lance les processus RPC appropriés pour permettre aux clients NFS de verrouiller des fichiers sur le serveur
- portmap
  - Le service RPC pour Linux
    - Il répond aux requêtes pour des services RPC et définit des connexions vers le service RPC
  - Il n'est pas utilisé avec NFSv4

123

Processus RPC

- rpc.nfsd
  - Processus serveur NFS qui travaille avec le noyau Linux pour, par exemple, fournir des threads afin de répondre aux requêtes des clients NFS
- rpc.mountd
  - Reçoit les demandes de montage du client NFS
  - Vérifie si les requêtes client correspondent à un système de fichiers actuellement exporté
  - Non utilisé avec NFSv4
- rpc.rquotad
  - Serveur RPC qui fournit des informations de quotas utilisateur pour les utilisateurs distants

124

Processus RPC

- rpc.statd
  - Processus qui avertit les clients NFS lorsqu'un serveur est redémarré sans avoir été préalablement arrêté correctement
  - Lancé automatiquement par le service nfslock
  - Non utilisé avec NFSv4
- rpc.lockd
  - Processus RPC qui correspond au service nfslock
  - Utile uniquement pour les anciens noyaux qui ne permettaient pas le verrouillage de fichiers NFS
  - Non utilisé avec NFSv4

125

Configuration de NFS

- Trois méthodes
  - Outils graphique de configuration
    - system-config-nfs
    - Webmin,…
  - En modifiant manuellement le fichier
    - /etc/exports
  - En exécutant la commande /usr/sbin/exportfs

126

Configuration de NFS

- /etc/exports
  - Spécifie quels fichiers doivent être exportés vers quels hôtes, et avec quelles options
  - Si plusieurs hôtes sont visés, ils doivent être séparés par des caractères d'espacement
  - Une ligne par fichier ou système de fichiers exporté
  - Structure :

Répertoire	hôte1(option1,option2)	hôte2(options)
/home	10.1.70.199(ro)	PC12(rw)

127

Nommer les hôtes

- Un hôte spécifique peut être renseigné avec
  - un nom de domaine,
  - un nom d'hôte,
  - une adresse IP.
- Les caractères génériques
  - * ou ? peuvent être utilisé
  - Attention : *.domaine.be et *.*.domaine.be donnent des résultats différents.
  - A ne pas utiliser avec les adresses IP
- Les réseaux et sous réseaux
  - 192.168.0.0
  - 192.168.0.0/28

128

Options de /etc/exports

- ro
  - Accès en lecture seule
- rw
  - Accès en lecture-écriture
- sync
  - Toute modification de fichier doit être engagée sur le disque avant que la demande d'écriture du client ne soit terminée. Cela peut ralentir les performances mais évite la perte de données.
- async
  - Permet au serveur d'écrire des données sur le disque lorsqu'il le juge opportun
- …
  - Voir man exports

129

Exporter les FS

- Exporter un système de fichiers
  - Signifie rendre le système de fichiers accessible à distance
- /var/lib/nfs/xtab
  - Les systèmes de fichiers devant être exportés sont écrits dans /var/lib/nfs/xtab
- rpc.mountd
  - Lit ce fichier pour accorder les privilèges d'accès à un système de fichiers

130

Exporter les FS

- La commande exportfs
  - Permet d'exporter ou d'annuler de façon sélective l'exportation des répertoires sans relancer les différents services NFS
  - Lit /etc/exports et écrit les FS exportés dans /var/lib/nfs/xtab
    → donne à rpc.mountd et rpc.nfsd les informations nécessaires afin d'accorder le montage à distance d'un système de fichiers à un hôte autorisé
- Exemples d'options de exportfs

  Sans options, exportfs affiche la liste des systèmes de fichiers actuellement exportés

  -r : actualise la liste d'exportation en relisant /rtc/exports

  -o *options* : permet de spécifier les répertoires devant être exportés qui ne sont pas inclus dans la liste de /etc/exports (pour test)

  -a : tous les répertoires sont exportés (ou non, selon les autres options transmises à exportfs)

131

Configurer le client NFS

- Utiliser la commande mount
  - Exemple

    $ mount –t nfs 10.1.70.199:/rép /mnt/nfs

    $ mount –t nfs4 10.1.70.199:/rép /mnt/nfs
- Configurer /etc/fstab
  - Le fichier /etc/fstab
    - est lu par le script /etc/rc.d/init.d/netfs au démarrage du système
    - ralenti le démarrage si le FS distant n'est pas disponible
    - utilise des ressources même si aucun client ne souhaite monter le FS partagé

132

Configurer le client NFS

- Utiliser le service autofs
  - Le paquetage autofs doit être installé
    - Le service est contrôlé par le script /etc/init.d/autofs et le fichier /etc/auto.master
    - L'éxécutable est /usr/sbin/automount
  - Principe
    - Avec autofs, tout programme ou utilisateur qui entrerait dans un répertoire qui est assigné à un périphérique provoque l'attachement du périphérique à ce répertoire (montage)
    - Ensuite, autofs les démontent automatiquement après une période d'inactivité

    → Permet de limiter les ressources employées
  - Peut être intéressant si
    - Le FS distant est peu ou pas utilisé
    - Beaucoup de FS distants sont montés

133

Configurer le client NFS

- Configuration de autofs
  - Editer le fichier /etc/auto.master
    - /pts-montage-général/ /fich-map options

```
/testnfs        /etc/auto.demonfs        --ghost,--timeout=60
```

  - Editer le fichier /fich-map
    - Mot-clé -options-de-montage localisation

```
tmp     -noexec,nosuid,ro,soft,ghost     192.168.226.128:/tmp
```

  - Divers
    - Vérifier l'activation du service
      - service autofs status
    - Créer les répertoire et les partages

134

Exemples d'options de montage

- noexec
  - N'autorise pas l'exécution de binaires sur le système de fichiers monté
- nosuid
  - Empêche l'utilisation des bits set-user-identifier ou set-group-identifier
- rsize=8192 et wsize=8192
  - Permet d'accélérer les communications NFS en lecture (rsize) et écriture (wsize) en paramétrant la taille des bloc de données à transférer
- nfsvers=2 ou nfsvers=3
  - Spécifie la version du protocole NFS à utiliser
- …
  - Voir man mount

135

Statuts des RPC

- La commande rpcinfo
  - Affiche des informations sur les service basé sur RPC
    - N° de port
    - N° de programme RPC
    - Version et son type de protocole (TCP ou UDP)
  - Permet de vérifier que les services rpc sont correctement démarrés
    - Exemple avec rpcinfo -p

```
program no_version protocole  no_port
 100000    2   tcp    111  portmapper
 100000    2   udp    111  portmapper
 100011    1   udp    957  rquotad
 100011    2   udp    957  rquotad
 100011    1   tcp    960  rquotad
 100011    2   tcp    960  rquotad
 100003    2   udp   2049  nfs
 100003    3   udp   2049  nfs
 100003    4   udp   2049  nfs
```

136

Sécurité

- Authentification
  - Lors du montage
    - NFS n'effectue d'authentification que lorsqu'un système client tente de monter une ressource NFS partagée
  - Sans Kerberos
    - Alors l'authentification s'effectue au niveau de l'hôte, pas du compte utilisateur
    - L'accès des utilisateurs peut être contrôlé par l'intermédiaire des permissions accordées aux fichiers et répertoires
  - TCP Wrapper (enveloppeurs)
    - Les enveloppeurs permettent de restreindre et tracer les accès à certains services (voir (x)inetd)
    - Pour déterminer si l'accès est autorisé, ils lisent les fichiers /etc/hosts.allow et /etc/hosts.deny
    - Une fois accepté, NFS se réfère à /etc/exports

137

Sécurité

- ACL
  - Par défaut, les listes de contrôle d'accès sont prises en charge par NFS
  - Option "no_acl" pour désactiver cette fonction lors de l'export du système de fichiers
- Privilèges
  - Le serveur utilise /etc/exports afin de connaître les privilèges de l'hôte concernant les FS disponibles
  - Les privilèges locaux des fichiers sont aussi vérifiés lors de l'accès au système de fichiers

138

Sécurité

- Privilèges
  - Les privilèges de montage NFS sont accordés de façon spécifique à un hôte et non à un utilisateur
    - → Attention aux permissions en rw
    - → Définition d'un utilisateur/groupe spécifique
  - La protection pour les fichiers partagés =
    - Leurs permissions d'accès
    - Leur appartenance à un utilisateur et un groupe
      - →Attention aux UID identiques d'un ordinateur à l'autre
      - →Utiliser les options anonuid et anongid

139

Sécurité

- root squashing
  - Tout accès client au système de fichiers exporté, exécuté en tant que super-utilisateur sur l'ordinateur client, se fait sous le compte "nobody"

  → option root_squash ou all_squash
- Attention aux caractères spéciaux et à la syntaxe
  - mon.domaine.be(rw) et mon.domaine.be (rw)
  - *.domaine.be et *.*.domaine.be
- Coté client
  - Utiliser l'option nosuid pour empêcher l'utilisateur root du serveur de se "connecter" en root sur le client

140

Résumé

- Démarrer le serveur
  - Activer le service systemctl enable nfs.service
- Configurer le serveur
  - Fichier /etc/exports
  - Commande exportfs
- Configurer le client
  - mount
  - /etc/fstab
  - autofs

141

Introduction au serveur Samba

Module 15

142

Bref historique

- IBM et Sytec développent NetBIOS
  - NetBIOS (Network Basic Input Output System)
    - Logiciel chargé en mémoire de manière à fournir une interface entre les programmes et le matériel réseau.
    - Adopte un schéma d'adressage utilisant des noms à 16 octets pour identifier les stations de travail et les applications réseau.
- Microsoft
  - SMB (Server Message Block)
    - Microsoft adapte le DOS pour permettre aux E/S des disques d'être redirigées vers l'interface NetBIOS.
      → L'espace disque devient partageable sur le réseau.
    - Ce protocole de partage de fichiers est appelé SMB puis deviendra CIFS.

143

Bref historique

- NetBIOS et TCP/IP
  - NetBIOS a été élaboré pour les LANs isolés.
  - TCP/IP peut être routé entre divers réseaux interconnectés (internetworks).
  - Conversion nom-adresse
    - La conversion des noms à 16 octets de NetBIOS en adresses IP permet que les messages puissent trouver leur chemin dans un interréseau IP routé.
- Microsoft améliore SMB
  - Le Browsing
    - Permet de voir les machines du réseau, annoncer des services
  - Le service d'authentification et d'autorisation centralisées
    - Connu sous le nom de Windows NT Domain Control.

144

Présentation de Samba

- Objectifs de Samba
  - Permet de transformer une machine Linux en serveur de fichiers et d'imprimantes pour les clients utilisant SMB/CIFS.
- Démons smbd
  - Démon qui fournit les services de partages de fichiers et d'imprimantes pour les clients Windows.
  - Responsable
    - de l'authentification des utilisateurs,
    - du verrouillage des ressources,
    - du partage des données par le biais du protocole SMB/CIFS.
  - Ecoute du trafic SMB sur les ports TCP 139 et 445

145

Présentation de Samba

- Démons nmbd
  - Responsable de la résolution de noms et de la navigation (browsing).
  - Par défaut le serveur attend du tracfic NMB sur le port UDP 137
  - Résolution de noms par serveur NBNS
    - NBNS = NetBIOS Name Service
    - WINS = Windows Internet Name Service. Nom de l'implémentation NBNS de Microsoft
    - Principe : contacter un serveur NBNS pour obtenir l'IP d'une machine. Le service n'est pas restreint au réseau local.
  - Résolution de nom par diffusion

146

Présentation de Samba

- LMB
  - Election d'un Local Master Browser parmi les machines participantes sur le segment.
  - Ce LMB conserve une liste des services disponibles (fichiers, imprimantes, …).
  - Cette liste est affichée dans le voisinage réseau.
- DMB
  - Des Domain Master Browsers coordonnent et propagent les listes de navigateurs des domaines.
  - Durées de synchronisation peuvent être longues.

147

Fichier /etc/smb.conf

- Section Global
  - Contient les variables globales qui affectent tous les partages
  - Ces variables sont décrites dans la « man page » smb.conf
- Exemples d'options globales
  - Workgroup
  - Security
  - Encrypt password
  - Smbpasswd file

148

Fichier smb.conf

- Section share
  - Contient les directives de configuration du partage
  - Les noms de partage sont indiqués entre crochets Par exemple :
    - [homes]
    - [printers]
    - [mon_repertoire]
  - La configuration des répertoires partagés se fait via des directives selon la syntaxe :

    directive=valeur

149

Exemple

- Exemple simple de partage

```
[tmp]
  Path = /tmp
  Read only = no
  Public = yes
```

- Man page

  $ man smb.conf

150

Types de serveurs Samba

- Serveur autonome (standalone)
  - Un tel serveur ne joue aucun rôle dans un domaine → groupe de travail
  - Sécurité
    - Au niveau du partage (security = share)
    - Au niveau de l'utilisateur (security = user)
  - Exemple

```
[global]
workgroup = DOCS
netbios name = DOCS_SRV
security = share
[data]
comment = Documentation Samba Server
path = /export
read only = Yes
guest only = Yes
```

Source : Red Hat Inc., RHEL4 – Guide de référence

151

Types de serveurs Samba

- Serveur membre d'un domaine
  - Semblable à un serveur autonome mais connecté à un contrôleur de domaine
    - donc soumis aux règles de sécurité de ce domaine
- Samba en contrôleur de domaine
  - tdbsam
    - Utilise la base de données de mot de passe nommée tdbsam (1 DC et moins de 250 users)
  - LDAP
    - Utilise la base de données LDAP
  - Samba ne peut pas
    - être un BDC d'un PDC Windows ou vice versa
    - fonctionner en tant que contrôleur de domaine Active Directory

152

L'option security

- Share
  - Pas d'authentification pour accéder au serveur.
  - Le client devra s'authentifier séparément pour chaque partage.
- User
  - L'utilisateur doit avant toute chose s'authentifier auprès du serveur afin de voir les partages disponibles.
- Server
  - Identique au mode user, mais l'authentification se fait via un autre serveur SMB.
- Domain
  - Permet à Samba de fonctionner dans un domaine en s'authentifiant via le contrôleur du domaine.
- ADS
  - Comme le mode Domain, permet à Samba de fonctionner dans un domaine mais le mode ADS permet aussi l'authentification Kerberos.

153

Mot de passe utilisateur

- Enregistrement des mots de passe
  - SMB requiert un fichier de mots de passe séparé de /etc/passwd
    - Usuellement /etc/samba/smbpasswd
  - Ajout d'un mot de passe samba

    # smbpasswd –a user1

154

Client samba

- Montage d'un FS samba sous Unix
  - mount -t smbfs -o username=youname,passwd=12345 //serveur/distant /mnt/smb
- Sous Windows
  - Accès par le voisinage réseau ou l'explorateur

155

Outils

- testparm
  - Teste le format du fichier smb.conf
- smbstatus
  - Liste les connexions en cours
- smbclient
  - Client smb en mode commande
- nmblookup
  - Client NetBIOS
- findsmb
  - Lister les machines visibles grâce aux requêtes smb de votre serveur
- smbpasswd
  - Crée ou modifie un mot de passe

156

Outils graphiques

- LinNeighborhood
  - Client graphique sous Linux pour avoir accès aux autres machines « windows » du réseau
- Swat
  - Samba Web Administration Tool.
  - Outil graphique d'administration de samba à distance
- Webmin
  - Outils d'administration qui permet de gérer de multiples serveurs via une interface Web

157

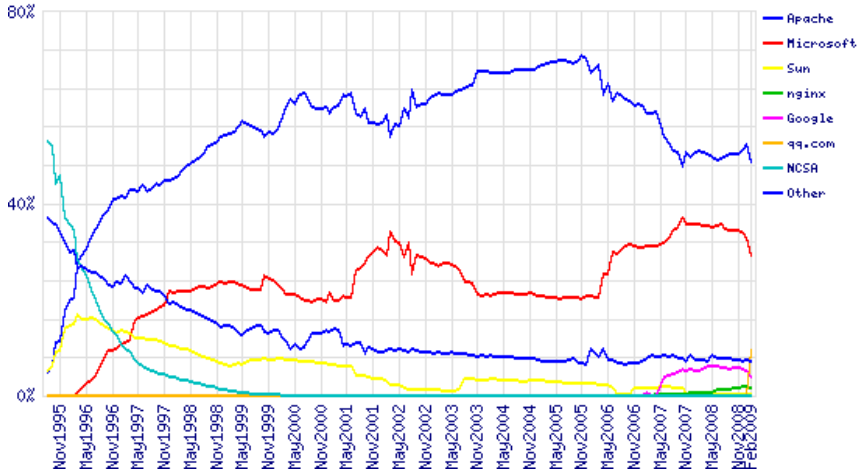
Serveur Web

Module 16

Introduction à HTTP Server (Apache)

158

Marché des serveurs Web



Source : http://news.netcraft.com/

159

Introduction

- Apache HTTP Server
  - Version actuelle : 2.2.11
  - Serveur Web
    - IIS, Sun Java system Web server, Boa…
  - Clients Web :
    - Firefox, IE, Opera, lynx,…
  - LAMP, WAMP, MAMP
- Composants
  - Apache = les éléments de base du serveur
  - Il est complété par une série de modules statiques et/ou dynamiques

160

Les modules

- But des modules
  - Ajouter des fonctionnalités au serveur
  - Exemple
    - Pour utiliser des connexions sécurisées il faut utiliser le module SSL (mod_SSL)
- Commandes

  # httpd –l (affiche les modules intégrés au serveur)

  # Apxs : permet d'ajouter des fonctionnalités sans devoir recompiler apache

161

http

- HTTP
  - HyperText Transfert Protocol
  - Protocole utilisé pour la communication entre le client et le serveur Web
  - Léger, rapide et adapté au transfert d'information multimédia (MIME)
- Ports
  - http écoute sur le port 80
  - https écoute sur le port 443

162

Les codes de statut http

- A chaque requête d'un client Web, le serveur Web retourne à ce client un nombre comportant 3 chiffres
- 1xx : Codes d'information
  - 100 : Attente de la suite de la requête
- 2xx : Codes de succès
  - 200 Ok : La requête HTTP a été traitée avec succès

163

Les codes de statut http

- 3xx : Codes de redirection
  - 301 : Moved Permanently. La ressource demandée possède une nouvelle adresse
- 4xx : Code d'erreur client
  - 403 : forbidden. Le serveur HTTP a compris la requête, mais refuse de la traiter
  - 404 page not found. Le serveur n'a rien trouvé qui corresponde à l'adresse (URL) demandée
- 5xx : Code d'erreur serveur
  - 500 : Erreur interne. Le serveur HTTP a rencontré une condition inattendue qui l'a empêché de traiter la requête

164

Exemple d'une page d'erreur

The page cannot be found

The page you are looking for might have been removed, had its name changed, or is temporarily unavailable.

Please try the following:

- Make sure that the Web site address displayed in the address bar of your browser is spelled and formatted correctly.
- If you reached this page by clicking a link, contact the Web site administrator to alert them that the link is incorrectly formatted.
- Click the Back button to try another link.

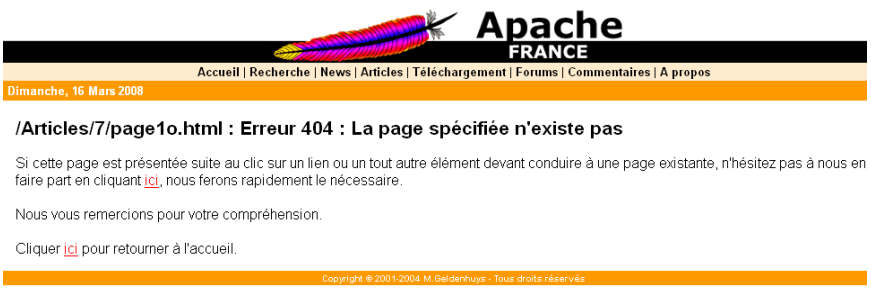
HTTP Error 404 - File or directory not found.
Internet Information Services (IIS)

Technical Information (for support personnel)

- Go to Microsoft Product Support Services and perform a title search for the words **HTTP** and **404**.
- Open **IIS Help**, which is accessible in IIS Manager (inetmgr), and search for topics titled **Web Site Setup**, **Common Administrative Tasks**, and **About Custom Error Messages**.

165

Exemple de personnalisation d'une page d'erreur



166

Autre exemple de personnalisation d'une page d'erreur

Objet non trouvé!

L'URL requise n'a pu etre trouvée sur ce serveur. Si vous avez tapé l'URL à la main, veuillez vérifier l'orthographe et réessayer.

Si vous pensez qu'il s'agit d'une erreur du serveur, veuillez contacter le gestionnaire du site.

Error 404

www.mediamarkt.be
Mon Mar 2 09:41:52 2009
Apache/2.0.59 (Linux/SuSE)

167

Processus et démons

- httpd
  - Démon principal exécuté via le compte root
- Spare server
  - Plusieurs processus fils sont démarrés afin de pouvoir répondre rapidement aux multiples requêtes des clients
  - Généralement ces processus sont démarrés par un compte spécifique (compte logique apache)

168

Configuration

- Fichier /etc/httpd/conf/httpd.conf
  - Configuration de directive
  - Exemple
    - *DocumentRoot* : Indique l'emplacement de l'arborescence du site Web
    - *ServerRoot* : Indique l'emplacement du répertoire de base du serveur apache
    - *PidFile* : Indique le fichier dans lequel est stocké le PID du démon httpd

169

Configuration

- La directive Directory
  - Permet de regrouper des directives utilisées par un répertoire spécifique
  - Ces directives sont insérées entre des balises
    <Directory /chemin_complet> et </Directory>
  - Les directives Directory ne peuvent pas être imbriquées

170

Configuration

- La directive Options
  - La directive Options contrôle quelles fonctions du serveur sont disponibles dans un répertoire particulier
  - Exemples d'arguments
    - ExecCGI : L'exécution des scripts CGI est autorisée
    - MultiViews : Permet de proposer plusieurs versions d'une page en fonction des préférences du client

171

Fichiers htaccess

- But
  - Permet de décentraliser la configuration pour adapter l'accès en fonction du répertoire
  - Une utilisation abusive de fichiers htaccess peut ralentir le système
  - L'usage des fichiers htaccess est contrôlée par la directive AllowOverride
    - *All* : toutes les directives possibles sont autorisées
    - *AuthConfig* : permet d'utiliser les directives liées à une procédure d'authentification

172

Contrôle d'accès

- Allow/Deny
  - Permet de préciser quels sont les hôtes qui peuvent/ne peuvent pas accéder à une ressource donnée
- Syntaxe
  - Deny,allow :
    - autorisé par défaut si l'hôte n'est pas dans deny
  - Allow,deny :
    - refusé par défaut si l'hôte n'est pas dans allow

173

Hôtes virtuels

- But
  - Permettre au serveur de gérer simultanément plusieurs arborescences Web
- Types
  - Hôtes virtuels basés sur l'adresse IP
    - Utilise plusieurs cartes réseaux (ou des alias)
  - Hôtes virtuels basés sur le numéro de port
    - Associations de différents ports d'une même adresse à des arborescences Web différentes
  - Hôtes virtuels basés sur le nom
    - Associations de différents noms DNS à des arborescences différentes mais à une seule adresse IP

174

Log

- Directives associées aux historiques
  - LogFormat
    - Définit le format d'un fichier de log
  - CustomLog
    - Permet de choisir le fichier dans lequel seront enregistrés les logs
  - ErrorLog
    - Permet de choisir le fichier dans lequel seront enregistrés les messages d'erreur
  - LogLevel
    - Permet de choisir le niveau de verbosité des messages d'erreur

175

Partitionnement

- Une partition pour les données
  - Par défaut il d'agit de /var/www
  - Intéressante pour mettre la partition en RAID et pour les sauvegardes
- Partitions en lecture seule
  - Selon les cas, les partitions /, /etc et /usr pourront être montées en lecture seule

176

Outils

- apachectl
  - Script qui permet le contrôle du démon httpd

  # apachectl start | stop | configtest …
- benchmark
  - Outil permettant de tester les performances du serveur web
  - Exemples : Ab (Apache benchmark), siege
- awstats
  - Outil fournissant des statistiques sur l'utilisation de son serveur apache

177

Introduction au shell BASH et scripting shell

Module 17

178

Introduction

- Shell
  - Interpréteur de commandes
    - Programme qui se charge de lire et d'exécuter les commandes dans l'environnement utilisateur
  - Environnement de programmation
    - Permet de définir des variables, des fonctions, des instructions complexes et des programmes complets
    - Il dispose de tous les mécanismes nécessaires pour une programmation structurée
      - Des variables pour le stockage des données
      - Des opérations pour modifier le contenu des variables
      - Des structures de contrôle du déroulement du programme

179

Introduction

- Shell en mode texte
  - Archaïque mais puissant
    - Les interfaces graphiques ne permettent pas toujours de réaliser toutes les tâches réalisables avec un shell
      - L'inverse est vrai également
  - Rapide
    - Pour toute une série d'opérations, il peut se révéler plus rapide qu'une interface graphique
  - Exemples de shell en mode texte
    - sh, ksh, csh, bash, zsh, ...
- Shell interactif
  - L'utilisateur dialogue avec le système, il saisit les commandes et le shell les exécute et affiche les résultats

180

Introduction

- Les scripts shell
  - Se sont des fichiers "texte" qui contiennent des instructions exécutables
  - Un script peut en appeler un autre, y compris lui même (script récursif)
  - – Utilisations
    - Ils permettent d'automatiser certaines tâches en démarrant séquentiellement plusieurs commandes
    - Ils ne conviennent pas pour des applications critiques ni des accès non séquentiels

181

BASH

- Bourne Again Shell
  - – Evolution du Bourne Shell (sh)
    - Prompt configurable
    - Autocomplétion de commande (touche tabulation)
    - Conforme aux normes POSIX
    - …
- Langage de programmation

182

Redirections

- Objectif
  - Transférer les données provenant d'un flux vers les données d'un autre flux
- Descripteur de fichier
  - Numéro utilisé par les programmes pour identifier les fichiers ouverts
    - 0 : flux d'entrée standard (stdin) ex :clavier
    - 1 : flux de sortie standard (stdout) ex : écran
    - 2 : flux d'erreur standard (stderr) ex : écran
  - Héritage
    - Les descripteurs de fichiers d'un processus sont généralement hérités par tous ses processus fils

183

Redirections

- Redirections de données en sortie
  - Intérêt
    - Permet d'enregistrer dans un fichier, les données écrites par un processus dans un de ses descripteurs de fichier
    - Utile pour garder une trace de certaines opérations ou à des fins d'analyse
  - Exemples
    - Rediriger la sortie d'un processus vers un fichier
      - $ ls > fichier
      - Le contenu précédent de "fichier" sera effacé
    - Pour ajouter les informations en fin de fichier
      - $ ls >> fichier
      - Ne supprime pas le contenu existant
    - Pour rediriger la sortie d'erreur
      - $ ls 2> fichier
      - $ ls &> fichier ou $ ls >& fichier

184

Redirections

- Redirections de données en entrée
  - Injecter des données provenant d'un fichier dans un descripteur de fichier d'un processus
    - $ n < fichier
    - $ sort –n < fichier
  - Injecter des données provenant d'un autre descripteur de fichier dans un descripteur de fichier
    - $ n < &s

185

Les pipes anonymes

- Les pipes
  - Intérêt
    - Un pipe permet d'injecter le résultat d'une commande dans le flux d'entrée standard d'une autre commande, sans passer par un fichier intermédiaire
      - L'opérateur | représente un tuyau canalisant les données issues d'une commande vers le flux d'entrée standard de la commande suivante
  - Exemple
    - $ ps | grep bash
  - Priorité
    - La redirection dans un pipe s'effectue après les autres types de redirections vues précédemment (priorité!)
  - Anonyme
    - Ces pipes sont dits anonymes, car ils sont créés directement par le shell pour une commande donnée

186

Les pipes nommés

- Pipes créés manuellement
  - Création : mkfifo
  - Suppression : rm
  - Cela permet de donner un nom aux pipes et de les utiliser a posteriori dans plusieurs commandes
  - Le système de fichiers doit être capable de les gérer

187

Divers

- Commandes multiples du shell
  - On utilise l'opérateur de concaténation " ; "
  - Exemple : $ cd destination ; ls
  - L'opérateur | est prioritaire sur l'opérateur ;
- La commande tee
  - Elle permet de dupliquer un flux de données. Elle est couramment utilisée en conjonction avec les pipes
- Raccourcis clavier
  - Flèches
    - Rappeler de l'historique d'une commande
    - ~/.bash_history
  - Commande stty
    - Suspendre : CTRL + Z
    - Interruption : CTRL + C
    - ...

188

Les variables d'environnement

- Les variables
  - Ces variables contiennent une valeur de type chaîne de caractères, dont la signification est propre à chaque variable
  - Le premier caractère ne doit pas être un chiffre
  - Variables d'environnement
    - Les variables d'environnement sont écrites complètement en majuscules, mais ce n'est pas une obligation
    - La liste des variables d'environnement peut être affichée à l'aide de la commande env

      $ env

189

Les variables d'environnement

- Gestion des variables
  - Définir une variable (affectation)
    - On utilise simplement le signe =
    - VARIABLE=valeur ou VARIABLE=chaîne
  - Récupérer la valeur d'une variable
    - # echo $VARIABLE
    - # echo ${VARIABLE}
  - Détruire une variable
    - # unset VARIABLE

190

Les variables d'environnement

- Environnement
  - L'environnement est un ensemble de variables d'environnement qui permettent de modifier le comportement d'un programme
  - Multiples environnement
    - Linux permet de définir un environnement d'exécution pour chaque programme en cours d'exécution
    - Le shell a deux environnements
      - Son propre environnement
        - » variables d'environnement normales + ses propres variables
      - L'environnement d'exécution
        - » l'environnement que le shell utilise pour exécuter des programmes
  - Héritage
    - Par défaut, les programmes sont lancés avec un environnement proche du programme qui les lance

191

Les variables d'environnement

- Les variables définies dans l'environnement
  - Elles ne font partie que de l'environnement (du shell), pas des programmes lancés par le shell
  - Exporter des variables
    - Pour les rendre accessibles aux programmes lancés par le shell, il faut les exporter dans l'environnement d'exécution
      - Pour exporter de manière permanente les variables du shell
        - » $ export variable
      - export permet aussi d'ajouter un argument à une variable
        - » $ export PATH=$PATH:/tmp
    - Pour définir des variables d'environnement uniquement pour l'environnement d'exécution d'une seule commande
      - $ variable=valeur commande

192

Les variables d'environnement

- La commande "env" liste les variabes d'environnement du contexte actuel
  - $BASH représente le chemin de l'exécutable de bash lui-même
  - $BASH_VERSION représente la version du bash
  - $EDITOR représente l'éditeur par défaut
  - $HOME représente le home directory de l'utilisateur
  - $HOSTNAME représente le nom de la machine
  - $OLDPWD représente le répertoire dans lequel l'utilisateur se trouvait précédemment
  - $PATH représente le chemin où le shell peut rechercher des commandes exécutables
  - $PPID représente le numéro du processus parent
  - $PS1 représente le prompt affiché à l'écran
  - $PWD représente le répertoire courant
  - …

193

Les variables d'environnement

- Variables spéciales
  - Pour recevoir des arguments, un script utilise les variables spéciales $1 jusque $9
  - La variable $0 correspond au nom du script
  - Si plus de 10 arguments doivent être utilisé, on utilise la commande shift

194

Opérations sur les variables

- Opérations
  - +, -, *, /
  - Modulo : %
  - Test d'égalité : ==, !=
  - Comparaisons : >, <, <=, >=,
- Exécution de commande en séquence &&, ||
  - command1 && command2
    - command2 n'est exécutée que si la première commande s'est exécutée correctement
  - command1 || command2
    - command2 n'est exécutée que si command1 ne s'est pas exécutée
  - Expressions à mettre entre parenthèse si l'on ignore les niveaux de priorités

195

Opérations sur les variables

- Test d'une condition (commande test)
  - Test sur un fichier
    - $ test –r fichier && echo "fichier accessible en lecture"
  - Test sur une chaîne de caractères
    - $ test "$a" = yes
  - Test numérique
    - $ test "$nombre" –gt 5

196

Le caractère d'échappement \

- Caractères spéciaux
  - Un certain nombre de caractères sont interprétés par le shell d'une manière spéciale.
    - C'est le cas des caractères génériques * et ?
    - Ces motifs sont remplacés par autant de résultats correspondant que possible (on parle d'expressions globales ou de wildcards)
    - Pour restreindre les possibilités on utilise les plages de caractères (entre crochets)
      - $ ls [a-p]*
      - $ ls [a-d,w-z].txt
  - Pour ne pas qu'ils soient interprétés, on les précède du caractère antislash \
    - hello="Hello world \!"
    - L'antislash permet également, lorsqu'il est placé en fin de ligne, de supprimer le saut de ligne qui le suit
      - Utilisé par exemple pour répartir une commande trop longue sur plusieurs lignes

197

Les substitutions

- Substitutions
  - Le shell permet d'effectuer des substitutions d'expressions par leur valeur
    - pour spécifier des motifs de chaîne de caractères,
    - pour récupérer la valeur des variables du shell,
    - pour calculer des expressions mathématiques,
    - …
  - Substitution du nom d'utilisateur
    - Le caractère tilde ~ est remplacé par le nom de l'utilisateur courant, ou, à défaut de nom, par le chemin sur le répertoire personnel de cet utilisateur
    - $ cp fichier ~pierre

198

Les substitutions

- Récupération de variables
  - Des variables du shell et des variables d'environnement peuvent être récupérées en préfixant le nom de la variable par le caractère dollars
    - $ echo $PATH
    - $ echo $USER
- Substitution du résultat d'une commande
  - $ kill `cat /var/run/httpd.pid`

- Répertoire . et ..
  - Le shell ignore systématiquement les répertoires . et .. dans les substitutions
  - Dans l'exemple qui suit, . et .. ne sont pas supprimés
    - $ rm –f *

Les chaînes de caractères

- Guillemets simple
  - Substitution
    - Ils ne permettent pas de bénéficier des fonctionnalités de substitutions du shell, comme par exemple le remplacement d'une variable par sa valeur dans la chaîne de caractères
    - Ils évitent d'utiliser \ trop souvent
  - Guillemet simple : ‘
    - Le guillemet simple sert à définir des chaînes de caractères
      - La chaîne de caractère est mise entre guillemet simple: 'chaîne'
    - La chaîne elle-même ne peut pas contenir de guillemet, même précédé d'un caractère d'échappement

Les chaînes de caractères

- Guillemet à droite
  - ` : guillemet inverse (ou à droite)
  - Sert à exécuter une commande et à en inclure le résultat dans une autre commande
- Guillemets doubles
  - Les caractères spéciaux $, ` et \ conservent leur signification initiale
  - Les guillememets doubles conservent donc les substitutions
- Exemples
  - $ echo 'Bonjour $USER'
  - $ echo "Bonjour $USER"
  - $ kill `cat /var/run/httpd.pid`

201

Structures de contrôle

- if, then, else, fi
- Case, esac
- For, do, done
- While, do, done
- Until, do, done
- …

202

Valeur de retour

- Valeur de retour
  - En se terminant, chaque processus linux envoie une valeur de retour
    - 0 si OK
    - Une autre valeur que 0 indique une erreur
  - Afficher la dernière valeur de retour
    - $ echo $?

203

Administration de réseau

Module 18

SNMP

204

Introduction

- Une fois configuré, il faut surveiller le réseau
  - l'utilisation de la bande passante,
  - l'état de fonctionnement des liens,
  - les éventuels goulets d'étranglement,
  - les problèmes de câblage,
  - le bon cheminement de l'information entre les machines,
  - etc.
- Points stratégiques à surveiller ?
  - Potentiellement tous les éléments constitutifs du réseau
    - Routeurs, les concentrateurs, les liens, les machines connectées au réseau (PC, imprimantes, …), …

205

Administration réseau

- Surveiller avec quel outil?
  - Complexité des réseaux
    - Les technologies utilisées dans le réseaux sont hétérogènes
      - Matériels différents : PC, switch, routeurs, imprimantes,..
      - OS différents : Unix, BSD, Windows, MAC, …
    - La quantité d'information à collecter et traiter est énorme
    - Ces informations sont réparties un peu partout dans le réseau
  - Architecture « de surveillance »
    - Nécessité d'une architecture permettant l'interopérabilité des équipements
    - Nécessité d'utiliser une application pour traiter les données de ces équipements

206

Administration réseau

- Surveiller avec quels outils?
  - Deux architectures de référence
    - SNMP
      - Simple Network Management Protocol
      - Utilisé pour gérer et surveiller à distance les équipements (principalement d'un LAN). Par exemple :
        - » Disposer d'une cartographie du réseau
        - » Fournir un inventaire précis de chaque machine
        - » Mesurer la consommation système d'une application
        - » Signaler des disfonctionnements
    - TMN
      - Telecommunication Management Network
      - Destiné à la gestion des réseaux d'opérateur (SDH, ATM,…)

207

Le protocole SNMP

- SNMP
  - Protocole client/serveur
    - Utilise la couche transport UDP (ports 161 et 162)
  - Il permet d'effectuer l'administration d'équipements réseau de manière centralisée
    - Supervision (Récupération de données)
      - Connaître l'état d'un appareil
      - Mesurer le trafic et les taux d'erreurs
      - Etre averti en cas d'événements exceptionnels
      - …
    - Contrôle (Action)
      - Modification de la configuration à distance
      - Action face à des événements exceptionnels
      - …

208

Le protocole SNMP

- Versions SNMP
  - Seules certaines versions sont supportées par le matériel
  - SNMPv1 : 1987
    - Sécurité basée uniquement sur une chaine de caractère nommée "community" fonctionnant comme un mot de passe
  - SNMPv2c : 1993
    - Amélioration de la sécurité mais aussi les opérations, les données,…
    - 4 formats différents : la version 2c est compatible avec SNMPv1
  - SNMPv3
    - Version plus sécurisée intégrant l'authentification, l'intégrité, la prévention du rejeu ainsi que la confidentialité
    - Permet d'introduire le protocole SNMP dans des réseaux étendus géographiquement

209

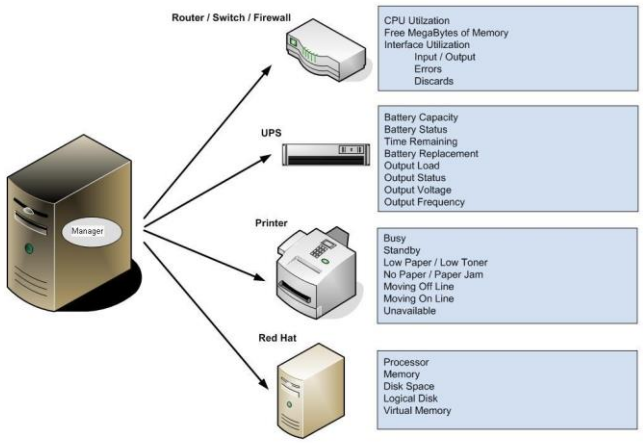
Principe SNMP

- Environnement de gestion SNMP
  - Base de données (MIB)
    - Chaque nœud administrable du réseau (switch, station,…) possède une base de données contenant des informations locales au nœud
  - La station de supervision (manager)
    - Exécute les applications de gestion qui contrôlent et questionnent les éléments réseaux
  - Agent SNMP
    - Un agent sur le noeud se charge de collecter les informations en réponse aux interrogations d'un manager

210

Principe SNMP

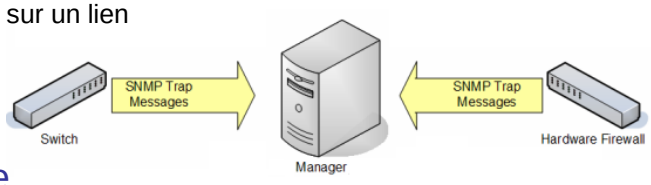
- Environnement de gestion SNMP



Source : http://www.finalitsolutions.com/images/SNMP%20Monitoring.jpg

Principe SNMP

- Environnement de gestion SNMP
  - Traps (alerte, notification)
    - Les alertes sont envoyées par l'agent quand un certain événement se produit
      - Par exemple la perte d'un lien, l'atteinte d'un seuil de trafic sur un lien



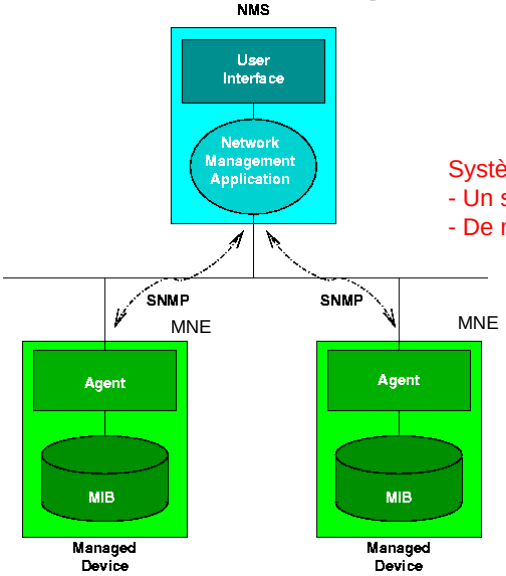
  - Protocole
    - Le transport des informations entre les agents et le manager se fait à l'aide du protocole SNMP

Nomenclature

- MNE
  - Managed Network Entity : Equipement géré en SNMP
- NMS
  - Network Management Station
  - Station contenant le logiciel de gestion SNMP (le manager SNMP)
- MIB
  - Management Information Base
  - Les informations (nom, versions, états,…) gérées en SNMP sont regroupées en MIB
- SMI
  - Structure of management Information
  - Système de représentation des objets de la MIB

213

Principe SNMP



Système client/serveur comprenant
- Un seul client (le manager)
- De multiples serveurs (les Agents)

214

Agent SNMP

- Agent SNMP
  - Logiciel contenu dans le MNE
    - Assure la collecte et l'envoi d'informations ainsi que le paramétrage de l'équipement
  - Il écoute sur le port UDP 161
    - Il répondra à une requête si elle est émise par une entité autorisée
- Dans quel matériel?
  - On trouve normalement des agents SNMP sur tout équipement dit administrable
    - Hub, switch, routeur, serveur, imprimante, AP, PC, UPS, …

215

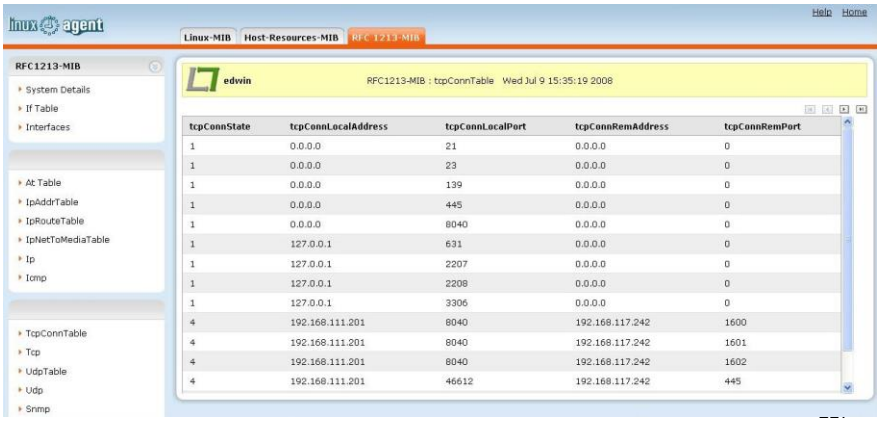
Les managers SNMP

- Manager
  - Logiciel gestionnaire avec lequel l'agent SNMP dialogue
  - Rôles
    - Questionner les agents et fournir à l'administrateur les informations récupérées
    - Gérer les traps reçus et prévenir l'administrateur
      - Il écoute les alarmes (traps) sur le port UDP 162
  - Le manager SNMP est au courant de l'état des agents par deux techniques
    - Le polling : le gestionnaire interroge les agents
    - Les traps : l'agent envoie de lui-même une alarme (trap) à l'administrateur lorsqu'un événement particulier arrive sur le réseau

216

Les managers SNMP

- Exemple d'affichage des connexions TCP sur une machine Linux distante



Source : http://www.webnms.com/linux-monitor/images/tcp_connection_details_ori.jpg

Les managers SNMP

- Exemple d'affichage de l'utilisation mémoire d'une machine Linux distante

Source : http://www.webnms.com/linux-monitor/images/memory_utilization_ori.jpg

SMI

- Structure of Management Information
  - Ensemble de règles utilisées pour la description d'objets manipulables par un protocole de gestion de réseau
  - Facilite la recherche d'informations
    - Structure définie pour s'y retrouver dans la foule d'informations proposées par chaque agent
      - Définition des structures des MIB
      - Méthode standard de définition (syntaxe, …) des objets des MIB
      - Méthode d'encodage standard des valeurs des objets
  - Chaque information de la MIB peut être retrouvée
    - soit à partir de son nom de variable
    - soit à partir d'une structure en arbre
      - Cela revient à parcourir des dossiers et sous-dossiers d'une arborescence

219

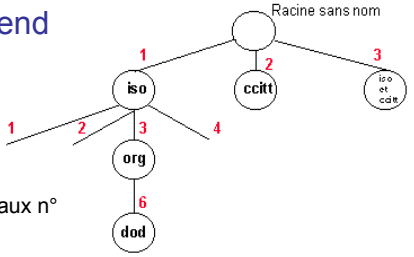
Bases de données SNMP

- MIB
  - Management Information base
    - Base de données contenant des informations de gestion
      - Ensemble d'objets gérables (Managed Objects) représentant divers types d'information
      - Décrits en ASN.1 (Abstract Syntax Notation 1)
      - Structurée par SMI
      - La MIB est maintenue par l'agent sur le matériel
  - Les objets gérables peuvent être
    - Statiques
      - Type de matériel, constructeur, version d'un équipement
    - Dynamiques
      - Etat à un instant donné (up, down, …)
    - Statistiques
      - Timer, compteurs, … (uptime, nombre d'octets sur un port, …)

220

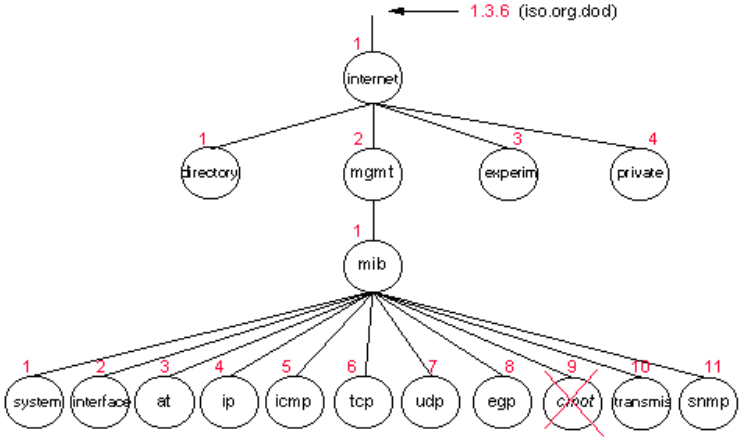
Description MIB

- Nommage
  - La MIB a une structure arborescente
    - Les informations des MIB sont nommées de manière unique suivant une notation hiérarchisée (SMI)
    - Chaque information est identifiée
      - par un nombre (OID : Object Identifier)
      - ou par un nom normalisé et hiérarchique (plus lisible)
  - La racine de l'arbre comprend trois branches
    - **iso.org.dod**
      - Le séparateur est un point
    - **1.3.6**
      - Suite d'entier correspondants aux n° des feuilles de l'arborescence



Description MIB

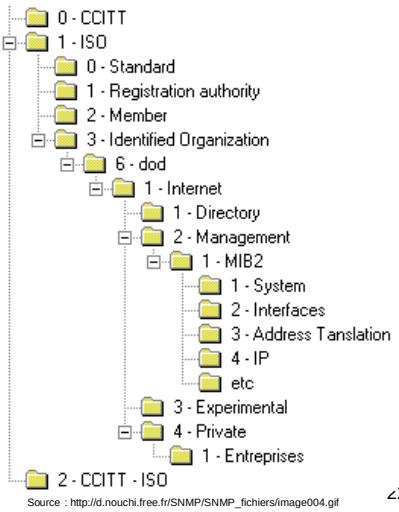
Préfixes des objets SNMP
(groupes)



222

Description MIB

- MIB Browser
  - Logiciel permettant d'explorer une arborescence MIB



Source : http://d.nouchi.free.fr/SNMP/SNMP_fichiers/image004.gif

223

Types de MIB

- MIB I et MIB II
  - Deux MIB publiques ont été normalisées: MIB I et MIB II
  - Les informations de ces MIB sont rangées dans des arborescences normalisée par l'ISO
  - MIB II
    - Evolution de la MIB I disposant d'objets supplémentaires
    - Elle contient les paramètres TCP/IP les plus courants
      →L'agent SNMP doit au moins supporter la MIB II
  - L'OID de la MIB2 commence par le préfixe
    - Iso(1).org(3).dod(6).internet(1).mgmt(2).mib(1).

224

Types de MIB

- Les "Private MIB"
  - Définies par les entreprises
    - Leur permet d'ajouter de nouvelles variables en fonctions des applications qu'elles veulent développer
  - Représentées en 1.3.6.1.4.1 dans la classification SMI
    - Exemples
      - Cisco : 1.3.6.1.4.1.9
      - IBM : 1.3.6.1.4.1.2
      - Microsoft : 1.3.6.1.4.1.311
      - RedHat : 1.3.6.1.4.1.2312
    - Liste :http://www.iana.org/assignments/enterprise-numbers

225

Types de MIB

- MIB RMON (Remote Monitoring)
  - Sonde RMON/RMON2
    - Logiciel installé sur les composants du réseau et collectant des données sur les interfaces réseau
      - Par exemple la quantité de trafic
    - Une sonde peut aussi être programmée pour déclencher une alarme (trap) si une anomalie survient
    - La norme actuelle est RMON2
    - HCRMON (High-Capacity RMON)
      - RFCs 2577 and 3273
      - Type de sonde utilisée pour les haut-débits full duplex
  - Types de sondes RMON
    - Equipements « dédiés » à cette fonction
      - Netscout, Sniffer distributed, OptiView® Network Analyzer
    - Equipement qui incluent des agents RMON
      - Certains commutateurs Ethernet
    - Logiciel à installer sur des O.S.
      - Sonde Network General, Sniffer Pro, RMONster

226

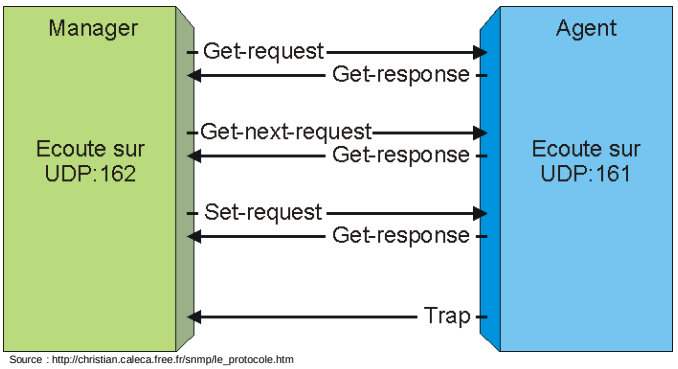
Polling et trap

- Polling
  - SNMP est basé sur le principe de scrutation (*Polling*)
  - Le client (station administration) questionne les différents agents ou serveurs
- Trap
  - Le protocole offre la possibilité aux agents de signaler au poste d'administration l'apparition de certains événements par l'envoi de trap
  - L'agent SNMP envoie de lui-même une alarme (trap) à l'administrateur lorsqu'un événement particulier arrive sur le réseau
  - A la réception d'un trap, le manager peut déclencher une action spécifique
  - Un trap n'est pas une variable de la MIB

227

Polling et trap

- Requêtes, réponses et traps



Source : http://christian.caleca.free.fr/snmp/le_protocole.htm

228

Messages SNMP

- SNMP autorise la consultation et la modification de variables via différentes requêtes
  - GetRequest
    - Le client demande une information à l'agent
  - GetNextRequest
    - Le client demande l'information suivante à l'agent
  - GetBulk
    - Permet de rechercher un ensemble de variables regroupées
  - SetRequest
    - Le client initialise une variable de l'agent

229

Messages SNMP

- Envois d'informations par l'agent
  - GetResponse
    - L'agent retourne l'information au client
  - Trap
    - Interruption - l'agent envoie une information à un client
- Format de trames SNMP

en-tête commune SNMP			en-tête à lire et à positionner			variables à lire et à positionner				
version (0)	communauté	Type PDU (0-3)	id. requête	statut d'erreur	index d'erreur	nom var.	valeur var.	nom var.	val var.	...

			en-tête de trap					variables		
version (0)	communauté	Type PDU (4)	entreprise	adresse agent	type de trap (0-7)	code spécifique	estampille horaire	nom var.	val var.	...

Source : http://www.htr.ups-tlse.fr/pedagogie/cours/admin/protos/snmp/snmp41.htm#S2

Messages SNMP

- Signification des champs
  - Version du protocole
    - Il correspond à la version SNMP moins 1 (valeur 0 pour la version 1 - RFC1157:
  - Nom de communauté
    - Semblable à un mot de passe, il permet de s'assurer de l'identité de l'expéditeur du message
  - Type de PDU (tag PDU)
    - GetRequest, GetNextRequest, SetRequest, …
  - Identificateur de la requête
    - Défini par le manager lors de l'envoi d'une requête et utilisé par l'agent dans sa réponse
    - Permet d'associer la réponse à la requête vu que le protocole transport utilisé est UDP

231

Messages SNMP

- Signification des champs
  - Le statut d'erreur
    - Est positionné par l'agent pour indiquer le code erreur associé à la requête d'une variable
    - Toujours à zéro dans une requête
  - L'index d'erreur
    - Indique la variable qui a provoqué l'erreur (uniquement dans les cas *noSuchName*, *badValue* et *readOnly*)
  - Variables (VarBindList)
    - Liste des noms de variables et de leurs valeurs
    - Lors d'une opération Get, les valeurs sont nulles

232

Messages SNMP

- Les types d'erreurs
  - 0 : noError (opération réalisée avec succès)
  - 1 : tooBig (variable trop grande pour être envoyée)
  - 2 : noSuchName
    - Variable inconnue de l'agent
    - Il n'y a plus de variable
    - La communauté spécifiée est incorrecte
  - 3 : badValue (type de données incorrect)
  - 4 : readOnly (variable en lecture seule)
  - 5 : genErr (autre erreur indiquant que la donnée ne peut être envoyée)
- Remarque
  - Attention, les messages d'erreurs ne sont pas toujours très clairs (signification multiple de NoSuchName)

233

Messages SNMP

- Signification des champs spécifiques aux Traps
  - Entreprise
    - Indique le type d'objet généré
  - Adresse de l'agent
    - Indique l'adresse IP de l'agent
  - Type de Trap
    - Positionné par l'agent, il indique l'évènement survenu
    - Dans le cas enterpriseSpecific (code spécifique à une entreprise) il faut regarder le champ qui suit (code spécifique)
  - Time stamp
    - L'estampille horaire indique le temps passé depuis l'initialisation de l'agent

234

Messages SNMP

- Type de trap

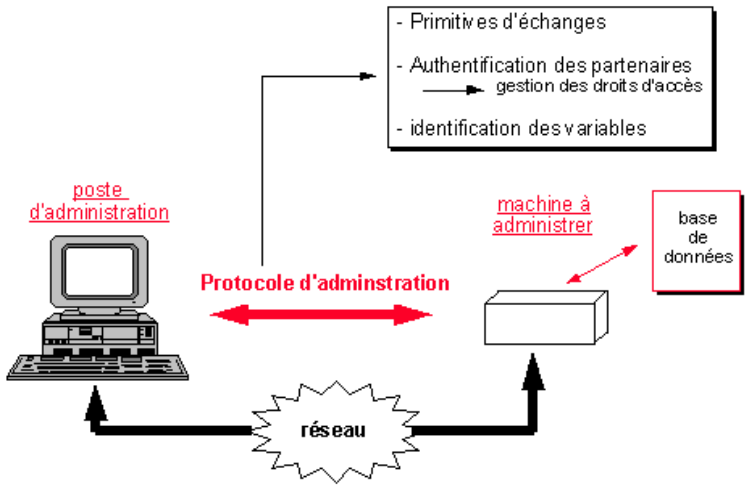
Type de trap	Nom	Description
0	coldStart	Initialisation de l'agent (démarrage)
1	warmStart	Réinitialisation de l'agent (reload)
2	linkDown	Passage de l'interface à l'état bas (première variable)
3	linkUp	Passage de l'interface à l'état haut (première variable)
4	authenticationFailure	Emission par le manager d'une communauté invalide *
5	egpNeighborLoss	Passage d'un homologue EGP à l'état bas (première variable indiquant l'@ IP de l'homologue) (Perte d'un voisin)
6	enterpriseSpecific	cf. champ spécifique pour avoir de l'information

Source : http://www.httr.ups-tlse.fr/pedagogie/cours/admin/protos/snmp/snmp41.htm#S2

235

* ou ne dispose pas des permissions

Mécanismes SNMP



236

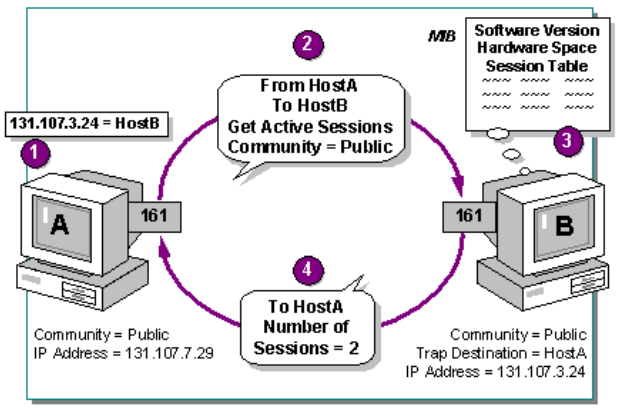
Les communautés

- Communauté
  - Domaines d'administration
    - Une communauté est un groupe auquel appartient les hôtes exécutant le service snmp
    - Les communautés servent donc à créer des domaines d'administration
    - Les communautés sont identifiée par un noms de communauté
  - Sécurité
    - Le nom de communauté est assimilée à un mot de passe
      - Les agent ne répondront qu'aux demandes en provenance de la même communauté
    - Par défaut la communauté est la chaine de caractères « PUBLIC »
    - Le concept est local à un agent
      - Un agent établit une communauté pour chaque combinaison d'authentification, de contrôle d'accès et de caractéristiques de proxies

237

Fonctionnement de SNMP

Principe de fonctionnement SNMP



Source : http://technet.microsoft.com/fr-fr/library/bb967438.aspx

238

Fonctionnement de SNMP

- Principe de fonctionnement SNMP
  - **Etape 1**
    - Un manager SNMP envoie une demande à un agent par le biais de son nom d'hôte (ou son adresse IP)
    - L'application transmet la demande au socket (port UDP) 161
    - Le nom d'hôte est converti en adresse IP par une des méthodes de résolution disponibles (fichier Hosts, DNS, WINS, …)
  - **Etape 2**
    - Formation d'un paquet SNMP contenant :
      - Une demande get, get-next ou set pour un ou plusieurs objets
      - Un nom de communauté ainsi que d'autres informations de validation
    - Le paquet est acheminé vers le socket (port UDP) 161 de l'agent

Source : http://technet.microsoft.com/fr-fr/library/bb967438.aspx

Fonctionnement de SNMP

- Principe de fonctionnement SNMP
  - **Etape 3**
    - L'agent SNMP reçoit le paquet
    - Le nom de communauté est vérifié
      - Si le nom de communauté n'est pas valide
        - » ou que le paquet n'est pas correctement formé, il est rejeté.
      - Si le nom de communauté est valide
        - » l'agent vérifie le nom d'hôte ou l'adresse IP source
        - » Si l'agent n'est pas autorisé à recevoir des paquets du système de gestion, le paquet est rejeté
    - La demande est transmise et l'agent prépare la réponse
  - **Etape 4**
    - Le paquet SNMP renfermant les informations demandées est renvoyé au gestionnaire SNMP

Source : http://technet.microsoft.com/fr-fr/library/bb967438.aspx

Les agents proxies

- Proxy
  - Un équipement non équipé d'un agent SNMP peut tout de même être géré grâce à un proxy qui utilise alors un agent de substitution
  - Cet agent doit donc être capable de travailler d'une part avec les protocoles de ces composants pour connaître les objets gérés et d'autre part dialoguer avec le manager (SNMP) pour les lui faire connaître

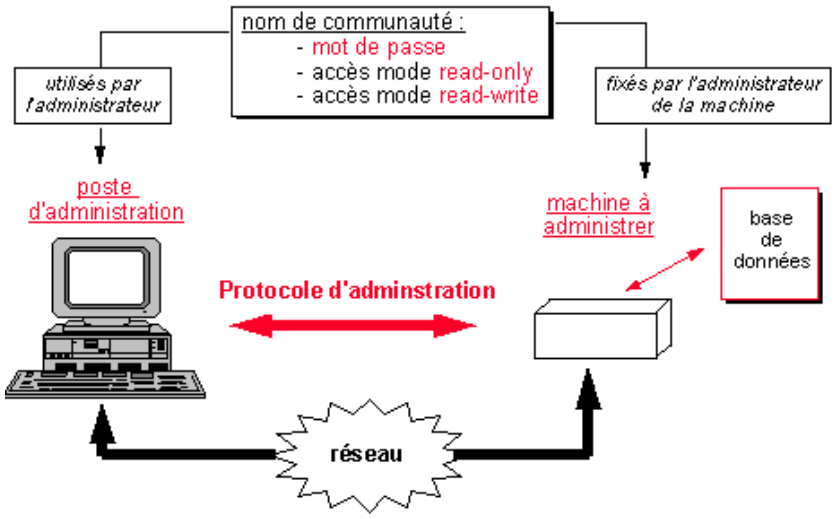
241

Sécurité SNMP

- Sécurité de base
  - Trois types d'accès sur les variables de la MIB
    - Pas d'accès
    - Accès en lecture seule: Read-Only
    - Accès en lecture et écriture : Read-Write
  - Nom de communauté
    - Circule en clair sur le réseau (SNMPv1)
    - Alternative
      - Limiter l'accès sur les MIBs en lecture seule
      - Se déplacer sur les équipements pour modifier certaines valeurs

242

Authentification



243

Sécurité SNMP

- SNMPv3
  - User Security Module
    - Authentification du message
      - Vérifie l'intégrité du message à l'aide d'une fonction de hachage
    - Cryptage
      - Empêche la lecture des informations contenues dans les paquets SNMPv3
    - Estampillage du temps
      - Empêche la réutilisation d'un paquet SNMPv3 déjà transmis

244

Plate-formes d'administration

- Plate-forme d'administration
  - Logiciel prévu pour administrer un réseau
    - Parfois nécessaire pour administrer du matériel propriétaire
    - Les fonctionnalités varient d'une plate-forme à l'autre
  - Exemples
    - Pour les systèmes distribués
      - HPOpenView de HP, Tivoli Netview d'IBM
    - Pour les réseaux locaux
      - Optivity de Nortel
      - Total Control de 3com
      - scotty (logiciel libre)
      - …

245

Plate-formes d'administration

- Exemples de fonctions
  - Récolte d'informations
    - Configuration des agents, paramétrages des agents et remontées d'alarmes
  - Cartographie du réseau
    - Découverte automatique des équipements du réseau
    - Outil graphique permettant de cartographier le réseau
  - Types de réseau
    - Supportent au moins Ethernet mais pas nécessairement d'autres réseaux
  - Programmation
    - Elles peuvent être programmées via des scripts
  - MIB
    - Support des MIB propriétaires

246

Simulateur de trap

- Le simulateur de Trap SNMP ou de Notification SNMP
  - Rôle
    - Permet de simuler des traps afin de tester les filtres de Trap
    - Permet de vérifier que les actions associées aux traps sont bien exécutées
  - Exemples
    - Webnms
    - SimpleAgentPRo (permet de simuler un réseau complet)
    - LoriotPro

247

NET-SNMP

- NET-SNMP
  - Ensemble d'outils et de bibliothèque SNMP
    - Snmpget, snmpgetnext, snmpset, …
  - Il supporte les versions 1, 2 et 3 de snmp
- Installation et activation de l'agent snmp
  - yum install net-snmp
  - yum install net-snmp-utils.i386
  - /etc/init.d/snmpd start
  - service snmpd start
  - Fichier de log
    - /var/log/snmpd.log
    - Options de verbosité : -a, -V, -d, -D

248

NET-SNMP

- Fichier de configuration de l'agent snmp
  - /etc/snmp/snmpd.conf
- Configuration minimale
  - Paramétrer les variables snmp du groupe system
  - Spécifier les communautés pour l'accès en lecture (rocommunity) et en écriture (rwcommunity)
  - La commande snmpconf permet de créer le fichier de configuration
- Commandes
  - Syntaxe
    - snmpxxx [option] agent [communauté] paramètres
    - snmpxxx = snmpget, snmptranslate, snmpwalk, …

249

Exemple

- Visualisation et modification des variables du point d'accès linksys WAP54G
  1. Installer net-snmp
  2. Activer le support snmp du point d'accès
     1. Voir l'onglet snmp
     2. Choisir enable
     3. Remplir les champs
        - Contact : admin
        - Device Name : aplinksys
        - Location : 217a
        - Snmp community : public en ro et private en rw

250

Exemple

- Lecture d'une variable
  - # snmpget -v1 -c public 192.168.1.245 system.sysContact.0
    Réponse : sysContact.0 admin
  - # snmpgetnext -v2c -c public 192.168.1.245 system.sysContact.0
    Réponse : sysName.0 aplinksys
- Capture de trame
  - Installer tcpdump

  # tcpdump udp port 161
    - On voit notre requête GetRequest ainsi que la réponse de type GetResponse
- Modification d'une variable
  - # snmpset -c private 192.168.1.245 system.sysContact.0 s admin@sysContact.0

251

Introduction à la sécurité

Module 19

252

Les menaces

- Virus
  - Programme caché dans le corps d'un fichier en apparence anodin
  - Lorsque ce fichier est ouvert/exécuté, le virus exécute automatiquement les actions que son auteur a programmées
- Vers (worms)
  - Virus qui se propagent généralement à travers un réseau. Ils s'autoreproduisent d'ordinateur à ordinateur
- Cheval de Troie (trojan)
  - Programme d'apparence légitime (souvent un petit jeu ou un utilitaire) qui comporte une routine nuisible exécutée sans l'autorisation de l'utilisateur
  - Ils sont souvent cachés dans des programmes d'apparence légitime
- Les backdoors (portes dérobées)
  - Programmes installés dans l'ordinateur à l'insu de l'utilisateur et qui donnent accès au contrôle de logiciels ou de l'ordinateur par des pirates

253

Les menaces

- Spyware (logiciels-espions)
  - Programme espion collectant, sans son autorisation, des données personnelles d'un utilisateur pour les envoyer à un tiers
    - Programme inclus dans un autre programme (le plus souvent un graticiel, un partagiciel ou un pilote de périphérique)
- Les adwares (publiciels)
  - Logiciel gratuit dont le créateur finance ses activités en affichant de la publicité lors de l'utilisation du logiciel
  - Il est malveillant si l'utilisateur n'est pas averti qu'il recevra de la publicité après avoir installé le logiciel
- Le keylogger
  - Logiciel espion capable d'enregistrer tout ce qui est tapé au clavier et qui le renvoie ensuite à un réseau de pirates
- Les exploits
  - Un exploit est un programme qui exploite une faille de sécurité informatique dans un système d'exploitation ou dans un logiciel afin de prendre le contrôle d'un ordinateur ou installer des malwares

254

Les menaces

- Les wabbits
  - Ils n'infectent pas les programmes ni les documents. Ils interviennent par exemple dans la saisie semi-automatique des navigateurs, en incorporant des termes censés amener l'internaute sur des sites payants
- Les rootkits
  - Ce sont des ensembles de programmes chargés de dissimuler l'activité nuisible d'un malware
- Les dialers (composeurs)
  - Un composeur branche l'ordinateur à un numéro de téléphone dont les frais d'utilisation sont très élevés
- Rogues/Scareware
  - Faux anti-spywares sont seulement créés pour faire de l'argent
  - Son principe est de convaincre un utilisateur que son ordinateur contient un logiciel malveillant, puis lui suggérer de télécharger un logiciel (payant) pour l'éliminer. Le virus est le plus souvent fictif et le logiciel téléchargé peut être inutile ou même malveillant

255

Les menaces

- Hoax (canulars)
  - Un hoax est une information fausse, périmée ou invérifiable propagée spontanément par les internautes
- Spam
  - Message intempestif envoyé à une personne ou à un groupe de personnes lors d'une opération de spamming
  - Supprimer sans lire et sans cliquer sur aucun lien (même désabonnement)
- Mailbombing
  - Consiste à envoyer des milliers de messages à un unique destinataire
  - Encombrement de boîte aux lettres, donc saturation de la capacité de stockage, voire crash machine ou déni de service
- Phishing (hameçonnage)
  - Courrier électronique semblant provenir d'une entreprise de confiance, typiquement une banque ou un site de commerce électronique alors qu'il provient d'escrocs usurpant l'identité d'une entreprise
- …

256

Les menaces

- Les sinistres
  - Usure des appareils provoquant des pannes
    - Alimentation, disque dur,…
  - Catastrophes naturelles
    - Incendie, inondation, …
  - Erreurs humaines
    - Effacer un fichier, un mail, renverser du liquide sur le matériel,…
  - Vol ou pertes
    - Clés USB, portables oubliés dans le train, …

257

Menaces principales

- Principales sources de menaces
  - Les utilisateurs du système
    - La **grande majorité** des problèmes liés à la sécurité d'un système d'information provient de l'utilisateur, que ce soit volontairement ou non
  - Les personnes malveillantes
    - Tentant de s'introduire sur le système et d'accéder à des données ou à des programmes auxquels elles ne sont pas censées avoir accès
  - Les programmes malveillants (Malware)
    - Logiciels destinés à nuire ou à abuser des ressources du système (offrent un accès au système, modifiant ou collectant des données, …)
  - Les sinistres (Pannes, pertes, vol, incendie, dégât des eaux)
    - Une mauvaise manipulation ou une malveillance entraînant une perte de matériel et/ou de données

258

Etudes et rapports

- Quelques études et cas répertoriés
  - Selon l'éditeur de solutions de sécurité Webroot
    - Fin 2005, 72% des ordinateurs connectés à Internet étaient infectés par au moins un logiciel espion
    - En 2006, le taux de machines infectées par les logiciels espions aurait atteint 89%
      - Etude sur 19.480 machines réparties à travers 71 pays dans le monde
    - En 2007, 268,5 millions d'euros ont été détournés par un système de fraude à la carte bleue sur Internet

259

Exemples d'attaques

- Les Botnets
  - Botnet Mariposa
    - Réseau de machines zombies qui regrouperait plus de 12 millions d'ordinateurs infectés
  - Wiseguy Tickets Inc
    - Cette société est accusée d'avoir, à l'aide d'un botnet, acheté frauduleusement 1,5 millions de tickets de concerts, dans le but de les revendre par la suite au marché noir
  - Iphone
    - Décembre 2009, l'institut SRI International Malware Threat Center a trouvé un réseau d'iPhone zombies dont l'objectif était de voler les données
- Scareware
  - Le cheval de Troie Win32/Renos se propageant via les faux logiciels de sécurité fin 2008, aurait infecté 4,4 millions d'ordinateurs distincts

260

Exemples d'attaques

- Tout est visé
  - Bluetooth
    - Le virus Cabir se propage par la téléphonie mobile Bluetooth (Symbian OS)
  - P2P
    - Le virus informatique MyDoom.A se propage par les courriels et le service P2P
  - Mail
    - Infection via les images et les pièces jointes
  - Facebook
    - Un message provenant de Facebook incite à voir une vidéo sur Youtube, en fait un faux site Youtube
    - Il propose alors une mise jour du lecteur flash → installation d'un cheval de troie
  - Jeux en ligne
    - Les jeux en ligne sont une des cibles privilégiées des keylogger
  - Clavier sans fil
    - Possible de sniffer la communication avec le PC
  - …

261

Exemples d'attaques

- USB
  - L'USB est devenu le cauchemar en terme de sécurité
    - Axé sur la convivialité, la connectivité, le faible coût et les performances
    - La sécurité des connexions USB est secondaire!
  - Exemple
    - *conficker*, qui exploite principalement une faille de svchost.exe
    - Il peut se propager via des périphériques qui déclenchent un AutoRun
    - En tentant de connaître les mots de passe il peut verrouiller des comptes utilisateurs (sur le serveur!!)
    - Il a été capable d'entrer dans les réseaux confidentiels de la Défense, depuis les périphériques USB y sont interdit

262

USB

- Le cauchemar de l'USB
  - Propagation des programmes malveillants
    - Très facile, dès qu'un périphérique est connecté son code peut s'exécuter
    - Il peuvent se copier sur tous les lecteurs raccordés aux machines infectées
  - Perte de données
    - Le format des clé USB les rendent très facile à perdre ou à voler
  - Piratage anonyme
    - Permet de disposer d'un laboratoire de piratage complet sur un périphérique USB. Il suffit de connecter la clé sur un PC sans surveillance puis de repartir sans laisser de trace

263

Exemples d'attaques

- Sapphire (Slammer)
  - Ver informatique (376 octets) se propageant grâce à une faille sur les serveurs Microsoft SQL Server
    - Après seulement dix minutes, le ver contaminait 12.000 serveurs, 130.000 une heure et demie plus tard
    - La chute de disponibilité du web mondial a atteint 15% (selon les données de Matrix.net)
    - Il a mis temporairement hors d'usage cinq des treize serveurs DNS "racines" de la planète
    - Idem pour les serveurs de nombreuses sociétés telles que Bank of America, HP, Skynet OVH, …
  - Et le meilleur pour la fin…
    - Le correctif à cette faille avait été publié 6 mois, jour pour jour, avant la diffusion du ver

264

Source : http://www.journaldunet.com/solutions/0301/030128_ver_sapphire.shtml

Conséquences possibles

- Ces problèmes peuvent notamment mener à
  - La perte de données
  - La divulgation/revente d'informations
  - L'impossibilité d'accéder à des services
    - Un programme ne fonctionne plus
    - Un service ne fonctionne plus ou est inaccessible
- Les conséquences peuvent être graves
  - Perte de performances ou détérioration du matériel
    - Mémoire, stockage, bande passante, …
  - Perte de productivité
    - Manque de ressources, hoax, …
  - Licenciement
  - Faillite?
    - Vol de documents, fichiers clients, …

265

Sécuriser

- Sécuriser, mais quoi?
  - Les accès (contrôle des accès, des connexions)
  - Les données (accès, intégrité)
  - Les services (accès, intégrité)
- Sécuriser selon quels critères
  - Disponibilité
    - Pour garantir que les éléments considérés sont accessibles au moment voulu par les personnes autorisées
  - Intégrité
    - Pour garantir que les éléments considérés sont exacts et complets
  - Confidentialité
    - Pour garantir que seules les personnes autorisées ont accès aux éléments considérés
  - Traçabilité
    - Pour garantir que les accès et tentatives d'accès aux éléments considérés sont tracés et que ces traces sont conservées et exploitables

266

Sécuriser

- Comment sécuriser?
  - Difficile et complexe
    - Possible de sécuriser à 100%?
    - Problème de coût
    - Problème de confort d'utilisation, de motivation,…
  - Agir
    - Mettre en place une politique de sécurité (Mentalité!)
    - Mettre en place des contremesures
    - Agir sur les vecteurs (FW, subdivision, …)
    - Agir sur les faiblesses (mise à jour, désactiver l'inutile, …)

267

Sécuriser

- Comment sécuriser?
  - Normes, méthodes et référentiels de bonnes pratiques en matière de sécurité
    - Norme ISO 27001, consacrée aux Systèmes de management de la sécurité de l'information (SMSI).
      - L'ISO 27001 porte moins sur l'efficacité des dispositions mises en place, que sur leur existence, et la mise en place d'une boucle d'amélioration (PDCA).
    - Norme ISO 27002 correspond à un niveau de détail plus fin que la 27001 et est un guide de Bonnes Pratiques (Best Practices) pour maîtriser la sécurité d'un système d'information
    - Autres : ISO 13335, ISO 15408, BS 7799, Méthode EBIOS, Méthode MEHARI, ISO 17799
  - Faire appel à des professionnels
    - Audit de sécurité

268

Sécurité physique

Sécurité physique

- Protéger l'accès au matériel
  - Empêcher le vol
    - Local sécurisé
  - Eviter les maladresses
    - Renverser du liquide, enfoncer un interrupteur, se prendre les pieds dans un câble, …
    - → Protéger et fixer les câbles
    - → Ne pas boire son café près des machines
  - Protection contre les coupures d'alimentation
    - Disjoncteur, onduleur, groupe électrogène, paratonnerre, …

Protéger les câbles

Source : http://www.promotelec.com/img/interface/illustration_d2.jpg

271

Protéger les câbles

Sources : http://www.cable-equipements.fr/images/photo%20GOULOTTE%20RIGIDE%20marron+chaussure%2007-2007%20rognee%20retournee%20WEB.JPG
http://www.cable-equipements.fr/images/schemaPCdanger.JPG

272

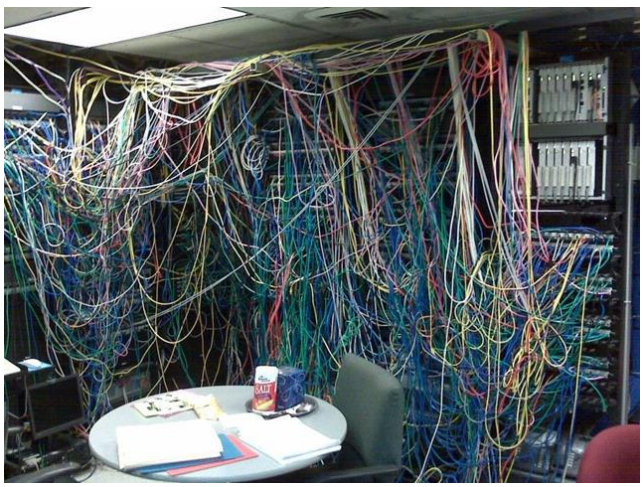
Fixer les câbles

Sources : http://www.amso.fr/informatique/img/baie-informatique2.jpg

http://www.oxer.fr/upload/galerie/IMG4888363a2d322-photo-001.jpg

273

No comment !



274

Sécurité physique

- Emplacement du matériel
  - Local spécialisé pour les serveurs
    - non poussiéreux,
    - bien aéré,
    - contrôle des accès.
  - Système d'alarme
    - Intrusion
    - Protection incendie et inondation
  - Local ou boîtier verrouillé

275

Source : http://www.journaldunet.com/solutions/securite/selection/07/0607-panorama-sso/ordinateur_cadenas_BU008089_getty.jpg

Avant d'installer

276

Choix d'une distribution

- Choix de la distribution
  - Orientée services, sécurité, stabilité…
- Équilibre entre fonctionnalités et sécurité
  - Compromis a trouver
    - (Sécurité + stabilité) et (fonctionnalités + facilité d'usage) ne vont généralement pas ensemble
- Rôle de la machine?
  - Poste de travail
  - Serveur (externe ou interne)
  - Infrastructure
  - Multi-usage
- Choix du noyau
  - Modulaire/monolitique
  - Recompiler un noyau sans les fonctionnalités inutiles

277

Pendant l'installation

278

Installation

- Choix du partitionnement
  - Ne pas tout mettre dans une seule partition
  - Utiliser les options de montage
    - Options noexec, nosuid, ro, …
  - Exemple de partitionnement
    - /tmp : nosuid, noexec
    - /usr : nosuid
    - /home : nosuid, noexec
    - /boot : nosuid, ro
    - N'oubliez pas de tester ces changements!

279

Installation

- Choix des paquetages
  - Connaissez votre système
    - Choix manuel des paquetages = travail lourd mais vous êtes certains de connaître tous les logiciels du système
    - Lisez chaque description de paquetage pour connaître son rôle
  - Installez et activez uniquement l'essentiel
    - Evitez les outils de développement sur les serveurs
    - Evitez les serveurs graphiques sur les serveurs
    - Evitez les logiciels inutiles

280

Installation

- Les sources
  - Vérifier toutes vos sources
    - Emballage scellé, site miroir approuvé, vérification de l'empreinte (MD5),…
- Installez les mises à jour
  - Au minimum les mise à jour de sécurité
  - Attention de connaître les modifications que la mise à jour va apporter au système
- Gardez toutes les sources
  - Toutes les sources utilisées lors de l'installation (OS, patch, mises à jour,…) doivent être sauvegardées afin de les avoir sous la main

281

Démarrage du système

282

Démarrage

- Protéger le BIOS
  - Par mot de passe
  - Empêcher le démarrage par un autre périphérique que le disque dur
  - Désactiver le flash du BIOS
- Protéger l'amorçage
  - Mot de passe, carte à puce,…
  - Empêcher "Linux single" ?
  - Empêcher la combinaison CTRL+ALT+DEL
  - Limiter l'utilisation de différentes commandes tels que reboot, halt…

283

Les services

- Quels services désactiver?
  - Il faut désactiver tous les services dont vous n'avez pas absolument besoin
  - Pour cela, vous devez pouvoir identifier chaque service et son rôle
- Supprimer un service
  - Supprimer le lien dans rc*.d
  - Supprimer le script lui-même
  - Utiliser la commande chkconfig
  - Désinstaller le service (Yum, readme, …)

284

Connexions des utilisateurs

285

Les comptes utilisateurs

- Shadow
  - Au minimum, il faut vérifier que les mots de passe shadow sont activés sur le système
- Mot de passe
  - Gestion des mots de passe
    - Changer le mot de passe lors de la première connexion ou régulièrement (chage)
    - Choisir un mot de passe compliqué (mais que vous arrivez à retenir!)
  - Mot de passe = sécurité?
    - Mot de passe générique
    - Enlever la pile du BIOS
    - Attaque par dictionnaire
    - Rayonnement électromagnétique (écrans, câbles)
    - Social engineering (machine à café = danger !!)
    - Keylogger
- Groupe
  - Les UPG peuvent conduire les utilisateurs à modifier les permissions d'accès afin de partager des données

286

Connexions

- Lire
  - Et vérifier la date de dernière connexion
- /etc/nologin
  - La présence de ce fichier interdit la connexion des utilisateurs
  - S'il existe, son contenu est affiché lors d'une tentative de connexion
- /var/log/secure
  - Liste les connexions qui ont échouées
- /etc/securetty
  - Liste les consoles sur lesquelles il est possible de se connecter en tant que root
- Utilisation de su
  - Exit après chaque su
  - # su root –c « commande »

287

Shell et déconnexion

- /etc/shells
  - Contient la liste des shells disponibles
  - Il est plus facile de surveiller et mettre à jour un shell que 3 ou 4
- Déconnexion
  - Se déconnecter dès que l'on est plus devant le PC
  - Mot de passe sur l'écran de veille
- /etc/shells et /etc/bash_logout
  - Nettoyer l'affichage lorsqu'un utilisateur se déconnecte
  - /usr/bin/clear dans /etc/skel/.bash_logout

288

Fichiers et systèmes de fichiers

289

Le système de fichiers

- umask
  - Cette commande permet de fixer les permissions par défaut pour les nouveaux fichiers créés
- SUID et SGID
  - Les fichiers SUID et SGID sont un risque pour la sécurité, ils sont dès lors à surveiller
    - find / -perm +6000 –follow
    - Liste à copier sur un support non réinscriptible pour comparaison ultérieure

290

Le système de fichiers

- Permissions et attributs de fichiers
  - chmod
  - lsattr, chattr
    - +i : empêche toute modification sur le fichier
    - +a : il est possible d'ajouter des données à un fichier ayant cet attribut, mais pas de le supprimer. Le contenu du fichier ne peut pas être écrasé.
- Montage des FS externes
  - Choix des options de montage en ro, noexec, nosuid, user, noauto, umask, …
  - Via une commande ou /etc/fstab

291

Le système de fichiers

- Suppression de fichiers
  - Un fichier supprimé peut être récupéré
  - Utilitaires de suppression de fichiers :
    - overwrite, wipe, bcwipe
- Chiffrer les données ou le FS
  - TCFS : Transparent cryptographic File System
- Sauvegarder fichiers et systèmes
  - Données, binaires, fichiers de configuration,…

  → Plan de sauvegarde

292

Contrôle de la sécurité

293

Logiciels

- Pare-feu
  - Netfilter/iptables,
- Anti-virus
  - NOD32, Sophos, McAfee, Avast, panda, …
- Crack et John The Ripper
  - Sont des logiciels de recherche de mot de passe
- Tripwire, AIDE
  - Ils vérifient l'intégrité du système. Il crée des signatures pour les fichiers que l'on désire surveiller
- Tiger, COPS, LSAT
  - Ils vérifient les permissions d'accès, le fichier de mot de passe, les fichiers SUID et SGID, …
- Scanlogd, portsentry
  - Détecteurs de balayage
- …

294

Logiciels

- Utilitaires pour renforcer le système
  - Bastille
    - Recherche les points considérés comme peu sécurisés et demande si l'on souhaite les renforcer
  - Openwall
    - Améliore la sécurité du noyau et apporte de nouvelles fonctionnalités
  - LIDS, snort
    - Systèmes de détection d'intrusion
  - Grsecurity
    - Inclus les fonctionnalités openwall ainsi que d'autres telles que les ACL, une journalisation améliorée, …
  - SELinux
    - Applique une sécurité plus stricte au niveau des utilisateurs (même root est considéré comme un utilisateur sans droit privilégié)
  - …

295

Remarques

- Un domaine très vaste
  - Ce cours ne se veut pas exhaustif
  - Il s'agit d'une sensibilisation, tous les aspects liés à la sécurité d'un système ne s'y trouvent pas
- Plan global
  - Cette présentation fournit un ensemble de points de réflexion et de conseils
  - Pour être efficace, la sécurisation doit être pensée globalement et toujours tenir compte des impératifs inhérents aux systèmes à sécuriser
- Audit
  - Plus votre réseau sera complexe, plus vous risquez de devoir faire appel à des spécialistes

296

Firewall

Module 20
Introduction à iptables

297

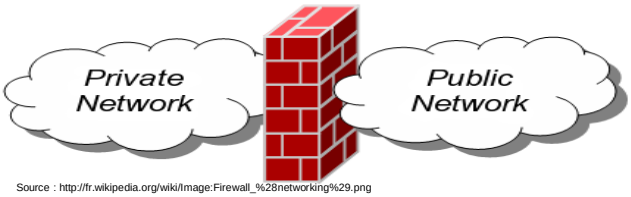
Objectifs

- Objectifs
  - Comprendre le fonctionnement du filtrage de paquet sous Linux
  - Pouvoir lire des configurations de pare-feu utilisant iptables
  - Savoir construire un pare-feu de base avec iptables
  - Proxy

298

Principe général

- Firewall (Pare-feu)
  - Un pare-feu est un dispositif logiciel ou matériel qui filtre le flux de données sur un réseau
  - Ce filtrage permet de définir quels sont les types de communications autorisés ou interdits



Source : http://fr.wikipedia.org/wiki/Image:Firewall_%28networking%29.png

299

Principe général

- Filtrage et accès aux ressources
  - Le pare-feu permet d'appliquer une politique d'accès aux ressources réseau en filtrant les paquets passant par lui
  - Critères de filtrage
    - L'origine ou la destination des paquets (adresse IP, ports TCP ou UDP, interface réseau, etc.)
    - Les options contenues dans les en-têtes des paquets (fragmentation, validité,...)
    - Les données elles-mêmes (taille, ...)
    - …

300

Types de pare-feu

- Nomenclature
  - Le terme Firewall est général et recouvre des architectures et techniques bien différentes
- Par exemple, on parle de firewall pour désigner
  - Un routeur filtrant
  - Un routeur filtrant + un serveur proxy cache
  - Un système Firewall logiciel ou matériel
  - Un système de filtrage de paquets + filtrage applicatif
  - Une architecture hybride avec routeurs filtrants, firewalls dédiés et proxy
  - …

301

Types de pare-feu

- Pare-feu filtre de paquets
  - Pare-feu sans états (stateless firewall)
    - Le pare-feu analyse chaque paquet indépendamment des autres et le compare à une liste de règles
  - Pare-feu avec états (Statefull)
    - Permet la distinction des paquets associés à une connexion des autres paquets et conserve la trace de ces connexions
      - Cela permet de vérifier la conformité des paquets à une connexion en cours (Le paquet 2 est bien la suite du 1,…)
    - Il offre plus de sécurité car il est possible de rejeter tous les paquets n'appartenant pas à une connexion

302

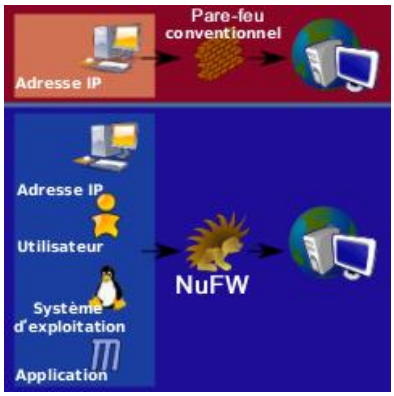
Types de pare-feu

- Pare-feu applicatif
  - Il vérifie la conformité du paquet à un protocole donné
    - Exemple
      - Permet de vérifier que seul du HTTP passe par le port TCP 80
    - Seul un nombre limité d'applications est supporté
      - Le pare-feu proxy doit connaître toutes les règles protocolaires des protocoles qu'il doit filtrer
      - Problèmes avec les protocoles « maisons » ou nouveaux
    - Traitement gourmand en temps de calcul

303

Types de pare-feu

- Pare-feu identifiant
  - Réalise l'identification des connexions ce qui permet de définir des règles de filtrage par utilisateur, OS, …



Source : http://www.nufw.org/

304

Type de pare-feu

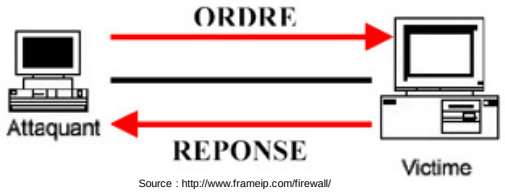
- Pare-feu personnel (Personal firewall)
  - Nom donné au pare-feu à états destiné à ne protéger qu'une machine, généralement une station de travail
- Pare-feu distribué
  - Le Firewall distribué se compose
    - d'un centre d'administration (hébergé sur un serveur)
    - d'un ensemble de firewalls personnels installés sur chaque poste à protéger
- Pare-feu Matériel/logiciel
- …

Fonctionnalités supplémentaires

- Les firewalls peuvent aussi intégrer
  - Un module antivirus
  - Un module antispam
  - Un module anti-Malware/spyware
  - Une sonde de détection d'intrusion
  - Un module VPN
  - …

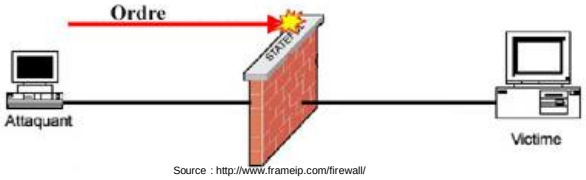
Principe général

- Principe d'une attaque



Source : http://www.frameip.com/firewall/

- Principe de la protection par FW



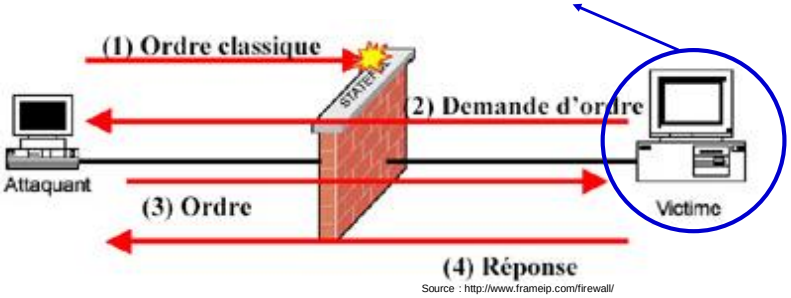
Source : http://www.frameip.com/firewall/

307

Principe général

- Le blocage des seuls flux entrants est insuffisant

La victime a installé sans le vouloir une backdoor qui initie elle-même une session avec l'attaquant



Source : http://www.frameip.com/firewall/

308

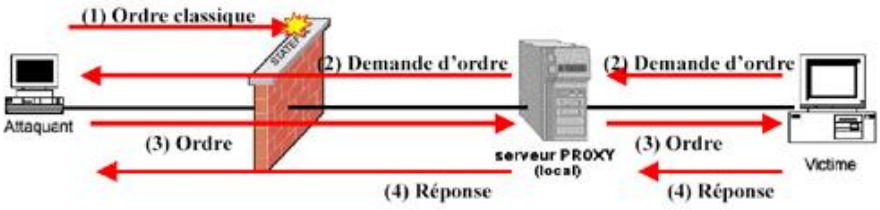
Proxy

- Principe
  - Connexion
    - Le proxy intercepte la demande de connexion du client puis établit, en lieu et place de l'utilisateur, une connexion avec le serveur demandé
  - Filtrage
    - Toutes les trames reçues peuvent être analysées pour en vérifier la cohérence
      - Une communication non conforme peut être immédiatement détectée, stoppée et générer une procédure d'alerte et d'identification de l'attaquant
    - Permet de filtrer au niveau des couches application du modèle OSI
      - Donc, sur base d'un protocole ou d'une application venant des clients d'un LAN

309

Proxy

- Protection par proxy
  - Un proxy joue le rôle d'un tampon entre un client et un serveur
  - Il héberge des applications qui filtrent ou traitent les messages avant de les transmettre à leur tour
  - Il est donc possible de contrôler le flux sortant
    - Le contrôle n'est pas efficace à 100%, une backdoor peut réussir à déjouer le proxy



Source : http://www.frameip.com/firewall/

310

Proxy

- Avantages
  - Contrôle, sécurité
    - Donne aux administrateurs le contrôle sur les applications et protocoles spécifiques
    - Possible de filtrer sur base de mots clé, d'applications, …
  - Mémoire cache
    - Un proxy peut mettre en cache des données, permettant de réduire la consommation de bande passante.

311

Proxy

- Avantages
  - Journalisation des requêtes (logging)
    - Les services proxy peuvent être journalisés permettant un contrôle plus strict de l'utilisation des ressources du réseau
  - Anonymat
    - Toutes les requêtes sont réalisées avec l'IP du proxy
  - Facilité de mise en place
    - Pas nécessaire de configurer les clients dans le cas d'un proxy transparent

312

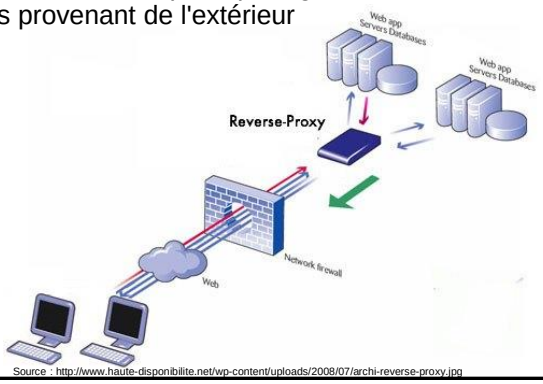
Proxy

- Inconvénients
  - Limitations
    - Les proxies sont souvent particuliers à des applications (HTTP, telnet, etc.) ou limités à des protocoles (la plupart fonctionnent seulement avec des services connectés à TCP)
  - Goulot d'étranglement
    - Toutes les requêtes et transmissions sont aiguillées vers le proxy plutôt que directement vers des connexions distantes

313

Reverse proxy

- Reverse proxy
  - Type de serveur proxy implémenté du côté des serveurs Internet
  - L'utilisateur du web passe par son intermédiaire pour accéder aux applications de serveurs internes
  - Habituellement utilisé pour protéger un serveur web des attaques provenant de l'extérieur

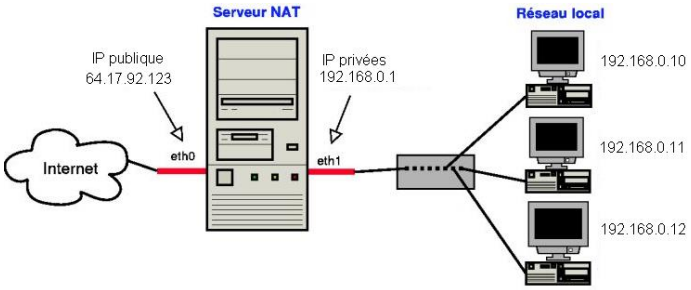


Source : http://www.haute-disponibilite.net/wp-content/uploads/2008/07/archi-reverse-proxy.jpg

314

NAT

- NAT : Network Address Translation
  - Correspondance entre les adresses IP internes non-uniques et souvent non routables d'un intranet à une (ou plusieurs) adresse externe unique et routable



315

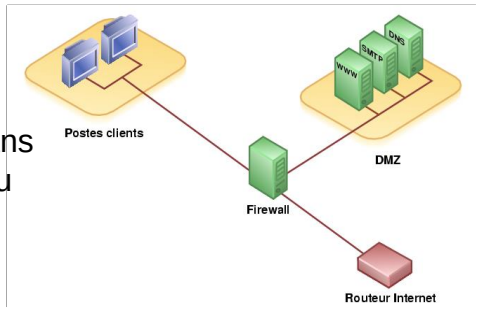
NAT

- Translation IP privées – IP publiques
  - Le NAT vous permet notamment de translater les adresses IP privées d'un LAN afin de les rendre "accessible" depuis un réseau public comme Internet
- IP Masquerading
  - Masquage IP : l'adresse IP du routeur/FW est la seule utilisée à l'extérieur
    - Des réseaux internes sont placés derrière une seule (ou plusieurs) adresse IP publique permettant ainsi de masquer toutes les requêtes comme provenant d'une source plutôt que plusieurs

316

Architectures

- Firewall et routeur
  - Le pare-feu permet de diviser un réseau en différentes zones de sécurité
  - On place le matériel dans une zone en fonction du niveau de confiance accordé



Source : http://www.drieu.org/docs/graphics/dmz.png

Architectures

- DMZ
  - Zone protégée par un FW
    - On y place les serveurs publics (Mail, FTP, Web)
    - Ces serveurs sont accessibles depuis Internet et depuis le réseau privé
  - DeMilitarized Zone (Zone démilitarisée)
    - Ces serveurs ne sont pas autant protégés que la partie privée (LAN) du réseau car ils doivent être accessible depuis Internet

LAN

- LAN
  - C'est le réseau privé, dans cette zone il n'y a que des machines (clients du réseau, serveurs, ...) qui sont "inaccessibles" publiquement depuis l'Internet
- Firewall dans le LAN
  - Protéger une partie d'un réseau des autres
    - Existence de sous-réseaux moins sûres que d'autres (réseaux de démonstration, de formation, de PC portables)
    - Existence de sous-réseaux nécessitant plus de sécurité (projets de développement secrets, données financières,…)

319

Architectures

- Nombreuses architectures
  - Il existe en fait de nombreuses manières de structurer un réseau pour protéger des systèmes à l'aide de pare-feu
- L'architecture de base définit 4 zones
  - Internet
  - DMZ
  - LAN
  - Wi-Fi

320

Architectures

- Le routeur peut être la propriété d'un FAI (fournisseur d'accès Internet), auquel cas on ne dispose pas du contrôle de celui-ci.
- Il faut demander au FAI d'y inclure des filtres
  - et avoir pleine confiance dans son FAI…

321

Architectures

- En utilisant un routeur filtrant, on peut créer une DMZ filtrée.
  - Cela donne un contrôle total sur les services Internet et maintient la séparation avec le réseau local "normal"

322

Architectures

- Il est possible de placer le serveur mandataire sur le réseau local
  - Dans ce cas, les règles du pare-feu ne doivent autoriser que le proxy à se connecter à Internet pour les services que celui-ci fournit
- Ainsi les utilisateurs ne peuvent accéder à Internet que par le mandataire
  - Cela nécessite de configurer chaque client

323

Architectures

- Un serveur mandataire peut relayer différentes requêtes et entretenir un cache des réponses. Architecture avec proxy transparent.

324

Architectures

- Architecture redondante de routeurs et pare-feux.

325

Architectures

- Utilisations de plusieurs pare-feu pour augmenter la sécurité (pare-feu dos à dos)

326

Filtre de paquets

- Les pare-feu de filtrage de paquets (routeur filtrant)
  - Ils peuvent lire et traiter les paquets en fonction des informations situées dans les en-têtes des paquets
  - Le filtrage se fait sur n'importe laquelle des interfaces
    - paquets entrants, paquets sortants ou les deux
  - Ils filtrent les paquets sur la base de règles programmables par l'administrateur
  - Une fonctionnalité de filtrage des paquets est incorporée dans le noyau Linux : netfilter

327

Filtre de paquets

- Avantages
  - Pas de configuration coté client
    - Ne nécessite aucune personnalisation du côté client car toute activité est filtrée au niveau du routeur/FW plutôt qu'au niveau de l'application
    - Netfilter est personnalisable par le biais de d'utilitaires graphiques ou de scripts
  - Performance
    - La performance réseau est plus rapide grâce à une connexion directe du client à l'hôte distant (Pas de proxy)
    - L'analyse des paquets se fait sur les couches basses

328

Filtre de paquets

- Inconvénient
  - Filtre les couches basses du modèle OSI
    - Traite des paquets au niveau du protocole de transport mais ne peut pas les filtrer au niveau d'une application ou du contenu
  - Configuration parfois complexe
    - Des architectures réseau complexes peuvent rendre difficile l'établissement de règles de filtrage

329

Netfilter et iptables

- Netfilter
  - Système qui, sous Linux, fournit le filtrage de paquets avec ou sans état ainsi que les services de masquage IP (NAT)
  - Netfilter permet aussi une gestion avancée du routage et des états de connexion
- iptables
  - iptables est une interface en ligne de commande qui permet d'implémenter Netfilter
  - Démarrage du service

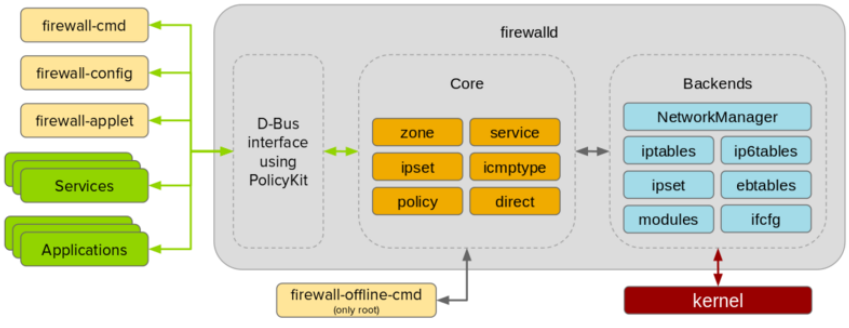
    ```
    # service  iptables  start
    ```
- ipchains
  - Prédécesseur de iptables

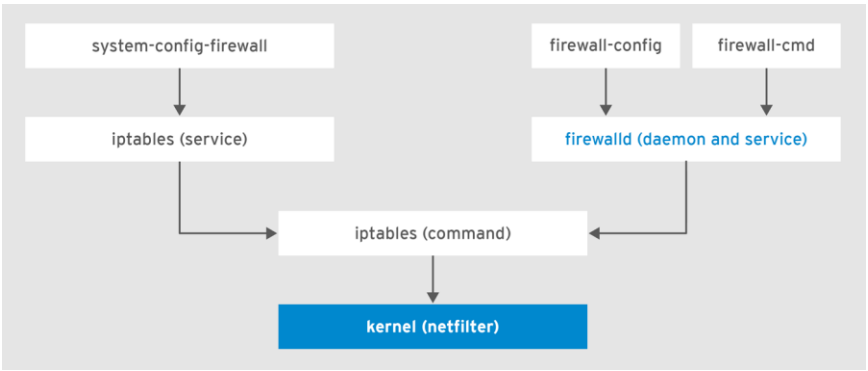
330

Firewalld

- Firewalld
  - FirewallD est un service SystemD de type dbus qui fournit une applet, une interface graphique, une interface en ligne de commande et une interface D-BUS.
  - FirewallD est lui-même une interface au module noyau netfilter et à ses modules complémentaires. En interne, les commandes iptables, ip6tables et ebtables sont alors utilisées pour configurer les règles du module noyaux netfilter et ses modules complémentaires.

Firewalld





333

Firewalld => iptables

- Procédure pour déactiver firewalld et activer iptables :
  - #systemctl stop firewalld
  - #systemctl disable firewalld
  - #yum remove firewalld
  - #yum install iptables-services
  - #systemctl enable iptables.service
  - #systemctl enable ip6tables.service

334

Tables et chaînes

- La notion de table
  - Les tables regroupent, en fonction de leur rôle respectif (filtrer, translater,…) les différentes règles de traitement des paquets
  - Chaque table est constituée d'une ou plusieurs chaînes
- La notion de chaîne
  - Une chaîne est constituée d'une suite de règles
  - Chaque table est constituée d'une ou plusieurs chaînes
    - Une chaîne qui correspond aux règles de filtrage des paquets en entrée
    - Une autre pour les paquets en sortie,
    - …
  - Avec iptables, le nom des chaines s'écrit en majuscule

335

Règles et politiques

- Règles
  - Chaque règle doit
    - Déterminer les paquets sur lesquels la règle doit s'appliquer (partie correspondance)
    - Préciser le sort des paquets (partie cible) (option –j)
  - Si la règle ne s'applique pas au paquet, on teste la règle suivante de la chaîne
- Politique
  - La politique d'une chaîne (option –P) détermine le sort des paquets qui atteignent le bout de la chaîne sans avoir été affecté d'une cible

Tout ce qui n'est pas explicitement autorisé doit être interdit

336

Les cibles

- Les cibles
  - Spécifient soit
    - l'action à appliquer à un paquet correspondant à une règle
    - la politique d'une chaîne
  - L'action (cible) est toujours écrite en majuscule

337

Exemples de cibles

- ACCEPT
  - Autorise le paquet à passer à l'étape suivante du traitement
- DROP
  - Le paquet est abandonné (arrêt de son traitement)
- REJECT
  - Le paquet est rejeté et une notification de rejet est envoyée
- LOG
  - Permet de loguer le paquet
- SNAT
  - Utilisée pour changer l'adresse source

338

Chaînes prédéfinies

- INPUT
  - Chaîne par laquelle passent les paquets entrant
- OUTPUT
  - Chaîne par laquelle passent les paquets sortant
- FORWARD
  - Chaîne utilisée par le paquet qui ne fait que transiter par le pare-feu
- PREROUTING
  - Permet de modifier un paquet dès qu'il rentre dans le système avant qu'il soit routé
- POSTROUTING
  - Permet de modifier un paquet juste avant sa sortie du système après être passé par le module de routage

339

Chaînes utilisateurs

- Chaînes utilisateurs
  - Ce sont de nouvelles chaînes qu'il est possible de créer
- Seules les chaînes prédéfinies sont utilisées
  - Les chaînes utilisateurs doivent donc être appelées à travers les règles des chaînes prédéfinies
  - Les chaînes utilisateur peuvent être réutilisée à plusieurs endroits

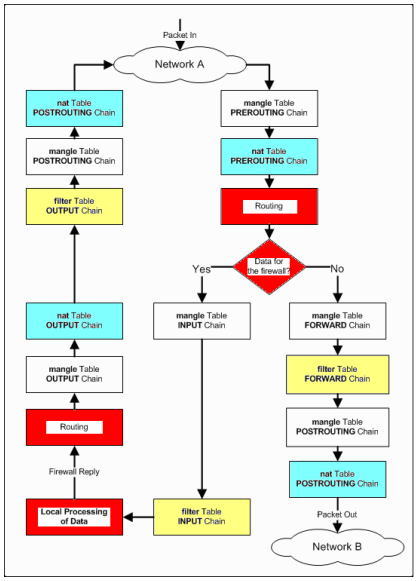
340

Les tables

- La table *filter*
  - C'est la table par défaut
  - Elle filtre les paquets sans en modifier le contenu
  - Elle contient les chaînes INPUT, OUTPUT et FORWARD
- La table *nat*
  - Elle réalise les translations d'adresse
  - Elle est constituée des chaînes PREROUTING, OUTPUT et POSTROUTING
- La table *mangle*
  - Cette table permet d'effectuer des modifications sur le paquet (la priorité du paquet par exemple)
  - Elle est constituée des chaînes INPUT, OUTPUT, FORWARD PREROUTING et POSTROUTING
- La table *raw*
  - Utilisée pour placer des marques sur les paquets qui ne doivent pas être vérifiés par le système de traçage de connexion
  - Cette table ne supporte que les chaînes PREROUTING et OUTPUT

341

Vue simplifiée



342

Syntaxe de iptables

iptables option_commande chaîne option_param action

- option_commande
  - Ce sont les options spécifiant une commande
    - -A : Ajouter une règle à une chaîne
    - -D : supprimer une règle d'une chaîne
    - -F : supprimer toutes les règles d'une chaîne
    - -P : fixer la politique d'une chaîne
- Chaîne
  - Indique la chaîne concernée par la l'option commande (INPUT,…)
- option_param
  - Ce sont la ou les option(s) utilisée(s) en paramètres d'une action
    - -p : indique le protocole concerné
    - -s : indique l'adresse source concernée
- Action
  - Il s'agit de la politique appliquée au paquet (ACCEPT, DROP,…)

343

Exemples

- # iptables –P INPUT DROP
  - Application de la politique par défaut pour la chaîne INPUT de la table filter consistant à refuser tout ce qui n'est pas autorisé
- # iptables –A INPUT –p icmp –j ACCEPT
  - Ajout d'une règle autorisant les accès icmp en entrée
- # iptables -A INPUT -p tcp --dport 22 -j ACCEPT
  - Acceptez les données envoyées par un serveur ssh distant

344

Ordre des règles

- Fonctionnement
  - L'ordre est important, si votre première règle est de tout refuser, les autres règles ne serviront jamais!
- Performance
  - Les règles les plus utilisées doivent être placées en premier
  - Exemple :
    - Les réponses aux requêtes vers le LAN
    - La navigation sur Internet

345

Script

- Nécessité du script
  - Les règles iptables sont effacées lors de l'arrêt du système car tout est directement géré au niveau du noyau
- Sauvegarde et chargement de la configuration
  - # iptables-save > fichierFW.sav
  - # iptables-restore < fichierFW.sav
  - Ou fichier de configuration /etc/sysconfig/iptables
- Conseils
  - Commencer le script en effaçant toutes les règles existantes

346

Exemples de logiciels

- Distributions pour transformer un PC en FW
  - SmoothWall
    - Distribution linux basée sur Netfilter intégrant notamment une interface web de management à distance, fichiers de log, NAT, DNS, Proxy, VPN, SSH, IPSEC...
  - IPCop-IPfire
    - Distribution linux basée développée à partir de smoothwall
    - Il peut également servir de serveur mandataire (proxy), serveur fournissant des adresses IP dynamique (DHCP), de relais DNS, de serveur de temps (NTP),
    - Firewall permettant aussi le contrôle de contenu, liste noire, liste d'accès, DNS dynamique, contrôle de trafic, etc...
  - Pfsense
    - Distribution firewall open source basée sur FreeBSD

347

Exemples de boitiers

Cisco ASA

SonicWALL

Check Point

Juniper

348

Exemples de proxy

- Squid
  - Proxy Open source le plus utilisé
- Delegate
  - Serveur proxy multi-plateformes multi-usages mandaté au niveau applicatif ou au niveau session
- SSH Proxy
  - Serveur mandataire pour le protocole SSH open source
- …

349

Divers

- Dansguardian et Squidguard
  - Logiciel de filtrage de contenu qui peut fonctionner avec Squid
- PortSentry
  - IDS (Système de Détection d'Intrusion) permettant de détecter les attaques de type "scan port" et ainsi réagir via iptables
- Inetd et xinetd
  - Démons qui permettent de gérer les connexions à des services réseau
- …

350

Protection efficace?

- **Un pare-feu = sécurité assurée?**
  - Il n'existe pas de firewall inviolable
    - Certains systèmes renferment des logiciels complexes qui ne peuvent pas être totalement exempts de failles
  - Etablissement des règles
    - Les règles de filtrage sont obligatoirement un compromis entre la sécurité maximale et une certaine liberté d'usage notamment de l'Internet
    - Là où il y a compromis, il y a aussi faiblesse

351

Protection efficace?

  - Proxy
    - Les proxys travaillent sur les couches hautes du modèle OSI, on peut les contourner en utilisant des failles au niveau des couches basses
  - Filtre de paquets
    - Les filtres de paquets travaillent sur les couches basses du modèle OSI, on peut les contourner en exploitant les failles des couches hautes

352

Attaques contre les firewalls

- firewalking
  - Technique visant à déterminer les règles de filtrage
    - des firewall, des routeurs ou des passerelles
  - Le but est simple : savoir quels ports ne sont pas filtrés
- Scanneur
  - nmap peut déterminer quels ports sont ou ne sont pas filtrés par un firewall

353

Attaques contre les firewalls

- Principe du firewalking
    - Basé sur les en-têtes de paquet
  - TTL
    - Le TTL permet de déterminer le nombre de routeurs filtrant entre la machine source et la machine cible (située derrière le firewall)
  - ICMP
    - Les routeurs renvoient des messages ICMP ce qui permet de récupérer des informations sur les routeurs (son adresse par exemple)
  - Règles
    - Ces attaques envoient ensuite différents types de paquets (TCP, UDP) pour essayer de déterminer les règles de firewalling

354

Conseils généraux

- Limiter les accès
  - Fermez tous les ports en écoute
    - sur les différents serveurs et ouvrez seulement ceux dont vous avez besoin
  - Filtrez les ports
    - c'est à dire rejetez toutes les requêtes sur les autres ports que ceux en écoute
  - Empêchez toutes les connexions sortantes sur des services non autorisés
  - Limiter l'accès à votre réseau à des utilisateurs connus

355

Conseils

- Travailler à tous les niveaux
  - Surveiller tout ce qu'il se passe
    - Aussi bien sur les serveurs que sur les filtres, pour détecter le plus rapidement possible toute activité anormale
    - Consulter les logs, faire du monitoring, …
  - Une machine, un service
    - Utilisez une machine dédiée au pare-feu, évitez de placer vos différents serveurs sur la même machine
  - Ne jamais se croire en sécurité
    - Il y a forcément quelque part une faille que l'on ne connaît pas, et qu'un pirate peut trouver

356

Exercices

1. Affichez les règles de la table filter
2. Interdisez les paquets en entrée sur le firewall pour la machine d'adresse MAC 11:22:33:44:55:66
3. Autorisez la transmission de paquets vers l'hôte 192.168.1.1
4. Solutions

   1) # iptables –t filter –L

   2) # iptables -A INPUT -m mac --mac-source 11-22-33-44-55-66 -j DROP

   3) # iptables -A FORWARD -d 192.168.1.1 –j ACCEPT

357

YUM

Module 21

Yellowdog Updater, Modified

358

Objectifs

- Objectifs
  - Pouvoir installer, mettre à jour, interroger et supprimer des packages logiciels avec l'utilitaire YUM

359

RPM

- RPM
  - RedHat package manager
  - Archives compressées contenant
    - Des programmes pré-compilés prêts à l'emploi, des informations sur le package, des icones, de la documentation, des scripts, une signature numérique, …
  - CentOS fourni ses logiciels sous forme de fichier .rpm
    - Nécessité de rechercher et installer manuellement chaque fichier .rpm (ainsi que les dépendances)
- Dépendances
  - Une dépendance est un paquetage devant être installé pour qu'un autre paquetage puisse l'être
    - Par exemple, lorsqu'un programme nécessite une librairie non incluse directement dans le paquetage du programme

360

YUM

- YUM
  - Yellowdog Updater, Modified

– Outils de gestion par défaut de CentOS
  - Permet de gérer les installations, désinstallations et mises à jour de paquetages au format RPM
  - Localise, « télécharge » et installe automatiquement un fichier .rpm ainsi que ses dépendances

361

YUM repository

- Notion de dépôt
  – Dépôt
    - Ce sont des espaces de stockage qui regroupent un ensemble de logiciels
      – Sur un disque dur, le réseau ou Internet
    - Repository en anglais (abréviation : repo)
  – Principe
    - YUM recherche les logiciels à installer dans ces dépôts
    - Pour être utilisé, chaque dépôt doit être ajouté à la configuration de YUM

362

YUM repository

- Les dépôts rpm distants
  - Les dépôts officiels et leurs mirroirs
    - Contiennent les paquetages rpm élaborés par le projet CentOS
    - Tous les paquetages du projet CentOS sont libres d'utilisation
  - Les dépôts tiers
    - Proposent des paquetages additionnels à ceux des serveurs officiels
    - Maintenus par un individu ou un groupe d'individus
    - Attention
      - Certains de ces dépôts proposent des versions différentes et incompatibles d'un même logiciel
      - Certains paquetages de ces dépôts peuvent fonctionner différemment des paquetages équivalent du dépôt de base
      - Certains paquetages de ces dépôts peuvent écraser des paquetages du dépôt de base
      - Les paquets construits pour une version de CentOS ne sont généralement pas compatibles avec d'autres versions de CentOS.

363

YUM repository

- Dépôts de base (généralement activés par défaut)
  - base
    - Dépôt où sont stockés tous les paquetages de base de la distribution CentOS
  - updates
    - Toutes les mises à jours des paquetages contenus dans le dépôts de base sont publiées dans ce dépôt
  - Addons
    - Contient des paquetages ajoutés dans CentOS mais qui ne sont pas disponibles dans la distribution Red Hat Entreprise Linux
  - Extras
    - Contient des paquetages logiciels maintenus et testés par les développeurs de CentOS
    - Offrent des fonctionnalités supplémentaires au dépôt de base mais n'ont subi que des tests de base
      - Il est déconseillé de les utiliser dans le cadre d'un serveur sans s'assurer de leur fiabilité.

364

YUM repository

- Dépôts de base (généralement désactivés par défaut)
  - centosplus
    - Maintenus par les développeurs de CentOS et des utilisateurs
    - Certains paquets contenus dans ce dépôt peuvent écraser des paquets de la distribution originale → A utiliser en connaissance ce cause
  - contrib
    - Contient des paquets qui proviennent exclusivement des utilisateurs et qui ne seront pas intégrés à la distribution
    - Ces paquets n'écraseront pas ceux des dépôts officiels
  - testing
    - Contient des paquetages devant encore être testé

365

YUM repository

- Ajouter un dépôt
  - Procédure
    - Il faut ajouter un fichier de définition du dépôt dans le répertoire /etc/yum.repos.d/

```
# vim /etc/yum.repos.d/nouveaudepot.repo
#yum repolist all
```

  - Signature des paquetages
    - gpgkey
      - Ce paramètre spécifie l'emplacement d'une clé publique qui vérifie les paquetages fournis par le dépôt
      - Cette clé publique est automatiquement importées la première fois qu'un logiciel est installé à partir d'un dépôt
      - Toutes les clés sont stockées dans un trousseau
    - Vérification manuelle
      - Trouver la signature sur le site (ex : fichier md5sum.txt)
      - Utiliser la commande md5sum

366

YUM repository

- Autoriser manuellement les sources de paquets
  - Ajout manuel d'une clé publique au trousseau de clés RPM

    # *wget http://mirror.centos.org/centos/package-CentOS.key*

    # rpm –import *package-CentOS.key*
  - Importer les clés publiques directement depuis un site web

    # rpm --import *http://mirror.centos.org/centos/RPM-GPG-KEY-CentOS-4*

367

YUM repository

- Dépôts complémentaires compatibles
  - Ils n'écrasent pas ceux des dépôts officiels
  - Lisez toujours les informations fournies sur la compatibilité
  - Dépôt *EPEL* (Extra Packages for Enterprise Linux)
    - Met à la disposition des utilisateurs de CentOS, des paquets conçus à l'origine pour Fedora.
    - http://fedoraproject.org/wiki/EPEL/FAQ#howtouse
  - Dépôt RPM Fusion
    - Contient toutes les applications et les codecs nécessaires pour lire les fichiers multimédia utilisant des formats non-libres.
    - Nécessite le dépôt EPEL pour fonctionner
    - http://rpmfusion.org/Configuration

368

YUM repository

- Adobe
  - Permet notamment d'installer le plugin Flash pour le navigateur Mozilla Firefox
  - http://linuxdownload.adobe.com/linux/i386/adobe-releasei386-1.0-1.noarch.rpm
- *Remi Collet*
  - Contient des versions récentes de MySQL et de PHP
  - http://rpms.famillecollet.com/
- Dépôt RPMForge
  - Il est incompatible avec *EPEL*. Ils ne doivent donc pas être activés en même temps. Contient des RPM issus des dépôts dag, dries et freshrpms.
- Liste des dépôts disponible pour CentOS
  - http://wiki.centos.org/AdditionalResources/Repositories?highlight=(repository)

369

YUM

- Dépôts recommandés par CentOS
  - Karanbir Singh's kbs-centos-extras Repository
    - http://centos.karan.org/
  - Dag Wieers' Repository
    - http://dag.wieers.com/rpm/FAQ.php#B4
  - Dries' Repository
    - http://dries.studentenweb.org/rpm/clientconfig.html
- Package group
  - Les packages peuvent aussi être manipulés en groupe
  - Liste des groupes de paquetage

    ```
    $ su -c 'yum grouplist'
    ```

370

YUM

- Exemple de fichier /etc/yum.conf

```
[main]
cachedir=/var/cache/yum
debuglevel=2
logfile=/var/log/yum.log
pkgpolicy=newest
installonlypkgs=kernel kernel-smp kernel-devel kernel-smp-devel kernel-largesmp
kernel-largesmp-devel kernel-hugemem kernel-hugemem-devel
distroverpkg=centos-release
tolerant=1
exactarch=1
retries=20
obsoletes=1
gpgcheck=1
plugins=1

# PUT YOUR REPOS HERE OR IN separate files named file.repo
# in /etc/yum.repos.d
```

371

YUM

- Exemple de fichier /etc/yum.conf
  - Deux sections
    - Main
      - Configure les options globales
    - Repository
      - Configure les options relatives à chaque dépôt
      - Par défaut, les définitions des dépôts sont placés dans les fichiers séparés (dans /etc/yum.repo.d)
  - Directives
    - distroverpkg : permet à yum de déterminer la version de la distribution (red-hat-release)
    - tolerant : yum tolère certaines erreurs dans les commandes
      - Par exemple, demander l'installation de trois paquetages dont un est déjà installé
    - exactarch : yum n'installe que les paquetages correspondant à l'architecture du système (ex : i686 et par i386)

372

YUM

- Exemple de fichier /etc/yum.conf
  - Directives
    - cachedir : indique l'emplacement du répertoire servant de cache
    - debuglevel indique le niveau de verbosité des logs (valeur de 0 à 10)
    - gpgcheck : indique si yum doit réaliser une vérification de signature des paquets
    - obsoletes = 1 : active le processus permettant à yum de surveiller l'obsolescence des paquetages
    - installonlypackage : paquetages ne devant pas être mis à jour (typiquement ceux correspondant au noyau)

373

YUM

- Empêcher l'utilisation d'un dépôt
  - Paramétrer la directive enable=0 dans le fichier de définition du dépôt
- Supprimer un dépôt
  - Supprimer le fichier correspondant dans /etc/yum.repos.d/
  - Effacer le contenu du cache /var/cache/yum/
- Effacer le cache
  - Par défaut, YUM conserve une trace des fichiers téléchargés pour une éventuelle utilisation future

```
# yum clean headers
# yum clean packages
# yum clean metadata
```

374

YUM

- Utiliser les plugins
  - Ajouter « plugins=1 » dans /etc/yum.conf
  - Le plugin fastestmirror
    - Permet de trier les sites miroirs par vitesse de connexion

    # yum install yum-plugin-fastestmirror
  - Le plugin protectbase
    - Utilisé pour protéger certains dépôts des mises à jour d'autres dépôts
    - Utile lorsque l'on utilise des dépôts tiers

    # yum install yum-plugin-protectbase
      - Ajouter protect=1(ou =0 selon la cas) dans chaque fichier /etc/yum.repos.d/*.repo
      - Seuls les dépôts protect=1 peuvent mettre à jour d'autres =1
  - Le plugin priorities
    - Permet d'assigner des priorités (de 1 à 99) aux dépôts YUM

375

YUM

- Exemple d'utilisation du plugin protectbase

```
[base]
name=CentOS-$releasever - Base
mirrorlist=http://mirrorlist.centos.org/?release=$releasever&arch=$basearch&repo=os
#baseurl=http://mirror.centos.org/centos/$releasever/os/$basearch/
gpgcheck=1
gpgkey=http://mirror.centos.org/centos/RPM-GPG-KEY-centos4
protect=1

#released updates
[update]
name=CentOS-$releasever - Updates
mirrorlist=http://mirrorlist.centos.org/?release=$releasever&arch=$basearch&repo=updates
#baseurl=http://mirror.centos.org/centos/$releasever/updates/$basearch/
gpgcheck=1
gpgkey=http://mirror.centos.org/centos/RPM-GPG-KEY-centos4
protect=1

#packages used/produced in the build but not released
[addons]
name=CentOS-$releasever - Addons
mirrorlist=http://mirrorlist.centos.org/?release=$releasever&arch=$basearch&repo=addons
#baseurl=http://mirror.centos.org/centos/$releasever/addons/$basearch/
gpgcheck=1
gpgkey=http://mirror.centos.org/centos/RPM-GPG-KEY-centos4
protect=0
```

Source : http://www.centos.org/docs/4/html/yum/sn-yum-maintenance.html

376

YUM

- Gestion des priorités
  - Extension pour Yum qui permet de gérer une hiérarchie de dépôts
    - Permet de protéger les paquets issus des dépôts de base
    - Permet d'attribuer un ordre de priorité aux dépôts tiers afin de préserver la stabilité de la machine
  - Installation/Configuration

    # su -c 'yum install yum- priorities'
    - Ajouter le niveau de priorité dans le fichier de configuration du dépôt: priority=1
    - Un dépôt sans priorité définie se voit attribuer la priorité la plus faible (99)
    - Exemple :
      - Priorité=1 pour les dépôts base, updates, addons, extras
      - Priorité=2 pour Centosplus et contrib qui sont moins fiables
      - Utiliser des n° de priorité plus élevés pour les autres dépôts

377

YUM

- Exemple d'utilisation des priorités des dépôts

```
[base]
name=CentOS-$releasever - Base
mirrorlist=http://mirrorlist.centos.org/?release=$releasever&arch=$basearch&repo=os
#baseurl=http://mirror.centos.org/centos/$releasever/os/$basearch/
gpgcheck=1
gpgkey=file:///etc/pki/rpm-gpg/RPM-GPG-KEY-CentOS-5
priority=1

#released updates
[updates]
name=CentOS-$releasever - Updates
mirrorlist=http://mirrorlist.centos.org/?release=$releasever&arch=$basearch&repo=updates
#baseurl=http://mirror.centos.org/centos/$releasever/updates/$basearch/
gpgcheck=1
gpgkey=file:///etc/pki/rpm-gpg/RPM-GPG-KEY-CentOS-5
priority=1

#additional packages that extend functionality of existing packages
[centosplus]
name=CentOS-$releasever - Plus
mirrorlist=http://mirrorlist.centos.org/?release=$releasever&arch=$basearch&repo=centosplus
#baseurl=http://mirror.centos.org/centos/$releasever/centosplus/$basearch/
gpgcheck=1
enabled=0
gpgkey=file:///etc/pki/rpm-gpg/RPM-GPG-KEY-CentOS-5
priority=2
```

Source : Marc Bessière, Configuration des dépôts, [On Line] http://www.linuxidentity.com/fr/down/articles/CentOS53_FR_depots.pdf

378

YUM

- Gestion des paquetages obsolètes
  - Problème
    - Un paquetage de moindre priorité pourrait signaler comme obsolète un autre paquetage provenant des dépôts officiels
      - Ainsi, le paquetage officiel pourrait être remplacé
  - Solution
    - Il faut le signaler à YUM en éditant le fichier de configuration des priorités

      # vim */etc/yum/pluginconf.d/priorities.conf*

      Ajouter « check_obsoletes=1 »

379

YUM

- Installer un programme avec YUM
  - Principe
    - Pour chaque opération, YUM recherche dans un dépôt les fichiers d'information
    - Ces fichiers permettent à YUM de déterminer les actions à réaliser pour obtenir le résultat demandé
    - YUM affiche un compte rendu et demande confirmation
    - Exemple : # yum install *tsclient*

```
================================================================================
 Package                 Arch       Version          Repository        Size
================================================================================
Installing:
 tsclient                i386       0.132-4          base              247 k
Installing for dependencies:
 rdesktop                i386       1.3.1-5          base              107 k

Transaction Summary
================================================================================
Install       2 Package(s)
Update        0 Package(s)
Remove        0 Package(s)
Total download size: 355 k
Is this ok [y/N]:
```

380

Source : http://www.centos.org/docs/4/html/yum/sn-managing-packages.html

YUM

- Exemples d'installation
  - Installer un paquetage
    - # yum install *nompaquetage*
    - # yum install *tsclient*
  - Installer un groupe de paquetages
    - # yum groupinstall "*nomdugroupe*"
    - # yum groupinstall "*MySQL Database*"
  - Remarque
    - Les services installés ne sont pas démarré par défaut

381

YUM

- Exemple de mise à jour
  - Mise à jour d'un paquetage
    - # yum update *nompaquetage*
    - # yum update *tsclient*
    - Si le logiciel à mettre à jour est en cours d'utilisation, il nécessite un redémarrage
  - Mise à jour de YUM
    - # yum update *yum*
  - Mise à jour du système complet
    - # yum update
    - # yum check-update

382

YUM

- Exemple de mise à jour de YUM

```
[root@localhost ~]# yum update yum
Setting up Update Process
Setting up repositories
update                    100% |=========================|  951 B    00:00
base                      100% |=========================| 1.1 kB    00:00
addons                    100% |=========================|  951 B    00:00
extras                    100% |=========================| 1.1 kB    00:00
Reading repository metadata in from local files
primary.xml.gz            100% |=========================| 232 kB    00:00
sqlite cache needs updating, reading in metadata
update     : ################################################## 733/733
Could not find update match for yum
No Packages marked for Update/Obsoletion
```

383

YUM

- Exemple de désinstallation
  - Désinstaller un paquetage
    - # yum remove *nompaquetage*
    - # yum remove *tsclient*
  - Désinstaller un groupe de paquetages
    - # yum groupremove "*nomdugroupe*"
    - # yum groupremove "*MySQL Database*"
  - Remarques
    - Vérifiez toujours la désinstallation du fichier de configuration
    - Les données utilisateurs ne sont pas supprimées

384

YUM

- Recherche de paquetages
  - Les recherches incluent les paquetages déjà installés ainsi que ceux disponibles sur les dépôts configurés
  - Recherche par nom de paquetage
    - # yum list *tsclient*
    - # yum list *tsclient-0.132*
  - Recherche sans connaître le nom de paquetage
    - # yum search *PalmPilot*
      - Vérifie les noms, les descriptions et résumés de paquets pour trouver ceux qui correspondent
    - # yum provides *libneon*
      - Vérifie à la fois les fichiers inclus dans les paquetages et les fonctions que le logiciel fournit
  - Utilisation des caractères génériques
    - # yum list tsc*

YUM

- Installation manuelle de paquetages
  - Programme non disponible dans un dépôt
    - Il faut l'installer manuellement
    - YUM permet d'installer manuellement un paquetage local :
      - # yum localinstall tsclient-0.132-4.i386.rpm
      - Si le paquetage était déjà installé dans une version antérieure, il sera mis à jour
      - Si des dépendances sont nécessaires, YUM les recherches sur les dépôts
  - Mise à jour
    - Yum update ne met pas à jour les paquetages installés manuellement

    → S'inscrire aux newsletters ou aux flux RSS du programme pour rester informé des nouvelles versions

Outils de gestion de paquetages

- up2date
  - Red Hat Update Agent
  - Outils utilisé dans les anciennes versions pour télécharger installer et mettre à jour le système
- Autres outils
  - Yum Extender (yumex), KYum, Smart
    - Front end graphique pour YUM

387