Guide de formation avec ateliers pratiques

Linux Administration

Jean-Francois Bouchaudy

Les Guides de formation Tsoft
Plus de 30 000 personnes formées à Unix et Linux

Tome 2

Administration système avancée

Tsoft EDITEUR

EYROLLES

Linux
Administration

CHEZ LE MÊME ÉDITEUR

Dans la collection Les guides de formation Tsoft

J.-F. BOUCHAUDY. – **Linux Administration. Tome 4 : les services applicatifs Internet (Web, FTP, email).**
N°12248, à paraître en 2009.

J.-F. BOUCHAUDY. – **Linux Administration. Tome 3 : sécuriser un serveur Linux.**
N°12245, 2008, 468 pages.

J.-F. BOUCHAUDY. – **Linux Administration. Tome 1 : les bases de l'administration système.**
N°12037, 2006, 288 pages.

A. BERLAT, J.-F. BOUCHAUDY, G. GOUBET. – **Unix Shell.**
N°11147, 2e édition, 2002, 412 pages.

J.-F. BOUCHAUDY, G. GOUBET. – **Unix Administration.**
N°11053, 2e édition, 2002, 580 pages.

J. BOUGEAULT. – **Java : la maîtrise.** *Java 5 et 6.*
N°12250, 2e édition, 2008, 550 pages.

J.-F. ROUQUIÉ. – **Lotus Domino 7 Administration.**
N°12106, 2007, 684 pages.

R. BIZOÏ. – **SQL pour Oracle 10g.**
N°12055, 2007, 620 pages.

R. BIZOÏ. – **PL/SQL pour Oracle 10g.**
N°12056, 2007, 340 pages.

Autres ouvrages

L. DRICOT. – **Ubuntu efficace.**
N°12362, 3e édition à paraître en 2009.

R. HERTZOG, R. MAS. – **Debian Lenny.** *Gnu/Linux.*
N°12443, 2009, 462 pages avec DVD-Rom (Cahiers de l'Admin).

C. BLAESS. – **Shells Linux et Unix par la pratique.**
N°12273, 2008, 262 pages (Collection Blanche).

I. HURBAIN. – **Mémento UNIX/Linux.**
N°11954, 2006, 14 pages.

B. BOUTHERIN, B. DELAUNAY. – **Sécuriser un réseau Linux.**
N°11960, 3e édition, 2007, 266 pages (Cahiers de l'Admin).

M. KRAFFT. – **Debian.** *Administration et configuration avancées.*
N°11904, 2006, 674 pages.

M. BÄCK *et al.* – **Monter son serveur de mails sous Linux.**
N°11931, 2006, 360 pages.

Linux
Administration

Tome 2

Administration système avancée

Jean-François Bouchaudy

Deuxième tirage 2009

Tsoft EDITEUR

EYROLLES

ÉDITIONS EYROLLES
61, bd Saint-Germain
75240 Paris Cedex 05
www.editions-eyrolles.com

TSOFT
10, rue du Colisée
75008 Paris
www.tsoft.fr



Le code de la propriété intellectuelle du 1er juillet 1992 interdit en effet expressément la photocopie à usage collectif sans autorisation des ayants droit. Or, cette pratique s'est généralisée notamment dans les établissements d'enseignement, provoquant une baisse brutale des achats de livres, au point que la possibilité même pour les auteurs de créer des œuvres nouvelles et de les faire éditer correctement est aujourd'hui menacée.

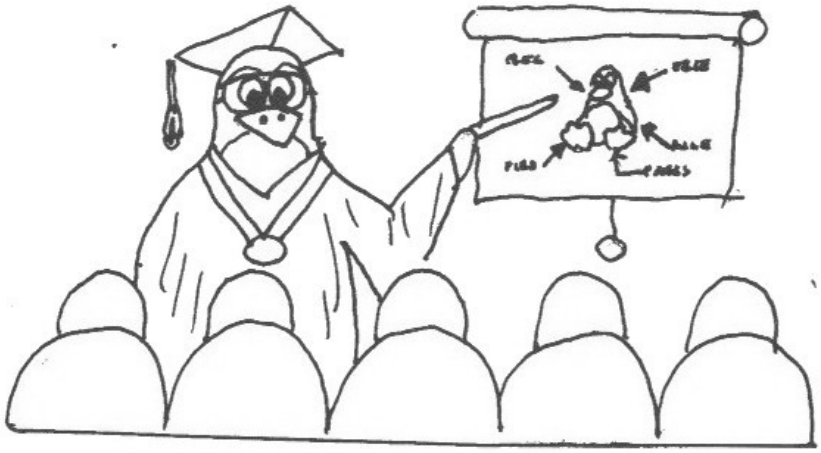
En application de la loi du 11 mars 1957, il est interdit de reproduire intégralement ou partiellement le présent ouvrage, sur quelque support que ce soit, sans autorisation de l'éditeur ou du Centre Français d'Exploitation du Droit de Copie, 20, rue des Grands-Augustins, 75006 Paris.

© Tsoft et Groupe Eyrolles, 2007, ISBN 13 : 978-2-212-12154-4



A mon fils Thomas

Avant-propos



Présentation de l'ouvrage

Aujourd'hui, il n'est plus besoin de présenter Linux, même les non informaticiens le connaissent, et certains l'utilisent à titre personnel. Dans les entreprises et les administrations, il est encore peu présent sur le poste de travail mais il envahit de plus en plus de serveurs… Et les serveurs doivent être administrés.

Ce livre traite de leur administration avancée. Il fait suite au tome 1, du même auteur, qui aborde les tâches élémentaires à accomplir par l'exploitant, sous le titre *Linux Administration : Bases de l'administration système*.

L'administration avancée, est associée à la conception (Design) et à l'organisation du système d'information. Elle est du ressort de l'administrateur système. Pour arriver à ses fins, ce dernier doit non seulement connaître toutes les possibilités techniques de l'OS (compilation du noyau, gestion des périphériques, démarrage), mais aussi maîtriser les techniques de structuration de l'espace disque (FS, RAID, LVM, sauvegarde et installation). Il doit savoir également améliorer les performances et dépanner son système.

L'informatique et Linux en particulier évoluent vite. Cet ouvrage est aussi l'occasion pour l'administrateur mais également pour l'utilisateur avancé ou même le développeur de découvrir les techniques récentes incluses dans Linux : Kickstart, Yum, SysFS, Udev, lm_sensors, SMART, ACLs, xattr, UnionFS, LVM, création de live-cd ou de clé USB bootable, etc.

Il existe de nombreux ouvrages sur Linux et son administration, en quoi ce livre est-il original ?

D'abord, il se veut manuel de formation. A ce titre, chaque module est divisé en deux parties : une partie "cours" et une partie "ateliers". La partie "cours" se divise elle-même en théorie et savoir pratique (commandes, fichiers, …). Les ateliers ne sont pas une accumulation d'exercices mais plutôt une séquence cohérente d'actions que le lecteur doit effectuer. Non seulement ils illustrent le cours mais ils représentent un savoir concret, en ce sens que la plupart des ateliers peuvent être considérés comme des "recettes pratiques d'administration". Les ateliers sont regroupés en "tâches". On n'est pas obligé de les réaliser toutes. Il faut privilégier celles qui correspondent à des concepts fondateurs ou à des sujets qui répondent à un besoin immédiat.

Contrairement au premier tome, ce livre développe beaucoup plus l'aspect théorique. C'est logique si l'on désire acquérir de solides bases conceptuelles. Il n'empêche que cet ouvrage reste très synthétique et très opérationnel.

Volontairement, ce livre privilégie le mode commande. Le système Windows a habitué l'utilisateur et l'administrateur à tout résoudre par des clics dans un environnement

graphique. Ce mode existe sous Linux, mais n'est pas celui utilisé par l'expert. Le mode commande en mode texte est plébiscité par l'ensemble des administrateurs Linux. Pourquoi ? Tout simplement parce qu'il est plus puissant, intemporel et même, à l'usage, plus simple. Ce mode permet l'administration complète d'un système Linux à distance avec une liaison inférieure à 9 600 bauds (débit supporté par un téléphone portable) ! Le mode commande est primordial dans l'approche automatisée de l'administration grâce à l'écriture de scripts shell. Il permet également une administration indépendante des distributions.

Ce livre ne se limite pas à une distribution particulière. Certes, pour les ateliers, il a bien fallu en choisir une. Nous avons opté pour CentOS, qui est un clone de la distribution RedHat, la plus utilisée dans les entreprises. Elle est binaire compatible avec elle. Dans les parties "cours", la rubrique "*Les particularités des distributions*" indique les commandes, les fichiers ou les aspects propres à une distribution particulière. Il a fallu faire un choix : seules les distributions RedHat, Debian et SuSE sont mentionnées.

Ce livre se veut le plus intemporel possible. Si de nouvelles commandes apparaissent avant une prochaine édition de cet ouvrage, le lecteur trouvera sur le site www.tsoft.fr (cf. plus loin) de nouvelles rubriques sur ces sujets.

Public

Le public visé par ce livre est d'abord les administrateurs de serveurs Linux. Du fait que les ateliers forment une sorte de recueil de "recettes pratiques d'administration", il peut être lu avec profit par tout administrateur, développeur ou utilisateur d'un système Linux.

Support de formation

Ce support convient à des formations sur l'administration avancée d'un système Linux d'une durée comprise entre deux et six jours. L'idéal est de cinq jours. La durée peut être écourtée ou allongée en fonction des modules et ateliers traités ainsi qu'en fonction du niveau des participants.

L'éditeur Tsoft (www.tsoft.fr) peut fournir aux organismes de formation et aux formateurs des "diapositives instructeurs" complémentaires destinés à aider le personnel enseignant.

Guide d'autoformation

Ce livre peut être également utilisé en tant que support d'auto-formation. L'élève doit disposer d'un ordinateur qui sera dédié à Linux (on le reformate complètement). Les modules installations et sauvegarde nécessite de disposer d'un poste configuré en serveur pour accomplir l'ensemble des ateliers.

Certifications

La certification LPI (Linux Professional Institute), indépendante des distributions, est supportée, parmi d'autres, par SuSe et IBM. L'ouvrage est une bonne préparation aux deux premiers niveaux du programme LPIC. Nous invitons les lecteurs à se renseigner auprès du LPI : http://www.lpi.org.

Un livre dynamique grâce à Internet

Le noyau Linux, les distributions Linux vont peut-être évoluer plus rapidement que cet ouvrage. Le site www.editions-eyrolles.com proposera sur la page de présentation du présent ouvrage des liens dédiés à des compléments sur ces évolutions.

Pour accéder à cette page, rendez-vous sur le site www.editions-eyrolles.com, dans la zone <Recherche> saisissez **G12154** et validez par <Entrée>.

Table des matières

PROGRESSION PÉDAGOGIQUE .....
QUELQUES CONVENTIONS DE NOTATION .....

1 ADMINISTRATION LINUX PROFESSIONNELLE 1-1
Philosophie de l'administration système 1-2
Automatiser les procédures. 1-4
Journaliser les changements et agir de manière réversible 1-5
Documenter son système 1-7
Communiquer 1-9
Segmenter l'administration 1-10
ATELIERS 1-11

2 L'INSTALLATION 2-1
Installer et organiser les données 2-2
L'installation automatique 2-3
L'installation Kickstart (RedHat) 2-6
Créer un serveur de dépôt 2-8
Créer un dépôt Debian 2-11
Créer un dépôt Yum, configurer le client 2-14
Créer un package RPM 2-16
Créer un package Debian 2-19
ATELIERS 2-21

3 LE NOYAU 3-1
Présentation du noyau 3-2
La compilation du noyau 3-4
Les modules du noyau 3-7
L'arborescence /proc 3-8
Le paramétrage à chaud du noyau 3-10
IPC System V 3-11
ATELIERS 3-13

4 LA GESTION DES PÉRIPHÉRIQUES 4-1
L'architecture d'un ordinateur 4-2
La gestion des périphériques sous Linux 4-8
Ajouter un périphérique 4-11
Le FS sysfs 4-13
La technologie Udev 4-15

La technologie lm_sensors 4-18
La technologie SMART 4-19
ATELIERS 4-20

5 LES SYSTÈMES DE FICHIERS 5-1
La vision théorique de la gestion des fichiers 5-2
Les FS à journalisation 5-6
Le FS XFS 5-7
Le FS ReiserFS 5-8
Le FS Ext3 5-9
Le FS NFS 5-13
FS spéciaux 5-15
Le montage à la volée 5-17
ATELIERS 5-18

6 COMPLÉMENTS SUR LES DISQUES ET LES FS 6-1
Les quotas 6-2
Les ACL 6-4
Les attributs Ext2 6-6
Les attributs étendus (xattr) 6-7
Les disques dédiés 6-8
SWAP 6-9
ATELIERS 6-12

7 LE RAID 7-1
La technologie RAID 7-2
Le RAID matériel 7-5
Solutions de type SAN 7-7
Les solutions de type DAS 7-9
Proliant (exemple de solution DAS) 7-10
Le RAID logiciel 7-11
Le RAID et le LVM 7-13
ATELIERS 7-15

8 LE LVM 8-1
LVM – les éléments clés 8-2
L'aspect dynamique des LV et des FS 8-6
Les snapshots 8-7
LVM – Compléments 8-8
ATELIERS 8-11

9 LA SAUVEGARDE 9-1
Vision générale de la sauvegarde 9-2
La sauvegarde incrémentale 9-4

Les commandes dump et restore 9-6
La commande dd 9-8
La sauvegarde réseau 9-9
La sauvegarde Bare-Metal 9-11
Le logiciel Mondo 9-12
Les cartouches 9-13
ATELIERS 9-14

10 LE DÉMARRAGE 10-1
Les grandes étapes du démarrage 10-2
Dépannage du démarrage 10-5
Init 10-6
Le script rc.sysinit 10-9
Démarrer avec LILO 10-10
Démarrer avec GRUB 10-12
Démarrer avec Syslinux 10-15
Initrd 10-16
Créer un live-cd, une live-key 10-17
ATELIERS 10-20

11 LA GESTION DES PERFORMANCES 11-1
Améliorer les performances, introduction 11-2
Surveiller le système 11-4
La commande vmstat 11-6
La commande sar 11-8
La comptabilité 11-10
Augmenter les ressources 11-11
Diminuer l'usage des ressources 11-13
ATELIERS 11-14

12 LE DÉPANNAGE 12-1
L'approche générale du dépannage 12-2
Etape n° 1 : Faire une description claire du problème 12-3
Etape n°2 : Imaginer les causes possibles 12-4
Etape n°3 : Rassembler des informations 12-5
Etape n°4 : Analyser les faits rassemblés, éliminer des causes 12-11
Un exemple de dépannage 12-12
Précautions 12-16
Régler le problème sans le résoudre 12-17
ATELIERS 12-18

13 COMPLÉMENTS 13-1
L'environnement 13-2
La localisation 13-4

L'heure 13-6
Le clavier 13-8
Chkconfig 13-10
Logrotate 13-11
Les travaux périodiques 13-13
Anacron 13-14
Les terminaux et la liaison série 13-15
ATELIERS 13-17

A ANNEXES A-1
Annexe A : L'éditeur vi A-2
Annexe B : Quelques numéros de ports udp/tcp A-3
@ Annexe C: Le programme breaksys

@ l'annexe C est à télécharger, rendez-vous sur le site www.editions-eyrolles.com, tapez **G12154** dans la zone <Recherche> et validez par <Entrée>.

INDEX I-1

Progression pédagogique

1) Administration Linux professionnelle
2) L'installation
3) Le noyau
4) La gestion des périphériques
5) Les systèmes de fichiers
6) Compléments sur les disques et les FS
7) Le RAID
8) Le LVM
9) La sauvegarde
10) Le démarrage
11) La gestion des performances
12) Le dépannage
13) Compléments

Administration Linux professionnelle

Ce chapitre donne des conseils pour administrer au mieux son système. De manière concrète, le lecteur apprend à utiliser le système de gestion de version RCS. Il apprend également à créer une page de manuel et à créer des comptes utilisateurs possédant des prérogatives d'administration grâce à la commande sudo.

L'installation

Après l'étude du chapitre, le lecteur sait automatiser l'installation de son système Linux. Il sait également créer un serveur d'installation ainsi qu'un serveur de dépôt de logiciel. Enfin il apprend à créer des packages.

Le noyau

Dans ce module, le lecteur apprend à compiler le noyau. Il étudie également son paramétrage à chaud. D'abord en apprenant la gestion de ses modules et ensuite en sachant modifier ses paramètres. Enfin, le lecteur améliore sa connaissance du noyau en étudiant les IPC et le FS /proc.

La gestion des périphériques

Dans ce module, le lecteur découvre comment sont gérés les périphériques sous Linux. En pratique, il apprend à recueillir des informations concernant le matériel à partir de plusieurs sources dont le FS Sysfs. Il apprend également à utiliser les technologies Udev, lm_sensors et SMART.

Les systèmes de fichiers

Après ce module, le lecteur maîtrise la gestion du système de fichier Ext3. Il a appris aussi à gérer les autres FS Linux : XFS, ReiserFS et NFS. Il sait comment le noyau gère les fichiers. Il sait également utiliser quelques FS exotiques comme tmpfs ou Unionfs. Enfin, il a étudié les techniques de montage à la volée des FS.

Compléments sur les disques et les FS

Ce module complète le précédent. On y apprend la gestion des quotas disques, des attributs et des ACL. La gestion de l'espace de Swap est étudiée également.

Le RAID

Dans ce module, le lecteur apprend le vocabulaire associé à la technique des disques RAID. Il découvre l'éventail des possibilités de gestion RAID. Il apprend à gérer le RAID logiciel inclus dans le système Linux.

Le LVM

Après l'étude du module, le lecteur sait gérer son espace disque de manière souple grâce à la technique LVM. Grâce à elle, il sait augmenter l'espace disque dédié à une application de manière dynamique. Il apprend aussi les techniques avancées du LVM (Snapshot, sauvegarde, déplacement des données, exportation, ...).

La sauvegarde

Ce module montre au lecteur les principales techniques de sauvegarde système et en premier lieu la sauvegarde incrémentale. Il étudie aussi les techniques de sauvegarde complète du système (bare metal). Enfin il apprend à gérer les lecteurs de cartouche.

Le démarrage

Après l'étude du chapitre, le lecteur comprend comment un système Linux démarre. Concrètement il sait paramétrer les principaux chargeurs (Grub, Lilo, Syslinux. Il sait également paramétrer la première application (init). Il a appris également à créer des live-cd ou des clés USB bootable.

La gestion des performances

Après l'étude du chapitre, le lecteur sait maintenir voire améliorer les performances de son système. Il a appris à surveiller l'usage des ressources par le biais de quantité de commandes, mais il a appris également à réagir à une baisse de performances.

Le dépannage

Après l'étude du chapitre, le lecteur connaît la méthodologie générale du dépannage. Il connaît également le fonctionnement de plusieurs commandes de diagnostic. Enfin, Il a pris connaissance de plusieurs conseils et heuristiques.

Compléments

Ce module traite de plusieurs sujets qui ne peuvent pas faire l'objet par eux-mêmes d'un chapitre, mais qui sont pourtant importants dans la maîtrise de l'administration d'un serveur. Ainsi, le lecteur apprend à régler les problèmes de localisation, de gestion de l'heure et les problèmes de clavier. Il apprend également les techniques de gestion périodiques complémentaires aux crontabs. Enfin il complète sa culture en apprenant les techniques de gestion de la liaison série et des terminaux sous Linux.

Quelques conventions de notation

Dans les parties théoriques

Dans les pages de ce livre, les commandes du système et les fichiers de paramètres sont expliqués en détail avec le contexte de leur utilisation.

Les noms des commandes sont imprimés en police Courier, par exemple :

`grub-install` Installe le chargeur Grub.

Les noms des fichiers sont imprimés en police Arial, par exemple :

/boot/grub/grub.conf Le fichier de configuration de GRUB.

Dans les ateliers

Ce livre fait une très large part aux ateliers pratiques à réaliser par le lecteur. Les commandes à passer et les résultats obtenus sont imprimés en police Courier sur un arrière-plan grisé. Les commandes mises en gras sont celles que vous devez taper au clavier, les autres informations sont celles qui sont affichées par le système.

```
[root@linux1 ~]# mount -t tmpfs tmpfs /mnt/disk
[root@linux1 ~]# free
             total       used       free     shared    buffers     cached
Mem:        449152     200580     248572          0      25832     144988
-/+ buffers/cache:      29760     419392
Swap:       522104          0     522104
[root@linux1 ~]# df -Th /mnt/disk
Filesystem    Type    Size  Used Avail Use% Mounted on
tmpfs         tmpfs   220M     0  220M   0% /mnt/disk
```

Dans certains exercices des lignes sont mises en italique, ce sont des lignes qu'il vous aura été demandé d'ajouter ou de modifier dans certains fichiers au cours des ateliers.

```
 [root@linux1 grub]# vi grub.conf
default=1
#timeout=5
title New Kernel (2.6.17)
        root (hd0,0)
        kernel /linux-2.6.17 ro root=LABEL=/1 rhgb
        initrd /initrd-2.6.17.img
title CentOS-4 i386 single
        root (hd0,0)
        kernel /vmlinuz-2.6.9-42.EL ro root=LABEL=/1 rhgb single
        initrd /initrd-2.6.9-42.EL.img
[root@linux1 grub]# cd
[root@linux1 ~]#
```

L'auteur a travaillé dans une version Linux RedHat avec la langue paramétrée pour le mode graphique (choix par défaut). Les caractères accentués sont alors mal restitués dans les sorties à l'écran en mode caractère, par exemple :

```
ArrÃªt de ntpd :          [ÃCHOUÃ]
```

Nous avons conservé ces sorties telles qu'elles sont, car c'est ce que verra le lecteur sur son système s'il se connecte de manière distante. Le module 13 donne toutes les explications et solutions.

Si vous travaillez en anglais (ce qui est conseillé dans le chapitre "installation"), vous n'aurez pas d'accents et donc pas de problèmes ! (L'anglais est ISO, les administrateurs Unix (et Windows) expérimentés travaillent généralement en anglais !).

- */etc/motd, wall*
- *script, diff, patch*
- *RCS, ci, co, ci -l*
- *groff, nroff, troff*
- */etc/sudoers, sudo*

1

Administration Linux professionnelle

Objectifs

Ce chapitre présente l'approche professionnelle de l'administration. Il donne des conseils pour administrer au mieux son système. De manière plus concrète, le lecteur apprend à utiliser le système de gestion de version RCS. Il apprend à créer une page de manuel. Il apprend à créer des comptes utilisateurs possédant des prérogatives d'administration grâce à la commande `sudo`.

Contenu

Philosophie de l'administration système

Automatiser les procédures

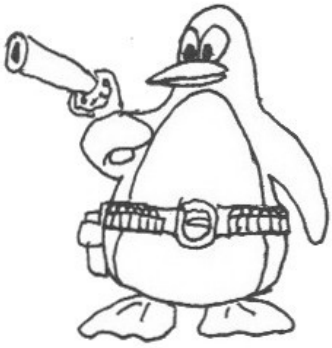
Journaliser les changements et agir de manière réversible

Documenter son système

Communiquer

Segmenter l'administration

Ateliers



Philosophie de l'administration système

La théorie

La philosophie de l'administration système en 10 points :

1. Votre exploitation doit répondre à un cahier des charges et à un budget.
2. Il faut faire en sorte que son système soit tolérant aux pannes.
3. Il faut automatiser les procédures.
4. Il faut journaliser les changements et agir de manière réversible.
5. Il faut documenter son système.
6. Il faut anticiper les problèmes.
7. Il faut communiquer.
8. Il faut connaître ses limites.
9. Il faut se documenter.
10. Il faut segmenter l'administration.

1. Votre exploitation doit répondre à un cahier des charges et à un budget.

On demande à l'administrateur que son système soit :

- Tolérant aux pannes.
- Performant.
- Sécurisé.
- Evolutif.
- Conforme aux standards.

On demande aussi directement à l'administrateur :

- Qu'il sache résoudre les problèmes qui surviennent.
- Qu'il soit réactif.

Enfin, l'administration n'est pas une fin en soi. Votre système sous-tend une application. Son bon fonctionnement est votre but principal.

L'ensemble de toutes ses doléances doit faire l'objet d'un cahier des charges qui indique clairement les priorités. D'autre part, ce cahier des charges doit être associé à un budget précis en temps/argent et personnel.

2. Il faut faire en sorte que son système soit tolérant aux pannes.

L'objectif minimaliste de l'administration système reste la tolérance aux pannes (« The show must go on »). Votre système est fragile. Un logiciel peut être bogué, votre matériel peut tomber en pannes, vous pouvez faire des erreurs d'exploitation. A tout moment, il faut être capable de réinstaller son système. Votre première mission doit être la continuité de l'exploitation. Selon le temps d'indisponibilité défini au niveau du cahier des charges, vous devez avoir soit une machine de remplacement directement utilisable et/ou être capable de réinstaller le système tel qu'il était avant le problème. Les solutions de type « cluster » ou de type « machine virtuelle » doivent être également envisagées.

3. Il faut automatiser les procédures (cf. section suivante)

4. Il faut journaliser les changements et agir de manière réversible (cf. section suivante)

5. Il faut documenter son système (cf. section suivante)

6. Il faut anticiper les problèmes.

Gouverner, c'est prévoir. Il faut imaginer les événements qui peuvent survenir et qui peuvent affecter l'exploitation. Il faut mesurer leur gravité et leur probabilité d'occurrence. Voici quelques exemples d'événements à prévoir :

- Vous pouvez tomber malade, l'exploitation doit continuer.
- Votre machine ne démarre plus, que faire ?
- Une nouvelle version majeure de votre système est imminente, remet-elle en cause les procédures actuelles ?
- Le local technique brûle, comment redémarrer l'exploitation au plus tôt ?

7. Il faut communiquer (cf. section suivante).

8. Il faut connaître ses limites.

Un administrateur ne sait pas tout et ne sait pas tout faire. Il doit connaître ses limites. Il doit éventuellement sous-traiter en partie l'administration et l'exploitation de son système. En cas de problème il doit s'adresser au support (hot-line, ...). Il doit donc avoir les contrats correspondants.

9. Il faut se documenter.

L'administration de système informatique est malheureusement une activité qui nécessite des connaissances en perpétuelle évolution. L'administrateur doit consacrer un temps non négligeable à se tenir informé de ces évolutions.

Les *mailings lists* constituent un élément de choix pour se tenir au fait des nouveautés.

10. Il faut segmenter l'administration (cf. section suivante)

Pour en savoir plus

Internet

Red Hat Enterprise Linux 4. Introduction à l'administration système, chapitre « Philosophie de l'administration système »
http://www.redhat.com/docs/manuals/enterprise/RHEL-4-Manual/fr/admin-guide/

Les Mailing lists respectivement des systèmes RedHat, SuSE et Debian
http://www.redhat.com/mailman/listinfo
http://lists.suse.com/archive/
http://www.debian.org/MailingLists/subscribe

The USENIX Special Interest group for Sysadmins
http://www.sage.org

Annuaires de Mailing List
http://tile.net/lists/
http://lists.topica.com

Automatiser les procédures.

La théorie

L'automatisation de l'exploitation permet de gagner un temps précieux et de rendre plus fiable les procédures. Toutes les tâches répétitives doivent être automatisées.

Le savoir concret

Les langages de script :

`bash`	Le shell est l'outil minimaliste d'automatisation des procédures.
`awk`	Le langage de création de rapport.
`perl`	Le langage le plus puissant de l'exploitation.
`phyton`	Un langage objet très clair et très puissant.

Les outils complémentaires d'automatisation.

`crontab`	La commande `crontab` permet d'activer périodiquement des tâches.
`make`	La commande `make` automatise la création de fichiers en fonction de règles de production.
`m4`	Préprocésseur généraliste.
`expect`	Automatise des traitements interactifs (du genre : question/réponse).

Pour en savoir plus

Les pages de manuel

awk(1), perl(1), phython(1), crontab(1), make(1), m4(1), expect(1)

Internet

Le langage Shell (Bash)
http://www.gnu.org/software/bash/manual/bash.html

Bash Guide for beginners
http://www.tldp.org/LDP/Bash-Beginners-Guide/html/

Le langage Perl
http://www.perl.org

Le langage Python
http://www.python.org

Livres

Unix Shell, par A. Berlat & all, chez Eyrolles. Ce livre traite du shell (bash, ksh) et de l'outil AWK.

Introduction à Perl, par R.Schartz & Tom Chistiansen, chez O'Reilly.

Apprendre à programmer avec Python, par Gérard Swinnen, chez O'Reilly.

Programmer avec les outils GNU, par M. Loukides & Andy Oram, chez O'Reilly. Ce livre traite notamment des commandes make, RCS, …

Journaliser les changements et agir de manière réversible

La théorie

Lorsque l'on fait du dépannage, la question récurrente est : quelles sont les modifications récentes qui ont été apportées au système. Il est important de les consigner dans un journal de bord.

Non seulement il faut consigner les changements, mais il faut (autant que faire se peut) agir de manière réversible. Les sauvegardes, notamment des configurations, sont dans cette optique, primordiales.

Le savoir concret

Les commandes

`script`	Enregistre une session shell.
`logger`	Envoie un message au système de journalisation (syslogd).
RCS	Ensemble de commandes permettant un suivi de fichiers texte.
`diff`	Affiche ou enregistre les modifications d'un ou plusieurs fichiers.
`patch`	Apporte une modification à un ou plusieurs fichiers.

Les commandes RCS

`ci`	Enregistre les modifications d'un fichier dans une archive, supprime l'original.
`ci -l`	Idem, mais on est immédiatement prêt pour faire évoluer le fichier.
`ci -u`	Idem, mais le fichier récupéré n'est pas verrouillé (il est en lecture seule).
`co`	Récupère une version d'un fichier, par défaut, la dernière.
`co -l`	Récupère une version d'un fichier pour la faire évoluer.
`rcs`	Change les caractéristiques d'un fichier.
`rcs -u`	Supprime le verrou de la version courante.
`rcs -l`	Verrouille la version courante.
`rlog`	Liste l'historique des versions.
`ident`	Affiche les commentaires RCS.

Focus : Le cycle de vie avec RCS

0) On crée un répertoire RCS.

1) On archive une version avec ci –l.

2) On utilise le fichier ou on le modifie.

On boucle sur les opérations 1 et 2.

Focus : Les commentaires RCS

Les commentaires RCS sont mis à jour automatiquement lors de l'extraction d'une version par la commande `co`.

`$Header$` ou `$Id$` Chemin du source, version, date.

`$Author$, $Date$, $Revision$, ...`

Alternatives aux systèmes de gestion de révisions

Solutions plus simple

Les systèmes de gestion de révision tels que RCS, CVS ou Subversion sont adaptés à gérer l'historique des modifications d'un fichier ainsi qu'à leur sauvegarde. A défaut de les utiliser, il faut au minimum garder une trace des modifications en copiant les fichiers d'administration avant de les modifier. Une solution simple est d'ajouter un suffixe à chaque copie qui indique un numéro d'ordre où l'époque de la copie :

```
# cp  fichier   fichier.001 ;  vi fichier
# cp  fichier   fichier.002 ;  vi fichier
```

etc...

```
# cp  fichier   fichier.`date '+%m%d%H%M'`   ; vi fichier
```

Gestion de révisions manuelles

Une autre possibilité est de périodiquement enregistrer les modifications effectuées sur un fichier ou sur une arborescence de fichiers par la commande `diff`. Il faut par contre conserver l'ancienne version des fichiers pour construire ces différences. Ensuite il est possible, avec la commande `patch` de reconstituer les fichiers à partir des fichiers d'origine.

On peut aussi conserver des sauvegardes différentielles de la configuration (/etc, ...). La sauvegarde de référence étant réalisée après l'installation complète.

Cfengine

Le logiciel libre Cfengine est un outil de configuration de parc de machines Unix/Linx en client/serveur. Cfengine utilise un langage spécifique qui décrit les stratégies de mises à jour (on peut répartir les machines en classes). Les stratégies sont stockées sur un serveur, mais ce sont les clients qui réalisent les opérations. Les fichiers modifiés sont sauvegardés et le logiciel tient à jour un journal spécifique.

Pour en savoir plus

Man

logger(1), script(1), rcsintro(1), ci(1), co(1), rcs(1), rlog(1), ident(1), cvs(1), diff(1), patch(1)

Internet

Cfengine
http://www.cfengine.org

Livres

Programmer avec les outils GNU, par M. Loukides & Andy Oram, chez O'Reilly. Ce livre traite notamment des commandes make, RCS, ...

Documenter son système

La théorie

Il faut conserver la documentation écrite. Il faut savoir accéder à la documentation électronique. Il faut si besoin écrire cette documentation par exemple sous forme de page de manuel.

Une page de manuel est un fichier texte compressé qui contient des macros `troff`. `Troff` est un langage de traitement de texte qui génère des commandes pour un affichage graphique (PostScript, ...). La commande `nroff` génère une vision texte d'un fichier contenant des macros `troff`.

Le savoir concret

Les macros troff d'en-tête d'une page de manuel

.Dd Month day, year

.Os [OPERATING_SYSTEM] [version/release]

.Dt DOCUMENT_TITLE [section number] [architecture/volume]

Les macros troff associées aux différentes sections d'une page de manuel

.Sh NAME

.Sh SYNOPSIS

.Sh DESCRIPTION

.Sh RETURN VALUES

.Sh ENVIRONMENT

.Sh FILES

.Sh EXAMPLES

.Sh COMPATIBILITY

.Sh ERRORS

.Sh SEE ALSO

.Sh STANDARDS

Les commandes

`troff`	Génère un fichier graphique (PS, ...) à partir d'une source troff.
`nroff`	Génère un fichier texte à partir d'un source nroff.
`groff`	La commande GNU qui émule les commandes nroff/troff.

La commande groff (Gnu-roff)

`groff -Tascii -man fichier`	Génère un fichier texte.
`groff -Tps -man fichier`	Génère un fichier PostScript.

Fichier : un fichier contenant des macros nroff/troff

Focus : imprimer une page de manuel

```
# man cal | col -b | lp
```

Pour en savoir plus.

Man

man(1), man(7), mdoc(7), man2html(1), col(1)

La page la plus importante (décrit les macros troff associées au manuel): mdoc.samples(7),

Internet

Docbook : système basé sur XML qui permet de générer une documentation dans le format de son choix (texte, PS, html, man, …)
http://www.docbook.org/

Communiquer

La théorie

L'administrateur est au centre d'un réseau humain. Sa hiérarchie qui désire que l'exploitation fonctionne correctement au moindre coût. Les responsables d'application qui vous demandent un maximum de ressources. Et enfin les utilisateurs dont vous êtes le premier interlocuteur. Il faut que vous communiquiez avec tous pour travailler le plus en harmonie. Vous devez faire connaître à tous vos contraintes et leur faire savoir les mesures que vous mettez en œuvre pour atteindre vos objectifs.

Le savoir concret

Les commandes

`mail`	Cette commande envoie un courrier.
`write`	Cette commande permet un dialogue avec un autre utilisateur.
`talk`	Cette commande permet un dialogue avec un utilisateur distant.
`wall`	Cette commande affiche un message sur l'ensemble des terminaux.
`ircii`	Outil de discussion à plusieurs (chat) sur Intranet/Internet.

Remarque : Les systèmes Unix disposent de la commande `news` qui affiche les fichiers contenus dans le répertoire /var/news. La commande n'affiche que les fichiers que l'on n'a pas encore consultés.

Les fichiers

/etc/issue	Le contenu de ce fichier apparaît avant la connexion.
/etc/motd	Le contenu de ce fichier apparaît en début de connexion.
/etc/nologin	Le contenu de ce fichier apparaît si les connexions sont interdites.
/etc/aliases	Ce fichier permet de définir de petites listes de diffusion. Après avoir modifié ce fichier, il faut activer la commande `newaliases`.

Pour en savoir plus

Les pages de manuel

mail(1), write(1), talk(1), wall(1), ircii(1), issue(5), motd(5), nologin(8), aliases(5), newaliases(1)

Segmenter l'administration

La théorie

Dans une petite exploitation, l'administrateur fait tout : il administre le système, les applications et passe le balai dans la salle où est enfermé le serveur.

Quand l'administration se complique, l'administration doit être répartie sur plusieurs têtes. Alors il faut bien sûr garder un responsable en chef.

L'outil `webmin` permet de définir des comptes d'administration spécialisés dans une tâche précise : l'impression, les sauvegardes, ... La commande `sudo` permet en final la même chose : telle commande nécessitant à priori les prérogatives de root peut être exécutée par un compte ordinaire.

Le savoir concret

La commande sudo

La commande `sudo` permet à un utilisateur d'exécuter des commandes qui nécessitent une identité différente de l'utilisateur, par exemple celle de root. Exemple :

```
pierre$ sudo useradd paul
```

Le fichier /etc/sudoers

Ce fichier décrit qui peut activer une commande avec `sudo` et sous quelle identité.

Syntaxe d'une ligne d'endossement de privilèges

Qui hôte = (identité) commande [, ...]

Exemple :

```
pierre  ALL = (root) /usr/bin/useradd, /usr/bin/usermod
```

L'utilisateur pierre peut activer les commandes `useradd` et `usermod` avec les droits de root.

Les alias

Pour simplifier l'écriture du fichier, on définit des alias (il faut spécifier la catégorie de l'alias en 1ere colone). Exemple :

```
User_Alias          ADM_USERS = pierre, paul
Runas_Alias    OPERATOR = root
Cmd_Alias      USER_CMDS = /usr/bin/useradd, /usr/bin/usermod
ADM_USERS  ALL = (OPERATOR) USER_CMDS
```

Mot de passe

Par défaut, la commande `sudo` demande le mot de passe de l'utilisateur. Le mot clé NOPASSWD peut précéder les commandes. Dans ce cas, le mot de passe n'est pas demandé.

Example :

```
%admins ALL = (root) NOPASSWD:  USER_CMDS
```

Remarque : un groupe est précédé du caractère %.

Pour en savoir plus

Man

sudo(8), sudoers(5), visudo(8)

ATELIERS



(À faire après l'installation)

Tâche 1 : Exemple d'automatisation d'exploitation 10mn

Tâche 2 : Exemple d'utilisation des commandes expect et m4 ..... 5mn

Tâche 3 : Journaliser avec script, logger, communiquer avec motd et wall 10mn

Tâche 4 : Tracer les modifications d'un fichier de configuration avec RCS 10mn

Tâche 5 : Ecrire une page de manuel 15mn

Tâche 6 : Utiliser la commande sudo 10mn

Tâche 7 : Les commandes diff et patch 5mn

Tâche 1 : Exemple d'automatisation d'exploitation

Créer un script qui envoie un mail à l'administrateur quand un des disques est plein (rempli à plus de x %). Automatiser son exécution par un crontab ;

1. Créer une commande qui surveille le taux de remplissage des disques.

```
[root@linux1 ~]# vi /usr/local/sbin/disk_full
#!/bin/sh
# disk_full - test si les disques sont plein
export LANG=C
:   ${SEUIL:=80}
message=/tmp/bidon$$

df -l | grep -v '^Filesystem' | sed 's/%//' |
        while read fs total used avail capacity mount
do
        if [ $capacity -gt $SEUIL ];then
                printf "%-20s %5d%%\n" $mount $capacity >> $message
        fi
done
if [ -r $message ];then
        mail -s "LES DISQUES SONT PLEINS" root < $message
        rm -f $message
fi
[root@linux1 ~]# chmod +x /usr/local/sbin/disk_full
```

2. Tester la commande.

```
[root@linux1 ~]# export SEUIL=20
[root@linux1 ~]# disk_full
[root@linux1 ~]# mail
Mail version 8.1 6/6/93.  Type ? for help.
"/var/spool/mail/root": 1 message 1 new
>N  1 root@linux1.pinguins  Sat Dec 30 12:42  16/629   "LES DISQUES SONT PLEINS"
& 1
Message 1:
From root@linux1.pinguins  Sat Dec 30 12:42:09 2006
Date: Sat, 30 Dec 2006 12:42:09 +0100
From: root <root@linux1.pinguins>
To: root@linux1.pinguins
```

```
Subject: LES DISQUES SONT PLEINS

/                        64%

& q
Saved 1 message in mbox
[root@linux1 ~]#
```

3. Créer un crontab pour automatiser la surveillance.

```
[root@linux1 ~]# crontab -e
0 12 * * * /usr/local/sbin/disk_full
[root@linux1 ~]#
```

Remarque : Sur les systèmes RedHat, en standard vous avez un `crontab` journalier qui active la commande `logwatch` et qui notamment affiche le taux d'utilisation des disques (cf. logwatch(8)).

Tâche 2 : Exemple d'utilisation des commandes expect et m4

1. Créer un script expect et on le teste.

```
[root@linux1 ~]# rpm -q expect
expect-5.42.1-1
[root@linux1 ~]# vi connexion.expect
#!/usr/bin/expect -f
set timeout 3
spawn ssh -l guest localhost
match_max 100000
expect -nocase "password: "
send -- "wwii1945\n"
expect "$ "
send -- "date > /tmp/date.log\r"
sleep 1
send -- "exit\r"
expect eof
[root@linux1 ~]# chmod +x connexion.expect
[root@linux1 ~]# ./connexion.expect
[root@linux1 ~]# cat /tmp/date.log
Sat Dec 30 14:38:53 CET 2006
[root@linux1 ~]#
```

2. Créer un fichier utilisant le préprocésseur M4.

```
[root@linux1 ~]# vi source.m4
dnl Ceci est un commentaire
dnl ========================
define(VER,`3.14')dnl
define(`fct1',`bonjour Mr $1')dnl
include(`date.txt')dnl
Version: VER
fct1(Dupont)
Goodbye

[root@linux1 ~]# date > date.txt
[root@linux1 ~]# m4 source.m4
Sat Mar 17 14:31:31 CET 2007
Version: 3.14
bonjour Mr Dupont
```

```
Goodbye
```

Tâche 3 :
Journaliser avec script, logger, communiquer avec motd et wall

1. Enregistrer ses actions avec la commande script.

```
[root@linux1 ~]# script -a
Script started, file is typescript
[root@linux1 ~]# date
Sun Dec 31 23:33:33 CET 2006
[root@linux1 ~]# id
uid=0(root) gid=0(root)
groups=0(root),1(bin),2(daemon),3(sys),4(adm),6(disk),10(wheel)
[root@linux1 ~]# uname -a
Linux linux1.pinguins 2.6.9-11.ELsmp #1 SMP Fri May 20 18:26:27 EDT 2005 i686
athlon i386 GNU/Linux
[root@linux1 ~]# uptime
 23:33:51 up 2 days, 10:55,  1 user,  load average: 0.00, 0.00, 0.17
[root@linux1 ~]# exit
exit
Script done, file is typescript
[root@linux1 ~]# cat typescript
Script started on Sun 31 Dec 2006 11:33:28 PM CET
[root@linux1 ~]# date
Sun Dec 31 23:33:33 CET 2006
[root@linux1 ~]# id
uid=0(root) gid=0(root)
groups=0(root),1(bin),2(daemon),3(sys),4(adm),6(disk),10(wheel)
[root@linux1 ~]# uname -a
Linux linux1.pinguins 2.6.9-11.ELsmp #1 SMP Fri May 20 18:26:27 EDT 2005 i686
athlon i386 GNU/Linux
[root@linux1 ~]# uptime
 23:33:51 up 2 days, 10:55,  1 user,  load average: 0.00, 0.00, 0.17
[root@linux1 ~]# exit
exit

Script done on Sun 31 Dec 2006 11:33:56 PM CET
[root@linux1 ~]#
```

Remarques :

1) L'option –a permet d'ajouter l'historique à la fin du fichier, par défaut ce dernier est écrasé.

2) La commande `script` peut être mise dans le fichier `~/.bash_profile` de l'administrateur. Ainsi, elle est déclenchée automatiquement ;

2. Journaliser les actions de l'administrateur avec syslog.

```
[root@linux1 ~]# echo "local1.*       /var/log/journal_de_bord.log" >>
/etc/syslog.conf
[root@linux1 ~]# pkill -1 syslogd
[root@linux1 ~]# logger -p local1.info  "aujourd'hui j'installe Oracle v10"
[root@linux1 ~]# logger -p local1.warn  "Les disques etaient pleins, j'ai fait
le menage"
[root@linux1 ~]# tail /var/log/journal_de_bord.log
Dec 31 23:27:46 linux1 root: aujourd'hui j'installe Oracle v10
Dec 31 23:31:39 linux1 root: Les disques etaient pleins, j'ai fait le menage
[root@linux1 ~]#
```

3. Prévenir les utilisateurs en utilisant le fichier motd.

```
[root@linux1 ~]# vi /etc/motd
============================

   Aujourd'hui, on arrete
   le systeme vers 16h
   pour installer la
   derniere version d'Oracle

============================
[root@linux1 ~]# ssh -l guest localhost
guest@localhost's password: wwii1945
Last login: Sat Dec 30 14:38:52 2006 from linux1.pinguins
============================

   Aujourd'hui, on arrete
   le systeme vers 16h
   pour installer la
   derniere version d'Oracle

============================

[guest@linux1 ~]$ exit
Connection to localhost closed.
[root@linux1 ~]#
```

4. Prévenir immédiatement les utilisateurs avec wall.

```
[root@linux1 ~]# echo "URGENT: les applications accedant au CDROM vont etre
tuees" |wall

Broadcast message from root (Sun Dec 31 23:46:15 2006):

URGENT: les applications accedant au CDROM vont etre tuees
[root@linux1 ~]#
```

Tâche 4 : Les commandes RCS, leur utilisation pour tracer les modifications d'un fichier de configuration

1. Créer un répertoire RCS qui contiendra l'historique des modifications.

```
[root@linux1 ~]# cd /etc/cups
[root@linux1 cups]# mkdir RCS
```

2. Créer le fichier RCS.

Le fichier RCS cupsd.conf,v contiendra l'historique des modifications du fichier cupsd.conf. La création de ce fichier entraîne la destruction du fichier de référence (cupsd.conf) si l'on n'utilise pas l'option –u ou -l. Avec l'option –l, le fichier peut être modifié de suite.

```
[root@linux1 cups]# ci -l cupsd.conf
RCS/cupsd.conf,v  <--  cupsd.conf
enter description, terminated with single '.' or end of file:
NOTE: This is NOT the log message!
>> The original of cupsd.conf file
>> .
initial revision: 1.1
done
```

```
[root@linux1 cups]# ls RCS
cupsd.conf,v
[root@linux1 cups]# ls -l cupsd.conf*
-rw-r-----  1 root root 22153 jan 30 15:30 cupsd.conf
-rw-r--r--  1 root root 21943 jan 20 10:21 cupsd.conf.save
```

3. Mettre à jour le fichier.

En l'occurrence, on veut modifier la configuration de CUPS. La modification concerne l'ouverture du port 631 de la carte réseau Ethernet. On pourra ainsi administrer à distance CUPS. On rajoute le commentaire RCS en tête du fichier (il y en a déjà un, mais on l'ignore). La ligne `Listen` est rajoutée après les commentaires concernant cette directive. Dans la balise `Location /admin`, on met en commentaire les directives associées à la sécurité (`Order`, `Deny` et `Allow`).

```
[root@linux1 cups]# netstat -an | grep 631
tcp        0      0 127.0.0.1:631               0.0.0.0:*
LISTEN
udp        0      0 0.0.0.0:631                 0.0.0.0:*
[root@linux1 cups]# vi cupsd.conf
# $Id$
...
Listen 192.168.0.1:631
...
<Location /admin>
...
## Restrict access to local domain
#Order Deny,Allow
#Deny From All
#Allow From 127.0.0.1

#Encryption Required
</Location>
```

5. Archiver la dernière version et on relance le service.

```
[root@linux1 cups]# ci -l cupsd.conf
RCS/cupsd.conf,v  <--  cupsd.conf
new revision: 1.2; previous revision: 1.1
enter log message, terminated with single '.' or end of file:
>> We open the port
>> .
done
[root@linux1 cups]# /etc/init.d/cups restart
Stopping cups:                                             [  OK  ]
Starting cups:                                             [  OK  ]
[root@linux1 cups]# netstat -an |grep 631
tcp        0      0 127.0.0.1:631               0.0.0.0:*                   LISTEN
tcp        0      0 192.168.0.1:631             0.0.0.0:*                   LISTEN
udp        0      0 0.0.0.0:631                 0.0.0.0:*
```

6. Visualiser les commentaires RCS.

```
[root@linux1 cups]# ident cupsd.conf
cupsd.conf:
     $Id: cupsd.conf,v 1.2 2006/12/31 21:38:22 root Exp $
```

7. Visualiser l'historique des modifications.

```
[root@linux1 cups]# rlog cupsd.conf
```

```
RCS file: RCS/cupsd.conf,v
Working file: cupsd.conf
head: 1.2
branch:
locks: strict
access list:
symbolic names:
keyword substitution: kv
total revisions: 2;     selected revisions: 2
description:
The original of cupsd.conf file
----------------------------
revision 1.2
date: 2006/12/31 21:38:22;  author: root;  state: Exp;  lines: +6 -6
We open the port
----------------------------
revision 1.1
date: 2006/12/31 21:27:40;  author: root;  state: Exp;
Initial revision
=============================================================================
[root@linux1 cups]#
```

Tâche 5 : Ecrire une page de manuel

1. Écrire la commande news.

```
[root@linux1 ~]# vi /usr/local/bin/news
#!/bin/sh
# news
option=$1

if [ -e ~/.news_time ];then
        FILES=`find /var/news -type f -newer ~/.news_time `
        if [ "$FILES" != "" ];then
                more $FILES
        fi
fi
if [ ! -e ~/.news_time -o N$option = "N-a" ];then
        more /var/news/*
fi
touch ~/.news_time

[root@linux1 ~]# chmod 555 /usr/local/bin/news
```

2. Écrire une page de manuel pour cette commande

```
[root@linux1 ~]# vi /usr/local/man/man1/news.1
.Dd August 1, 2006
.Dt NEWS 1
.Sh NAME
news \- print news items
.Sh SYNOPSIS
news [-a]
.Sh DESCRIPTION
news is used to keep the user informed of current events. By convention, these
events are described by files in the directory /var/news.
```

```
.Pp
When invoked without arguments, news prints the contents of all current files in
/var/news. news stores the ``currency'' time as the modification date of a file
named .news_time in the user's home directory; only files more recent than this
currency time are considered ``current.''
.Sh OPTIONS
-a
Print all items, regardless of currency.
.Sh FILES
.nf
/var/news/*
$HOME/.news_time
```

3. Tester si la page s'affiche correctement.

```
[root@linux1 ~]# groff -Tascii -man /usr/local/man/man1/news.1
NEWS(1)                   BSD General Commands Manual                   NEWS(1)

NAME
     news - print news items

SYNOPSIS
     news [-a]

DESCRIPTION
     news is used to keep the user informed of current events. By convention,
     these events are described by files in the directory /var/news.

     When invoked without arguments, news prints the contents of all current
     files in /var/news. news stores the ``currency'' time as the modification
     date of a file named .news_time in the user's home directory; only files
     more recent than this currency time are considered ``current.''

OPTIONS
     -a Print all items, regardless of currency.

FILES
     /var/news/*
     $HOME/.news_time


BSD                             August 1, 2006                             BSD4.
On la compresse.
[root@linux1 ~]# gzip /usr/local/man/man1/news.1
[root@linux1 ~]# ls -l /usr/local/man/man1/new*
-rw-r--r--  1 root root 402 Dec 30 11:26 /usr/local/man/man1/news.1.gz
```

5. Tester son accès avec la commande man.

```
[root@linux1 ~]# man news
No manual entry for news
[root@linux1 ~]# MANPATH=/usr/share/man:/usr/local/man
[root@linux1 ~]# export MANPATH
[root@linux1 ~]# man news
NEWS(1)                   BSD General Commands Manual                   NEWS(1)

NAME
```

```
     news - print news items

SYNOPSIS
     news [-a]

DESCRIPTION
....
```

6. Incorporer la commande dans l'index des commandes.

```
[root@linux1 ~]# makewhatis /usr/local/man
[root@linux1 ~]# man -k news
news                 (1)  - print news items
[root@linux1 ~]#
```

7. Visualiser le texte source d'une page de manuel.

```
[root@linux1 ~]# zcat /usr/share/man/man8/init.8.gz  | more
.\"{{{}}}
.\"{{{  Title
.TH INIT 8 "18 April 2003" "" "Linux System Administrator's Manual"
.\"}}}
.\"{{{  Name
.SH NAME
init, telinit \- process control initialization
.\"}}}
.\"{{{  Synopsis
.SH SYNOPSIS
...
```

8. Tester la commande news.

```
[root@linux1 ~]# mkdir /var/news
[root@linux1 ~]# echo "Oracle V9 est obsolete" > /var/news/msg1
[root@linux1 ~]# echo "Oracle V10 va etre installee" > /var/news/msg2
[root@linux1 ~]# su - guest
[guest@linux1 ~]$ news
:::::::::::::::
/var/news/msg2
:::::::::::::::
Oracle V10 va etre installee
:::::::::::::::
/var/news/msg1
:::::::::::::::
Oracle V9 est obsolete
[guest@linux1 ~]$ exit
[root@linux1 ~]# echo "le systeme sera arrete Lundi" > /var/news/msg3
[guest@linux1 ~]$ news      # on ne voit que les fichiers récents !
le systeme sera arrete Lundi
[guest@linux1 ~]$ news -a
:::::::::::::::
/var/news/msg1
:::::::::::::::
Oracle V9 est obsolete
:::::::::::::::
/var/news/msg2
:::::::::::::::
Oracle V10 va etre installee
```

```
::::::::::::::
/var/news/msg3
::::::::::::::
le systeme sera arrete Lundi
[guest@linux1 ~]$
```

Tâche 6 : Utiliser sudo

1. Créer un compte.

```
[root@linux1 ~]# useradd -m pierre
[root@linux1 ~]# echo pierre | passwd --stdin pierre
Changing password for user pierre.
passwd: all authentication tokens updated successfully.
[root@linux1 ~]#
```

2. Modifier la configuration : pierre peut créer des comptes utilisateurs.

```
[root@linux1 ~]# cp /etc/sudoers /etc/sudoers.000
[root@linux1 ~]# visudo
...
pierre  linux1.pinguins = (root) /usr/sbin/useraddd
```

3. Se connecter sous le compte pierre.

```
[root@linux1 ~]# ssh -l pierre localhost
pierre@localhost's password: pierre
Last login: Sat Dec 30 10:33:02 2006 from raichu.pokemon
[pierre@linux1 ~]$ sudo /usr/sbin/useradd -m titeuf

We trust you have received the usual lecture from the local System
Administrator. It usually boils down to these two things:

        #1) Respect the privacy of others.
        #2) Think before you type.

Password: pierre
[pierre@linux1 ~]$
```

4. Créer une configuration plus réaliste.

```
[root@linux1 ~]# cp /etc/sudoers /etc/sudoers.`date '+%m%d%H%M'`
[root@linux1 ~]# ls -l /etc/sudoers*
-r--r-----  1 root root 621 Dec 30 10:30 /etc/sudoers
-r--r-----  1 root root 580 Dec 30 10:14 /etc/sudoers.000
-r--r-----  1 root root 621 Dec 30 10:39 /etc/sudoers.12301039
[root@linux1 ~]# visudo
...
#pierre linux1 = (root) /usr/sbin/useradd

Cmnd_Alias CMDS_USER  =  /usr/sbin/useradd, \
                         /usr/sbin/userdel, \
                         /usr/sbin/usermod, \
                         /usr/sbin/groupadd, \
                         /usr/bin/passwd [A-z]*, ! /usr/bin/passwd root
%admins ALL = (root) NOPASSWD: CMDS_USER
```

5. Créer des administrateurs.

```
[root@linux1 ~]# groupadd admins
```

```
[root@linux1 ~]# useradd -m -g admins cathy
[root@linux1 ~]# echo cathy | passwd --stdin cathy
Changing password for user cathy.
passwd: all authentication tokens updated successfully.
[root@linux1 ~]#
```

6. Tester.

```
[root@linux1 ~]# su - cathy
[cathy@linux1 ~]$ sudo -l
User cathy may run the following commands on this host:
    (root) NOPASSWD: /usr/sbin/useradd, /usr/sbin/userdel, /usr/sbin/usermod,
/usr/sbin/groupadd, /usr/bin/passwd [A-z]*, !/usr/bin/passwd root
[cathy@linux1 ~]$ PATH=$PATH:/usr/sbin
[cathy@linux1 ~]$ sudo useradd -m vomito
[cathy@linux1 ~]$ id vomito
uid=506(vomito) gid=506(vomito) groups=506(vomito)
[cathy@linux1 ~]$
```

Tâche 7 : Les commandes diff et patch

On suppose que l'atelier sur RCS a été accompli.

```
[root@linux1 cups]# diff -c cupsd.conf cupsd.conf.save > patch.txt
[root@linux1 cups]# head patch.txt
*** cupsd.conf  2007-01-30 15:30:22.000000000 +0100
--- cupsd.conf.save     2007-01-20 10:21:52.000000000 +0100
***************
*** 1,6 ****
  #
! # $Id: cupsd.conf,v 1.2 2007/01/30 14:30:10 root Exp root $
! # "$Id: cupsd.conf,v 1.2 2007/01/30 14:30:10 root Exp root $"
  #
  #   Sample configuration file for the Common UNIX Printing System (CUPS)
  #   scheduler.
[root@linux1 cups]# cp cupsd.conf.save cupsd.conf.000
[root@linux1 cups]# cp cupsd.conf.save cupsd.conf.001
[root@linux1 cups]# patch -p0 cupsd.conf.001 < patch.txt
patching file cupsd.conf.001
Reversed (or previously applied) patch detected!  Assume -R? [n] y
[root@linux1 cups]# diff cupsd.conf cupsd.conf.001
[root@linux1 cups]#
```

La commande `diff` signale les différences entre deux fichiers. Si l'on utilise l'option –c et si l'on mémorise ces différences avec une redirection, on peut reconstituer le fichier à partir du fichier d'origine et du fichier `diff` (appelé ici `patch.txt`).

Remarque :

Les commandes `diff` et `patch` peuvent agir sur des arborescences (le répertoire courant correspond à l'arborescence avant les modifications et a_jour, celui ayant subi les modifications) :

```
# diff   -u  -r  .   ../a_jour  | gzip > patch.gz
# gunzip -c patch.gz  |  patch -p0
```

- *KickStart, AutoYaST*
- *repository*
- *Yum, Apt-get*
- *repodata,Packages.gz*
- *Spec, rpmbuild*

2

L'installation

Objectifs

Ce chapitre traite de L'installation du système. L'objectif principal est de savoir créer un serveur local d'installation et de mise à jour. On apprend également à créer des packages.

Contenu

Installer et organiser les données

L'installation automatique

L'installation Kickstart (RedHat)

Créer un serveur de dépôt

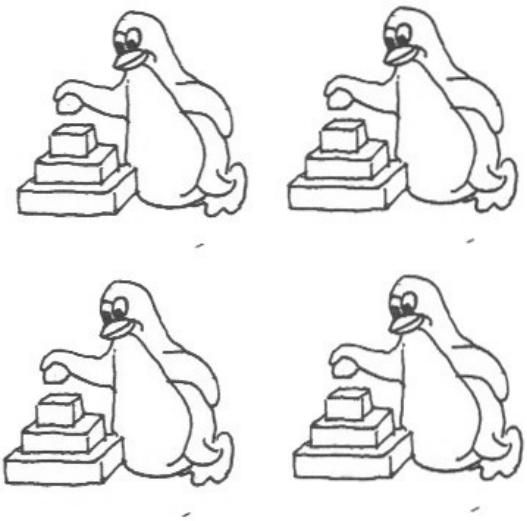
Créer un dépôt Debian

Créer un dépôt Yum, configurer le client

Créer un package RPM

Créer un package Debian

Ateliers



Installer et organiser les données

La théorie

Objectifs

Quand on installe un système, on veut simplifier l'exploitation. En clair on veut :

- Réinstaller le système facilement.
- Faciliter la réinstallation partielle du système, par exemple la réinstallation uniquement du système ou de l'application principale.
- Faciliter les mises à jour du système.
- On veut un système fiable, par exemple disposer du système sur un miroir.
- Avoir de bonnes performances, par exemple en mettant les données sur un RAID-5.

Stratégies

Stratégies globales

On structure l'espace disque, on sépare les éléments suivants :

- Le démarrage du système, le reste des fichiers et le swap (/boot, / et swap)
- Système et application (/, /usr, /var et /home, /usr/local/apache).
- Fichiers stables et fichiers fréquemment modifiés (/, /usr et /var, /tmp).
- Pour une application, les programmes, la configuration, les journaux, les données, … (./bin, ./etc, ./log, ./data, …)

Les éléments structurants

- Les arborescences de fichiers (/, /usr, /home, …).
- Les systèmes de fichiers (FS), chacun abritant une arborescence de fichiers.
- Les groupes de volumes (VG). Le VG système, le VG de l'application principal, le VG des autres applications. Un VG regroupant un ou plusieurs FS.
- Les disques internes ou externes. Les disques seront partitionnés et chaque partition abritera un FS, du swap ou un PV d'un VG. Les disques peuvent aussi être vus uniquement en tant que PV. Les disques d'un serveur sont gérés en RAID.

Exemples de stratégies

Remarque : On prend Apache comme exemple d'application.

Système mono disque :

- Partitionnement physique : /boot, /, swap, /usr/local/apache.
- Mixte (partitionnement physique et VG) : /boot, /, /usr, /var, /tmp et un PV associé à un VG apache lequel abritant les FS d'apache (./log, ./htdocs, …).
- VG : /boot (obligatoire) et le reste sous forme d'un PV associé à un VG divisé en LV (/, /usr/local/apache, swap, …). C'est la solution par défaut à l'installation.

Système possédant plusieurs disques :

Un disque système en miroir et les disques externes (RAID) répartis dans un ou plusieurs VG applicatif.

L'installation automatique

La théorie

L'installation, la réinstallation et la mise à jour des systèmes peuvent devenir répétitives. Dans ce cas il faut la simplifier, voire l'automatiser.

L'idée générale est de créer un serveur d'installation qui contient les logiciels systèmes et les applications ainsi que les mises à jour. Le poste à installer accède au serveur via le réseau. Grâce au protocole DHCP il obtient sa configuration réseau et via FTP, HTTP ou NFS il accède au dépôt des logiciels à installer.

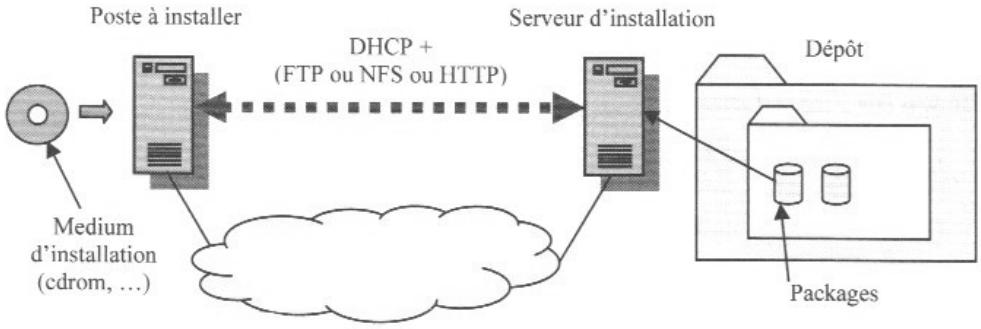


Fig. Installation réseau

Il faut activer l'installation à partir des postes à installer. Il faut que ceux-ci possèdent un système Linux en mémoire et qu'ils déclenchent la procédure d'installation. Pour cela, on démarre les postes à partir du premier CDrom d'installation et on spécifie au chargeur les paramètres permettant l'installation automatique. Il est également possible d'utiliser PXE. Dans ce cas, quand le poste démarre, il recherche un serveur DHCP compatible PXE et après avoir chargé un mini système grâce au protocole TFTP, l'installation commence. Dans ce cas, la procédure d'installation peut être entièrement automatique. Il est même possible que le démarrage des postes à installer soit piloté à distance.

Remarque : En théorie l'installation automatique n'est pas forcement réalisée en réseau. Mais son intérêt et la facilité de mise en œuvre l'impliquent presque obligatoirement.

Particularités des distributions

RedHat, Fedora, CentOS

L'installation automatique d'un système RedHat utilise la technologie Kickstart.

SuSE

L'installation automatique d'un système SuSE utilise la technologie AutoYaST.

Debian

L'installation automatique d'un système Debian utilise le logiciel debian-installer.

L'installation automatique d'un système Debian peut également utiliser le logiciel libre FAI (Fully Automatic Installation). Il est beaucoup plus sophistiqué (mais plus complexe) que le logiciel standard. Il avait été créé pour réaliser l'installation de cluster de calculs.

Le savoir concret

Le logiciel debian-installer (Debian)

Principe

Au lieu de répondre aux questions du logiciel standard d'installation (debian-installer), on peut lui fournir un fichier qui contient les paramètres d'installation.

Création du fichier de pré configuration (preseed.cfg)

La manière la plus simple pour créer le fichier de pré configuration est d'abord d'installer le système et de générer le fichier qui correspond à l'installation réalisée :

```
# debconf-get-selections --installer  > preseed.cfg
# debconf-get-selections   >> preseed.cfg
```

Eventuellement on édite ensuite le fichier.

Démarrer l'installation

Lors du démarrage à partir du medium d'installation (CDrom par exemple), on ajoute en paramètre du noyau l'emplacement du fichier. Par exemple :

```
boot:  linux  preseed/url=http://serveur/pub/preseed.cfg
```

Le logiciel AutoYaST (SuSE)

Rappels sur l'installation d'un système SuSE

L'installation d'un système SuSE se déroule en trois phases :

- La préparation : Le logiciel d'installation recueille les paramètres d'installation.
- L'installation : YaST partitionne, crée les FS, installe les packages et le chargeur.
- La configuration : Des scripts de post-installation configure le matériel, le réseau, ...

Principe

Au lieu de répondre aux questions lors de la première phase, on crée un fichier qui contient les paramètres d'installation.

Création du fichier de contrôle (autoyast.xml)

La manière la plus simple pour créer le fichier de pré configuration est d'abord d'installer le système et de générer le fichier qui correspond à l'installation réalisée.

Pour ce faire, on active l'`Auto-installation` du menu `Misc` de `Yast`. Ensuite, éventuellement on édite le fichier XML généré. On peut vérifier sa syntaxe :

```
# xmllint  autoyast.xml
```

Création du serveur d'installation

On recopie les CDroms d'installation dans un répertoire d'un serveur HTTP ou NFS.

Démarrer l'installation

Lors du démarrage à partir du medium d'installation (CDrom par exemple), on ajoute en paramètre du noyau l'emplacement du fichier. Par exemple :

a) Le fichier est sur une disquette :

```
autoyast=floppy://autoyast.xml
```

b) Le fichier est sur un serveur http :

```
autoyast=http://serveur/pub/suse/autoyast.xml
```

Pour en savoir plus

Howto

Network-installation-HOWTO (SuSe, RedHat et Debian)

Internet – Installation automatique d'une distribution Debian

L'installation automatique
http://www.debian.org/releases/stable/i386/ch04s07.html.fr

Un exemple de fichier preseed.cfg
http://www.debian.org/releases/stable/i386/apcs01.html.fr

Le manuel d'installation
http://www.debian.org/releases/stable/i386/index.html

Le logiciel FAI – Home Page
http://informatik.uni-kolen.de/fai

Annexe B. Automatisation de l'installation et préconfiguration d'un système Debian
http://d-i.alioth.debian.org/manual/fr.i386/apbs01.html

Internet – Installation automatique d'une distribution SuSE

AutoYast – la documentation officielle
http://forgeftp.novell.com/yast/doc/SL10.0/autoinstall/

Supinfo - Un exemple complet d'utilisation d'AutoYaST
http://www.supinfo-projects.com/fr/2005/autoinstall_linux_en/3/

L'installation Kickstart (RedHat)

La théorie

Les distributions RedHat, Fedora et CentOS utilisent le logiciel Anaconda pour leur installation. Il est possible de créer un fichier « Kickstart » qui contient toutes les réponses aux questions posées lors de l'installation. Ce fichier permet une installation automatique. Ce fichier peut être présent sur une disquette ou un serveur d'installation. Le système à installer, lui, peut être présent sur un DVD ou un serveur d'installation.

Le savoir concret

Les commandes

`ksconfig`, `redhat-config-kickstart`
: Cette commande graphique génère de toute pièce un fichier kickstart.

`ksconfig --generate fichier`
: Cette commande en mode texte génère un fichier kickstart qui reprend la liste des packages installés sur le système.

Les fichiers

/root/anaconda-ks.cfg
: Ce fichier kickstart est créé automatiquement lors de l'installation. Ce qui concerne le partitionnement est en commentaire.

Focus : Format d'un fichier Kickstart

Les options générales

Le type d'installation (cf. plus bas), le langage (lang), le clavier (keyboard), la configuration réseau (network), la sécurité (authconfig, firewall, selinux), la zone géographique (timezone), le mot de passe de root (rootpw), le partitionnement (par), le chargeur (bootloader), etc.

Remarque : Contrairement aux autres sections, celle-ci n'est pas introduite par un entête.

%packages

Cette ligne est suivie des packages à installer. Ce sont soit des packages individuels, soit des groupes de packages. Dans ce dernier cas, leur nom est précédé de « @ ».

%pre

Cette ligne est suivie des commandes à exécuter avant l'installation.

%post

Cette ligne est suivie des commandes à exécuter après l'installation.

Focus : Le type d'installation

Installation à partir du cdrom

```
install
cdrom
```

Installation réseau

a) En NFS

```
install
nfs --server=venus --dir=/var/ftp/pub
```

c) En FTP ou FTP (on spécifie une URL)

```
install
url --url http://server/pub
```

Remarque : dans les différents exemples, le répertoire indiqué contient l'arborescence RedHat. On peut créer celle-ci en recopiant l'arborescence RedHat des CDrom d'installation.

Focus : Démarrer l'installation

A l'invite du chargeur (boot:) on entre une des lignes suivantes :

a) Si on utilise un CDrom (ou un DVDrom) et une disquette qui contient le fichier kickstart.

```
linux ks=floppy:/ks.cfg
```

b) Si on utilise un serveur réseau NFS, HTTP ou FTP :

```
linux ks=nfs:server:/pub/ks.cfg
linux ks=http://server/pub/ks.cfg
linux ks=ftp://server/pub/ks.cfg
```

c) Si on utilise un fichier initrd :

```
linux ks=file:/<file>
```

Pour en savoir plus

Fichiers

La documentation du package /usr/share/doc/system-config-kickstart-*/.

Internet

RedHat (RHEL4) - System Administration Guide (inclus l'installation PXE)
http://www.redhat.com/docs/manuals/enterprise/RHEL-4-Manual/sysadmin-guide/ch-kickstart2.html

Installation Guide for x86, Itanium™, AMD64 and EM64T
http://www.redhat.com/docs/manuals/enterprise/RHEL-4-Manual/x8664-multi-install-guide/

Linux Journal - How to create a single CD for fast and easy customized installation.
http://www.linuxjournal.com/article/6473

KickStart Configurations & CFEngine
http://sial.org/howto/kickstart/

Créer un serveur de dépôt

La théorie

Pourquoi utiliser un serveur de dépôt

L'installation et la mise à jour des packages sont simplifiées si l'administrateur utilise des logiciels évolués comme Yum ou Apt-get qui gèrent les dépendances de package et qui automatiquement téléchargent ces packages à partir de dépôts et enfin les installent. L'administrateur peut configurer ces logiciels pour qu'ils accèdent à des dépôts officiels présents sur Internet ou configurer son propre serveur de dépôt.

L'administrateur tire plusieurs avantages de gérer son propre serveur de dépôt :

- Son serveur d'installation peut jouer le rôle de serveur de dépôt. Il réalise ainsi une centralisation de l'installation et une bonne utilisation des ressources.
- Il contrôle de manière absolue les mises à jour. Il peut par exemple ne mettre à disposition qu'une partie d'entre elles, celles qui ont été pleinement testées et qui sont donc sans risque.
- Il utilise son réseau interne qui est plus sur et plus performant.

Remarque : L'administrateur peut décider de transformer son serveur de dépôt en véritable miroir des serveurs officiels de sa distribution. Dans ce cas, son serveur est automatiquement à jour mais l'administrateur perd alors le contrôle des packages qu'il peut mettre ou non à disposition. Il peut tout au moins adopter cette attitude pour les mises à jour de sécurité.

La création d'un serveur de dépôt

La création d'un serveur de dépôt suppose plusieurs étapes :

- Il faut créer un dépôt correspondant aux packages de base de la distribution. Ce sont les même utilisés pour l'installation du système. On peut les obtenir en les téléchargeant à partir d'Internet ou en les recopiant à partir des CDroms d'installation.
- Il faut créer un dépôt correspondant aux mises à jour du système. Il faut une procédure pour récupérer ces mises à jour à partir d'Internet.
- Il faut installer et configurer un service pour mettre à disposition les dépôts. Selon le cas, on utilise HTTP, FTP, RSYNC, ou NFS.
- Pour chaque dépôt il faut créer les Métadata qui mémorisent entre autre les relations de dépendances entre packages. Chaque outil utilise des Métadata de nature différente. Eventuellement on peut créer plusieurs Métadata. Ainsi le même dépôt pourrait être accessible aussi bien par Yum que par Apt-get.

Particularités des distributions

RedHat, Fedora, CentOS

Les distributions Fedora et CentOS utilisent en standard l'outil `Yum` pour l'installation et la mise à jour des packages. Par défaut la configuration inclue le référencement de serveurs de dépôt présents sur Internet.

La distribution RedHat Enterprise utilise en standard l'outil `up2date` pour l'installation et la mise à jour des packages. Le poste doit être enregistré chez RedHat pour accéder à son réseau de mises à jour. Il est tout à fait possible d'utiliser Yum comme serveur local de dépôt. D'autre part, `up2date` peut utiliser un dépôt Yum.

Debian, Ubuntu

La distribution Debian et ses dérivées utilisent en standard l'outil `apt-get` pour l'installation et la mise à jour des packages.

SuSE

Le cas de SuSE est assez complexe. Les versions récentes utilisent l'outil `rug` pour l'installation et la mise à jour des packages. Cet outil accepte de dialoguer avec plusieurs types de dépôt, des dépôts au format spécifiquement SuSE (YaST ou ZENWorks) ou même des dépôts Yum. On peut aussi directement utiliser Yum pour accéder aux dépôts officiels.

Le savoir concret

SuSE

La configuration du client

Dans la terminologie SuSE, un dépôt est appelé « service ».

1) On ajoute un dépôt (un service).

```
# rug sa --type=yum http://serveur/pub  racine
```

Syntaxe : rug sa --type=<TYPE> URL nom_du_depot

La valeur TYPE précise le type de dépot : yum, zypp (YaST repository), nu, rce, zenworks.

2) On y souscrit.

```
# rug  sub  racine
```

On peut ensuite l'utiliser (lister son contenu, télécharger des packages, …).

Créer un dépôt SuSE

On peut tout simplement créer un dépôt Yum. La création d'un dépôt de type Zypp (YaST repository) est assez complexe (cf. « Creating YaST Installation Sources » donné en référence).

La mise à jour du miroir

Panorama des techniques

Techniques de bas niveau

`rsync` Protocole adapté au miroring de site.

`lftp` Outil de transfert de fichiers supportant les protocoles HTTP et FTP.

Techniques de haut niveau

`Yum-pull` et `mrepo` (anciennement `yam`) sont des outils qui permettent de mirorer des dépôts yum, apt ou up2date.

Remarque : Ces outils transfèrent un dépôt intégralement à la première utilisation. Ensuite, ils ne transfèrent que les nouveaux fichiers.

lftp – exemple

```
# cat script.lft
set  ftp:list-options -a
open serveur/pub/racine
lcd  /var/ftp/pub/racine
mirror --delete --exclude *.src.rpm  --verbose
quit
# lftp  -f  script.lftp
```

On transfère l'arborescence pub/racine présente sur le serveur serveur dans le répertoire local /var/ftp/pub/racine. L'option `--delete` supprime en local les fichiers qui ne sont plus présents sur le site distant. On ne transfert pas les fichiers ayant l'extension `src.rpm`.

rsync – exemple

```
# rsync serveur::      # lister les domaines d'un serveur rsync
ftp             ftp area
# rsync -av --exclude *.src.rpm serveur::ftp/pub/racine /var/ftp/pub
```

On transfère l'arborescence pub/racine du domaine ftp du serveur serveur dans le répertoire local /var/ftp/pub/racine. On ne transfert pas les fichiers ayant l'extension src.rpm.

Voici la configuration du serveur rsync correspondante :

```
serveur# cat /etc/rsyncd.conf
use chroot = yes
[ftp]
        path = /var/ftp/
        comment = ftp area
serveur# chkconfig rsync on
serveur# ls -F /var/ftp/pub/
racine/
```

Pour en savoir plus

Les pages de manuel

rsync(1), lftp(1)

Internet

Yum-pull
http://www.stearns.org/mirror/

Mrepo (yam)
http://dag.wieers.com/home-made/mrepo/

Creating YaST Installation Sources - openSUSE
http://fr.opensuse.org/Creating_YaST_Installation_Sources

Livre

Linux Patch Management : Keeping Linux System Up to date, par Michael Jang, chez Prentice Hall

Créer un dépôt Debian

Le savoir concret

La structure d'un dépôt Debian

Un dépôt Debian est constitué d'au moins un répertoire et de fichiers particuliers :

- Packages.gz Catalogue des packages binaires.
- Sources.gz Catalogue des packages sources.
- Release Fichier associé au dépôt lui-même (fichier optionnel).

Il y a deux sortes de dépôt Debian :

- Dépôt trivial Sa structure est très simple.
- Dépôt auto Il est compatible avec la structure des dépôts officiels Debian.

Dépôt trivial

Il est composé de deux répertoires, l'un pour les packages binaires, l'autre pour les packages sources :

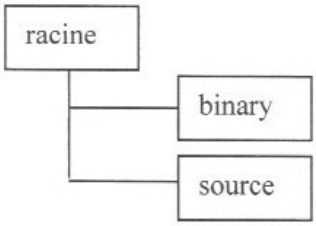


Fig. Dépôt trivial

Dépôt auto

Un dépôt auto reproduit la structure d'un dépôt officiel.

Chaque répertoire binary-* (un par type de plateforme matériel) contient un fichier Packages.gz et éventuellement un fichier Releases.

Chaque répertoire source contient un fichier Sources.gz et éventuellement un fichier Releases.

Remarques :

1) Les packages n'ont pas besoin d'être dans le même répertoire que les catalogues, car le chemin de chaque package est contenu dans les catalogues. Ceci permet la création de « pool ». Un pool est une arborescence de packages regroupés selon des critères différents des catalogues.

2) On peut créer autant de distributions ou de composant que l'on veut et on peut les nommer comme on veut.

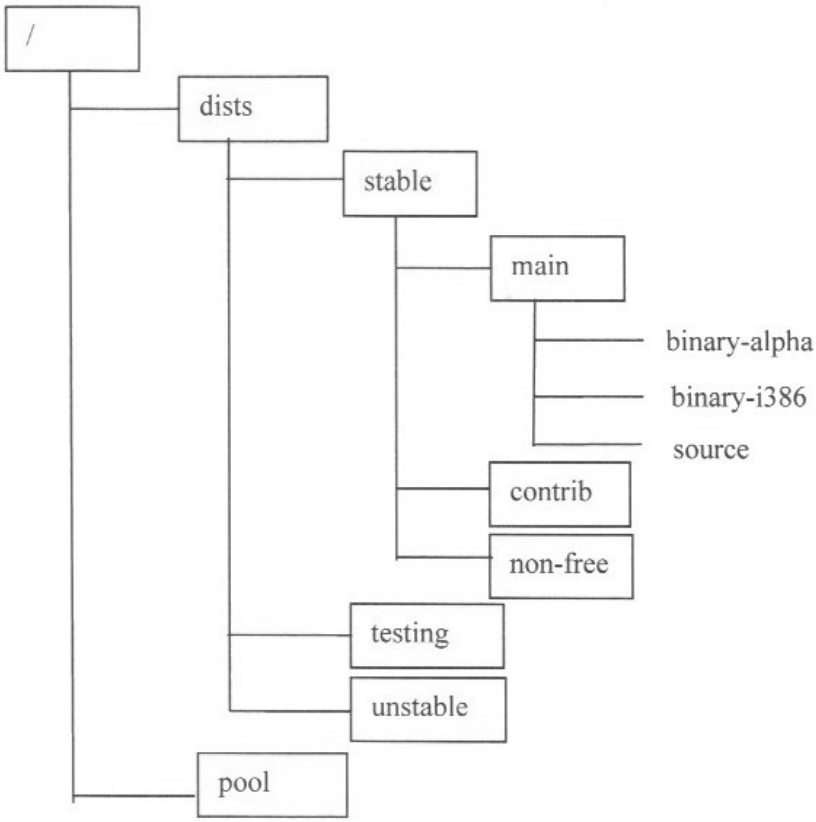


Fig. Dépôt auto.

Les commandes de gestion d'un dépôt

`dpkg-scanpackages`	Génère le fichier Packages.
`dpkg-scansources`	Génère le fichier Sources.
`apt-ftparchive`	Génère les catalogues (fait appel aux commandes de bas niveau dpkg-scanpackages et dpgk-scansources).
`apt-mirror`	Permet de mirorer un Dépôt Debian en partie ou en totalité.

Exemple - création du catalogue d'un dépôt trivial

```
# cd  pub
# dpkg-scanpackages binary /dev/null | gzip -9c > binary/Packages.gz
```

La configuration du poste client

C'est le fichier /etc/apt/sources.list qui contient la liste des dépôts où sont recherchés les packages à installer.

Chaque ligne à la structure suivante :

```
(deb| deb-src) <URL> <distribution> {<composant>}
```

La ligne commence par `deb` pour un dépôt de package binaire et par `deb-src` pour un dépôt de packages sources. La distribution spécifie la stabilité des packages (stable, testing, unstable ou frozen). Enfin sont indiqués, le ou les composants que l'on peut trouver sur le dépôt (main, conrtib, non-free, non-US/main, non-US/contrib).

Si on a un dépôt non-Debian (trivial), distribution et composant sont remplacés par le sous-répertoire contenant les packages.

Exemple – Un fichier /etc/apt/sources.list :

```
deb ftp://ftp.debian.org/debian stable contrib
deb http://nonus.debian.org/debian-non-US stable/non-US main contrib
deb ftp://serveur/pub    binary    /
```

Pour en savoir plus

Les pages de manuel

apt-get(1), sources.list(5), apt-mirror(1)

Internet

Debian Repository HOWTO
http://www.debian.org/doc/manuals/repository-howto/repository-howto

APT HOWTO
http://www.debian.org/doc/manuals/apt-howto/index.fr.html

How To Create A local Debian/Ubuntu Mirror with apt-mirror
http://www.howtoforge.com/local_debian_ubuntu_mirror

Créer un dépôt Yum, configurer le client

Le savoir concret

Créer un dépôt Yum

Les étapes

1. On installe les RPM dans une arborescence de fichiers.
2. On crée les métadata avec la commande `createrepo`.
3. On installe un service HTTP, FTP ou RSYNC pour mettre à disposition le dépôt.

Remarques :

1) L'arborescence où l'on installe les packages peut se résumer à un seul répertoire (RPMS) ou être très complexe. Par exemple, un répertoire pour les binaires fonctionnant sur les systèmes i386, un autre contenant des scripts (no-arch), etc.

2) L'option –g de `createrepo` permet de mettre à disposition des groupes de packages. Il faut fournir un fichier XML qui les décrit.

Les métadata

Les métadata, créés par la commande `createrepo`, sont formés de fichiers XML présents dans le répertoire repodata qui est dans le répertoire contenant les packages.

La configuration du client

Structure de la configuration

La configuration se divise en sections :

- `[main]` La section principale.
- `[<un_depot>]` Une section par dépôt, par exemple [base], [updates], …

Les fichiers de configuration

/etc/yum.conf Le fichier principal de configuration.

/etc/yum.repos.d/, /etc/yum/repos.d/
Les répertoires où sont stockés les fichiers de description des dépôts.

*.repo Les fichiers décrivant les dépôts (un par dépôt).

Remarque : On peut utiliser uniquement yum.conf et y décrire tous les dépôts. On peut également utiliser des noms de répertoires différents de ceux mentionnés en paramétrant l'option `reposdir` de la section `[main]`.

Les principales options de la section [main]

`reposdir=` La liste des répertoires qui contiennent la description des dépôts. Par défaut `/etc/yum.repos.d/` et `/etc/yum/repos.d/`.

`errorlevel=` Le niveau d'erreur de 0 à 10.

`logfile=` Le fichier journal.

Les principales options d'une section décrivant un dépôt

`name=` Une description du dépôt.

`baseurl=` Une ou plusieurs URL qui indique comment accéder au dépôt.

`enabled=` Le dépôt doit être pris en compte (1) ou non (0).

Remarque : quand on active Yum, on peut activer ou désactiver des dépôts :

```
# yum --enablerepo=extra --disablerepo=base  install un_logiciel
```

Les principales options communes aux différentes sections

`gpgcheck=`	Vérification des signatures (1) ou non (0) des packages.
`exclude=`	Liste des packages interdit d'installation ou de mise à jour.
`proxy=`	L'URL d'un proxy.

Pour en savoir plus

Les pages de manuel

yum(8), yum.conf(5), createrepo(8)

Internet

Yum – Le site officiel
http://linux.duke.edu/projects/yum/

Fedora – Software Management with Yum
http://fedora.redhat.com/docs/yum

Creating a local Update Repository for FC6
http://news.softpedia.com/news/Creating-a-Local-Update-Repository-for-FC6-43559.shtml

Créer un package RPM

La théorie

La création d'un RPM peut faciliter grandement le déploiement de logiciels maison ou de logiciels Open-Source.

La démarche

1) Obtenir le code source.
2) Obtenir (éventuellement) les patchs.
3) Créer le fichier Spec.
4) Placer les fichiers au bon endroit.
5) Construire le RPM (en fait il construit le RPM binaire et le RPM source).

Le savoir concret

Le fichier Spec

Le fichier Spec décrit le logiciel et renferme les instructions pour le construire. Il liste également les binaires qui seront installés.

Nom d'un fichier Spec

Syntaxe :

nom_du_logiciel-version-release.spec

Exemple :

example-2.0.2-1.spec

Remarque : La release spécifie la version du package.

La structure d'un fichier Spec

La section principale	Elle décrit le logiciel, sa version et donne l'emplacement du source et des patchs.
%description	Description du logiciel.
%prep	
%setup	Ces deux sections ont pour objectif de tout préparer avant la compilation.
%build	Les commandes qui créent le binaire, habituellement `make`.
%install	Les commandes qui installent le logiciel, habituellement `make install`.
%clean	Les commandes réalisées après l'installation. Généralement on détruit l'arborescence où a été détaré et compilé le logiciel.
%files	Liste les fichiers qui sont installés.
%changelog	L'historique des modifications du package.

Remarques :

1) On peut avoir d'autres sections, notamment associées à l'exécution de scripts de pré-installation (%pre), de post-installation (%post), de pré-désinstallation (%preun) et de post-désinstallation (%psostun).

2) Dans chaque section on peut utiliser tout un ensemble de macros.

Un exemple de fichier Spec

```
Summary: An example
Name: example
Version: 2.0.2
Release: 1
Copyright: GPL
Group: Games
Source: http://server/pub/example/example-2.0.2.tar.gz
BuildRoot: /var/tmp/%{name}-buildroot
%description
It is a example
%prep
%setup -q
%build
make
%install
rm -rf $RPM_BUILD_ROOT
%makeinstall
%clean
rm -rf $RPM_BUILD_ROOT
%files

/usr/bin/example
/usr/man/man1/example.1
%changelog
* Sun Mar 21 2001 Paul Hochon
- ver 2.0.2
```

Les répertoires utilisés lors de la construction d'un RPM

Par défaut, les sources et les packages générés sont dans le répertoire /usr/src/redhat (Sur un système RedHat). Ce répertoire est composé de plusieurs sous répertoires :

SOURCES	C'est dans ce répertoire que l'on copie le source (*.tar.gz) et les patchs.
BUILD	Ce répertoire contient l'arborescence des sources après que la phase %prep soit terminée.
SRPMS	Ce répertoire contient le source RPM (*.src.rpm) après sa construction.
RPMS	Cette arborescence contient le package final (le binaire). Il y a un sous répertoire par type de plateforme (i386, i486, …).
SPECS	Ce répertoire contient le fichier Spec.

Remarque : Lors de la construction du RPM, il y a aussi un répertoire, spécifié dans le fichier Spec (BuildRoot) qui est créé et qui contient temporairement les fichiers destinés à être incorporés dans le RPM (les binaires, les fichiers de documentation, …).

Les commandes

`rpm`	Le gestionnaire de package.
`rpmbuild`	Construit un nouveau package.
`rpmsign`	Signe un package.
`rpm2cpio`	Convertit un RPM au format cpio.
`rpmdeps`	Recherche pour des fichiers les dépendances de packages.

Les fichiers

/usr/lib/rpm/rpmrc, /usr/lib/rpm/redhat/rpmrc, /etc/rpmrc, ~/.rpmrc
Les fichiers de configuration de construction des RPM.

/usr/lib/rpm/macros, /usr/lib/rpm/redhat/macros, /etc/rpm/macros, ~/.rpmmacros
Emplacement des macros.

Focus : La commande rpmbuild

Construire le package

```
# rpmbuild -ba example-2.0.1-1.spec
```

Construire le package à partir du package source

```
# rpmbuild --rebuild example-2.0.1-1.src.rpm
```

Le package binaire résultant est créé dans l'arborescence /usr/src/redhat.

Pour en savoir plus

Les pages de manuel

rpm(8), rpm2cpio(8), rpmbuild(8), rpmdeps(8)

Howto

RPM-HOWTO

Internet

GURULABS – Creating RPM Guide (Excellent !)
http://www.rpm.org/hintskinks/buildtree/GURULABS-RPM-GUIDE-v1.0.PDF

Maximum RPM - Chapter 11. Building Packages: A Simple Example
http://www.rpm.org/max-rpm/ch-rpm-build.html

Packaging software with RPM, Part 1
http://www-128.ibm.com/developerworks/library/l-rpm1/

Distributing Python Modules - 5.2 Creating RPM packages
http://docs.python.org/dist/creating-rpms.html

Créer un package Debian

Le savoir pratique

Nom d'un package

Nom_version_arch.deb (version : majeur.mineur.révision)

Exemple :

axel_1.0b-1_i386.deb

La structure d'un package

Un package Debian est une archive Unix (`ar`) composé de trois fichiers :

- Un fichier texte debian-binary qui contient la version Debian (2.0<saut_de_ligne>).
- Un fichier tar compressé control.tar.gz. Il contient les fichiers assurant la gestion du package.
- Un autre fichier tar compressé data.tar.gz. Il contient les fichiers devant être installés.

Le fichier control.tar.gz contient au format tar compressé les fichiers suivants :

- control Il décrit le package. Ce fichier est requis.
- md5sums Sommes MD5 des fichiers devant être installés.
- postinf Script exécuté après l'installation.
- prerm Script exécuté avant la désinstallation.
- postrm Script exécuté après la désinstallation.

Le fichier control

Le fichier control est obligatoire. Il décrit le logiciel et les dépendances vis-à-vis des autres packages.

Exemple

```
Package : example
Section : misc
Priority : optional
Maintener : Paul Hochon <paul@dormi.fermez>
Architecture : all
Version : 1.0-1
Description : example
```

La structure des répertoires

On crée un répertoire racine de l'application dans lequel on installe tous les fichiers qui devront être installés. Le chemin d'installation est par rapport à ce répertoire. Par exemple, si l'on veut installer `example` dans /usr/bin et si le répertoire racine est /root/example-1.0/ son chemin est /root/example-1.0/usr/bin/example.

Dans le répertoire racine de l'application on crée le répertoire DEBIAN, il doit contenir les fichiers qui seront mis dans le fichier control.tar.gz.

Les commandes

`ar` Crée un package ou visualise, extrait les fichiers d'un package. Commande de bas niveau.

`dpkg`	Gère un package (installe, visualise…).
`dpkg-deb`	Gère un fichier package (crée, …).
`fakeroot`	Permet à un utilisateur ordinaire de créer un package.
`lintian`	Vérifie un package (est-ce qu'il manque des fichiers, …).

Focus : Créer un package avec la commande dpkg-deb

```
# dpkg-deb --build  example-1.0
```

L'argument correspond au fichier racine de l'application (qui contient le sous répertoire DEBIAN).

Pour en savoir plus

Les pages de manuel

dpkg(1), dpkg-deb(1), deb(5), deb-control(5), ar(1)

Howto

Debian Binary Package Building HOWTO

Internet

Guide du nouveau responsable Debian (construire le paquet, …)
http://www.debian.org/doc/maint-guide/

Create Debian Linux Packages
http://www.ibm.com/developerworks/linux/library/l-debpkg.html

ATELIERS



Tâche 1 : Créer un serveur d'installation 20mn
Tâche 2 : Réaliser une installation Kickstart 30mn
Tâche 3 : Créer un serveur de dépôt 10mn
Tâche 4 : Configurer un poste en client d'un dépôt 10mn
Tâche 5 : Créer un RPM 40mn

Tâche 1 : Créer un serveur d'installation

1. Copier les cdroms d'installation dans l'arborescence FTP.

Pour chacun des cdrom, on effectue les opérations suivantes :

```
[root@serveur ~]# mount /media/cdrom
[root@serveur ~]# command cp -rfa  /media/cdrom/CentOS  ~ftp/pub
[root@serveur ~]# eject cdrom
```

Remarque : Si l'on travaille sur une distribution RedHat, on remplace CentOS par RedHat.

2. Activer le serveur ftp.

Au préalable on vérifie sa présence et si besoin, on l'installe.

```
[root@serveur ~]# rpm -q vsftpd
vsftpd-2.0.3-1
[root@serveur ~]# chkconfig vsftpd on
[root@serveur ~]# service vsftpd start
Starting vsftpd for vsftpd:                                [  OK  ]
[root@serveur ~]#
```

3. Configurer et activer le serveur DHCP.

```
[root@serveur ~]# rpm -q dhcp
dhcp-3.0.2-12
[root@serveur ~]# vi /etc/dhcpd.conf
ddns-update-style ad-hoc;
subnet 192.168.0.0 netmask 255.255.255.0 {
        range 192.168.0.100 192.168.0.150;
        default-lease-time 86400;
        max-lease-time 172800;
        option routers 192.168.0.254;
        option domain-name-servers 192.168.0.200;
        option domain-name "pinguins";
}
[root@serveur ~]# chkconfig dhcpd on
[root@serveur ~]# service dhcpd start
Starting dhcpd:                                            [  OK  ]
[root@serveur ~]# ps -e |grep dhcpd
 5637 ?        00:00:00 dhcpd
[root@serveur ~]#
```

Tâche 2 :
Réaliser une installation automatique KickStart

1. Créer sur le serveur d'installation un fichier KickStart.

```
[root@serveur ~]# cd ~ftp/pub
[root@serveur pub]# cp /root/anaconda-ks.cfg   ks.cfg
[root@serveur pub]# vi ks.cfg
# Kickstart file automatically generated by anaconda.

install
url --url ftp://192.168.0.200/pub
lang en_US.UTF-8
langsupport --default=en_US.UTF-8 ar_MA.UTF-8 en_US.UTF-8 fr_FR.UTF-8
keyboard fr
xconfig --startxonboot --resolution 1024x768
network --device eth0 --bootproto static --ip 192.168.0.1 --netmask 255.255.255.
0 --gateway 192.168.0.254 --nameserver 192.168.0.200 --hostname linux1.pinguins
rootpw secret
firewall --disabled
selinux --disabled
authconfig --enableshadow --enablemd5
timezone Europe/Paris
bootloader --location=mbr --append="rhgb quiet"
# The following is the partition information you requested
# Note that any partitions you deleted are not expressed
# here so unless you clear all partitions first, this is
# not guaranteed to work
clearpart --all
part /boot --fstype "ext3" --size=100
part / --fstype "ext3" --size=10000
part /home --fstype "ext3" --size=512
part swap --size=512

%packages
@ admin-tools
@ text-internet
@ legacy-network-server
@ gnome-desktop
@ dialup
@ ftp-server
@ games
@ compat-arch-support
@ smb-server
@ base-x
@ web-server
@ printing
@ kde-desktop
@ french-support
@ server-cfg
@ development-tools
rsh-server
telnet-server
grub
e2fsprogs
```

```
kernel-smp
kernel-smp-devel
firefox

%post
useradd -m guest
echo  "wwii1945" | passwd --stdin guest
```

Remarques :

- La façon la plus simple de créer le fichier kickstart est de réaliser une installation manuelle. Après celle-ci, sur le système nouvellement installé, le fichier /root/anaconda-ks.cfg contient les paramètres d'installation.

- Dans le fichier kickstart généré automatiquement, les ordres de partitionnement sont mis en commentaire. Il faut retirer les symboles commentaires si l'on désire partitionner le disque.

- Le principal ordre à modifier est l'instruction `url` dans lequel on précise l'adresse réseau du serveur d'installation, le répertoire qui contient le répertoire RedHat (/pub dans le cas présent) ainsi que le protocole réseau utilisé (ici `ftp`).

- Le mot de passe de l'administrateur peut être crypté ou non. Dans le fichier généré automatiquement il l'est, mais on peut le saisir en clair directement dans le fichier.

- La configuration graphique peut poser problème si l'on réalise l'installation sur un système différent de celui utilisé pour l'installation manuelle. On peut, comme dans l'exemple ne pas préciser la carte graphique, ou même sauter la configuration du graphique en remplaçant l'ordre `xconfig` par `skipx`.

- Dans l'exemple on a ajouté le package `firefox` et en post-intallation on a ajouté les commandes pour créer l'utilisateur guest.

2. Activer l'installation automatique.

On démarre le poste à installer à partir du 1er cdrom d'installation. A l'invite du chargeur, au lieu de valider, on saisie l'adresse du fichier kickstart en argument du noyau.

```
boot:  linux  ks=ftp://192.168.0.200/pub/ks.cfg
```

Tâche 3 : Créer un serveur de dépôt

1. Vérifier la présence du logiciel createrepo, et si besoin on l'installe (cf. fin de la tâche 4).

```
[root@linux1 ~]# rpm -q createrepo
createrepo-0.4.3-1
```

2. Recopier les cdroms d'installation et on active le service FTP.

(cf. item 1 et 2 de la tâche 1)

3. Créer les métadata associés aux packages.

```
[root@linux1 ~]# cd  ~ftp/pub/CentOS/RPMS
[root@linux1 RPMS]# createrepo  .   # « . » répertoire courant
1499/1499 - ctags-5.5.4-1.i386.rpm
Saving Primary metadata
Saving file lists metadata
Saving other metadata
[root@linux1 RPMS]# ls repodata/
filelists.xml.gz  other.xml.gz  primary.xml.gz  repomd.xml
```

4. Créer les métadata associés aux groupes de packages.

```
[root@linux1 RPMS]# cp ../base/comps.xml .
[root@linux1 RPMS]# createrepo -g comps.xml .
```

```
1499/1499 - ctags-5.5.4-1.i386.rpm
Saving Primary metadata
Saving file lists metadata
Saving other metadata
[root@linux1 RPMS]# ls repodata/
comps.xml  filelists.xml.gz  other.xml.gz  primary.xml.gz  repomd.xml
[root@linux1 RPMS]# cd
```

Tâche 4 :
Configurer un poste en client d'un dépôt

1. Modifier la configuration du client (on la sauvegarde au préalable).

Remarque : on adapte l'adresse du serveur à celle de son binôme ou à la sienne.

```
[root@linux1 ~]# mv /etc/yum.conf /etc/yum.conf.old
[root@linux1 ~]# mv /etc/yum.repos.d /etc/yum.repos.d_old
[root@linux1 ~]# vi  /etc/yum.conf
[base]
name=CentOS $releasever - $basearch - Base
baseurl=ftp://localhost/pub/CentOS/RPMS
```

Remarque : On utilise le nom RedHat dans le cas d'une distribution RedHat.

2. Faire un test : on désinstalle le logiciel Lynx et on le réinstalle.

```
[root@linux1 ~]# yum search lynx
Searching Packages:
Setting up repositories
base                      100% |=========================| 1.1 kB    00:00
Reading repository metadata in from local files
primary.xml.gz            100% |=========================| 570 kB    00:00
base      : ################################################## 1499/1499
Added 1499 new packages, deleted 0 old in 7.47 seconds
...
lynx.i386                                2.8.5-18.2              installed
Matched from:
lynx
Lynx is a text-based Web browser. Lynx does not display any images,
but it does support frames, tables, and most other HTML tags. One
advantage Lynx has over graphical browsers is speed; Lynx starts and
exits quickly and swiftly displays webpages.
http://lynx.isc.org/

[root@linux1 ~]# yum remove lynx
...
[root@linux1 ~]# yum install lynx
Setting up Install Process
Setting up repositories
Reading repository metadata in from local files
Parsing package install arguments
Resolving Dependencies
--> Populating transaction set with selected packages. Please wait.
---> Downloading header for lynx to pack into transaction set.
lynx-2.8.5-18.2.i386.rpm  100% |=========================|  21 kB    00:00
---> Package lynx.i386 0:2.8.5-18.2 set to be updated
--> Running transaction check
```

```
Dependencies Resolved

================================================================================
 Package                Arch       Version          Repository        Size
================================================================================
Installing:
 lynx                   i386       2.8.5-18.2       base              1.8 M

Transaction Summary
================================================================================
Install       1 Package(s)
Update        0 Package(s)
Remove        0 Package(s)
Total download size: 1.8 M
Is this ok [y/N]: y
Downloading Packages:
(1/1): lynx-2.8.5-18.2.i3 100% |=========================| 1.8 MB    00:00
Running Transaction Test
Finished Transaction Test
Transaction Test Succeeded
Running Transaction
  Installing: lynx                         ########################## [1/1]

Installed: lynx.i386 0:2.8.5-18.2
Complete!
[root@linux1 ~]#
```

3. Lister les groupes de logiciels.

```
[root@linux1 ~]# yum grouplist
Setting up Group Process
Setting up repositories
comps.xml                 100% |=========================| 557 kB    00:00
Installed Groups:
   Administration Tools
   Compatibility Arch Support
   Engineering and Scientific
   FTP Server
   GNOME Desktop Environment
   Graphics
   KDE (K Desktop Environment)
   Legacy Network Server
   Mail Server
   MySQL Database
   Office/Productivity
   PostgreSQL Database
   Printing Support
   Server Configuration Tools
   Text-based Internet
   Web Server
   Windows File Server
   X Window System
Available Groups:
   Authoring and Publishing
```

```
   Compatibility Arch Development Support
   DNS Name Server
   Development Tools
   Editors
   GNOME Software Development
   Games and Entertainment
   Graphical Internet
   KDE Software Development
   Legacy Software Development
   Network Servers
   News Server
   Sound and Video
   System Tools
   X Software Development
Done
[root@linux1 ~]#
```

Remarque : Sur un système RedHat, on peut récupérer le logiciel yum à partir d'un miroir de la distribution CentOS, par exemple :

```
# wget 'http://isoredirect.centos.org/centos/4/os/i386/CentOS/RPMS/yum-2*.rpm'
```

Il faut récupérer également les RPM suivant :

python-urlgrabber, python-sqlite, python-elementtree, sqlite, createrepo.

Enfin on installe les RPM dans l'ordre des dépendances :

```
# rpm -Uvh  sqlite-*.rpm
# rpm -Uvh python-*.rpm
# rpm -Uvh yum-2.4.3-1.c4.noarch.rpm --nodeps
# rpm -Uvh createrepo-0.4.3-1.noarch.rpm
```

Tâche 5 : Créer un RPM

1. Créer le logiciel source.

a) On crée le répertoire racine du logiciel.

```
[root@linux1 ~]# mkdir nbmister-1.0
[root@linux1 ~]# cd nbmister-1.0
```

b) On saisit les sources.

```
[root@linux1 nbmister-1.0]# vi nbmister.c
/* nbmiser.c */
# include <stdio.h>
# include <string.h>

int random_number();

main() {
 int misterious_number = random_number();
 int a_try, i;
 char a_string[256];

 for(i=1;;i++) {
        printf("What is the misterious number (between 1 and 100) ? ");
        fgets( a_string, 255, stdin);
        a_try = atoi( a_string);
```

```
        printf("===>Try number:%d,  Try:%d\n", i, a_try);
        if ( a_try == misterious_number ) break;
        if ( a_try <  misterious_number ) puts("=== too small ===");
        if ( a_try >  misterious_number ) puts("=== too high  ===");
 }
 printf("YOUPI!!!!\n");
}
[root@linux1 nbmister-1.0]# vi bib.c
/* bib.c */
# include <stdlib.h>
# include <time.h>

int random_number() {
 srandom( time(0) );
 return random() % 100 + 1;
}

[root@linux1 nbmister-1.0]# vi Makefile
nbmister: nbmister.o bib.o
   cc -o nbmister nbmister.o bib.o
install: nbmister
   install -m 755 nbmister /usr/local/bin
clean:
   rm -f *.o
```

Remarque : les lignes indentées d'un fichier Makefile doivent commencer par une tabulation.

c) On teste le fonctionnement.

```
[root@linux1 nbmister-1.0]# make install
cc    -c -o nbmister.o nbmister.c
cc    -c -o bib.o bib.c
cc -o nbmister nbmister.o bib.o
install -m 755 nbmister /usr/local/bin
[root@linux1 nbmister-1.0]# nbmister
What is the misterious number (between 1 and 100) ?
^C
[root@linux1 nbmister-1.0]# make clean
rm -f *.o
[root@linux1 nbmister-1.0]# rm -f nbmister  /usr/local/bin/nbmister
```

d) On crée le tarball.

```
[root@linux1 nbmister-1.0]# cd
[root@linux1 ~]# tar cvzf nbmister-1.0.tar.gz nbmister-1.0/
nbmister-1.0/
nbmister-1.0/Makefile
nbmister-1.0/nbmister.c
nbmister-1.0/bib.c
[root@linux1 ~]#
```

2. Créer le fichier Spec.

```
[root@linux1 ~]# vi nbmister-1.0-1.spec
Summary: find the misterious number
Name: nbmister
Version: 1.0
Release: 1
```

```
Copyright: GPL
Group: Games
Source: nbmister-1.0.tar.gz
BuildRoot: /var/tmp/%{name}-buildroot

%description
nbmister software is a game
you must find the misterious number

%prep
%setup -q

%build
make
%install
rm -rf $RPM_BUILD_ROOT
mkdir -p $RPM_BUILD_ROOT/usr/local/bin
install -s -m  755 nbmister $RPM_BUILD_ROOT/usr/local/bin/
%clean
rm -rf $RPM_BUILD_ROOT
%files
/usr/local/bin/nbmister
%changelog
* Fri Feb 16 2007 JF B
- ver 1.0
[root@linux1 ~]#
```

3. Créer le RPM.

```
[root@linux1 ~]# ls /usr/src/redhat/
BUILD  RPMS  SOURCES  SPECS  SRPMS
[root@linux1 ~]# cp nbmister-1.0.tar.gz /usr/src/redhat/SOURCES/
[root@linux1 ~]# rpmbuild -ba nbmister-1.0-1.spec  > err 2>&1
[root@linux1 ~]# more err
…
Requires: libc.so.6 libc.so.6(GLIBC_2.0)
Processing files: nbmister-debuginfo-1.0-1
Checking for unpackaged file(s): /usr/lib/rpm/check-files /var/tmp/nbmister-
buildroot
Wrote: /usr/src/redhat/SRPMS/nbmister-1.0-1.src.rpm
Wrote: /usr/src/redhat/RPMS/i386/nbmister-1.0-1.i386.rpm
Wrote: /usr/src/redhat/RPMS/i386/nbmister-debuginfo-1.0-1.i386.rpm
Executing(%clean): /bin/sh -e /var/tmp/rpm-tmp.43644
+ umask 022
+ cd /usr/src/redhat/BUILD
+ cd nbmister-1.0
+ rm -rf /var/tmp/nbmister-buildroot
+ exit 0
[root@linux1 ~]# find /usr/src/redhat/
/usr/src/redhat/
/usr/src/redhat/SRPMS
/usr/src/redhat/SRPMS/nbmister-1.0-1.src.rpm
/usr/src/redhat/SPECS
/usr/src/redhat/RPMS
/usr/src/redhat/RPMS/i386
```

```
/usr/src/redhat/RPMS/i386/nbmister-1.0-1.i386.rpm
/usr/src/redhat/RPMS/i386/nbmister-debuginfo-1.0-1.i386.rpm
/usr/src/redhat/RPMS/i486
...
```

4. Utiliser les RPM générés (le source et le binaire).

a) On recopie les RPM générés dans le répertoire courant.

```
[root@linux1 ~]# cp /usr/src/redhat/RPMS/i386/nbmister-1.0-1.i386.rpm  .
[root@linux1 ~]# cp /usr/src/redhat/SRPMS/nbmister-1.0-1.src.rpm  .
[root@linux1 ~]# ls *.rpm
nbmister-1.0-1.i386.rpm  nbmister-1.0-1.src.rpm
```

b) On installe le logiciel.

```
[root@linux1 ~]# rpm -Uvh nbmister-1.0-1.i386.rpm
Preparing...                ########################################### [100%]
   1:nbmister               ########################################### [100%]
[root@linux1 ~]# rpm -q nbmister
nbmister-1.0-1
```

c) On liste les fichiers qui le composent.

```
[root@linux1 ~]# rpm -ql nbmister
/usr/local/bin/nbmister
```

d) On affiche son descriptif

```
[root@linux1 ~]# rpm -qi nbmister
Name        : nbmister                     Relocations: (not relocatable)
Version     : 1.0                              Vendor: (none)
Release     : 1                            Build Date: Fri Feb 16 18:49:25 2007
Install Date: Fri Feb 16 18:52:07 2007      Build Host: linux1.pinguins
Group       : Games                        Source RPM: nbmister-1.0-1.src.rpm
Size        : 3324                            License: GPL
Signature   : (none)
Summary     : find the misterious number
Description :
nbmister software is a game
you must find the misterious number
```

e) On affiche les évolutions du package.

```
[root@linux1 ~]# rpm -q --changelog nbmister
* Fri Feb 16 2007 JF B

- ver 1.0
```

f) On utilise le logiciel.

```
[root@linux1 ~]# nbmister
What is the misterious number (between 1 and 100) ? 50
===>Try number:1,  Try:50
=== too high  ===
What is the misterious number (between 1 and 100) ? 26
===>Try number:2,  Try:26
=== too high  ===
What is the misterious number (between 1 and 100) ? 13
===>Try number:3,  Try:13
=== too small ===
```

```
What is the misterious number (between 1 and 100) ? 19
===>Try number:4,  Try:19
=== too small ===
What is the misterious number (between 1 and 100) ? 21
===>Try number:5,  Try:21
YOUPI!!!!
```

g) On supprime le logiciel.

```
[root@linux1 ~]# rpm -e nbmister
```

h) On supprime le package binaire et on le reconstruit à partir du package source.

```
[root@linux1 ~]# rm /usr/src/redhat/RPMS/i386/nbmister-*
[root@linux1 ~]# rpmbuild --rebuild nbmister-1.0-1.src.rpm
[root@linux1 ~]# ls /usr/src/redhat/RPMS/i386
nbmister-1.0-1.i386.rpm  nbmister-debuginfo-1.0-1.i386.rpm
[root@linux1 ~]#
```

5. Inspecter un RPM source (ici un noarch).

```
[root@linux1 ~]# wget
'ftp://rpmfind.net/linux/redhat/enterprise/4/en/os/i386/SRPMS/katana-2.0.0-
1.src.rpm'
[root@linux1 ~]# rpm2cpio katana-2.0.0-1.src.rpm | cpio -itv '*spec'
-rw-rw-r--   1 root     root         2799 Sep 27  2004 katana.spec
70 blocks
[root@linux1 ~]# rpm2cpio katana-2.0.0-1.src.rpm | cpio -iv '*spec'
katana.spec
70 blocks
[root@linux1 ~]# more katana.spec
Summary: Helper scripts for Java rpms
Name: katana
Version: 2.0.0
Release: 1
Source1: javaconfig.sh
Source2: katana.sh
Source3: classfile.py
Source4: stage1.py
Source5: stage2.py
Source6: stage3.py
License: GPL
Group: System Environment/Libraries
BuildRoot: %{_tmppath}/%{name}-%{version}-root
Obsoletes: redhat-java-rpm-scripts
Provides: redhat-java-rpm-scripts = %{version}-%{release}
BuildArch: noarch

%description
A collection of scripts used to perform common tasks for rpm packages
that contain Java libraries and executables.

%package build
%package build
Summary: Build tool for Java rpms
Group: Development/Libraries
Requires: katana = %{version}-%{release}
Requires: coreutils, python >= 2.2, gcc-java >= 3.4
```

```
%description build
A collection of scripts used to build rpm packages that contain Java
libraries and executables.

%install
[ "$RPM_BUILD_ROOT" != "/" ] && rm -rf $RPM_BUILD_ROOT

mkdir -p ${RPM_BUILD_ROOT}%{_sbindir}
install -m 755 $RPM_SOURCE_DIR/javaconfig.sh \
    $RPM_BUILD_ROOT%{_sbindir}/javaconfig

mkdir -p ${RPM_BUILD_ROOT}%{_bindir}
install -m 755 $RPM_SOURCE_DIR/katana.sh \
    $RPM_BUILD_ROOT%{_bindir}/katana

mkdir -p ${RPM_BUILD_ROOT}%{_datadir}/katana/build
install -m 644 $RPM_SOURCE_DIR/*.py \
    $RPM_BUILD_ROOT%{_datadir}/katana/build

%clean
[ "$RPM_BUILD_ROOT" != "/" ] && rm -rf $RPM_BUILD_ROOT

%files
%defattr(-,root,root)
%{_sbindir}/*
%dir %{_datadir}/katana

%files build
%defattr(-,root,root)
%{_bindir}/*
%{_datadir}/katana/build

%changelog
* Mon Sep 27 2004 Gary Benson <gbenson@redhat.com> 2.0.0-1
- Rewrite to be JPackage friendly.
…
```

- *uname –a ; uname -r*
- *make menuconfig*
- */proc/cpuinfo*
- *echo 1 > ip_forward*
- *IPC, ipcs, ipcrm*

3

Le noyau

Objectifs

Le noyau est le cœur des systèmes Linux. Dans ce chapitre on apprend à le recompiler si besoin. On étudie également la gestion dynamique du noyau. D'abord, en apprenant à gérer ses modules mais aussi en apprenant à modifier à chaud ses paramètres. Enfin, on apprend à mieux connaître le noyau notamment à travers le FS /proc.

Contenu

Présentation du noyau

La compilation du noyau

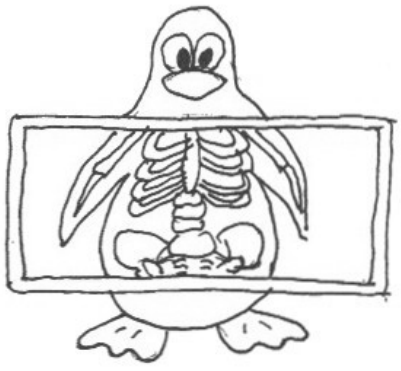
Les modules du noyau

L'arborescence /proc

Le paramétrage à chaud du noyau

IPC system V

Ateliers



Présentation du noyau

La théorie

Le rôle du noyau

Le noyau (kernel) est le cœur du système d'exploitation (OS) Linux. Les applications se reposent sur lui pour accomplir les tâches primordiales que l'on demande à un OS :

- La gestion des processus.
- La gestion des fichiers.
- La gestion des périphériques.

La structure logique du noyau

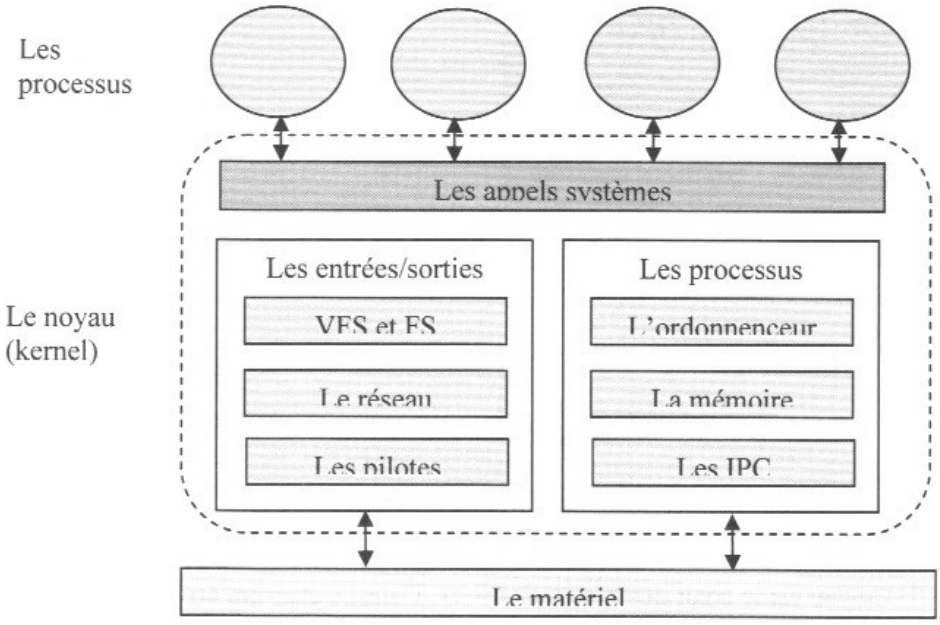


Fig. La structure du noyau.

Le noyau est composé de différents sous-systèmes :

- L'interface avec les applications : les appels systèmes (System calls).
- L'Ordonnanceur de tâches (Scheduler) : c'est lui qui attribue du temps CPU à un processus.
- La gestion de la mémoire.
- La communication interprocessus (Inter Process Communication ou IPC).
- La gestion du réseau.
- Le VFS (Virtual File System) permet une vision uniforme des FS même si chaque FS possède son propre pilote.
- Les pilotes de périphériques. Chacun gère les entrées/sorties d'un périphérique. Ils sont indispensables pour accéder au matériel.

La composition physique du noyau

Concrètement, le noyau se présente essentiellement comme un fichier chargé par le chargeur (Grub, Lilo, …) au démarrage du système. Il reste en mémoire jusqu'à l'arrêt complet du système. Si le noyau est compilé de manière dynamique (c'est le cas le plus fréquent), les pilotes ou même des sous-systèmes peuvent être stockés dans des modules qui sont des fichiers avec l'extension .ko (kernel object).

Le savoir concret

Les commandes

`uname -r`	Affiche la version du noyau courant.
`uname -a`	Affiche des informations générales sur le système (version du noyau, date de compilation, …).
`arch`	Affiche l'architecture matérielle.
`strace`	Trace les appels système d'une application.
`lsmod`	Liste les modules actuellement chargés en mémoire.
`dmesg`	Affiche les messages du noyau, notamment ceux du démarrage.

Les fichiers

/boot/vmlinuz*	La partie statique du noyau chargé par le chargeur.
/lib/modules/$(uname –r)	L'arborescence qui contient les modules (.ko) du noyau.
/boot/grub/grub.conf	Le fichier de configuration de Grub. Il spécifie le noyau à charger et ses paramètres.

Focus : les versions du noyau

```
# uname -r
2.6.9-42.EL
```

Une version de noyau est composée de plusieurs éléments :

- La version (ici 2).
- Le niveau de patch (ici 6).
- Le sous-niveau (ici 9).
- L'extra-version (ici 42.EL).

Les niveaux pairs de patch correspondent à une version stable du noyau : 2.0, 2.2, 2.4, 2.6.

Pour en savoir plus

Howto

Kernel-analysis-howto (comment fonctionne le noyau)

Internet

The Linux Kernel (un peu ancien, mais très intéressant)
http://en.tldp.org/LDP/tlk/

Livre

Understanding the Linux Kernel, par D. Bovet & M. Cesati, chez O'Reilly.

La compilation du noyau

La théorie

Pourquoi compiler le noyau Linux ?

La raison essentielle pour laquelle on est amené à recompiler le noyau est la volonté d'utiliser de nouvelles fonctionnalités du noyau. Par exemple, il y a quelques années, l'utilisation des ACL nécessitait la recompilation du noyau.

En fait, de nos jours cette opération est (heureusement) de moins en moins nécessaire. Et ceci pour plusieurs raisons :

- Les principales fonctionnalités qui manquaient à Linux (ACL, HotPlug, ...) font maintenant parties en standard du noyau.
- Le noyau est dynamique : si l'on désire ajouter un module ou changer les paramètres systèmes, il n'est plus nécessaire de le recompiler.
- Les distributions offrent tout un choix de noyau sous forme packagée : noyau pour plateforme multiprocesseurs, noyau supportant de grande capacité mémoire, noyaux correspondant à des mises à jour, ...
- L'utilisation de fichiers initrd permet le chargement des modules nécessaires au démarrage par le chargeur. En conséquence, ces modules ne sont plus obligatoirement compilés de manière statique.
- La recompilation peut même être éventuellement interdite. Ainsi, si vous utiliser une version commerciale de Linux, la recompilation vous fait perdre le support.

Pourquoi compiler un module ?



Si la compilation du noyau est maintenant rarement nécessaire, la compilation des modules reste souvent une obligation. En effet, **Le noyau ne peut charger un module que si ce dernier a été compilé pour la version exacte de votre noyau** (version, niveau de patch, sous-niveau et extra-version).

Remarque : Certains fabricants de matériels ne livrent leur pilote que sous forme binaire et uniquement pour telle ou telle version d'un noyau RedHat ou SUSE. Cette situation diminue encore les raisons de recompiler votre noyau : vous ne pourriez plus utiliser ces pilotes.

Les grandes étapes de la compilation du noyau

1) Récupérer et installer les sources du noyau soit à partir du site officiel (kernel.org) ou à partir du site de votre distribution.

2) Patcher (éventuellement) les sources.

3) Configurer le noyau.

4) Compiler le noyau (très long).

5) Créer (éventuellement) un fichier initrd.

6) Installer les modules.

7) Configurer le chargeur (`Grub`, `Lilo`, ...) pour permettre le démarrage du noyau.

8) On teste : on redémarre le système en utilisant le nouveau noyau.

Remarque : Les patchs correspondent à des modifications des sources. Une nouvelle fonctionnalité du noyau se présente soit sous forme de source d'un module ou sous forme

d'un Patch. Les noyaux des distributions Linux comportent de nombreux patchs (SELinux, etc.)

La configuration du noyau

La configuration est l'étape la plus importante. On peut choisir globalement d'avoir un noyau statique ou dynamique. Mais concrètement, on va faire le choix des modules qui compose votre noyau. On choisit aussi si l'inclusion est statique ou sous forme d'un module chargé dynamiquement. La configuration permet également la prise en compte ou non d'une option d'un module ou d'un sous-système.

Le savoir concret

Les commandes

`patch`	Cette commande met à jour d'une arborescence de fichiers texte.
`make`	Cette commande crée le fichier de configuration du noyau. Il permet également de déclencher la compilation du noyau et l'installation des modules.
`mkinitrd`	Crée un fichier initrd associé à une version du noyau.

La configuration du noyau

`make config`	Configuration en mode texte, on vous demande de vous prononcer sur chaque option (déconseillé).
`make menuconfig`	Configuration en mode texte mais sous forme de menus.
`make xconfig,`	
`make gconfig`	Configuration en mode graphique.
`make defconfig`	Crée une configuration par défaut.
`make oldconfig`	Configure le noyau en se basant sur le fichier de configuration existant (.config).

Remarque : En face d'une fonctionnalité, la lettre M indique que l'on a choisi une compilation dynamique (sous forme de module).

Les autres cibles de make

`make`	Si l'on ne précise pas de cible, cette commande déclenche la compilation du noyau et des modules.
`make modules_install`	Installe les modules dans l'arborescence /lib/modules/<version_du_nouveau_noyau>.
`make clean`	Supprime les fichiers générés lors de la dernière compilation du noyau mais garde le fichier de configuration.
`make mproper`	Nettoyage plus complet. Détruit le fichier de configuration.
`make distclean`	Nettoyage encore plus complet.

Les fichiers

.config	Le fichier de configuration du noyau.
arch/i386/bzImage	Le noyau généré lors de la compilation.

Makefile	Le fichier de directives utilisé par la commande `make`. Contient notamment la version du futur noyau.
/boot/config-*	Les fichiers de configuration des différents noyaux installés. Leur présence est conventionnelle.

Remarque : Les chemins relatifs sont par rapport au répertoire d'installation des sources.

Particularités des distributions

Debian

`make-kpkg`	Génère un paquet Debian traitant du noyau. Il peut contenir un noyau complet (noyau statique, modules et configuration) ou simplement un module.

Pour en savoir plus

Les pages de manuel

mkinitrd(8), patch(1)

Fichiers

Documentation/	Cette arborescence, faisant partie des sources du noyau, contient la documentation du noyau. Dans le cas d'un noyau packagé, cette documentation est dans un package séparé.

Internet

Le site officiel du noyau Linux, contient ses sources.
http://www.kernel.org

Kernel Rebuild Guide (noyau 2.6, c'est le futur Howto)
http://www.digitalhermit.com/linux/Kernel-Build-HOWTO.html

Le linux-kernel HOWTO (noyau 2.4, Howto obsolète)
http://www.ibiblio.org/pub/linux/docs/HOWTO/translations/fr/pdf/Kernel-HOWTO.pdf

Livre

Linux Kernel in a nutshell, par Greg Kroah-Hartman, chez O'Reilly.

Les modules du noyau

La théorie

La plupart des pilotes (et quelques sous-systèmes) peuvent être compilés sous forme de module. L'ajout d'un périphérique, n'oblige pas alors de recompiler le noyau. Il suffit de récupérer le module binaire et de l'installer. Eventuellement, on le récupère sous forme source et on le recompile avant de l'installer.

Les modules permettent également une mise à jour aisée du noyau : il suffit de remplacer, supprimer ou ajouter des modules au lieu d'obtenir un nouveau noyau. Enfin, la structure modulaire du noyau permet également d'avoir en mémoire uniquement les modules actuellement utilisés.

Remarque : Le noyau ne peut charger un module que si ce dernier à été compilé pour la version exacte de votre noyau (version, niveau de patch, sous-niveau et extra-version).

Le savoir concret

Les commandes

`lsmod`	Liste les modules actuellement chargés.
`insmod`	Charge un module, on peut passer des paramètres au module.
`modprobe`	Idem, mais prend en compte les dépendances.
`depmod`	Crée le fichier qui mémorise les dépendances entre modules.
`rmmod`	Décharge un module.
`modinfo`	Affiche des informations concernant un module.

Les fichiers

/etc/modules.conf	Le fichier de configuration des modules (noyau 2.4).
/etc/modprobe.conf	Le fichier de configuration des modules (noyau 2.6).
/lib/modules/$(uname –r)/	L'arborescence où est stockée les modules d'une version du noyau.
/lib/modules/$(uname –r)/modules.dep	Le fichier qui contient les dépendances entre modules.

Pour en savoir plus

Les pages de manuel

lsmod(8), modprobe(8), depmod(8), rmmod(8), modinfo(8), modprobe.conf(5), modules.dep(5)

Howto

Module-HOWTO

L'arborescence /proc

La théorie

Le FS /proc est une interface au noyau sous forme de fichiers. Ces fichiers permettent d'obtenir des informations provenant du noyau en temps réel ou même, pour certains, de configurer à chaud le noyau.

Remarque : La plupart des informations présentes dans /proc sont accessibles par de simples commandes (`ps`, `lsmod`, `mount`, `uptime`, ...). Bien sur, elles nécessitent que /proc soit monté.

Le savoir concret

Les principaux fichiers à la racine du FS

cmdline	Les arguments passés au noyau lors du démarrage.
cpuinfo	La description du processeur.
filesystems	La liste des FS supportés par le noyau.
locks	Les verrous fichiers.
modules	Les modules actuellement chargés
mounts	Les FS actuellement montés.
partitions	Les partitions connues du noyau.
swaps	Les espaces de swap utilisés.
version	La version du noyau.

Informations liées aux performances :

meminfo, uptime, loadavg, uptime, stat

Informations liées à la gestion des périphériques :

bus/, devices, dma, ioports, irq, driver/, fb, ide/, interrupts, isapnp,

Les principaux fichiers associés à un processus

Chaque processus est documenté par des fichiers présents dans le répertoire /proc/<PID>/, où PID est l'identifiant du processus, par exemple : /proc/381/.

cmdline	Les arguments de la commande.
cpu	Le CPU courant du processus et le dernier utilisé.
cwd	Le répertoire courant de l'application.
root	Le répertoire racine (ne pas oublier chroot()!)
environ	L'environnement (les variables d'environnement)
exe	Le fichier exécutable
maps	La cartographie de l'espace virtuelle (bibliothèques dynamiques, ...).
status	L'état du processus (PID, UID, GIDs, gestion des signaux, etc, ...)

Pour en savoir plus

filesystems/proc.txt	Ce fichier, appartenant à la documentation du noyau, décrit l'arborescence /proc.

Le paramétrage à chaud du noyau

La théorie

De nombreux paramètres du noyau Linux peuvent être déterminés à chaud. C'est grâce au FS /proc que cela est possible. En effet, non seulement cette arborescence de fichiers nous informe sur l'état et la configuration du système, mais certains fichiers, accessibles en lecture/écriture permettent une configuration dynamique du noyau.

Remarque : Les arguments passés au noyau grâce à la configuration du chargeur constitue une autre technique de paramétrage du noyau.

Le savoir concret

Les commandes

`sysctl`	Cette commande affiche ou modifie un paramètre du noyau.
`sysctl -a`	Liste l'ensemble des paramètres modifiables du noyau et leur valeur.

Les fichiers

/proc/sys/	L'arborescence comprenant les paramètres configurables du noyau.
/etc/sysctl.conf	Contient les valeurs des paramètres configurés au démarrage.

Quels paramètres

`file-max`	Le nombre maximum de fichiers ouverts.
`inode-max`	Le nombre maximum d'inodes ouvertes par processus.
`shmmax`	Le nombre maximum de segments de mémoires partagées (IPC).
`ip_forward`	Le système est un routeur (1) ou non (0).
icmp_echo_ignore_all	Ne répond pas (0) aux paquets ping.
icmp_echo_ignore_broadcasts	Ne répond pas (0) aux paquets ping de diffusion.

Focus : modifier à chaud un paramètre du noyau

```
# cat /proc/sys/net/ipv4/ip_forward
0
# echo 1 > /proc/sys/net/ipv4/ip_forward
# cat /proc/sys/net/ipv4/ip_forward
1
```

Pour en savoir plus

Les pages de manuel

sysctl(8), sysctl.conf(5)

Fichier

filesystems/proc.txt	Ce fichier, appartenant à la documentation du noyau, décrit l'arborescence /proc.

IPC System V

La théorie

Description

Les IPC System V sont un ensemble de mécanismes de communication inter-process standard chez les systèmes Unix. Leur originalité est d'utiliser la mémoire comme moyen de communication, ce qui fait que ce sont les méthodes les plus rapides pour échanger des données ou synchroniser des processus.

Beaucoup de logiciels à contrainte de temps les utilisent. Par exemple les bases de données Oracle ou PostgreSQL, ou encore le jeu Quake.

Les catégories d'IPC

Il y a trois catégories d'IPC system V :

- Les tableaux de sémaphores pour réaliser la synchronisation de processus.
- Les segments de mémoire partagée.
- Les files d'attente de messages.

Remarque : Les tableaux de sémaphores et les segments de mémoire partagée vont souvent de pair : les sémaphores permettant l'accès exclusif des segments de mémoire partagées.

IPC et paramétrage du noyau

Pour des raisons de construction, les IPC System V sont composées de tables système de dimension finies. Les dimensions courantes de ces tables sont éventuellement incompatibles avec les pré-requis d'une application (comme Oracle). Il faut dans ce cas modifier leur valeur en modifiant des fichiers de l'arborescence /proc.

IPC et exploitation

Normalement un administrateur n'a pas à se soucier des IPC : elles sont créées automatiquement au démarrage des applications qui les utilisent et elles sont détruites lors de leur arrêt.

Mais les IPC forment des ressources limitées. Une application boguée peut très bien ne pas détruire ses IPC ce qui empêchera cette application ou une autre de démarrer faute de ressources. L'administrateur doit, dans ce cas détruire manuellement les IPC fautives.

Le savoir concret

Les commandes

`ipcs`	Liste les IPC courantes.
`ipcrm`	Permet de détruire une IPC.

Les fichiers

/proc/sysvipc/*	Informations de base concernant les IPC.
/proc/sys/kernel/*	La dimension des tables IPC (éléments paramétrables).

Pour en savoir plus

Les pages de manuel

ipcs(8), ipcrm(8)

Internet

Configuration de PostgreSQL – Liste des paramètres IPC et conseils sur leur configuration
http://docs.postgresqlfr.org/pgsql-8.2.1-fr/kernel-resources.html

Livre

Chapitre 18 (Process Communication) du livre « Understanding the Linux Kernel », chez O'Reilly (déjà cité).

ATELIERS



Tâche 1 : Compiler et patcher un noyau 2.6, en partant des sources officiels..... 45mn
Tâche 2 : Ajouter un nouveau module à votre nouveau noyau 10mn
Tâche 3 : Ajouter un module au noyau packagé 10mn
Tâche 4 : Installer un noyau packagé 10mn
Tâche 5 : Compiler le noyau à partir des sources de la distribution..... 30mn
Tâche 6 : Les modules dynamiques du noyau..... 10mn
Tâche 7 : L'arborescence /proc 10mn
Tâche 8 : Paramétrer à chaud le noyau..... 10mn
Tâche 9 : Les IPC System V..... 10mn

Tâche 1 :
Compiler et patcher un noyau 2.6, en partant des sources officiels.

1. Afficher la version courante du noyau.

```
[root@linux1 ~]# uname -r
2.6.9-11.ELsmp
```

2. Télécharger le noyau.

```
[root@linux1 ~]# cd
[root@linux1 ~]# wget 'ftp://ftp.kernel.org/pub/linux/kernel/v2.6/linux-
2.6.16.tar.bz2'
--12:11:39--  ftp://ftp.kernel.org/pub/linux/kernel/v2.6/linux-2.6.16.tar.bz2
           => `linux-2.6.16.tar.bz2'
Resolving ftp.kernel.org... 204.152.191.37, 204.152.191.5
Connecting to ftp.kernel.org[204.152.191.37]:21... connected.
Logging in as anonymous ... Logged in!
==> SYST ... done.    ==> PWD ... done.
==> TYPE I ... done.  ==> CWD /pub/linux/kernel/v2.6 ... done.
==> PASV ... done.    ==> RETR linux-2.6.16.tar.bz2 ... done.
Length: 40,845,005 (unauthoritative)
54% [==============>                ] 22,186,080   121.62K/s    ETA 02:19
100%[==============================>] 40,845,005   263.26K/s    ETA 00:00

12:16:43 (132.39 KB/s) - `linux-2.6.16.tar.bz2' saved [40,845,005]
```

3. Télécharger (éventuellement) des patchs (mises à jour ou nouvelles fonctionnalités).

```
[root@linux1 ~]# wget 'ftp://ftp.kernel.org/pub/linux/kernel/v2.6/patch-
2.6.17.gz'
…
100%[====================================>] 7,411,732     76.66K/s    ETA 00:00

16:01:22 (78.64 KB/s) - `patch-2.6.17.gz' saved [7,411,732]

[root@linux1 ~]#
```

4. Installer les sources dans l'arborescence /usr/src/<version-noyau>.

```
[root@linux1 ~]# cd /usr/src
[root@linux1 src]# tar xvjf /root/linux-2.6.16.tar.bz2
```

```
...
linux-2.6.16/usr/
linux-2.6.16/usr/.gitignore
linux-2.6.16/usr/Kconfig
linux-2.6.16/usr/Makefile
linux-2.6.16/usr/gen_init_cpio.c
linux-2.6.16/usr/initramfs_data.S
[root@linux1 src]#
```

5. «Patcher » le noyau.

```
[root@linux1 src]# cd linux-2.6.16
[root@linux1 linux-2.6.16]# gunzip < /root/patch-2.6.17.gz | patch -s -p1
[root@linux1 linux-2.6.16]#
```

6. Configurer le noyau.

On prend en charge les systèmes de fichiers ReiserFs, XFS et NTFS.

```
[root@linux1 linux-2.6.16]# make menuconfig
Linux Kernel v2.6.16 Configuration

                          Linux Kernel Configuration
  Arrow keys navigate the menu.  <Enter> selects submenus --->.  Highlighted
letters are
  hotkeys.  Pressing <Y> includes, <N> excludes, <M> modularizes features.
Press <Esc><Esc> to exit, <?> for Help, </> for Search.  Legend: [*] built-in  [
] excluded  <M> module  < >

      Code maturity level options  --->
      General setup  --->
      Loadable module support  --->
      Block layer  --->
      Processor type and features  --->
      Power management options (ACPI, APM)  --->
      Bus options (PCI, PCMCIA, EISA, MCA, ISA)  --->
      Executable file formats  --->
      Networking  --->
      Device Drivers  --->
      File systems  --->
      Instrumentation Support  --->
      Kernel hacking  --->
Security options  --->
      Cryptographic options  --->
      Library routines  --->
  ---
      Load an Alternate Configuration File
      Save Configuration to an Alternate File

                        <Select>    < Exit >    < Help >
```

On navigue dans les menus avec les flèches, on valide ses choix grâce à « Select ».

```
File systems  --->
<M> Reiserfs support
[ ]   Enable reiserfs debug mode (NEW)
[*]   Stats in /proc/fs/reiserfs
[*]   ReiserFS extended attributes
```

```
[*]     ReiserFS POSIX Access Control Lists
<M> XFS filesystem support
[*]   XFS Quota support
[*]   XFS Security Label support
[*]   XFS POSIX ACL support
[ ]   XFS Realtime support (EXPERIMENTAL) (NEW)
DOS/FAT/NT Filesystems  --->
    <M> MSDOS fs support
    <M> VFAT (Windows-95) fs support
    (437) Default codepage for FAT
    (ascii) Default iocharset for FAT
    <M> NTFS file system support
          [ ]   NTFS debugging support (NEW)
          [*]   NTFS write support
```

Remarque : Pour chaque composant on a le choix entre :

M	La fonctionnalité sera compilée sous forme de module.
Y	La fonctionnalité sera compilée sous forme statique.
N	La fonctionnalité ne sera pas prise en compte.

On termine en sortant des différents menus par « Exit », et on enregistre le fichier de configuration (choix « Yes »).

```
    Do you wish to save your new kernel configuration?
          < Yes >      <  No  >

*** End of Linux kernel configuration.
*** Execute 'make' to build the kernel or try 'make help'.

[root@linux1 linux-2.6.16]# grep -i NTFS .config
CONFIG_NTFS_FS=m
# CONFIG_NTFS_DEBUG is not set
CONFIG_NTFS_RW=y
```

7. Démarrer la compilation du noyau et des modules (TRES LONG !).

```
[root@linux1 linux-2.6.16]# make
  CHK     include/linux/version.h
  UPD     include/linux/version.h
  SYMLINK include/asm -> include/asm-i386
  HOSTLD  scripts/kconfig/conf
...
  LD [M]  sound/usb/snd-usb-audio.ko
  CC      sound/usb/snd-usb-lib.mod.o
  LD [M]  sound/usb/snd-usb-lib.ko
  CC      sound/usb/usx2y/snd-usb-usx2y.mod.o
  LD [M]  sound/usb/usx2y/snd-usb-usx2y.ko
[root@linux1 linux-2.6.16]#
```

8. Installer le noyau.

```
[root@linux1 linux-2.6.16]# cp arch/i386/boot/bzImage /boot/linux-2.6.17
```

9. Installer les modules dans /lib/modules/<version_du_noyau>/.

```
[root@linux1 linux-2.6.16]# make modules_install
...
  INSTALL sound/pci/ymfpci/snd-ymfpci.ko
```

```
  INSTALL sound/soundcore.ko
  INSTALL sound/synth/emux/snd-emux-synth.ko
  INSTALL sound/synth/snd-util-mem.ko
  INSTALL sound/usb/snd-usb-audio.ko
  INSTALL sound/usb/snd-usb-lib.ko
  INSTALL sound/usb/usx2y/snd-usb-usx2y.ko
if [ -r System.map -a -x /sbin/depmod ]; then /sbin/depmod -ae -F System.map
2.6.17; fi
[root@linux1 linux-2.6.16]#
```

10. Créer un initrd (éventuellement) et on l'installe dans l'arborescence /boot.

```
[root@linux1 linux-2.6.16]# mkinitrd /boot/initrd-2.6.17.img   2.6.17
[root@linux1 linux-2.6.16]# cd /boot/grub
```

11. Configurer le chargeur (ici Grub) pour prendre en compte le nouveau noyau.

On ajoute les lignes en italique.

```
[root@linux1 grub]# cp grub.conf grub.conf.$(date '+%m%d%H%M')
[root@linux1 grub]# vi grub.conf
#boot=/dev/hda
default=0
timeout=5
splashimage=(hd0,0)/grub/splash.xpm.gz
hiddenmenu
title Red Hat Enterprise Linux ES (2.6.9-11.ELsmp)
        root (hd0,0)
        kernel /vmlinuz-2.6.9-11.ELsmp ro root=LABEL=/ rhgb quiet
        initrd /initrd-2.6.9-11.ELsmp.img
title Red Hat Enterprise Linux ES-up (2.6.9-11.EL)
        root (hd0,0)
        kernel /vmlinuz-2.6.9-11.EL ro root=LABEL=/ rhgb quiet
        initrd /initrd-2.6.9-11.EL.img
title New Kernel (2.6.17)
        root (hd0,0)
        kernel /linux-2.6.17 ro root=LABEL=/ rhgb quiet
        initrd /initrd-2.6.17.img
```

12. Tester

```
[root@linux1 grub]# reboot

Broadcast message from root (pts/1) (Tue Jan  2 14:03:53 2007):

The system is going down for reboot NOW!
[root@linux1 grub]#
```

A l'invite de Grub, on sélectionne le nouveau noyau. Après le démarrage et après s'être connecté, on affiche la Version du noyau.

```
[root@linux1 ~]# uname -a
Linux linux1.pinguins 2.6.17 #1 Tue Jan 2 16:31:05 CET 2007 i686 athlon i386
GNU/Linux
[root@linux1 ~]# uname -r
2.6.17
```

Tâche 2 : Ajouter un module à notre nouveau noyau

On prend comme exemple, l'ajout du module UNIONFS.

1. Télécharger le source.

```
[root@linux1 ~]# wget 'ftp://ftp.fsl.cs.sunysb.edu/pub/unionfs/unionfs-
1.3.tar.gz'
…
16:48:59 (146.44 KB/s) - `unionfs-1.3.tar.gz' saved [162,909]

[root@linux1 ~]#
```

2. Extraire les fichiers de l'archive.

```
[root@linux1 ~]# tar xvzf unionfs-1.3.tar.gz
…
unionfs-1.3/man/unionfs.4
unionfs-1.3/man/unionimap.8
[root@linux1 ~]#
```

3. Construire le module du noyau (unionfs.ko) et l'utilitaire (unionctl).

```
[root@linux1 ~]# cd unionfs-1.3
[root@linux1 unionfs-1.3]# make
*******************************************************************************
If you have questions about or problems building Unionfs, please read INSTALL.
*******************************************************************************
…
DUNIONFS_VERSION=\"1.3\" -DSUP_MAJOR=2 -DSUP_MINOR=6 -DSUP_PATCH=17
make[1]: Leaving directory `/root/unionfs-1.3/utils'
[root@linux1 unionfs-1.3]# file utils/unionctl
utils/unionctl: ELF 32-bit LSB executable, Intel 80386, version 1 (SYSV), for
GNU/Linux 2.2.5, dynamically linked (uses shared libs), not stripped
# [root@linux1 unionfs-1.3]# ls -l unionfs.ko
-rw-r--r--  1 root root 1580927 Jan  2 16:55 unionfs.ko
```

Remarques :

1) Si on lit la documentation (les fichiers README et INSTALL) il suffit pour créer le module du noyau d'activer la commande make. Cette commande, via le fichier makefile, recherche les fichiers d'entête du noyau dans l'arborescence /usr/src en se basant sur le noyau actif actuellement.

2) Si l'on désire inclure le module pour un noyau particulier, il faut lancer la commande patch-kernel.sh en donnant en argument la racine de l'arborescence des sources du noyau considéré.

4. Installer le logiciel.

```
[root@linux1 unionfs-1.3]#  make install
…
mkdir -p /usr/local/sbin
cp unionctl /usr/local/sbin
…
mkdir -p //lib/modules/2.6.17/kernel/fs/unionfs
cp unionfs.ko //lib/modules/2.6.17/kernel/fs/unionfs
/sbin/depmod -a
[root@linux1 unionfs-1.3]#
```

5. Essayer de charger le module.

```
[root@linux1 unionfs-1.3]# insmod unionfs.ko
[root@linux1 unionfs-1.3]# lsmod|grep unionfs
unionfs                85152  0

[root@linux1 unionfs-1.3]# tail /var/log/messages
…
Jan  2 17:26:16 linux1 kernel: Registering unionfs 1.3
```

```
Jan  2 17:26:16 linux1 kernel: unionfs: debugging is not enabled
[root@linux1 unionfs-1.3]#
```

Conclusion : L'ajout d'un module au noyau suit une logique similaire à l'ajout d'une bibliothèque dynamique à une application. Essentiellement, on crée le module en activant la commande `make`. Il faut que les sources du noyau ou au minimum ses fichiers d'entête soient installés.

Tâche 3 : Ajouter un module au noyau d'origine (packagé)

On redémarre éventuellement pour travailler sous le noyau d'origine (2.6.9).

1. Vérifier si les fichiers d'en-tête du noyau courant sont installés.

```
[root@linux1 ~]# uname -r
2.6.9-11.ELsmp
[root@linux1 ~]# rpm -q kernel-smp-devel
kernel-smp-devel-2.6.9-11.EL
```

2. Télécharger les sources du module.

```
[root@linux1 ~]# wget 'ftp://ftp.fsl.cs.sunysb.edu/pub/unionfs/unionfs-
1.1.5.tar.gz'
...
17:44:24 (64.34 KB/s) - `unionfs-1.1.5.tar.gz' saved [153,220]
```

Remarque : On vérifie bien sur le site proposant Unionfs, que l'on télécharge la version compatible avec la version du noyau.

3. Extraire les fichiers de l'archive.

```
[root@linux1 ~]# tar xvzf unionfs-1.1.5.tar.gz
...
unionfs-1.1.5/man/unionfs.4
unionfs-1.1.5/man/unionimap.8
```

4. Compiler le module.

```
[root@linux1 ~]# cd unionfs-1.1.5
[root@linux1 unionfs-1.1.5]# make
...
make -C /lib/modules/2.6.9-11.ELsmp/build SUBDIRS=/root/unionfs-1.1.5
FISTDEVMK=/root/unionfs-1.1.5/fistdev.mk modules
...
```

5. L'installer et essayer de le charger.

```
[root@linux1 unionfs-1.1.5]# make install
...
[root@linux1 unionfs-1.1.5]# insmod unionfs.ko
[root@linux1 unionfs-1.1.5]# lsmod |grep unionfs
unionfs                279584  0
[root@linux1 unionfs-1.1.5]# cd
```

Tâche 4 : Installer un noyau packagé

1. Installer le paquetage.

```
[root@linux1 ~]# yum install kernel-hugemem
...
Install      1 Package(s)
Update       0 Package(s)
Remove       0 Package(s)
Total download size: 9.2 M
```

```
Is this ok [y/N]: y
Downloading Packages:
Running Transaction Test
Finished Transaction Test
Transaction Test Succeeded
Running Transaction
  Installing: kernel-hugemem                ########################## [1/1]

Installed: kernel-hugemem.i686 0:2.6.9-11.EL
Complete!
```

2. Vérifier si tout est correct.

```
[root@linux1 ~]# ls -ld /lib/modules/2.6.9-11.ELhugemem/
drwxr-xr-x  3 root root 4096 Jan  2 17:50 /lib/modules/2.6.9-11.ELhugemem/
[root@linux1 ~]# ls -l /boot/vmlinuz-2.6.9-11.ELhugemem
-rw-r--r--  1 root root 1381680 May 21  2005 /boot/vmlinuz-2.6.9-11.ELhugemem
[root@linux1 unionfs-1.1.5]# tail /boot/grub/grub.conf
…
title Red Hat Enterprise Linux ES (2.6.9-11.ELhugemem)
        root (hd0,0)
        kernel /vmlinuz-2.6.9-11.ELhugemem ro root=LABEL=/ rhgb quiet
        initrd /initrd-2.6.9-11.ELhugemem.img
…
```

Remarque : Et c'est tout ! L'installation du paquetage a installé le nouveau noyau, les modules et a également mis à jour la configuration du chargeur.

Tâche 5 : Compiler le noyau à partir des sources de la distribution

1. Récupérer les sources.

```
[root@linux1 ~]# wget
'ftp://updates.redhat.com/enterprise/4ES/en/os/SRPMS/kernel-2.6.9-11.EL.src.rpm'
```

2. Installer le paquetage.

```
[root@linux1 ~]# rpm -ivh kernel-2.6.9-11.EL.src.rpm
   1:kernel                 ########################################### [100%]
[root@linux1 ~]#
```

Remarque : On pouvait aussi installer le package grâce à la commande yum:
yum localinstall kernel-2.6.9.*.src.rpm

```
[root@linux1 ~]# cd /usr/src/redhat/SPECS/
[root@linux1 SPECS]# rpmbuild -bp --target=i686  kernel-2.6.spec
Building target platforms: i686
Building for target i686
Executing(%prep): /bin/sh -e /var/tmp/rpm-tmp.23691
+ umask 022
…
removed `./fs/read_write.c.orig'
removed `./fs/char_dev.c.orig'
+ find . -name '*~' -exec rm -fv '{}' ';'
+ exit 0
[root@linux1 SPECS]# cd /usr/src/redhat/BUILD/kernel-2.6.9/linux-2.6.9/
[root@linux1 linux-2.6.9]# diff .config configs/kernel-2.6.9-i686.config
[root@linux1 linux-2.6.9]# make oldconfig
…
```

```
CRC-CCITT functions (CRC_CCITT) [M/y/?] m
CRC32 functions (CRC32) [Y/?] y
CRC32c (Castagnoli, et al) Cyclic Redundancy-Check (LIBCRC32C) [M/y/?] m
[root@linux1 linux-2.6.9]#
[root@linux1 linux-2.6.9]# vi .config
...
# CONFIG_NTFS_DEBUG is not set
CONFIG_NTFS_FS=m
CONFIG_NTFS_RW=y
...
[root@linux1 linux-2.6.9]# vi Makefile
...
EXTRAVERSION = -ntfs
[root@linux1 linux-2.6.9]# make
...
  CC      sound/usb/usx2y/snd-usb-usx2y.mod.o
  LD [M]  sound/usb/usx2y/snd-usb-usx2y.ko
[root@linux1 linux-2.6.9]# cp arch/i386/boot/bzImage /boot/linux-2.6.9-ntfs
[root@linux1 linux-2.6.9]# make modules_install
...
  INSTALL sound/usb/snd-usb-lib.ko
  INSTALL sound/usb/usx2y/snd-usb-usx2y.ko
if [ -r System.map ]; then /sbin/depmod -ae -F System.map  2.6.9-ntfs; fi
[root@linux1 linux-2.6.9]# mkinitrd /boot/initrd-2.6.9-ntfs.img   2.6.9-ntfs
[root@linux1 linux-2.6.9]# cd /boot/grub
[root@linux1 grub]# cp grub.conf grub.conf.$(date '+%m%d%H%M')
[root@linux1 grub]# vi grub.conf
#boot=/dev/hda
default=0
timeout=5
splashimage=(hd0,0)/grub/splash.xpm.gz
hiddenmenu
title Red Hat Enterprise Linux ES (2.6.9-11.ELsmp)
        root (hd0,0)
        kernel /vmlinuz-2.6.9-11.ELsmp ro root=LABEL=/ rhgb quiet
        initrd /initrd-2.6.9-11.ELsmp.img
title Red Hat Enterprise Linux ES-up (2.6.9-11.EL)
        root (hd0,0)
        kernel /vmlinuz-2.6.9-11.EL ro root=LABEL=/ rhgb quiet
        initrd /initrd-2.6.9-11.EL.img
title Red Hat Kernel et NTFS (2.6.9-ntfs)
        root (hd0,0)
        kernel /linux-2.6.9-ntfs ro root=LABEL=/ rhgb
        initrd /initrd-2.6.9-ntfs.img

[root@linux1 linux-2.6.9]# reboot
```

Après le redémarrage (on a choisi le nouveau noyau) et après s'être connecté :

```
[root@linux1 ~]# uname -a
Linux linux1.pinguins 2.6.9-ntfs #1 Tue Jan 2 23:05:45 CET 2007 i686 athlon i386
GNU/Linux
[root@linux1 ~]#
```

Tâche 6 :
Les modules dynamiques du noyau

1. Lister les modules présents actuellement en mémoire.

```
[root@linux1 ~]# lsmod |more
Module                  Size  Used by
lp                     16456  0
ipv6                  243488  14
parport_pc             30148  0
parport                38600  1 parport_pc
autofs4                24580  0
sunrpc                145468  1
...
8139too                29568  0
mii                     9600  1 8139too
floppy                 61188  0
ext3                  124680  3
jbd                    57876  1 ext3
```

2. Décharger un module (non utilisé).

```
[root@linux1 ~]# rmmod lp
[root@linux1 ~]# lsmod | grep lp
[root@linux1 ~]#
```

3. Essayer de décharger un module utilisé.

```
[root@linux1 ~]# rmmod parport
ERROR: Module parport is in use by lp,parport_pc
[root@linux1 ~]# rmmod lp
[root@linux1 ~]# rmmod parport_pc
[root@linux1 ~]# rmmod parport
[root@linux1 ~]# lsmod | grep parport
[root@linux1 ~]#
```

4. Charger un module avec la commande modprobe.

La commande `modprobe` charge les modules dépendants.

```
[root@linux1 ~]# modprobe lp
[root@linux1 ~]# lsmod | grep lp
lp                     16456  0
parport                38600  2 lp,parport_pc
[root@linux1 ~]# rmmod lp
```

5. Charger un module avec insmod.

```
[root@linux1 ~]# uname -r
2.6.17
[root@linux1 ~]# grep lp /lib/modules/2.6.17/modules.dep
/lib/modules/2.6.17/kernel/drivers/char/lp.ko:
/lib/modules/2.6.17/kernel/drivers/parport/parport.ko
...
[root@linux1 ~]# rm /lib/modules/2.6.17/modules.dep
rm: remove regular file `/lib/modules/2.6.17/modules.dep'? y
[root@linux1 ~]# rmmod lp
[root@linux1 ~]# modprobe lp
FATAL: Could not load /lib/modules/2.6.17/modules.dep: No such file or directory
[root@linux1 ~]# insmod /lib/modules/2.6.17/kernel/drivers/char/lp.ko
[root@linux1 ~]# lsmod |grep lp
lp                     16456  0
```

```
parport                38600  2 lp,parport_pc
[root@linux1 ~]#
```

6. Régénérer le fichier de dépendance des modules.

```
[root@linux1 ~]# depmod
[root@linux1 ~]# rmmod lp
[root@linux1 ~]# modprobe lp
[root@linux1 ~]# lsmod |grep lp
lp                     16456  0
parport                38600  2 lp,parport_pc
[root@linux1 ~]#
```

7. Afficher des informations sur un module.

```
[root@linux1 ~]# modinfo lp
filename:       /lib/modules/2.6.17/kernel/drivers/char/lp.ko
parmtype:       parport:array of charp
parmtype:       reset:bool
alias:          char-major-6-*
license:        GPL
vermagic:       2.6.17 SMP mod_unload 686 REGPARM gcc-3.4
depends:        parport
```

8. Essayer de charger un module construit pour une autre version de noyau.

```
[root@linux1 ~]# insmod /lib/modules/2.6.9-42.EL/kernel/drivers/char/lp.ko
insmod: error inserting '/lib/modules/2.6.9-42.EL/kernel/drivers/char/lp.ko': -1
Invalid module format
[root@linux1 ~]#
```

Tâche 7 : L'arborescence /proc

1. Afficher les paramètres du noyau courant et sa version.

```
[root@linux1 ~]# cat /proc/cmdline
ro root=LABEL=/1 rhgb quiet
[root@linux1 ~]# cat /proc/version
Linux version 2.6.17 (root@linux1.pinguins) (gcc version 3.4.6 20060404 (Red Hat
3.4.6-3)) #1 Tue Feb 6 17:20:08 CET 2007
[root@linux1 ~]#
```

2. Afficher les caractéristiques du processeur.

```
[root@linux1 ~]# more /proc/cpuinfo
processor       : 0
vendor_id       : AuthenticAMD
cpu family      : 15
model           : 44
model name      : AMD Sempron(tm) Processor 3100+
stepping        : 2
cpu MHz         : 1800.895
cache size      : 256 KB
fdiv_bug        : no
hlt_bug         : no
f00f_bug        : no
coma_bug        : no
fpu             : yes
fpu_exception   : yes
cpuid level     : 1
wp              : yes
```

```
flags           : fpu vme de pse tsc msr pae mce cx8 apic sep mtrr pge mca cmov
pat pse36 clflush mmx fxsr sse sse2 syscall nx mmxext fxsr_opt lm 3dnowext 3dnow
 up pni lahf_lm ts fid vid ttp tm stc
bogomips        : 3606.30
```

3. Afficher des informations sur un processus.

```
[root@linux1 ~]# ps
  PID TTY          TIME CMD
 4085 pts/0    00:00:00 bash
23446 pts/0    00:00:00 ps
```

a) Ses arguments.

```
[root@linux1 ~]# more /proc/4085/cmdline
-bash
```

b) Son environnement.

```
[root@linux1 ~]# more /proc/4085/environ
USER=rootsr/sbin:/usr/bin17 4560 22TTY=/dev/pts/0
```

c) La cartographie de son espace virtuel.

```
[root@linux1 ~]# more /proc/4085/maps
005ca000-005df000 r-xp 00000000 03:02 331425     /lib/ld-2.3.4.so
005df000-005e0000 r-xp 00015000 03:02 331425     /lib/ld-2.3.4.so
005e0000-005e1000 rwxp 00016000 03:02 331425     /lib/ld-2.3.4.so
005e8000-0070d000 r-xp 00000000 03:02 331426     /lib/tls/libc-2.3.4.so
0070d000-0070f000 r-xp 00124000 03:02 331426     /lib/tls/libc-2.3.4.so
0070f000-00711000 rwxp 00126000 03:02 331426     /lib/tls/libc-2.3.4.so
00711000-00713000 rwxp 00711000 00:00 0
00715000-00717000 r-xp 00000000 03:02 331428     /lib/libdl-2.3.4.so
00717000-00718000 r-xp 00001000 03:02 331428     /lib/libdl-2.3.4.so
00718000-00719000 rwxp 00002000 03:02 331428     /lib/libdl-2.3.4.so
00821000-00824000 r-xp 00000000 03:02 326045     /lib/libtermcap.so.2.0.8
00824000-00825000 rwxp 00002000 03:02 326045     /lib/libtermcap.so.2.0.8
08047000-080d8000 r-xp 00000000 03:02 649046     /bin/bash
080d8000-080de000 rwxp 00090000 03:02 649046     /bin/bash
080de000-08125000 rwxp 080de000 00:00 0          [heap]
b7cb4000-b7cb5000 r-xp 00000000 03:02 16364      /usr/lib/gconv/ISO8859-1.so
b7cb5000-b7cb7000 rwxp 00000000 03:02 16364      /usr/lib/gconv/ISO8859-1.so
b7cb7000-b7cd5000 r-xp 00000000 03:02 1250593    /usr/share/locale/fr/LC_MESSAGE
S/libc.mo
b7cd5000-b7cd7000 rwxp b7cd5000 00:00 0
b7cd7000-b7cdd000 r-xs 00000000 03:02 16439      /usr/lib/gconv/gconv-modules.ca
che
b7cdd000-b7cde000 r-xp 02a55000 03:02 1236299    /usr/lib/locale/locale-archive
b7cde000-b7d19000 r-xp 00d62000 03:02 1236299    /usr/lib/locale/locale-archive
b7d19000-b7f19000 r-xp 00000000 03:02 1236299    /usr/lib/locale/locale-archive
b7f19000-b7f22000 r-xp 00000000 03:02 324530     /lib/libnss_files-2.3.4.so
b7f22000-b7f23000 r-xp 00008000 03:02 324530     /lib/libnss_files-2.3.4.so
b7f23000-b7f24000 rwxp 00009000 03:02 324530     /lib/libnss_files-2.3.4.so
b7f24000-b7f26000 rwxp b7f24000 00:00 0
bf8d9000-bf8ee000 rw-p bf8d9000 00:00 0          [stack]
ffffe000-fffff000 ---p 00000000 00:00 0          [vdso]
```

d) L'état du processus.

```
[root@linux1 ~]# more /proc/4085/status
Name:   bash
```

```
State:  S (sleeping)
SleepAVG:       98%
Tgid:   4085
Pid:    4085
PPid:   4083
TracerPid:      0
Uid:    0       0       0       0
Gid:    0       0       0       0
FDSize: 256
Groups: 0 1 2 3 4 6 10
VmPeak:     4796 kB
VmSize:     4796 kB
VmLck:         0 kB
VmHWM:      1584 kB
VmRSS:      1584 kB
VmData:      308 kB
VmStk:        84 kB
VmExe:       580 kB
VmLib:      3768 kB
VmPTE:        20 kB
Threads:        1
SigQ:   0/7151
SigPnd: 0000000000000000
ShdPnd: 0000000000000000
SigBlk: 0000000000010000
SigIgn: 0000000000384004
SigCgt: 000000004b813efb
CapInh: 0000000000000000
CapPrm: 00000000fffffeff
CapEff: 00000000fffffeff
[root@linux1 ~]#
```

Tâche 8 : Paramétrer à chaud le noyau

1. Lister les paramètres modifiables du noyau.

```
[root@linux1 ~]# sysctl -a
...
fs.quota.drops = 0
fs.quota.lookups = 0
fs.suid_dumpable = 0
fs.inotify.max_queued_events = 16384
fs.inotify.max_user_watches = 8192
fs.inotify.max_user_instances = 128
fs.aio-max-nr = 65536
fs.aio-nr = 0
fs.lease-break-time = 45
fs.dir-notify-enable = 1
fs.leases-enable = 1
fs.overflowgid = 65534
fs.overflowuid = 65534
fs.dentry-state = 6529  5225    45      0       0       0
fs.file-max = 44380
fs.file-nr = 384        0       44380
```

```
fs.inode-state = 6242   951     0       0       0       0       0
fs.inode-nr = 6242      951
error: unknown error 22 reading key 'fs.binfmt_misc.register'
fs.binfmt_misc.status = enabled
[root@linux1 ~]#
```

2. Lire la documentation du noyau.

Remarque : Tout ce qui concerne le paramétrage à chaud du noyau est associé au nom « SYSCTL ».

```
[root@linux1 ~]# rpm -q kernel-doc
kernel-doc-2.6.9-42.EL
[root@linux1 ~]# cd /usr/share/doc/kernel-doc*/Documentation
[root@linux1 Documentation]# ls -F
...
exception.txt           mono.txt                sysctl/
fb/                     moxa-smartio            sysrq.txt
filesystems/            MSI-HOWTO.txt           telephony/
firmware_class/         mtrr.txt                time_interpolators.txt
floppy.txt              nbd.txt                 tipar.txt
ftape.txt               networking/             tty.txt
hayes-esp.txt           nfsroot.txt             uml/
highuid.txt             nmi_watchdog.txt        unicode.txt
hpet.txt                numastat.txt            usb/
hw_random.txt           oops-tracing.txt        VGA-softcursor.txt
i2c/                    paride.txt              video4linux/
i2o/                    parisc/                 vm/
i386/                   parport-lowlevel.txt    voyager.txt
ia64/                   parport.txt             watchdog/
ide.txt                 pci.txt                 x86_64/
infiniband/             pm.txt                  xterm-linux.xpm
initrd.txt              pnp.txt                 zorro.txt
[root@linux1 Documentation]# ls sysctl/
abi.txt  fs.txt  kernel.txt  README  sunrpc.txt  vm.txt
[root@linux1 Documentation]# more sysctl/kernel.txt
...
- suid_dumpable
- sysrq                         ==> Documentation/sysrq.txt
- tainted
- threads-max
- version
...
[root@linux1 Documentation]# more sysrq.txt
Linux Magic System Request Key Hacks
Documentation for sysrq.c version 1.15
Last update: $Date: 2001/01/28 10:15:59 $

*  What is the magic SysRq key?
~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
It is a 'magical' key combo you can hit which the kernel will respond to
regardless of whatever else it is doing, unless it is completely locked up.

*  How do I enable the magic SysRq key?
~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
You need to say "yes" to 'Magic SysRq key (CONFIG_MAGIC_SYSRQ)' when
```

```
configuring the kernel. When running on a kernel with SysRq compiled in, it
may be DISABLED at run-time using following command:

        echo "0" > /proc/sys/kernel/sysrq

Note that previous versions disabled sysrq by default, and you were required
to specifically enable it at run-time. That is not the case any longer.

*  How do I use the magic SysRq key?
~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
On x86    - You press the key combo 'ALT-SysRq-<command key>'. Note - Some
            keyboards may not have a key labeled 'SysRq'. The 'SysRq' key is
            also known as the 'Print Screen' key. Also some keyboards cannot
            handle so many keys being pressed at the same time, so you might
            have better luck with "press Alt", "press SysRq", "release Alt",
            "press <command key>", release everything.
...

*  What are the 'command' keys?
~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
'r'     - Turns off keyboard raw mode and sets it to XLATE.

'k'     - Secure Access Key (SAK) Kills all programs on the current virtual
          console. NOTE: See important comments below in SAK section.

'b'     - Will immediately reboot the system without syncing or unmounting
          your disks.

'c'     - Intentionally crash the system without syncing or unmounting
          your disks.  This is most useful if the NETDUMP client package
          and/or the DISKDUMP package have been installed.

'o'     - Will shut your system off (if configured and supported).

's'     - Will attempt to sync all mounted filesystems.

'u'     - Will attempt to remount all mounted filesystems read-only.

'p'     - Will dump the current registers and flags to your console.

't'     - Will dump a list of current tasks and their information to your
          console.

'm'     - Will dump current memory info to your console.

'v'     - Dumps Voyager SMP processor info to your console.
'0'-'9' - Sets the console log level, controlling which kernel messages
          will be printed to your console. ('0', for example would make
          it so that only emergency messages like PANICs or OOPSes would
          make it to your console.)

'e'     - Send a SIGTERM to all processes, except for init.

'i'     - Send a SIGKILL to all processes, except for init.
```

```
'l'     - Send a SIGKILL to all processes, INCLUDING init. (Your system
          will be non-functional after this.)

'h'     - Will display help ( actually any other key than those listed
          above will display help. but 'h' is easy to remember :-)
...
[root@linux1 Documentation]# cd
```

3. Visualiser la valeur d'un paramètre.

```
[root@linux1 ~]# cat /proc/sys/kernel/sysrq
0
```

4. Le modifier à chaud.

```
[root@linux1 ~]# echo 1 > /proc/sys/kernel/sysrq
[root@linux1 ~]# cat /proc/sys/kernel/sysrq
1
```

5. Tester sysrq.

Sur la console maîtresse ([Ctrl]+[Alt]+[F1]), on tape simultanément sur : [Alt]+[Impr écran]+m

Attention : le clavier est considéré comme QWERTY.

On doit voir la configuration de la mémoire :

```
...
Free swap = 5122104kB
Total swap = 512104kB
```

6. Visualiser la valeur d'un paramètre avec la commande sysctl.

```
[root@linux1 ~]# sysctl kernel.sysrq
kernel.sysrq = 1
```

7. Modifier la valeur d'un paramètre avec la commande sysctl.

```
[root@linux1 ~]# sysctl kernel.sysrq=0
kernel.sysrq = 0
[root@linux1 ~]# sysctl kernel.sysrq
kernel.sysrq = 0
```

8. Modifier de manière permanente un paramètre (le paramètre sera configuré à chaque démarrage).

```
[root@linux1 ~]# echo "kernel.sysrq = 1" >> /etc/sysctl.conf
```

Tâche 9 : Les IPC System V

1. Lister les IPC courantes.

```
[root@linux1 ~]# ipcs

------ Shared Memory Segments --------
key        shmid      owner      perms      bytes      nattch     status

------ Semaphore Arrays --------
key        semid      owner      perms      nsems

------ Message Queues --------
key        msqid      owner      perms      used-bytes   messages
```

2. Lister le paramétrage des tables systèmes IPC.

```
[root@linux1 ~]# ipcs -l
```

```
------ Shared Memory Limits --------
max number of segments = 4096
max seg size (kbytes) = 32768
max total shared memory (kbytes) = 8388608
min seg size (bytes) = 1

------ Semaphore Limits --------
max number of arrays = 128
max semaphores per array = 250
max semaphores system wide = 32000
max ops per semop call = 32
semaphore max value = 32767

------ Messages: Limits --------
max queues system wide = 16
max size of message (bytes) = 8192
default max size of queue (bytes) = 16384
```

3. Lister les fichiers de /proc qui mémorisent la configuration des IPC.

```
[root@linux1 ~]# ls -l /proc/sys/kernel |grep -e sem -e shm -e msg
-rw-r--r--  1 root root 0 Feb  7 16:15 msgmax
-rw-r--r--  1 root root 0 Feb  7 16:15 msgmnb
-rw-r--r--  1 root root 0 Feb  7 16:15 msgmni
-rw-r--r--  1 root root 0 Feb  7 16:15 sem
-rw-r--r--  1 root root 0 Feb  7 16:15 shmall
-rw-r--r--  1 root root 0 Feb  7 16:15 shmmax
-rw-r--r--  1 root root 0 Feb  7 16:15 shmmni
[root@linux1 ~]# more /proc/sys/kernel/msgmax
8192
```

4. Créer une IPC de type tableau de sémaphores (on utilise un petit programme Perl).

```
[root@linux1 ~]# vi ipc.pl
use IPC::SysV qw(IPC_CREAT);
$IPC_KEY = 123;
$id = semget($IPC_KEY, 10, 0666 | IPC_CREAT ) || die "$!";
print "shm key $id\n";
[root@linux1 ~]# perl ipc.pl
shm key 32769
```

5. Visualiser ses caractéristiques.

```
[root@linux1 ~]# ipcs -s

------ Semaphore Arrays --------
key        semid      owner      perms      nsems
0x0000007b 32769      root       666        10
```

6. Détruire l'IPC.

```
[root@linux1 ~]# ipcrm -h
ipcrm: invalid option -- h
ipcrm: illegal option -- ?
usage: ipcrm [ [-q msqid] [-m shmid] [-s semid]
          [-Q msgkey] [-M shmkey] [-S semkey] ... ]
[root@linux1 ~]# ipcrm -s 32769
[root@linux1 ~]#
```

- *IRQ, DMA, ACPI*
- *SCSI, PCI-X, PCIe*
- */dev/hda*
- */sys/block/sda*
- *Udev, lm_sensors, SMART*

4

La gestion des périphériques

Objectifs

Ce chapitre traite de la gestion des périphériques. On apprend comment sont gérés les périphériques sous Linux. Il y a beaucoup de vocabulaire et de concepts associés (et donc beaucoup de théorie). En pratique on apprend à recueillir des informations de plusieurs sources concernant les périphériques. On apprend également à utiliser quelques technologies connexes comme Udev, lm_sensors ou SMART.

Contenu

L'architecture d'un ordinateur

La gestion des périphériques sous Linux

Ajouter un périphérique

Le FS sysfs

La technologie Udev

La technologie lm_sensors

La technologie SMART

Ateliers



L'architecture d'un ordinateur

La théorie

Les composants d'un ordinateur

Un ordinateur est composé logiquement de différentes entités :

- Une unité centrale ou CPU (Central Processing Unit) qui est le cœur de l'ordinateur : c'est elle qui effectue les traitements. Un ordinateur multiprocesseur comporte plusieurs CPU.
- La mémoire vive ou RAM (Random Access Memory) où sont conservés les programmes et les données qui sont accessibles immédiatement par le ou les CPU.
- Des périphériques qui conservent de manière durable les informations, par exemple les disques dur, ou qui sont le lien avec le monde externe : clavier, écran, souris ou qui permettent l'échange de données avec d'autres ordinateurs comme les réseaux.
- Les contrôleurs qui relient les périphériques aux autres composants : la mémoire, le CPU. Les ponts (bridge) sont des contrôleurs particuliers : ils permettent de passer d'un bus à un autre.
- Les bus qui relient les composants de l'ordinateur : CPU, mémoire, contrôleurs. Le bus ISA est le bus primitif des premiers PC. Actuellement, le bus PCI fournit l'ossature du système d'E/S d'un PC, même si d'autres bus sont utilisés : ISA, EISA, USB, SCSI, …

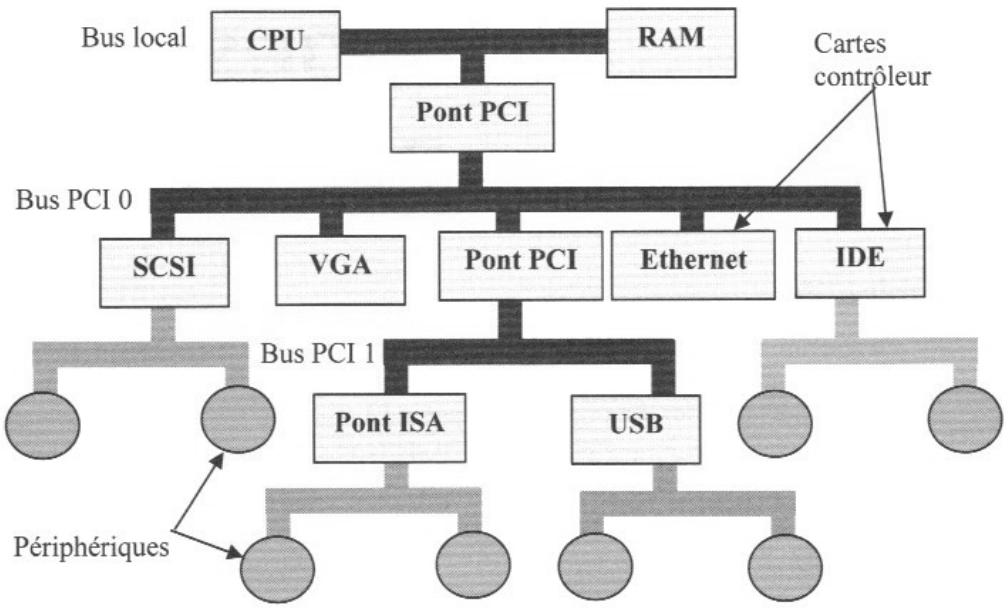


Fig. Arborescence matériel (Bus, ponts, contrôleur et périphériques)

Un ordinateur est composé physiquement des entités suivantes :

- Un microprocesseur (voir plusieurs) : c'est le CPU qui sur les ordinateurs actuels est composé d'une seule puce (chipset).
- Une carte mère (motherboard) : c'est le support des autres composants. Elle est essentiellement composée des principaux bus du système qui sont gravés dessus. Quelques contrôleurs (bridge ou non) sont également intégrés à la carte. On a aussi des slots et des connecteurs qui permettent d'enficher le microprocesseur, des barrettes de mémoire (RAM), des cartes contrôleurs ou de se relier directement à des périphériques.

- Des cartes contrôleurs enfichées dans un des slots de la carte mère.
- Des chipsets : ce sont des puces présentes soit directement sur la carte mère ou soit sur une carte contrôleur. Elles peuvent correspondre aux principales fonctionnalités d'un contrôleur ou intégrer plusieurs fonctionnalités. Un numéro les identifie (il apparaît sur le boîtier renfermant la puce).
- Un boîtier qui contient la carte mère, l'alimentation et quelques périphériques, comme le disque dur, le lecteur de cdrom.
- Des périphériques externes reliés au boîtier, comme l'écran, le clavier, la souris, le scanner, l'imprimante, le réseau.

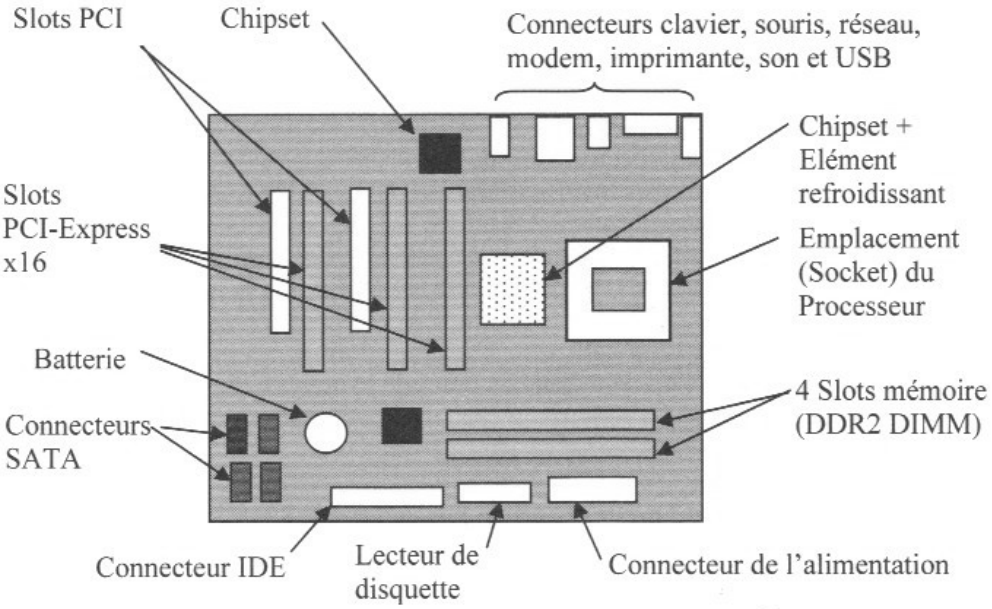


Fig. Carte mère (motherboard)

Remarque : certains ordinateurs, appelés « lames », sont composés essentiellement d'une carte mère enfichée dans un rack normalisé. Ils ne possèdent ni écran ni clavier et ne sont accessibles que par le réseau.

Gestion matérielle des périphériques

Les échanges de données entre un périphérique et les autres composants sont effectués grâce à un ensemble de mécanismes :

Les interruptions matérielles

Une interruption ou IRQ (Interrupt ReQuest) est générée par un contrôleur de périphérique à destination du CPU. Son rôle est de prévenir le système qu'une donnée est disponible au niveau du contrôleur. Si l'interruption est acceptée, le programme en cours est interrompu, et un gestionnaire est déclenché qui gère l'interruption : par exemple, lecture des données et déblocage de l'application en attente de la fin de l'entrée/sortie. Les anciens PC étaient limités à 8 ou 15 IRQ. Un chipset, le PIC (Programmable Interrupt Controler) gère les interruptions. Actuellement, grâce à l'APIC (Advanced PIC), on peut gérer 255 niveaux d'IRQ. Leur numéro, indique leur priorité : ainsi pour un système donné, l'interruption du clavier (IRQ 1) est prioritaire par rapport au disque dur (IRQ 14). Normalement un contrôleur est associé à une IRQ, mais plusieurs contrôleurs peuvent partager la même IRQ (si on utilise différentes astuces comme les bus PCI). Un contrôleur peut même n'avoir pas d'IRQ du tout. Dans les contrôleurs anciens associés au bus primitif ISA, l'IRQ était

paramétrée de manière matérielle, par exemple par un cavalier. Les contrôleurs PCI, eux, peuvent avoir leur IRQ fixé dynamiquement par le logiciel embarqué (Firmware ou BIOS).

Remarque : il existe aussi les interruptions logicielles qui proviennent de logiciels.

Les adresses d'entrées/sorties

Pour pouvoir échanger des données entre le contrôleur et le CPU, un contrôleur est associé à une plage spécifique d'adresses d'entrées/sorties. Ces adresses correspondent soit à de vraies adresses mémoires, on parle alors d'E/S mappé en mémoire, soit à des ports d'E/S dissociés de la mémoire. Comme pour les IRQ, les contrôleurs ISA doivent être configurés de manière matérielle et inversement, les contrôleurs PCI ont leur adresse d'E/S fixé au démarrage.

Le DMA (Direct Memory Access)

Ce système permet le transfert de données directement d'un contrôleur ISA à la mémoire. Il existe plusieurs canaux DMA chacun identifié par un numéro. Les contrôleurs PCI utilisent une technique similaire le « bus mastering » qui permet également un transfert de données directement du contrôleur vers la mémoire. Cette technique est beaucoup plus transparente pour le système que le DMA.

Les bus

Un bus est composé de plusieurs fils reliant des équipements électroniques. Ces fils sont soit gravés sur une carte soit ils forment une « nappe ».

Le bus local

C'est le nom que l'on donne au bus qui relie le ou les CPU, la mémoire et les ponts (contrôleur de bus).

Le bus ISA

C'est le bus primitif du PC. C'est un bus de largeur 8 bits de données. Il est toujours utilisé pour certains périphériques ancestraux comme le clavier, les ports séries et parallèles. Si votre PC dispose de slots ISA ou EISA (Extented ISA) vous pouvez ajouter une carte contrôleur ISA. Le problème c'est qu'il faut paramétrer l'IRQ, les adresses d'E/S et éventuellement le canal DMA pour ne pas avoir de conflit matériel. Si la carte est récente, elle supporte le protocole PnP (Plug and Play) qui permet à l'OS de fixer ces valeurs.

Le bus PCI

C'est le bus le plus courant pour effectuer des E/S, en effet beaucoup de cartes contrôleurs ou de chipset sont PCI. C'est un standard de l'industrie, il est présent dans la plupart des ordinateurs et pas seulement dans les PC. C'est également lui qui forme la colonne vertébrale du PC : grâce à des ponts, on accède à d'autres bus, comme ISA, USB ou SCSI. Il est par nature PnP : les IRQ, les adresses d'E/S sont fixés au démarrage par le logiciel embarqué (firmware). C'est un bus 32 bits de données à 66 MHz.

Une machine peut contenir un ou plusieurs domaines PCI numéroté de 0000 à ffff. Chaque domaine contient un ou plusieurs bus racine, chacun étant numéroté de 00 à ff. Un bus racine peut être connecté à un ou plusieurs bus via des ponts. Chaque bus est connecté à un ou plusieurs contrôleurs (jusqu'à 32), chacun pouvant supporter une ou plusieurs fonctions (jusqu'à 8). On identifie donc une fonction par le nombre : dddd:bb:xx.y (dddd=domaine, bb=bus racine, xx=contrôleur, y=fonction). Un slot PCI est évidemment associé à une adresse fixe.

Un contrôleur PCI publie son constructeur et un identifiant.

Il existe plusieurs variantes ou évolutions du bus PCI : AGP, PCI-X et PCIe.

Le bus AGP

Le bus AGP (Accelerated Graphics Port) est dérivé du bus PCI, mais plus rapide et dédié à la connexion d'une carte graphique.

Le bus PCI-X

C'est une évolution du bus PCI. C'est un bus à 64 bits de données à 133MHz. Il est compatible avec les périphériques PCI anciens. Il est maintenant supplanté par le bus PCI Express.

Le bus PCI Express

Le bus PCI Express (ou PCIe) C'est une évolution du bus PCI mais à très haut débit et série. Les slots sont différents évidemment des slots PCI classiques, par exemple PCI Express x1 et x16 respectivement pour une liaison et seize liaisons PCIe. Il permet des transferts à 50 Mbits/s.

Le bus USB

C'est un bus série qui a remplacé les liaisons COM (série) et LPT (parallèle) des premiers PC. Il permet également de brancher des scanners, un clavier, une souris, et des unités de stockage comme les clés USB. Il supporte un mécanisme PnP qui permet de brancher ou retirer un périphérique à chaud. Les périphériques sont reliés entre par des Hubs. L'ensemble forme un arbre. La racine de cet arbre est le Hub faisant partie du système. Chaque périphérique et chaque Hub possèdent une adresse. Un Hub permet de relier des périphériques et d'autres Hub au Hub hiérarchiquement supérieur. Quand on connecte un nouveau périphérique USB, ce dernier publie différentes informations : sa classe (Hub, interface humaine, audio, communication, imprimante, scanner, outils de stockage, ...), son constructeur, son nom et sa version. D'autre part, il se voit attribuer un numéro de périphérique unique de 1 à 127 (device number). Les contrôleurs USB sont intégrés à la carte mère ou à une carte contrôleur PCI. Ils sont compatibles avec l'interface OHCI (Open Host Controller Interface) de Compaq ou avec l'interface UHCI (Universal Host Controller Interface) d'Intel.

Le bus SCSI

Le bus SCSI (Small computer System Interface) permet de relier des périphériques de différentes sortes au système, mais principalement des disques. La plupart des serveurs sont équipés de disques SCSI. Chaque périphérique présent sur un bus SCSI parallèle est identifié par un « SCSI ID » de 0 à 7 ou de 0 à 15 en fonction de la largeur du bus. Le contrôleur lui-même possède un « SCSI ID », c'est la valeur la plus grande (7 ou 15). Pour les autres périphériques, le choix se fait par des cavaliers ou par l'emplacement du périphérique (son slot). Chaque périphérique possède aussi au moins un LUN (0). Mais il peut en posséder plusieurs comme par exemple un juke box où chaque plateau a un LUN attribué.

Le BIOS

Le BIOS (Basic Input Output System) est un programme en mémoire morte (ROM) qui est exécuté au démarrage du système. Si le BIOS, appelé aussi firmware, est stocké en EEPROM, il peut être mis à jour (« flasher le Bios »). Le BIOS, en final, essaye d'activer un OS.

La gestion de l'alimentation

APM

L'APM (Advanced Power Management) est une spécification du BIOS qui permet de gérer l'alimentation des ordinateurs, dont la batterie des ordinateurs portables.

ACPI

L'ACPI (Advanced Configuration and Power Interface) est un standard développé par Compaq, Intel, Microsoft, Phoenix et Toshiba. L'ACPI veut remplacer l'APM, l'ancien standard. Il intègre de nouvelles fonctionnalités, comme le contrôle thermique. Contrairement à l'APM géré par le BIOS, l'ACPI est contrôlé par l'OS.

Le savoir concret

Les fichiers de /proc

Remarque : Informations provenant du noyau.	
/proc/cpuinfo	Renseignement sur le CU.
/proc/interrupts	Liste les interruptions.
/proc/ioports	Liste les ports d'entrées/sorties et les pilotes associés.
/proc/iomem	Renseigne sur la mémoire associée aux entrée/sorties.
/proc/dma	Renseigne sur le DMA (pour les disques IDE).
/proc/acpi/	Renseigne sur l'ACPI.
/proc/bus/pci/*	Renseignements sur les périphériques PCI.
/proc/bus/usb/*	Renseignements sur les périphériques USB.
/proc/bus/scsi/*	Renseignements sur les périphériques SCSI.
/proc/bus/isapnp/*	Renseignements sur les périphériques ISA PnP.

Les commandes

Remarque : Les commandes présentées sont une abstraction des informations de /proc.	
`lspci`	Liste les contrôleurs PCI.
`lsusb`	Liste les périphériques USB.
`lsscsi`	Liste les périphériques SCSI.
`scsiinfo`	Affiche des informations concernant les périphériques SCSI.
`lsdev`	Liste les périphériques avec l'indication des ports, IRQ et DMA utilisés.
`dmidecode`	Affiche le contenu des tables DMI (SMBIOS). Elles contiennent des informations matérielles provenant de différent chipset comme le BIOS, le CPU, ...
`hdparm`	Liste des informations sur les disques IDE (DMA utilisée ou non, ...).
`pnpdump`	Visualise les informations des cartes ISA PnP.
`dmesg`	Affiche les messages provenant du noyau.
`x86info`	Affiche des informations sur le processeur.

Pour en savoir plus

Les pages de manuel

lspci(8), lsusb(8), lsdev8), lscsi(8), lsscsi(8), scsiinfo(8), hdparm(8), pnpdump(8), dmesg(8), x86info(1)

Howto

Plug-and-Play-HOWTO – lire les chapitres concernant le matériel.

ACPI-HOWTO

Battery-Powered

Internet

Computer Hardware - Learn all the basics
http://www.infosyssec.net/infosyssec/comphard.htm

Computer Hardware
http://en.wikipedia.org/wiki/Computer_hardware

Les manuels en ligne des cartes mères : chaque constructeur d'ordinateur ou de carte mère, propose une documentation de ses cartes mère (motherboard). On y trouve l'architecture logique, physique, les données numériques (vitesse de transfert, largeur de bus, …), les CPU et les mémoires compatibles, les connecteurs, les chipsets présents, l'utilisation du bios et comment effectuer sa mise à jour, … Voici quelques adresses pour l'exemple :
http://www.wikihow.com/Find-Documentation-for-Your-Motherboard-Online
http://www.x86.org/intel.doc/intelmotherboards.htm
http://www.intel.com/support/motherboards/

SCSI
http://en.wikipedia.org/wiki/Small_Computer_System_Interface
http://www.scsita.org

PCI, PCI-X, PCIe
http://fr.wikipedia.org/wiki/Peripheral_Component_Interconnect

La gestion des périphériques sous Linux

La théorie

L'approche Unix de la gestion des périphériques

Sur un système Unix, il y un logiciel qui gère un contrôleur et via ce dernier un périphérique, c'est le pilote de périphérique (device driver). Ce logiciel système est un module du noyau. Les applications dialoguent avec un pilote grâce à un fichier périphérique (/dev/…).

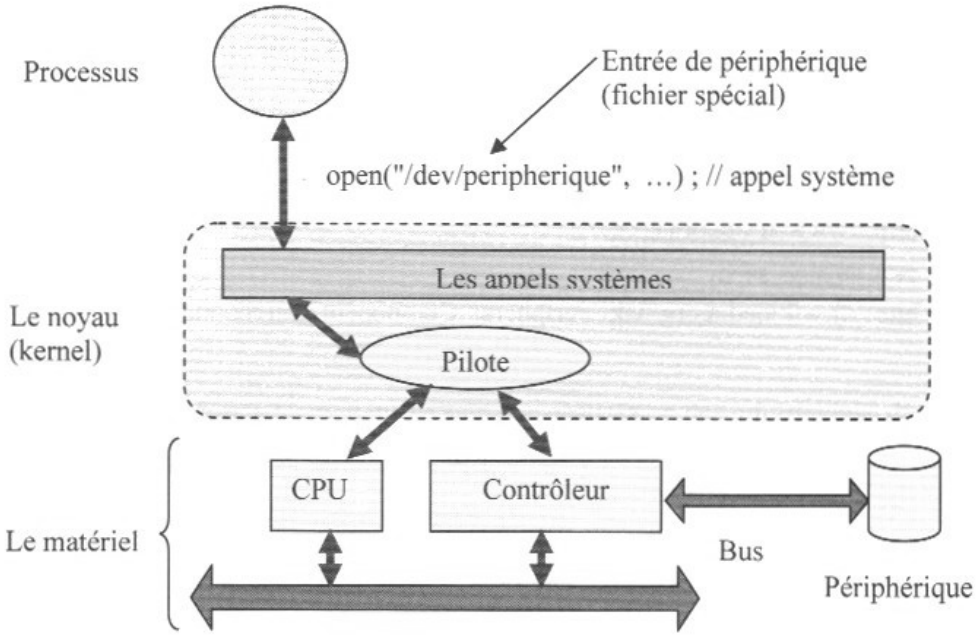


Fig. L'approche Unix de la gestion des périphériques.

Les particularités de Linux

Globalement Linux reprend l'approche Unix pour la majorité des périphériques. La principale exception est la gestion du réseau. Dans ce cas, les applications référent une interface réseau non par un fichier spécial mais par un nom comme eth0. Autre particularité, l'interface graphique X-Window. Le gestionnaire de la carte graphique n'est pas un pilote du noyau, mais une bibliothèque du serveur X.

Linux est un système en Open Source. Celui qui compile le noyau décide l'ajout ou le retrait d'un pilote, ainsi que son mode d'inclusion : statique ou dynamique. Dans le cas où il est dynamique, le pilote peut être chargé ou déchargé sur demande.

Contrairement à beaucoup d'autres systèmes, un module est spécifique d'une version bien précise du noyau. En conséquence, l'ajout d'un pilote exotique nécessite sa recompilation au sein de l'environnement de développement du noyau.

Les pilotes chargés dynamiquement

La plupart des pilotes de périphériques sous Linux sont chargés dynamiquement par le noyau. Le module KML (Kernel Module Loader) du noyau déclenche le chargement du module suite à un événement. KLM peut gérer également le retrait du pilote.

Les types de pilotes

Un pilote de périphérique Linux appartient à un des trois types suivants :

Pilote de type caractère

C'est un pilote accessible via un fichier spécial de type caractère. L'accès aux données se fait octet par octet ou suite d'octets par suite d'octets. La majorité des périphériques sont gérés par ce type de pilote.

Pilote de type bloc

C'est un pilote accessible via un fichier spécial de type bloc. L'accès aux données se fait bloc par bloc de manière aléatoire (on accède à n'importe quel bloc). Seules, les disques appartiennent à cette catégorie.

Pilote d'interface réseau

C'est un pilote spécifique des cartes réseaux.

Les caractéristiques associées à un fichier périphérique

Un fichier spécial (ou périphérique) mémorise trois informations :

- Le type de pilote : caractère ou bloc.
- L'adresse du pilote : dans le noyau Linux, il existe une table pour chaque type et dans chaque table, un pilote est repéré par un numéro : le majeur. La combinaison type et majeur identifie un pilote.
- Le mineur : ce nombre est transmis au pilote lors de l'ouverture du périphérique. C'est donc lui qui l'interprète. Il peut signifier aussi bien un numéro de périphérique qu'une option de gestion, par exemple, rembobiner ou non un lecteur de cartouche. Il peut aussi signifier toutes ces choses à la fois. En conséquence, bien qu'il n'y ait pas beaucoup de pilotes, il y a souvent beaucoup de fichiers spéciaux correspondant aux différentes combinaisons des différentes options ainsi qu'aux différentes instances de périphériques.

Udev

Dans la version 2.4 du noyau, les entrées de périphériques étaient créées par la commande MAKEDEV. Dans la version 2.6 du noyau, elles sont créées par Udev.

Grâce au système Udev, les entrées de périphériques sont créées dynamiquement. Il est possible également de leur attribuer des noms indépendants de l'ordre dans lequel ils sont branchés (c'est le problème avec les disques SCSI).

Vision avancée de la gestion de périphérique

Les applications accèdent aux périphériques via un fichier spécial, par exemple /dev/tty1. Pour accéder au périphérique, le développeur dispose des fonctions de gestion de fichier ordinaires : open(), read(), write(), close() ainsi qu'un appel spécifique pour les périphériques : ioctl(). L'accès aux interfaces réseau se fait par l'intermédiaire des fonctions sockets.

Le savoir concret

Les commandes

`dmesg`	Cette commande affiche les messages récents du noyau (ils sont stockés dans un tampon circulaire). Chaque message est précédé du pilote qui la généré.
`lsmod`	Liste les modules dynamiques actuellement chargés. Plusieurs de ces modules sont des pilotes de périphériques.
`modinfo`	Affiche des informations provenant d'un module.

`ls -l`	Liste les caractéristiques d'un fichier, notamment d'un fichier périphérique.
`MAKEDEV`	Crée les entrées de périphérique de manière statique (noyau 2.4).
`mknod`	La commande primitive qui crée une entrée de périphérique.
`usbmodules`	Liste les modules disponibles pour un périphérique USB.
`hdparm`	Visualise ou modifie les paramètres d'un disque IDE.
`blockdev`	Provoque l'appel système ioctl() pour un périphérique bloc.

Focus : Les caractéristiques d'un fichier spécial

```
# ls -l /dev/hda3 /dev/mice
brw-rw----  1 root disk  3,  3 Feb  6 18:28 /dev/hda3
crw-------  1 root root 13, 63 Feb  6 19:28 /dev/mice
```

Le premier caractère indique le mode :

- b Bloc
- c Caractère

La taille du fichier est remplacé par le couple : Majeur, Mineur. Dans l'exemple, hda3 a le majeur 3 et le mineur 3 et mice a le majeur 12 et le mineur 63.

Les fichiers

/proc & /sys	Arborescences de fichiers gérés par le noyau.
/proc/devices	La liste des majeurs des périphériques bloc et caractère.
/proc/sys/kernel/modprobe	La commande activée automatiquement par le noyau pour charger un pilote suite à un événement (le branchement du périphérique par exemple).

Remarque : Le système Syslog permet d'enregistrer les messages en provenance du noyau via la facilité `kern`. Cette facilité est associée au fichier /var/log/messages dans le système RedHat.

Pour en savoir plus

Les pages de manuel

ls(1), MAKEDEV(8), dmesg(8), mknod(8), modinfo(8), modprobe(8), hdparm(8), blockdev(8)

Les pages de manuel de la section 4 décrivent les fichiers spéciaux permettant d'accéder aux pilotes et grâce à eux aux périphériques. Ils sont stockés dans le répertoire /usr/share/man/man4. Voici quelques exemples :

Le lecteur de disquette : fd(4), les disques IDE : hd(4), les disques SCSI : sd(4), les lecteurs de cartouches SCSI : st(4), Pilotes de carte graphique : ati(4), i810(4), … L'interface parallèle aux imprimantes : lp(4), …

Ajouter un périphérique

La théorie

Gestion des périphériques sous Linux en résumé

Pour accéder à un périphérique sous Linux, il faut :

- Un pilote de périphérique intégré de manière statique ou dynamique (module).
- La configuration de ce pilote.
- Un fichier spécial (pas pour le réseau) crée de manière statique ou dynamique par Udev.

Ajouter un périphérique sous Linux

On vient de le voir, l'aspect essentiel de l'ajout d'un périphérique passe pas l'ajout d'un pilote et dans le cas où on dispose de ce pilote, cela passe par sa configuration.

Le problème reformulé devient donc : le pilote existe-t-il pour Linux ? Le choix d'un PC haut de gamme comme un Proliant (HP), un xSeries (IBM) ou un PowerEdge (Dell) résout souvent le problème lorsque l'on veut utiliser Linux comme serveur. En effet, ces matériels sont supportés nativement par les constructeurs. Pour un poste de travail, c'est plus complexe, il faut souvent faire confiance aux « listes de compatibilité » des différentes distributions ou provenant d'autres horizons.

Normalement l'ajout d'un périphérique sous Linux est automatique. Des logiciels, propres aux distributions, permettent la découverte des nouveaux périphériques ainsi que leur configuration.

Le savoir concret

Les commandes

`lsmod`, `modprobe`, `modinfo`, ...	Les commandes gérant les modules dynamique du noyau.
`kudzu`	Détecte et configure automatiquement les nouveaux périphériques (RedHat)
`discover`	Idem pour Debian.
`hwinfo`	Idem pour SUSE.

Focus : Kudzu (RedHat)

Kudzu est le système RedHat de détection et de configuration automatique de périphériques. Activé au démarrage, il recherche le matériel présent et vérifie son existence dans sa base de données, le fichier /etc/sysconfig/hwconf. Il en déduit le matériel ajouté ou retiré. Dans ce cas, il demande à l'exploitant de configurer ou de retirer le matériel et il met à jour sa base de données. Kudzu est également capable de configurer automatiquement la base de données de `modprobe` (/etc/modprobe.conf).

Les fichiers

/etc/modules.conf	Configuration des modules (noyau 2.4)
/etc/modprobe.conf	Configuration des modules (noyau 2.6)

Focus : un exemple de fichier modprobe.conf

```
alias eth0 8139too
alias snd-card-0 snd-via82xx
options snd-card-0 index=0
```

Les deux ordres principaux que l'on trouve dans le fichier modprobe.conf sont alias et option. Alias permet de donner des noms génériques à un pilote et option permet de fournir aux pilotes des options de configuration.

Pour en savoir plus

Les pages de manuel

kudzu(1), discover(8), hwinfo(8)

Howto

Hardware-HOWTO – liste le matériel compatible Linux

PCI-HOWTO – Informations sur les cartes PCI fonctionnant sous Linux

PCMCIA-HOWTO

Scanner-HOWTO

Sound-HOWTO

Sound-Playing-HOWTO

Soundbalster-AWE

DVD-HOWTO

DVD-Playback-HOWTO

FAX-Server

Modem-HOWTO

Webcam-HOWTO

Wireless-HOWTO

ZIP drive-HOWTO

Winmodems-and-Linux-HOWTO

Internet

Linux Hardware Compatibility Lists & Linux Drivers
(Centralise les listes de compatibilité : Redhat, Suse, Debian, ...)
http://www.linux-drivers.org/

RedHat HCL (Hardware Compatibility List)
http://hardware.redhat.com/hcl/

SuSE HCL
http://en.opensuse.org/Hardware

Debian GNU/Linux device driver check page (vous indiquez votre matériel)
http://kmuto.jp/debian/hcl/

Lea-linux - Installer et configurer son matériel (modems, son, …)
http://www.lea-linux.org/cached/index/Hardware-index.html

Le FS sysfs

La théorie

Le FS sysfs, habituellement monté sur /sys, est une interface au noyau en mémoire vive, comme /proc, mais ne traitant que des périphériques. Il apparaît avec les noyaux 2.6. Ses objectifs sont les suivants :

- Publier les périphériques découverts au démarrage.
- Supporter le branchement à chaud de périphériques (HotPlug).
- Supporter la gestion de l'économie d'énergie.

Le savoir concret

Les principaux répertoires

/sys/devices/	L'arborescence (au sens matériel) des périphériques. Affiche donc l'arborescence des bus qui connectent les périphériques.
/sys/devices/platform/	Liste les périphériques spécifiques du PC (clavier, souris, lecteur de disquette, …).
/sys/devices/system	Liste le matériel non périphérique (CPU, APIC, timer, …).
/sys/bus/	Les périphériques, mais regroupés par type de bus (ide, pci, scsi,scsi_host usb, usb_host, …). Les périphériques « host » (usb_host, scsi_host) correspondent à des cartes contrôleur.
/sys/class/	Les périphériques, mais regroupés par classe fonctionnelle (net, printer, input, graphics, sound, …).
/sys/block/	Les périphériques blocs (disques, ram, …).
/sys/firmware/	Interface avec le firmware (BIOS), comme l'ACPI.
/sys/modules/	Contient un répertoire pour chaque module du noyau actuellement chargé.
/sys/power/	Traite de l'alimentation. Le fichier state liste les états supportés.

Remarque : La plupart des fichiers ou des répertoires sont en fait des liens symboliques pointant vers des fichiers de l'arborescence /sys/devices/.

La commande systool

La commande `systool` du package `sysfsutils` présente les éléments de l'arborescence /sys de manière conviviale. Sans option, la commande liste les différents Bus, les différentes classes et les répertoires de premier niveau de /sys/devices. Voici les principales options :

-r device	Affiche l'arborescence matérielle. Si l'on donne le bus racine (pci0000:00) en paramètre, la commande affiche l'arborescence complète.
-b bus	Liste les périphériques reliés au système par un bus spécifique (pci, scsi, usb, ide, …).
-c classe	Liste les périphériques appartenant à une classe (net, printer, …).
-a	Affiche les attributs.
-v	Affiche tous les attributs et leur valeur.

-A att	Affiche la valeur d'un attribut.
-p	Visualise le chemin /sys de la ressource.
-D	Affiche le pilote.

Focus : le montage de sysfs

```
# mount -t sysfss  sysfs /sys
```

Pour en savoir plus

Les pages de manuel

systool(1)

Fichier

filesystems/sysfs.txt	Fichier de la documentation du noyau qui décrit sysFS.

Internet

Linux diagnostics tools
http://linux-diag.sourceforge.net/Sysfsutils.html

Le sysfs Filesystem
http://www.kernel.org/pub/linux/kernel/people/mochel/doc/papers/ols-2005/mochel.pdf

Livre

/proc & /sys, par Oliver Daudel, chez O'Reilly.

La technologie Udev

La théorie

Udev est une technologie qui permet la gestion dynamique de l'arborescence /dev. Elle repose sur l'utilisation de sysFS et est donc liée au noyau 2.6. Elle permet notamment la création des fichiers spéciaux (/dev/*) des périphériques connectés à chaud (hotplug) USB ou SCSI. Udev peut aussi résoudre élégamment le problème des noms persistants de périphérique. C'est à dire qu'un même périphérique peut avoir toujours le même nom indépendamment de son ordre de branchement.

Remarque : Normalement, un disque SCSI (ou USB géré comme un disque SCSI) a son nom qui dépend de l'ordre de branchement. Le 1er disque SCSI branché est appelé /dev/sda, le deuxième /dev/sdb, etc.

Le savoir concret

Les commandes

`udevd`	Démon qui sérialise les événements hotplug. Pour chaque événement, le démon active une instance de la commande `udev`. Tous les événements du même type sont bloqués tant que l'instance d'`udev` associée ne soit terminée.
`udev`	Ajoute ou retire un périphérique (crée ou supprime son fichier spécial). La commande se base sur des fichiers de configuration et sur des scripts.
`hotplug`	Commande activée directement par le noyau dés qu'un événement hotplug survient. La commande active la commande `udevsend`.
`udevsend`	Cette commande envoie l'événement reçu au démon `udevd`. Si ce dernier n'est pas présent, la commande l'active.
`udevstart`	Rempli initialement le répertoire /dev. Cette commande est typiquement activée automatiquement au démarrage.
`udevinfo`	Affiche des informations provenant d'`udev`.
`udevtest`	Simule une activation de la commande `udev`. Utilisé pour tester des règles de configuration.

Les fichiers

/dev	FS en mémoire qui contient les entrées de périphériques.
/etc/udev/udev.conf	Le fichier de configuration principal d'Udev.
/etc/udev/rules.d/	Le répertoire contenant les règles de configuration d'Udev.
/proc/sys/kernel/hotplug	L'exécutable que le noyau active quand un événement hotplug survient (par exemple le branchement ou le retrait d'une clé USB).
/etc/dev.d/$(DEVNAME)/*.dev	Programmes exécutés après que le fichier spécial soit crée ou supprimé

La création de règles Udev

La prise en compte des règles

Les fichiers de règles sont dans le répertoire /etc/udev/rules.d/. Ils doivent avoir l'extension .rules. Le fichier de règles par défaut se nomme 50-udev.rules. Les fichiers sont pris en compte par `Udev` dans l'ordre ASCII. Il est conseillé de créer ses propres règles dans le fichier 10-local.rules, ainsi, elles seront prioritaires.

La syntaxe d'une règle

```
clé [, ...] assignation [, ...]
```

Si le ou les clés sont vrais, le ou les assignations sont réalisées.

Les principales clés

KERNEL	Prend comme critère un nom noyau de périphérique. On peut utiliser les jockers. Par exemple : fd0, sd*, hd[a-z], mouse, lp[0-9], …
BUS	Un nom de bus : scsi, usb,
SYSFS{mot}	Un attribut du périphérique tel que donné par le FS sysfs, par exemple le fabricant (vendor), le numéro de série (serial), le modèle (model), …
ENV{var}	Une variable d'environnement. La variable ACTION notamment indique si le périphérique est connecté ou déconnecté (valeurs « add » ou « remove ») ; La variable DEVPATH indique le chemin du périphérique dans /sys (mais sans /sys).
PROGRAM	Un programme externe dont la clé est valide si la valeur de sortie est zéro. La chaîne retournée peut être comparée à la clé RESULT.

Les assignations

NAME	Le nom que l'on attribut au périphérique (nom réseau ou nom dans /dev).
SYMLINK	Un lien symbolique que l'on ajoute en plus du nom officiel du périphérique.
OWNER, GROUP, MODE	Le propriétaire, le groupe et les droits du fichier périphérique.

Les macros accessibles dans NAME, SYMLINK et PROGRAM

%n	Le n° du périphérique, par exemple 3 pour sda3.
%k	Le nom noyau du périphérique, par exemple hda.

Exemples

```
#  Crée le fichier spécial /dev/sauvegarde à la place de /dev/hdb
KERNEL="hdb",  NAME="sauvegarde"
# Crée le fichier spécial /dev/ide et le lien /dev/un_ide
KERNEL="hdb",  NAME="ide", SYMLINK="un_ide"
# Crée des fichiers spéciaux associés au numéro du lecteur de
# disquette, par ex: /dev/disquette/0 et le lien /dev/fd0
KERNEL="fd[0-9]",  NAME="disquette/%n", SYMLINK="%k"
```

Pour en savoir plus

Les pages de manuel

udev(8), udevinfo(8), hotplug(8), udevd(8), udevsend(8), udevtest(8), udevinfo(8)

Internet

Udev
http://www.kernel.org/pub/linux/utils/kernel/hotplug/udev.html

La technologie lm_sensors

La théorie

La technologie lm_sensors est un ensemble d'outils et de pilotes qui donnent accès aux détecteurs de la carte mère pour surveiller la température du CPU et de la carte mère. Eventuellement, on peut également surveiller le voltage et la ventilation.

Le noyau 2.6 supporte la technologie lm_sensors, mais il faut évidemment que la carte mère la supporte également, le plus souvent grâce aux bus I2c et SMBus.

Le savoir concret

Les commandes

`sensors-detect`	Script qui détermine les pilotes du noyau associés aux détecteurs de la carte mère.
`sensors -s`	Etalonne les détecteurs en se basant sur le fichier de configuration. Les valeurs par défaut de ce fichier suffisent généralement.
`sensors`	Affiche les données provenant des détecteurs.
`xsensors`, `ksensors`	Idem, mais en mode graphique.
/etc/init.d/lm_sensors	Script RC permettant le chargement des pilotes et l'activation de la commande `sensors -s`.

Les fichiers

/etc/sensors.conf	Fichier de configuration permettant l'étalonnage des valeurs provenant des détecteurs. Ce fichier est lu par la commande `sensors -s`.

Pour en savoir plus

Les pages de manuel

sensors(1), sensors.conf(5)

Internet

Lm_sensors – le site officiel
http://www.lm-sensors.org
http://www2.lm-sensors.nu/~lm78

Hardware-hard plus-sensors - Lea Linux
http://www.lea-linux.org/cached/index/Hardware-hard_plus-sensors.html

La technologie SMART

La théorie

La technologie SMART (Self-Monitoring, Analysis and Reporting Technology) permet de surveiller la fiabilité des disques durs et de prédire leurs pannes. Cette technologie est maintenant incluse dans la plupart des disques ATA, IDE et SCSI.

Parmi les informations recueillies, on trouve : la température du disque, le nombre d'heures de fonctionnement, le nombre d'arrêt/démarrage, le taux d'erreurs, …

Le savoir concret

Les commandes

`smartd`	Démon qui active la surveillance des disques, recueille périodiquement leur erreurs et leurs modifications et les transmet à Syslog.
/etc/init.d/smartd	Script de démarrage (RC) qui active le démon `smartd`.
`smartctl`	Gère SMART (active/désactive SMART pour un disque, déclenche des tests, récupère les compte rendues, affiche des informations, …).

Les principales options de smartctl

-i	Affiche des informations générales concernant le disque. Indique notamment si il supporte SMART et si oui si SMART est activé pour ce disque.
-s on	Active SMART pour ce périphérique.
-s off	Désactive SMART pour ce périphérique.
-c	Liste les tests supportés par le périphérique.
-t Type	Active un test. Type a pour valeur : short, long ou conveyance.
-l Type	Visualise un journal. Type a pour valeur : error, selftest, …
-a	Affiche toutes les informations concernant le disque.
-A	Affiche uniquement les attributs spécifiques du fabricant.

Remarque : Les commandes –s on/off se trouvent normalement dans un script de démarrage.

Les fichiers

/etc/smartd.conf Le fichier de configuration du démon smartd.

Pour en savoir plus

Les pages de manuel

smartd(8), smartctl(8), smartd.conf(5)

Internet

Smartmontools – home page
http://smartmontools.sourceforge.net

Lea-linux – Hardware
http://www.lea-linux.org/cached/index/Hardware-index.htm

ATELIERS



Tâche 1 : L'architecture de l'ordinateur 10mn
Tâche 2 : Linux et les périphériques 10mn
Tâche 3 : La configuration des périphériques 5mn
Tâche 4 : Le FS sysfs 10mn
Tâche 5 : Udev 15mn
Tâche 6 : lm_sensors 5mn
Tâche 7 : La technologie SMART 10mn

Tâche 1 : L'architecture de l'ordinateur

1. Afficher quelques informations provenant de /proc.

```
[root@linux1 ~]# cat /proc/cpuinfo
processor       : 0
vendor_id       : AuthenticAMD
cpu family      : 15
model           : 44
model name      : AMD Sempron(tm) Processor 3100+
stepping        : 2
cpu MHz         : 1800.895
cache size      : 256 KB
fdiv_bug        : no
hlt_bug         : no
f00f_bug        : no
coma_bug        : no
fpu             : yes
fpu_exception   : yes
cpuid level     : 1
wp              : yes
flags           : fpu vme de pse tsc msr pae mce cx8 apic sep mtrr pge mca cmov
pat pse36 clflush mmx fxsr sse sse2 syscall nx mmxext fxsr_opt lm 3dnowext 3dnow
up pni lahf_lm ts fid vid ttp tm stc
bogomips        : 3606.30

[root@linux1 ~]# cat /proc/interrupts
           CPU0
  0:   79921551    IO-APIC-edge  timer
  1:         69    IO-APIC-edge  i8042
  7:          3    IO-APIC-edge  parport0
  8:          1    IO-APIC-edge  rtc
  9:          0   IO-APIC-level  acpi
 12:        139    IO-APIC-edge  i8042
 14:      82440    IO-APIC-edge  ide0
 15:        211    IO-APIC-edge  ide1
177:      76650   IO-APIC-level  eth0
185:          0   IO-APIC-level  VIA8237
193:       1529   IO-APIC-level  ehci_hcd:usb1, uhci_hcd:usb2, uhci_hcd:usb3,
uhci_hcd:usb4, uhci_hcd:usb5
```

```
NMI:          0
LOC:   79926679
ERR:          0
MIS:          0
[root@linux1 ~]# cat /proc/ioports
0000-001f : dma1
0020-0021 : pic1
0040-0043 : timer0
0050-0053 : timer1
0060-006f : keyboard
0070-0077 : rtc
0080-008f : dma page reg
00a0-00a1 : pic2
00c0-00df : dma2
00f0-00ff : fpu
0170-0177 : ide1
01f0-01f7 : ide0
0376-0376 : ide1
0378-037a : parport0
03c0-03df : vga+
03f6-03f6 : ide0
03f8-03ff : serial
4000-407f : motherboard
  4000-4003 : PM1a_EVT_BLK
  4004-4005 : PM1a_CNT_BLK
  4008-400b : PM_TMR
  4020-4023 : GPE0_BLK
5000-500f : motherboard
  5000-500f : pnp 00:02
    5000-5007 : vt596_smbus
d000-d0ff : 0000:00:0a.0
  d000-d0ff : 8139too
d400-d40f : 0000:00:0f.0
  d400-d407 : ide0
  d408-d40f : ide1
d800-d81f : 0000:00:10.0
  d800-d81f : uhci_hcd
dc00-dc1f : 0000:00:10.1
  dc00-dc1f : uhci_hcd
e000-e01f : 0000:00:10.2
  e000-e01f : uhci_hcd
e400-e41f : 0000:00:10.3
  e400-e41f : uhci_hcd
e800-e8ff : 0000:00:11.5
  e800-e8ff : VIA8237
[root@linux1 ~]# cat /proc/dma
 4: cascade
[root@linux1 ~]# cat /proc/iomem
00000000-0009f7ff : System RAM
0009f800-0009ffff : reserved
000a0000-000bffff : Video RAM area
000c0000-000c81ff : Video ROM
000cc000-000d13ff : Adapter ROM
000f0000-000fffff : System ROM
```

```
00100000-1beeffff : System RAM
  00100000-002da1ff : Kernel code
  002da200-00399d6f : Kernel data
1bef0000-1bef2fff : ACPI Non-volatile Storage
1bef3000-1beffffff : ACPI Tables
20000000-2000ffff : 0000:00:0a.0
e0000000-e7ffffff : 0000:00:00.0
  e0000000-e7ffffff : aperture
e8000000-ebffffff : PCI Bus #01
  e8000000-ebffffff : 0000:01:00.0
ec000000-edffffff : PCI Bus #01
  ec000000-ecffffff : 0000:01:00.0
  ed000000-ed00ffff : 0000:01:00.0
ee010000-ee0100ff : 0000:00:0a.0
  ee010000-ee0100ff : 8139too
ee011000-ee0110ff : 0000:00:10.4
  ee011000-ee0110ff : ehci_hcd
fec00000-fec00fff : reserved
fee00000-fee00fff : reserved
ffff0000-ffffffff : reserved
[root@linux1 ~]# ls /proc/bus/pci/
00  01  devices
[root@linux1 ~]# ls /proc/bus/usb
001  002  003  004  005  devices
[root@linux1 ~]# ls /proc/acpi
ac_adapter  button              event  info            thermal_zone
alarm       dsdt                fadt   power_resource  video
battery     embedded_controller fan    processor       wakeup
[root@linux1 ~]#
```

Remarque : Sur un portable, on peut connaître le taux de charge de la batterie grâce au fichier /proc/acpi/battery/*/state.

2. Afficher l'architecture PCI.

```
[root@linux1 ~]# lspci -t
-[0000:00]-+-00.0
           +-00.1
           +-00.2
           +-00.3
           +-00.4
           +-00.7
           +-01.0-[0000:01]----00.0
           +-0a.0
           +-0f.0
           +-10.0
           +-10.1
           +-10.2
           +-10.3
           +-10.4
           +-11.0
           +-11.5
           +-18.0
           +-18.1
           +-18.2
           \-18.3
```

3. Ai-je une carte Ethernet ? Afficher la liste des contrôleurs et des périphériques PCI.

```
[root@linux1 ~]# lspci |grep -i ethernet
00:0a.0 Ethernet controller: Realtek Semiconductor Co., Ltd. RTL-
8139/8139C/8139C+ (rev 10)
[root@linux1 ~]# lspci
00:00.0 Host bridge: VIA Technologies, Inc. K8M800 Host Bridge
00:00.1 Host bridge: VIA Technologies, Inc. K8M800 Host Bridge
00:00.2 Host bridge: VIA Technologies, Inc. K8M800 Host Bridge
00:00.3 Host bridge: VIA Technologies, Inc. K8M800 Host Bridge
00:00.4 Host bridge: VIA Technologies, Inc. K8M800 Host Bridge
00:00.7 Host bridge: VIA Technologies, Inc. K8M800 Host Bridge
00:01.0 PCI bridge: VIA Technologies, Inc. VT8237 PCI bridge [K8T800/K8T890
South]
00:0a.0 Ethernet controller: Realtek Semiconductor Co., Ltd. RTL-
8139/8139C/8139C+ (rev 10)
00:0f.0 IDE interface: VIA Technologies, Inc.
VT82C586A/B/VT82C686/A/B/VT823x/A/C PIPC Bus Master IDE (rev 06)
00:10.0 USB Controller: VIA Technologies, Inc. VT82xxxxx UHCI USB 1.1 Controller
(rev 81)
00:10.1 USB Controller: VIA Technologies, Inc. VT82xxxxx UHCI USB 1.1 Controller
(rev 81)
00:10.2 USB Controller: VIA Technologies, Inc. VT82xxxxx UHCI USB 1.1 Controller
(rev 81)
00:10.3 USB Controller: VIA Technologies, Inc. VT82xxxxx UHCI USB 1.1 Controller
(rev 81)
00:10.4 USB Controller: VIA Technologies, Inc. USB 2.0 (rev 86)
00:11.0 ISA bridge: VIA Technologies, Inc. VT8237 ISA bridge
[KT600/K8T800/K8T890 South]
00:11.5 Multimedia audio controller: VIA Technologies, Inc. VT8233/A/8235/8237
AC97 Audio Controller (rev 60)
00:18.0 Host bridge: Advanced Micro Devices [AMD] K8 [Athlon64/Opteron]
HyperTransport Technology Configuration
00:18.1 Host bridge: Advanced Micro Devices [AMD] K8 [Athlon64/Opteron] Address
Map
00:18.2 Host bridge: Advanced Micro Devices [AMD] K8 [Athlon64/Opteron] DRAM
Controller
00:18.3 Host bridge: Advanced Micro Devices [AMD] K8 [Athlon64/Opteron]
Miscellaneous Control
01:00.0 VGA compatible controller: VIA Technologies, Inc. S3 Unichrome Pro VGA
Adapter (rev 01)
```

Remarque : La commande `lspci` est la commande la plus importante pour connaitre l'architecture matérielle d'un PC. Le bus PCI formant l'ossature de ses entrées/sorties.

4. Afficher plus d'informations concernant les périphériques PCI.

```
[root@linux1 ~]# lspci -v |tail

01:00.0 VGA compatible controller: VIA Technologies, Inc. S3 Unichrome Pro VGA
Adapter (rev 01) (prog-if 00 [VGA])
        Subsystem: FIRST INTERNATIONAL Computer Inc: Unknown device 6001
        Flags: bus master, 66Mhz, medium devsel, latency 32, IRQ 10
        Memory at e8000000 (32-bit, prefetchable) [size=64M]
        Memory at ec000000 (32-bit, non-prefetchable) [size=16M]
        [virtual] Expansion ROM at ed000000 [disabled] [size=64K]
        Capabilities: [60] Power Management version 2
        Capabilities: [70] AGP version 3.0
```

5. Lister les périphériques USB.

```
[root@linux1 ~]# lsusb
Bus 005 Device 002: ID 058f:9360 Alcor Micro Corp.
Bus 005 Device 001: ID 0000:0000
Bus 004 Device 001: ID 0000:0000
Bus 003 Device 001: ID 0000:0000
Bus 002 Device 001: ID 0000:0000
Bus 001 Device 001: ID 0000:0000
```

6. Lister l'arborescence des périphériques USB.

```
[root@linux1 ~]# lsusb -t
Bus#  5
`-Dev#   1 Vendor 0x0000 Product 0x0000
  `-Dev#   2 Vendor 0x058f Product 0x9360
Bus#  4
`-Dev#   1 Vendor 0x0000 Product 0x0000
Bus#  3
`-Dev#   1 Vendor 0x0000 Product 0x0000
Bus#  2
`-Dev#   1 Vendor 0x0000 Product 0x0000
Bus#  1
`-Dev#   1 Vendor 0x0000 Product 0x0000
```

7. Lister les périphériques USB en affichant beaucoup d'informations.

```
[root@linux1 ~]# lsusb -v |tail -15
        bInterval               12
Hub Descriptor:
  bLength              11
  bDescriptorType      41
  nNbrPorts             8
  wHubCharacteristic 0x000a
    No power switching (usb 1.0)
    Per-port overcurrent protection
    TT think time 8 FS bits
  bPwrOn2PwrGood       10 * 2 milli seconds
  bHubContrCurrent      0 milli Ampere
  DeviceRemovable    0x00 0x00
  PortPwrCtrlMask    0x00  0x00
  Language IDs: (length=4)
     0409 English(US)
```

8. Afficher les informations concernant les périphériques SCSI.

```
[root@linux1 ~]# wget
'ftp://141.30.228.4/pub/mirrors/redhat/contrib/libc6/i386/scsiinfo-1.7-
4.i386.rpm'
…
11:40:01 (182.03 KB/s) - `scsiinfo-1.7-4.i386.rpm.1' saved [57719]
[root@linux1 ~]# rpm -Uvh scsiinfo-1.7-4.i386.rpm
…
[root@linux1 ~]# scsiinfo
Error: scsiinfo - too few arguments
Usage: scsiinfo [-options] [device]
        Allowed options are:
        -c    Display information from Caching Page.
        -C    Display information from Control Mode Page.
```

```
        -d    Display defect lists.
        -Farg Format of the defect list:
                -Flogical  - logical blocks
                -Fphysical - physical blocks
                -Findex    - defect bytes from index
        -e    Display information from Error Recovery page.
        -f    Display information from Format Device Page.
        -g    Display information from Rigid Disk Drive Geometry Page.
…
[root@linux1 ~]# scsiinfo -l
/dev/sde
```

Remarque : On peut brancher une clé pour voir des périphériques SCSI.

9. Afficher le contenu des tables DMI (SMBIOS).

```
[root@linux1 ~]# dmidecode | more
# dmidecode 2.2
SMBIOS 2.3 present.
35 structurès occupying 1065 bytes.
Table at 0x000F0000.
Handle 0x0000
        DMI type 0, 20 bytes.
        BIOS Information
                Vendor: Phoenix Technologies, LTD
                Version: 6.00 PG
                Release Date: 10/28/2005
                Address: 0xE0000
                Runtime Size: 128 kB
                ROM Size: 512 kB
                Characteristics:
                 ISA is supported
                 PCI is supported
                 PNP is supported
                 APM is supported
                 BIOS is upgradeable
…
                 Function key-initiated network boot is supported
Handle 0x0001
        DMI type 1, 25 bytes.
        System Information
                Manufacturer: Packard Bell NEC
                Product Name: ISTART 2055
                Version: SYS-xxxxxx
                Serial Number: S100470010223
                UUID: 80469465-803F-D811-8000-4E45435F4349
                Wake-up Type: Power Switch
…
Handle 0x0004
        DMI type 4, 35 bytes.
        Processor Information
                Socket Designation: Socket 940
                Type: Central Processor
                Family: Athlon 64
                Manufacturer: AMD
                ID: C2 0F 02 00 FF FB 8B 07
```

```
                Version: AMD Sempron(tm) Processor 3100+
                Voltage: 1.4 V
                External Clock: 200 MHz
                Max Speed: 3400 MHz
                Current Speed: 1800 MHz
…
                Part Number:
Handle 0x0005
        DMI type 5, 20 bytes.
        Memory Controller Information
                Error Detecting Method: 64-bit ECC
                Error Correcting Capabilities:
                 None
                Supported Interleave: One-way Interleave
                Current Interleave: One-way Interleave
                Maximum Memory Module Size: 1024 MB
                Maximum Total Memory Size: 2048 MB
                Supported Speeds:
                 70 ns
                 60 ns
                 50 ns
                Supported Memory Types:
                 Standard
                 DIMM
                Memory Module Voltage: 2.9 V
                Associated Memory Slots: 2
                 0x0006
                 0x0007
                Enabled Error Correcting Capabilities: None
…
Handle 0x0017
        DMI type 9, 13 bytes.
        System Slot Information
                Designation: PCI1
                Type: 32-bit PCI
                Current Usage: Available
                Length: Long
                ID: 2
                Characteristics:
                 5.0 V is provided
                 PME signal is supported
…
```

10. Afficher les messages en provenance du noyau (on vient de brancher une clé USB).

```
[root@linux1 ~]# dmesg |tail -20
ide0: reset: success
process `sysctl' is using deprecated sysctl (syscall)
net.ipv6.neigh.eth0.base_reachable_time; Use
net.ipv6.neigh.eth0.base_reachable_time_ms instead.
usb 1-6: new high speed USB device using ehci_hcd and address 3
usb 1-6: configuration #1 chosen from 1 choice
scsi1 : SCSI emulation for USB Mass Storage devices
usb-storage: device found at 3
usb-storage: waiting for device to settle before scanning
```

```
  Vendor: Intuix    Model: U3                Rev: 6.50
  Type:   Direct-Access                      ANSI SCSI revision: 00
SCSI device sde: 1961695 512-byte hdwr sectors (1004 MB)
sde: Write Protect is off
sde: Mode Sense: 45 00 00 08
sde: assuming drive cache: write through
SCSI device sde: 1961695 512-byte hdwr sectors (1004 MB)
sde: Write Protect is off
sde: Mode Sense: 45 00 00 08
sde: assuming drive cache: write through
 sde: sde1
sd 1:0:0:0: Attached scsi removable disk sde
usb-storage: device scan complete
```

11. Liste des périphériques et IRQ, I/O ports et DMA associés.

```
[root@linux1 ~]# wget
'ftp://rpmfind.net/linux/fedora/core/3/i386/os/Fedora/RPMS/procinfo-18-
13.i386.rpm'
...
[root@linux1 ~]# rpm -Uvh procinfo-18-13.i386.rpm
...
[root@linux1 ~]# procinfo
Linux 2.6.9-ntfs (root@linux1.pinguins) (gcc 3.4.6 20060404 ) #1 1CPU [linux1]

Memory:      Total        Used        Free      Shared     Buffers
Mem:        450080      341576      108504           0       12188
Swap:       522104           0      522104

Bootup: Sat Apr  7 17:38:47 2007    Load average: 0.95 0.74 0.34 1/53 4054

user  :       0:00:34.13   0.6%  page in :         0
nice  :       0:00:00.00   0.0%  page out:         0
system:       0:00:09.26   0.2%  swap in :         0
idle  :       1:35:46.17  99.2%  swap out:         0
uptime:       1:36:29.77         context :    885210

irq  0:   5790478 timer                 irq  9:          0 acpi
irq  1:        89 i8042                 irq 10:          0 uhci_hcd, uhci_hcd
irq  2:         0 cascade [4]           irq 11:     245218 VIA8233, ehci_hcd, u
irq  4:         5                       irq 12:         84 i8042
irq  6:         2                       irq 14:      16792 ide0
irq  7:         6                       irq 15:      51750 ide1
irq  8:         1 rtc

[root@linux1 ~]# lsdev
Device            DMA   IRQ  I/O Ports
------------------------------------------------
0000:00:0a.0                 d000-d0ff
0000:00:0f.0                 d400-d40f
0000:00:10.0                 d800-d81f
0000:00:10.1                 dc00-dc1f
0000:00:10.2                 e000-e01f
0000:00:10.3                 e400-e41f
0000:00:11.5                 e800-e8ff
```

```
8139too                      d000-d0ff
acpi                    9
cascade                 4
dma                          0080-008f
dma1                         0000-001f
dma2                         00c0-00df
eth0                  177
fpu                          00f0-00ff
GPE0_BLK                       4020-4023
i8042                 1 12
ide0                    14   01f0-01f7 03f6-03f6   d400-d407
ide1                    15   0170-0177 0376-0376   d408-d40f
keyboard                     0060-006f
motherboard                  4000-407f 5000-500f
parport0                 7   0378-037a
pic1                         0020-0021
pic2                         00a0-00a1
PM1a_CNT_BLK                   4004-4005
PM1a_EVT_BLK                   4000-4003
PM_TMR                         4008-400b
pnp                            5000-500f
rtc                      8   0070-0077
serial                       03f8-03ff
timer                    0
timer0                       0040-0043
timer1                       0050-0053
uhci_hcd                       d800-d81f   dc00-dc1f   e000-e01f   e400-e41f
uhci_hcd:usb5         193
vga+                         03c0-03df
VIA8237               185      e800-e8ff
vt596_smbus                      5000-5007
```

12. Informations sur le CPU.

```
[root@linux1 ~]# x86info
x86info v1.13.  Dave Jones 2001-2003
Feedback to <davej@redhat.com>.

Found 1 CPU
--------------------------------------------------------------------------
Family: 15 Model: 12 Stepping: 2
CPU Model : Athlon 64
Feature flags:
 fpu vme de pse tsc msr pae mce cx8 apic sep mtrr pge mca cmov pat pse36 clflsh
mmx fxsr sse sse2
Extended feature flags:
 syscall mmxext lm 3dnowext 3dnow
```

13. Informations sur un disque IDE.

```
[root@linux1 ~]# hdparm /dev/hda

/dev/hda:
 multcount    = 16 (on)
 IO_support   =  1 (32-bit)
 unmaskirq    =  1 (on)
 using_dma    =  1 (on)
```

```
 keepsettings =  0 (off)
 readonly     =  0 (off)
 readahead    = 256 (on)
 geometry     = 65535/16/63, sectors = 40060403712, start = 0
[root@linux1 ~]# hdparm -g /dev/hda

/dev/hda:
 geometry     = 65535/16/63, sectors = 40060403712, start = 0
[root@linux1 ~]#
```

14. Utiliser blockdev.

```
[root@linux1 ~]# blockdev --getro /dev/hda    # lecture seule (1)
0
[root@linux1 ~]# blockdev --getsize /dev/had    # taille du disque en secteurs
78242976
[root@linux1 ~]# blockdev --getra /dev/hda   # lecture anticipée en secteurs
256
[root@linux1 ~]#
```

Tâche 2 : Linux et les périphériques

1. Afficher les messages du noyau.

a) Messages associés à la mémoire.

```
[root@linux1 ~]# dmesg | grep -i memory
Memory: 449316k/457664k available (1896k kernel code, 7708k reserved, 766k data,
176k init, 0k highmem)
Freeing initrd memory: 412k freed
Total HugeTLB memory allocated, 0
Freeing unused kernel memory: 176k freed
```

b) Messages associées à l'USB.

```
[root@linux1 ~]# dmesg | grep -i usb
usbcore: registered new driver usbfs
usbcore: registered new driver hub
usbmon: debugfs is not available
usbcore: registered new driver hiddev
usbcore: registered new driver usbhid
drivers/usb/input/hid-core.c: v2.6:USB HID core driver
PCI0 USB3 USB4 USB5 USB6 USB7 AC97 UAR1
ehci_hcd 0000:00:10.4: new USB bus registered, assigned bus number 1
ehci_hcd 0000:00:10.4: USB 2.0 started, EHCI 1.00, driver 10 Dec 2004
usb usb1: configuration #1 chosen from 1 choice
hub 1-0:1.0: USB hub found
USB Universal Host Controller Interface driver v3.0
...
USB Mass Storage support registered.
  Vendor: Generic   Model: USB SD Reader     Rev: 1.00
  Vendor: Generic   Model: USB CF Reader     Rev: 1.01
  Vendor: Generic   Model: USB SM Reader     Rev: 1.02
  Vendor: Generic   Model: USB MS Reader     Rev: 1.03
usb-storage: device scan complete
```

c) Messages associés au CPU.

```
[root@linux1 ~]# dmesg | grep -i cpu
Initializing CPU#0
CPU 0 irqstacks, hard=c03cd000 soft=c03cc000
CPU: After generic identify, caps: 078bfbff e3d3fbff 00000000 00000000 00000001
00000000 00000001
CPU: After vendor identify, caps: 078bfbff e3d3fbff 00000000 00000000 00000001
00000000 00000001
CPU: L1 I Cache: 64K (64 bytes/line), D cache 64K (64 bytes/line)
CPU: L2 Cache: 256K (64 bytes/line)
CPU: After all inits, caps: 078bfbff e3d3fbff 00000000 00000410 00000001
00000000 00000001
Intel machine check reporting enabled on CPU#0.
CPU: AMD Sempron(tm) Processor 3100+ stepping 02
```

d) Messages associés au SCSI.

```
[root@linux1 ~]# dmesg | grep -i scsi
SCSI subsystem initialized
scsi0 : SCSI emulation for USB Mass Storage devices
  Type:   Direct-Access                      ANSI SCSI revision: 00
  Type:   Direct-Access                      ANSI SCSI revision: 00
  Type:   Direct-Access                      ANSI SCSI revision: 00
  Type:   Direct-Access                      ANSI SCSI revision: 00
sd 0:0:0:0: Attached scsi removable disk sda
sd 0:0:0:1: Attached scsi removable disk sdb
sd 0:0:0:2: Attached scsi removable disk sdc
sd 0:0:0:3: Attached scsi removable disk sdd
```

e) Messages associés aux terminaux.

```
[root@linux1 ~]# dmesg |grep -i  tty
serial8250: ttyS0 at I/O 0x3f8 (irq = 4) is a 16550A
00:09: ttyS0 at I/O 0x3f8 (irq = 4) is a 16550A
```

f) Messages divers.

```
[root@linux1 ~]# dmesg |more
Linux version 2.6.17 (root@linux1.pinguins) (gcc version 3.4.6 20060404 (Red Hat
 3.4.6-3) #1 Tue Feb 6 17:20:08 CET 2007
BIOS-provided physical RAM map:
 BIOS-e820: 0000000000000000 - 000000000009f800 (usable)
 BIOS-e820: 000000000009f800 - 00000000000a0000 (reserved)
…
0MB HIGHMEM available.
446MB LOWMEM available.
found SMP MP-table at 000f56d0
On node 0 totalpages: 114416
  DMA zone: 4096 pages, LIFO batch:0
  Normal zone: 110320 pages, LIFO batch:31
DMI 2.3 present.
ACPI: RSDP (v000 FIC                                    ) @ 0x000f7170
ACPI: RSDT (v001 FIC    K8M-800M 0x42302e31 AWRD 0x00000000) @ 0x1bef3000
ACPI: FADT (v001 FIC    K8M-800M 0x42302e31 AWRD 0x00000000) @ 0x1bef3040
```

Remarque : la commande `dmesg` est une des commande les plus importante associée à la gestion des périphériques : elle nous montre ce que les développeurs du noyau veulent nous faire savoir des pilotes et par leur biais des périphériques.

2. Lister les modules du noyau actuellement chargés.

```
[root@linux1 ~]# lsmod | head
Module                  Size  Used by
xfs                   492068  0
reiserfs              218736  0
ipv6                  222368  12
parport_pc             26180  1
lp                     12488  0
parport                34632  2 parport_pc,lp
autofs4                20484  0
sunrpc                140604  1
sd_mod                 19584  0
```

3. Lister les modules pouvant être chargés.

```
[root@linux1 ~]# modprobe -l |head
/lib/modules/2.6.17/kernel/net/sched/sch_ingress.ko
/lib/modules/2.6.17/kernel/net/sched/sch_prio.ko
/lib/modules/2.6.17/kernel/net/sched/cls_fw.ko
/lib/modules/2.6.17/kernel/net/sched/cls_rsvp6.ko
/lib/modules/2.6.17/kernel/net/sched/sch_sfq.ko
/lib/modules/2.6.17/kernel/net/sched/sch_dsmark.ko
/lib/modules/2.6.17/kernel/net/sched/cls_route.ko
/lib/modules/2.6.17/kernel/net/sched/sch_htb.ko
/lib/modules/2.6.17/kernel/net/sched/sch_teql.ko
/lib/modules/2.6.17/kernel/net/sched/sch_atm.ko
```

4. Lister les caractéristiques de quelques fichiers spéciaux.

```
[root@linux1 ~]# ls -l /dev/hda /dev/mice /dev/cdrom
lrwxrwxrwx  1 root root      3 Feb  6 19:28 /dev/cdrom -> hdc
brw-rw----  1 root disk  3,  0 Feb  6 19:28 /dev/hda
crw-------  1 root root 13, 63 Feb  6 19:28 /dev/mice
```

5. Créer un fichier spécial (on se base sur un fichier existant).

```
[root@linux1 ~]# ls -l /dev/null
crw-rw-rw-  1 root root 1, 3 Feb  6 19:28 /dev/null
[root@linux1 ~]# mknod /dev/trou_noir c 1 3
[root@linux1 ~]# chmod a=rw /dev/trou_noir
[root@linux1 ~]# ls -l /dev/trou_noir
crw-rw-rw-  1 root root 1, 3 Feb 10 12:22 /dev/trou_noir
[root@linux1 ~]# echo "RIEN QUE POUR VOS YEUX" > /dev/trou_noir
[root@linux1 ~]#
```

6. Lister les majeurs utilisés.

```
[root@linux1 ~]# cat /proc/devices
Character devices:
  1 mem
  4 /dev/vc/0
  4 tty
  4 ttyS
...
Block devices:
  1 ramdisk
  2 fd
  3 ide0
  8 sd
  9 md
```

```
 22 ide1
 65 sd
...
[root@linux1 ~]#
```

Tâche 3 : La configuration des périphériques

1. Afficher la configuration des modules de périphériques.

```
[root@linux1 ~]# more /etc/modprobe.conf
alias eth0 8139too
alias snd-card-0 snd-via82xx
options snd-card-0 index=0
install snd-via82xx /sbin/modprobe --ignore-install snd-via82xx && /usr/sbin/als
actl restore >/dev/null 2>&1 || :
remove snd-via82xx { /usr/sbin/alsactl store >/dev/null 2>&1 || : ; }; /sbin/mod
probe -r --ignore-remove snd-via82xx
alias usb-controller ehci-hcd
alias usb-controller1 uhci-hcd
```

2. Afficher la configuration d'un module et ses options.

```
[root@linux1 ~]# modinfo 8139too
filename:       /lib/modules/2.6.17/kernel/drivers/net/8139too.ko
author:         Jeff Garzik <jgarzik@pobox.com>
description:    RealTek RTL-8139 Fast Ethernet driver
license:        GPL
version:        0.9.27
parmtype:       multicast_filter_limit:int
parmtype:       media:array of int
parmtype:       full_duplex:array of int
parmtype:       debug:int
parm:           debug:8139too bitmapped message enable number
parm:           multicast_filter_limit:8139too maximum number of filtered
multicast addresses
parm:           media:8139too: Bits 4+9: force full duplex, bit 5: 100Mbps
parm:           full_duplex:8139too: Force full duplex for board(s) (1)
vermagic:       2.6.17 mod_unload 686 REGPARM 4KSTACKS gcc-3.4
depends:        mii
alias:          pci:v000010ECd00008139sv*sd*bc*sc*i*
alias:          pci:v000010ECd00008138sv*sd*bc*sc*i*
[root@linux1 ~]# modinfo -p 8139too
debug:8139too bitmapped message enable number
multicast_filter_limit:8139too maximum number of filtered multicast addresses
media:8139too: Bits 4+9: force full duplex, bit 5: 100Mbps
full_duplex:8139too: Force full duplex for board(s) (1)
```

3. Activer la recherche des périphériques réseaux.

On affiche le résulat à l'écran, on ne met pas à jour la configuration.

```
[root@linux1 ~]# kudzu -p -c NETWORK
-
class: NETWORK
bus: PCI
detached: 0
device: eth0
driver: 8139too
```

```
desc: "Realtek Semiconductor Co., Ltd. RTL-8139/8139C/8139C+"
network.hwaddr: 00:40:CA:B1:26:EF
vendorId: 10ec
deviceId: 8139
subVendorId: 1509
subDeviceId: 6001
pciType: 1
pcidom:    0
pcibus:  0
pcidev:  a
pcifn:  0
[root@linux1 ~]#
```

Remarque : En plus de la classe NETWORK, on peut aussi utiliser les classes VIDEO, MOUSE et OTHER.

4. Visualiser la base de données créée par Kudzu.

Au démarrage, cette base sert de référence pour savoir s'il y a de nouveaux périphériques qui ont été ajoutés.

```
[root@linux1 ~]# tail -15 /etc/sysconfig/hwconf
-
class: IDE
bus: PCI
detached: 0
driver: unknown
desc: "VIA Technologies, Inc. VT82C586A/B/VT82C686/A/B/VT823x/A/C PIPC Bus
Master IDE"
vendorId: 1106
deviceId: 0571
subVendorId: 1509
subDeviceId: 6001
pciType: 1
pcidom:    0
pcibus:  0
pcidev:  f
pcifn:  0
```

Tâche 4 : Le FS sysfs

1. Lister les périphériques associés au bus IDE.

```
[root@linux1 ~]# df -Ta |grep /sys
none          sysfs           0         0         0   -  /sys
[root@linux1 ~]# ls /sys/bus/ide/devices/
0.0  1.0
[root@linux1 ~]# rpm -q sysfsutils
sysfsutils-1.2.0-1
[root@linux1 ~]# systool -b ide
Bus = "ide"

  Device = "0.0"
  Device = "1.0"
```

2. Visualiser les périphériques réseaux.

```
[root@linux1 ~]# ls /sys/class/net
eth0  lo  sit0
[root@linux1 ~]# systool -c net
Class = "net"

  Class Device = "eth0"
    Device = "0000:00:0a.0"

  Class Device = "lo"

  Class Device = "sit0"
```

3. Lister les périphériques bloc.

```
[root@linux1 ~]# ls /sys/block
dm-0  hda  md0   ram1   ram11  ram13  ram15  ram3  ram5  ram7  ram9  sdb  sdd
dm-1  hdc  ram0  ram10  ram12  ram14  ram2   ram4  ram6  ram8  sda   sdc  sde
```

4. Afficher la taille d'un disque.

```
[root@linux1 ~]# cat /sys/block/hda/size
78242976
```

5. Afficher des informations concernant les disques.

```
[root@linux1 ~]# systool -b ide -v
Bus = "ide"

  Device = "0.0"
  Device path = "/sys/devices/pci0000:00/0000:00:0f.0/ide0/0.0"
    drivename           = "hda"
    media               = "disk"
    modalias            = "ide:m-disk"
    uevent              = <store method only>

  Device = "1.0"
  Device path = "/sys/devices/pci0000:00/0000:00:0f.0/ide1/1.0"
    drivename           = "hdc"
    media               = "cdrom"
    modalias            = "ide:m-cdrom"
    uevent              = <store method only>
```

6. Lister les différents bus, classes et périphériques sysfs supportés.

```
[root@linux1 ~]# systool
Supported sysfs buses:
        ac97
        i2c
        ide
        pci
        platform
        pnp
        scsi
        serio
        usb
Supported sysfs classes:
```

```
        block
        drm
        firmware
        graphics
        i2c-adapter
        input
        mem
        misc
        net
        pci_bus
        printer
        raw
        scsi_device
        scsi_disk
        scsi_host
        sound
        tty
        usb_device
        usb
        usb_host
        vc
Supported sysfs devices:
        pci0000:00
        platform
        pnp0
        system
```

7. L'arborescence des périphériques (à partir du bus PCI maître).

```
[root@linux1 ~]# systool -r pci0000:00 |head
Root Device = "pci0000:00"

  Device = "0000:00:00.0"
  Device = "0000:00:00.1"
  Device = "0000:00:00.2"
  Device = "0000:00:00.3"
  Device = "0000:00:00.4"
  Device = "0000:00:00.7"
  Device = "0000:00:01.0"
    Device = "0000:01:00.0"
```

8. Lister les périphériques traditionnels du PC.

```
[root@linux1 ~]# ls /sys/devices/platform/
i8042  pcspkr  power  serial8250  uevent  vesafb.0
[root@linux1 ~]# systool -r platform -a |head
Root Device = "platform"

  Device = "i8042"
    uevent

    Device = "serio0"
      bind_mode
      description
      drvctl
      modalias
```

```
[root@linux1 ~]# systool -r platform -v |head
Root Device = "platform"

  Device = "i8042"
  Device path = "/sys/devices/platform/i8042"
    uevent              = <store method only>

    Device = "serio0"
    Device path = "/sys/devices/platform/i8042/serio0"
      bind_mode           = "auto"
      description         = "i8042 Aux Port"
```

Remarque : i8042 est le contôleur clavier et vesafb gère l'affichage en mode console sous forme Bitmap.

9. Lister le matériel système.

```
[root@linux1 ~]# ls /sys/devices/system/
acpi  cpu  i8237  i8259  ioapic  irqrouter  lapic  timer
[root@linux1 ~]# systool -r system  | head
Root Device = "system"

  Device = "acpi"
    Device = "acpi0"
  Device = "cpu"
    Device = "cpu0"
  Device = "i8237"
    Device = "i82370"
  Device = "i8259"
    Device = "i82590"
```

Remarque : l'i8259 est le gestionnaire des interruptions et l'i8237 le controleur DMA.

10. Afficher la valeur d'un attribut, de tous les attributs SCSI.

```
[root@linux1 ~]# systool -b scsi -A state
Bus = "scsi"

  Device = "0:0:0:0"
    state               = "running"
[root@linux1 ~]# systool -vb scsi
Bus = "scsi"

  Device = "0:0:0:0"
  Device path = "/sys/devices/pci0000:00/0000:00:10.3/usb5/5-1/5-
1:1.0/host0/target0:0:0/0:0:0:0"
    delete              = <store method only>
    device_blocked      = "0"
    iocounterbits       = "32"
    iodone_cnt          = "0x64"
    ioerr_cnt           = "0x63"
    iorequest_cnt       = "0x64"
    max_sectors         = "240"
    model               = "USB SD Reader   "
    queue_depth         = "1"
    queue_type          = "none"
    rescan              = <store method only>
    rev                 = "1.00"
```

```
    scsi_level          = "3"
    state               = "running"
    timeout             = "30"
    type                = "0"
    uevent              = <store method only>
    vendor              = "Generic "
```

11. Visualiser les pilotes associés aux périphériques IDE.

```
[root@linux1 ~]# systool -Dpb ide
Bus = "ide"

  Driver = "ide-cdrom"
  Driver path = "/sys/bus/ide/drivers/ide-cdrom"
    Devices using "ide-cdrom" are:
      Device = "1.0"
      Device path = "/sys/devices/pci0000:00/0000:00:0f.0/ide1/1.0"

  Driver = "ide-disk"
  Driver path = "/sys/bus/ide/drivers/ide-disk"
    Devices using "ide-disk" are:
      Device = "0.0"
      Device path = "/sys/devices/pci0000:00/0000:00:0f.0/ide0/0.0"

  Driver = "ide-floppy"
  Driver path = "/sys/bus/ide/drivers/ide-floppy"
```

Tâche 5 : Udev

1. Est-ce que le démon udevd est présent ?

```
[root@linux1 ~]# ps -e |grep udevd
 1283 ?        00:00:00 udevd
```

2. Brancher une clé USB et visualiser ses caractéristiques.

```
[root@linux1 ~]# dmesg
…
usb 1-6: new high speed USB device using ehci_hcd and address 4
usb 1-6: configuration #1 chosen from 1 choice
scsi2 : SCSI emulation for USB Mass Storage devices
usb-storage: device found at 4
usb-storage: waiting for device to settle before scanning
  Vendor: Intuix    Model: U3                Rev: 6.50
  Type:   Direct-Access                      ANSI SCSI revision: 00
SCSI device sde: 1961695 512-byte hdwr sectors (1004 MB)
sde: Write Protect is off
sde: Mode Sense: 45 00 00 08
sde: assuming drive cache: write through
SCSI device sde: 1961695 512-byte hdwr sectors (1004 MB)
sde: Write Protect is off
sde: Mode Sense: 45 00 00 08
sde: assuming drive cache: write through
 sde: sde1
sd 2:0:0:0: Attached scsi removable disk sde
usb-storage: device scan complete
[root@linux1 ~]# udevinfo -a -p /block/sde
```

```
udevinfo starts with the device the node belongs to and then walks up the
device chain, to print for every device found, all possibly useful attributes
in the udev key format.
Only attributes within one device section may be used together in one rule,
to match the device for which the node will be created.

  looking at class device '/sys/block/sde':
    SYSFS{dev}="8:64"
    SYSFS{range}="16"
    SYSFS{removable}="1"
    SYSFS{size}="1961695"
    SYSFS{stat}="       1        7        8        0        0        0        0
0        0        0        0"

follow the class device's "device"
  looking at the device chain at '/sys/devices/pci0000:00/0000:00:10.4/usb1/1-
6/1-6:1.0/host2/target2:0:0/2:0:0:0':
    BUS="scsi"
    ID="2:0:0:0"
    SYSFS{device_blocked}="0"
    SYSFS{iocounterbits}="32"
    SYSFS{iodone_cnt}="0xb"
    SYSFS{ioerr_cnt}="0x0"
    SYSFS{iorequest_cnt}="0xb"
    SYSFS{max_sectors}="240"
    SYSFS{model}="U3                "
    SYSFS{queue_depth}="1"
    SYSFS{queue_type}="none"
    SYSFS{rev}="6.50"
    SYSFS{scsi_level}="3"
    SYSFS{state}="running"
    SYSFS{timeout}="30"
    SYSFS{type}="0"
    SYSFS{vendor}="Intuix  "

[root@linux1 ~]#
```

3. Lister les règles standards d'Udev.

```
[root@linux1 devices]# ls /etc/udev/rules.d/
[root@linux1 devices]# tail /etc/udev/rules.d/50-udev.rules
KERNEL="hd[a-z]", BUS="ide", SYSFS{removable}="1",
PROGRAM="/etc/udev/scripts/check-cdrom.sh %k DVD-R", SYMLINK="dvdwriter%e"
KERNEL="sr[0-9]*", BUS="scsi", PROGRAM="/etc/udev/scripts/check-cdrom.sh %k DVD-
R", SYMLINK="dvdwriter%e"

# rename sr* to scd*
KERNEL="sr[0-9]*", BUS="scsi", NAME="scd%n"
KERNEL="hd[a-z]*", BUS="ide", SYSFS{removable}="1", NOREMOVE="1"

KERNEL="dasd*[a-z]", SYMLINK="dasd/%b/disc"
KERNEL="dasd*[0-9]", SYMLINK="dasd/%b/part%n"

[root@linux1 devices]#
```

4. Créer une règle Udev : on remplace le nom officiel du périphérique. On teste virtuellement.

```
[root@linux1 devices]# vi /etc/udev/rules.d/20-perso.rules
BUS="scsi", SYSFS{vendor}="Intuix  ",NAME="cleperso"
[root@linux1 devices]# udevtest /block/sde
version 039
looking at '/block/sde'
configured rule in '/etc/udev/rules.d/20-perso.rules' at line 2 applied, 'sde'
becomes 'cleperso'
creating device node '/dev/cleperso', major = '8', minor = '64', mode =
'060600', uid = '0', gid = '0'
```

5. Tester réellement : on met la clé.

```
[root@linux1 devices]# ls -l /dev/sde
ls: /dev/sde: No such file or directory
[root@linux1 devices]# ls -l /dev/cleperso
brw-------  1 root root 8, 65 Feb 10 19:27 /dev/cleperso
[root@linux1 devices]#
```

6. Modifier la règle : on ajoute un lien sur le nom officiel.

```
[root@linux1 devices]# vi /etc/udev/rules.d/20-perso.rules
BUS="scsi", SYSFS{vendor}="Intuix  ",SYMLINK="cleperso"
[root@linux1 devices]# udevtest /block/sde
version 039
looking at '/block/sde'
configured rule in '/etc/udev/rules.d/20-perso.rules' at line 2 applied, added
symlink 'cleperso'
configured rule in '/etc/udev/rules.d/51-by-id.rules' at line 5 applied, added
symlink 'disk/by-path/%c'
creating device node '/dev/sde', major = '8', minor = '64', mode = '060660', uid
= '0', gid = '6'
[root@linux1 devices]#
```

7. Tester à nouveau : on enlève et on remet la clé.

```
[root@linux1 devices]# ls -l /dev/cleperso
lrwxrwxrwx  1 root root 4 Feb 10 19:32 /dev/cleperso -> sde1
[root@linux1 devices]# ls -l /dev/sde
brw-rw----  1 root disk 8, 64 Feb 10 19:32 /dev/sde
```

8. Créer une règle complexe avec un programme qui décide de la création du périphérique.

```
[root@linux1 devices]# vi /etc/udev/rules.d/20-perso.rules
BUS="scsi", SYSFS{vendor}="Intuix  ",PROGRAM="/root/cle_guest.sh %k",
SYMLINK="cleperso"
[root@linux1 devices]# vi /root/cle_guest.sh
#!/bin/sh
if who |grep guest > /dev/null 2>&1 ; then
        exit 0
else
        exit 1
fi

[root@linux1 devices]# chmod u+x /root/cle_guest.sh
[root@linux1 devices]# udevtest /block/sde
version 039
looking at '/block/sde'
configured rule in '/etc/udev/rules.d/51-by-id.rules' at line 5 applied, added
symlink 'disk/by-path/%c'
```

```
creating device node '/dev/sde', major = '8', minor = '64', mode = '060660', uid
= '0', gid = '6'
```

Sur une autre console, guest se connecte.

```
[root@linux1 devices]# udevtest /block/sde
version 039
looking at '/block/sde'
configured rule in '/etc/udev/rules.d/20-perso.rules' at line 3 applied, added
symlink 'cleperso'
configured rule in '/etc/udev/rules.d/51-by-id.rules' at line 5 applied, added
symlink 'disk/by-path/%c'
creating device node '/dev/sde', major = '8', minor = '64', mode = '060660', uid
= '0', gid = '6'
[root@linux1 devices]#
```

On teste réellement.

```
[root@linux1 devices]# who |grep guest
guest    pts/1        Feb 10 19:48 (raichu.pokemon)
[root@linux1 devices]# ls -l /dev/cleperso
lrwxrwxrwx  1 root root 3 Feb 10 19:58 /dev/cleperso -> sde
[root@linux1 devices]# ls -l /dev/sde
brw-rw----  1 root disk 8, 64 Feb 10 19:58 /dev/sde
```

Tâche 6 : lm_sensors

1. Déterminer les pilotes nécessaires à l'utilisation de sensors.

La commande sensors-detect vous pose des questions. On prend les valeurs par défaut.

```
[root@linux1 ~]# sensors-detect

This program will help you determine which I2C/SMBus modules you need to
load to use lm_sensors most effectively. You need to have i2c and
lm_sensors installed before running this program.
Also, you need to be `root', or at least have access to the /dev/i2c-*
files, for most things.
If you have patched your kernel and have some drivers built in, you can
safely answer NO if asked to load some modules. In this case, things may
seem a bit confusing, but they will still work.

It is generally safe and recommended to accept the default answers to all
questions, unless you know what you're doing.

 We can start with probing for (PCI) I2C or SMBus adapters.
 You do not need any special privileges for this.
 Do you want to probe now? (YES/no):
...
To make the sensors modules behave correctly, add these lines to
/etc/modules.conf:

#----cut here----
# I2C module options
alias char-major-89 i2c-dev
#----cut here----

To load everything that is needed, add this to some /etc/rc* file:
```

```
#----cut here----
# I2C adapter drivers
modprobe i2c-viapro
modprobe i2c-isa
# I2C chip drivers
modprobe eeprom
modprobe it87
# sleep 2 # optional
/usr/bin/sensors -s # recommended
#----cut here----

WARNING! If you have some things built into your kernel, the list above
will contain too many modules. Skip the appropriate ones! You really should
try these commands right now to make sure everything is working properly.
Monitoring programs won't work until it's done.

Do you want to generate /etc/sysconfig/lm_sensors? (YES/no):
Copy prog/init/lm_sensors.init to /etc/rc.d/init.d/lm_sensors
for initialization at boot time.
```

Remarque : La commande se termine en affichant le script d'activation de lm_sensors : les pilotes devant être chargés, leur configuration et la commande `sensors -s`.

2. Initialiser lm_sensors (on charge les pilotes et on étalonne les compteurs).

```
[root@linux1 ~]# /etc/init.d/lm_sensors start
DÃ©marrage de lm_sensors :                                  [  OK  ]
```

3. Interroger les détecteurs.

```
[root@linux1 ~]# sensors
it87-isa-0290
Adapter: ISA adapter
VCore 1:   +1.39 V  (min =  +1.42 V, max =  +1.57 V)   ALARM
VCore 2:   +3.34 V  (min =  +2.40 V, max =  +2.61 V)   ALARM
+3.3V:     +6.02 V  (min =  +3.14 V, max =  +3.46 V)   ALARM
+5V:       +4.97 V  (min =  +4.76 V, max =  +5.24 V)
+12V:     +12.22 V  (min = +11.39 V, max = +12.61 V)
-12V:      -4.17 V  (min = -12.63 V, max = -11.41 V)   ALARM
-5V:       -1.45 V  (min =  -5.26 V, max =  -4.77 V)   ALARM
Stdby:     +2.88 V  (min =  +4.76 V, max =  +5.24 V)   ALARM
VBat:      +0.00 V
fan1:     2136 RPM  (min =    0 RPM, div = 8)
fan2:        0 RPM  (min = 3013 RPM, div = 8)          ALARM
fan3:        0 RPM  (min = 3013 RPM, div = 8)          ALARM
M/B Temp:    +29 C  (low  =   +15 C, high =   +40 C)   sensor = thermistor
CPU Temp:    +48 C  (low  =   +15 C, high =   +45 C)   sensor = diode
Temp3:       +29 C  (low  =   +15 C, high =   +45 C)   sensor = thermistor

eeprom-i2c-0-51
Adapter: SMBus Via Pro adapter at 5000
Memory type:            DDR SDRAM DIMM
Memory size (MB):       256

eeprom-i2c-0-50
Adapter: SMBus Via Pro adapter at 5000
```

```
Memory type:            DDR SDRAM DIMM
Memory size (MB):       256

[root@linux1 ~]#
```

Tâche 7 : La technologie SMART

1. Le démon smartd est-il actif ?

```
[root@linux1 ~]# ps -e |grep smartd
 3330 ?        00:00:00 smartd
[root@linux1 ~]# service smartd status
smartd (pid 3330) en cours d'exÃ©cution...
```

2. Afficher des informations générales sur le disque.

```
[root@linux1 ~]# smartctl -i /dev/hda
smartctl version 5.33 [i686-redhat-linux-gnu] Copyright (C) 2002-4 Bruce Allen
Home page is http://smartmontools.sourceforge.net/

=== START OF INFORMATION SECTION ===
Device Model:     SAMSUNG SP4002H
Serial Number:    0411J1ET704025
Firmware Version: QU100-60
User Capacity:    40,060,403,712 bytes
Device is:        In smartctl database [for details use: -P show]
ATA Version is:   6
ATA Standard is:  ATA/ATAPI-6 T13 1410D revision 1
Local Time is:    Sat Feb 10 14:17:36 2007 CET

==> WARNING: May need -F samsung or -F samsung2 enabled; see manual for details.

SMART support is: Available - device has SMART capability.
SMART support is: Enabled

[root@linux1 ~]#
```

Remarque : Le disque supporte SMART, et SMART est activé sur ce disque.

3. Afficher la syntaxe et des exemples !

```
[root@linux1 ~]# smartctl -h |tail -15
        Abort any non-captive test on device

=================================================== SMARTCTL EXAMPLES =====

  smartctl -a /dev/hda                          (Prints all SMART information)

  smartctl --smart=on --offlineauto=on --saveauto=on /dev/hda
                                                 (Enables SMART on first disk)

  smartctl -t long /dev/hda                (Executes extended disk self-test)

  smartctl --attributes --log=selftest --quietmode=errorsonly /dev/hda
                                        (Prints Self-Test & Attribute errors)
  smartctl -a --device=3ware,2 /dev/sda
          (Prints all SMART info for 3rd ATA disk on 3ware RAID controller)
```

```
[root@linux1 ~]# smartctl -A /dev/hda
smartctl version 5.33 [i686-redhat-linux-gnu] Copyright (C) 2002-4 Bruce Allen
Home page is http://smartmontools.sourceforge.net/

=== START OF READ SMART DATA SECTION ===
SMART Attributes Data Structure revision number: 16
Vendor Specific SMART Attributes with Thresholds:
ID# ATTRIBUTE_NAME          FLAG     VALUE WORST THRESH TYPE      UPDATED
WHEN_FAILED RAW_VALUE
  1 Raw_Read_Error_Rate     0x000b   100   100   051    Pre-fail  Always       -
10
  4 Start_Stop_Count        0x0032   100   100   000    Old_age   Always       -
92
  5 Reallocated_Sector_Ct   0x0033   253   253   010    Pre-fail  Always       -
0
  7 Seek_Error_Rate         0x000b   253   253   051    Pre-fail  Always       -
0
  8 Seek_Time_Performance   0x0024   253   253   000    Old_age   Offline      -
0
  9 Power_On_Hours          0x0032   099   099   000    Old_age   Always       -
908657
 12 Power_Cycle_Count       0x0032   100   100   000    Old_age   Always       -
56
194 Temperature_Celsius     0x0022   130   088   000    Old_age   Always       -
36
197 Current_Pending_Sector  0x0033   253   253   010    Pre-fail  Always       -
0
198 Offline_Uncorrectable   0x0031   253   253   010    Pre-fail  Offline      -
0
199 UDMA_CRC_Error_Count    0x000a   100   100   000    Old_age   Always       -
29
200 Multi_Zone_Error_Rate   0x000b   100   100   051    Pre-fail  Always       -
0
201 Soft_Read_Error_Rate    0x000b   100   100   051    Pre-fail  Always       -
0
```

4. Afficher toutes les informations concernant le disque.

```
[root@linux1 ~]# smartctl -a /dev/hda |tail -20

  After command completion occurred, registers were:
  ER ST SC SN CL CH DH
  -- -- -- -- -- -- --
  84 51 00 c8 70 62 e1  Error: ICRC, ABRT at LBA = 0x016270c8 = 23228616

  Commands leading to the command that caused the error were:
  CR FR SC SN CL CH DH DC   Powered_Up_Time  Command/Feature_Name
  -- -- -- -- -- -- -- --  ----------------  --------------------
  c8 00 00 c8 70 62 e1 00  21d+19:45:45.471  READ DMA
  c8 00 80 c8 70 63 e1 00  21d+19:45:45.471  READ DMA
  ca 00 00 c7 7f d2 e1 00  21d+19:45:45.471  WRITE DMA
  ca 00 00 c7 7f d1 e1 00  21d+19:45:45.471  WRITE DMA
  c8 00 00 48 70 61 e1 00  21d+19:45:45.471  READ DMA

SMART Self-test log structure revision number 1
No self-tests have been logged.  [To run self-tests, use: smartctl -t]
```

```
Device does not support Selective Self Tests/Logging
[root@linux1 ~]#
```

5. Démarrer un test sur un disque.

On déclenche le test et après la période indiqué on peut voir les résultats du test.

```
[root@linux1 ~]# smartctl -t short /dev/hda
smartctl version 5.33 [i686-redhat-linux-gnu] Copyright (C) 2002-4 Bruce Allen
Home page is http://smartmontools.sourceforge.net/

=== START OF OFFLINE IMMEDIATE AND SELF-TEST SECTION ===
Sending command: "Execute SMART Short self-test routine immediately in off-line
mode".
Drive command "Execute SMART Short self-test routine immediately in off-line
mode" successful.
Testing has begun.
Please wait 1 minutes for test to complete.
Test will complete after Sat Feb 10 14:30:13 2007

Use smartctl -X to abort test.
[root@linux1 ~]# date
Sat Feb 10 14:32:50 CET 2007
[root@linux1 ~]# smartctl -l selftest /dev/hda
smartctl version 5.33 [i686-redhat-linux-gnu] Copyright (C) 2002-4 Bruce Allen
Home page is http://smartmontools.sourceforge.net/

=== START OF READ SMART DATA SECTION ===
SMART Self-test log structure revision number 1
No self-tests have been logged.  [To run self-tests, use: smartctl -t]
```

Remarque : Pas d'erreur.

6. Regarder le journal des erreurs.

```
[root@linux1 ~]# smartctl -l error /dev/hda | tail

  Commands leading to the command that caused the error were:
  CR FR SC SN CL CH DH DC   Powered_Up_Time  Command/Feature_Name
  -- -- -- -- -- -- -- --  ----------------  --------------------
  c8 00 00 c8 70 62 e1 00  21d+19:45:45.471  READ DMA
  c8 00 80 c8 70 63 e1 00  21d+19:45:45.471  READ DMA
  ca 00 00 c7 7f d2 e1 00  21d+19:45:45.471  WRITE DMA
  ca 00 00 c7 7f d1 e1 00  21d+19:45:45.471  WRITE DMA
  c8 00 00 48 70 61 e1 00  21d+19:45:45.471  READ DMA

[root@linux1 ~]#
```

7. Lister les tests supportés par le disque.

```
[root@linux1 ~]# smartctl -c /dev/hda
smartctl version 5.33 [i686-redhat-linux-gnu] Copyright (C) 2002-4 Bruce Allen
Home page is http://smartmontools.sourceforge.net/

=== START OF READ SMART DATA SECTION ===
General SMART Values:
Offline data collection status:  (0x00) Offline data collection activity
                                        was never started.
```

```
                                   Auto Offline Data Collection: Disabled.
Self-test execution status:         (   0) The previous self-test routine completed
                                   without error or no self-test has ever
                                   been run.
Total time to complete Offline
data collection:                    (1800) seconds.
Offline data collection
capabilities:              (0x1b) SMART execute Offline immediate.
                                   Auto Offline data collection on/off support.
                                   Suspend Offline collection upon new
                                   command.
                                   Offline surface scan supported.
                                   Self-test supported.
                                   No Conveyance Self-test supported.
                                   No Selective Self-test supported.
SMART capabilities:               (0x0003) Saves SMART data before entering
                                   power-saving mode.
                                   Supports SMART auto save timer.
Error logging capability:           (0x01) Error logging supported.
                                   No General Purpose Logging support.
Short self-test routine
recommended polling time:           (   1) minutes.
Extended self-test routine
recommended polling time:           (  30) minutes.

[root@linux1 ~]#
```

- *Ext3, ReiserFS, XFS*
- *Journal interne/externe*
- *Cd : mkisofs, cdrecord*
- *Dvd : growisofs*
- *Automount_media*

5

Les systèmes de fichiers

Objectifs

Ce chapitre traite des systèmes de fichiers. On présente comment Linux accède aux fichiers. On apprend concrètement à gérer les FS XFS, ReiserFS, NFS. On étudie plus complètement le FS Ext3. On présente aussi d'autres FS plus exotiques comme UnionFS ou tmpfs. Enfin on présente les différentes techniques de montage à la volée de FS.

Contenu

La vision théorique de la gestion des fichiers

Les FS à journalisation

Le FS XFS

Le FS ReiserFS

Le FS Ext3

Le FS NFS

FS spéciaux

Le montage à la volée

Ateliers



La vision théorique de la gestion des fichiers

La théorie

La vision applicative de la gestion des fichiers

Du point de vue des applications, les fichiers sont référencés par un chemin, par exemple : /var/spool/cron/root. L'ensemble des fichiers forment une seule arborescence qui débute par le répertoire racine (/).

Un fichier a également des caractéristiques (des attributs), principalement :

- Son type : fichier ordinaire, répertoire, périphérique, lien symbolique, tube, etc.
- Ses droits.
- Son propriétaire, son groupe.
- Les dates de dernière modification, de dernier accès et de création.

Un fichier ordinaire est vu comme un espace adressable d'octets. Sa taille indique en fait l'adresse de son dernier octet + 1.

Une application peut réaliser un certain nombre d'actions sur un fichier :

- Créer ou supprimer un fichier (en fait créer ou supprimer une entrée d'un répertoire).
- Obtenir ses caractéristiques.
- Changer certaines de ses caractéristiques (propriétaire, groupe, droits, dates).
- Ouvrir ou fermer un fichier. Lire ou écrire des octets, se déplacer dans l'espace adressable du fichier, verrouiller une plage d'octets du fichier. Changer la taille du fichier, éventuellement supprimer l'intégralité de ses octets.
- Synchroniser le fichier : mettre à jour le disque par rapports aux tampons mémoire.
- Spécifiquement pour un répertoire, une application peut lire, créer ou supprimer des entrées d'un répertoire ainsi que créer ou détruire un répertoire et créer ou supprimer des liens.

Une application est également associée à un répertoire courant et à un répertoire racine. Les applications qui seront ses descendantes hériteront des références à ces répertoires ainsi que globalement à tous les fichiers ouverts par l'application. Une application accède à un fichier, après son ouverture grâce à un descripteur de fichier (handle). C'est une valeur numérique. Par défaut, les trois premiers descripteurs (0, 1 et 2) forment les entrées/sorties standards (entrée standard, sortie standard, erreur standard).

Remarque : La sémantique d'accès aux fichiers que l'on vient de présenter provient des premiers systèmes Unix. Pour l'essentiel, elle est maintenant la base des normes ISO de conceptions des applications. Elle est notamment valable pour Windows.

La vision noyau de la gestion des fichiers

Le VFS

A l'intérieur du noyau, c'est le module VFS (Virtual File System) qui a pour charge de gérer les fichiers. C'est lui, par l'intermédiaire des appels systèmes, qui offre aux applications la sémantique que l'on vient de décrire plus haut.

En fait, ce n'est pas le VFS qui réalise concrètement les opérations, par exemple, créer un répertoire. Chaque fichier est géré à l'intérieur d'un système de fichier particulier.

Un FS correspond à une sous arborescence gérée comme un tout par le noyau. Avant de pouvoir accéder à un des fichiers d'un FS il faut que ce dernier soit « monté ». Cette opération correspond donc à son activation. Concrètement, on confond la racine du FS à un répertoire (dit répertoire de montage) de l'arborescence active (appartenant à un FS déjà monté). L'opération inverse, le démontage, empêche par la suite l'accès aux fichiers du FS.

Pour chaque FS, le VFS maintient plusieurs tables systèmes (cf. plus bas). La plus importante est la table des inodes. Une inode contient les caractéristiques et l'emplacement d'un fichier. Le noyau référence une inode par le FS qui l'abrite et par son numéro (numéro d'inode). Ce sont les entrées de répertoires qui permettent au noyau de traduire un nom de fichier en numéro d'inode.

En bref, c'est le VFS qui offre une vision unifiée de la gestion de fichiers. C'est lui qui sert d'intermédiaires à la gestion concrète des fichiers réalisés au sein des FS.

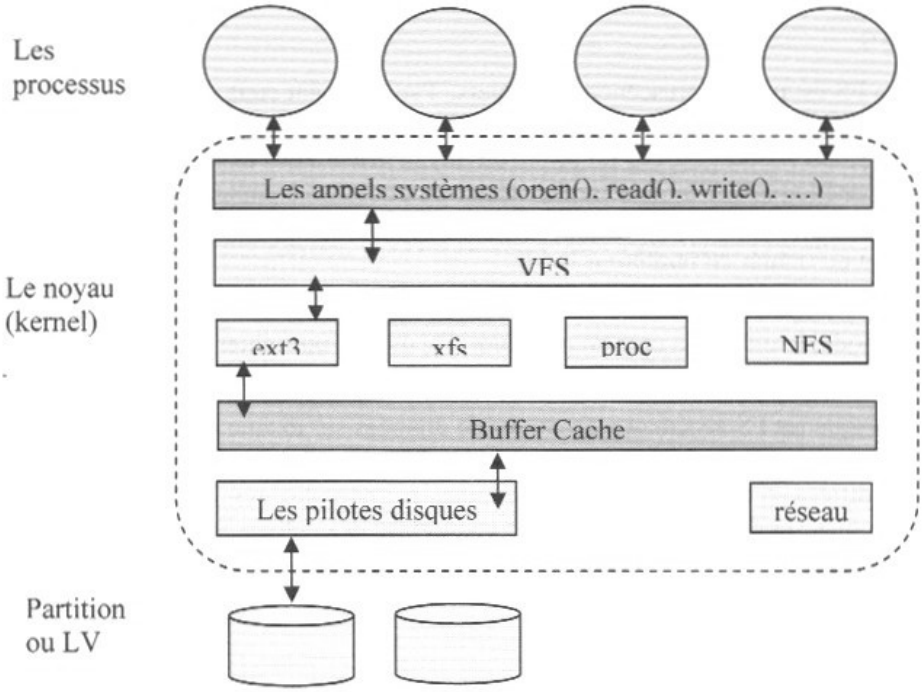


Fig. Le noyau et les FS.

Les tables systèmes du noyau

- La table des fichiers ouverts par processus. Chaque entrée est associée à une entrée de la table des fichiers ouverts par l'ensemble des processus.
- La table des fichiers ouverts par l'ensemble des processus. Chaque entrée est associée à une entrée de la table des inodes ouvertes.
- La table des inodes ouvertes. L'inode d'un répertoire de montage pointe sur le répertoire racine du FS monté.
- La table des FS actuellement montés. Chaque entrée référence les inodes du répertoire racine et du répertoire de montage ainsi que le Superbloc.

Quelques appels systèmes

`creat()`, `unlink()`
: Créer un fichier, le supprimer.

`mkdir()`, `rmdir()`, `readdir()`
: Créer, supprimer un répertoire, accéder à ses entrées.

`stat()`
: Lire les caractéristiques d'un fichier.

`open()`, `close()`, `read()`, write(), `lseek()`, `flock()`	Ouvrir, fermer un fichier, lire ou écrire des octets, se déplacer dans le fichier, verrouiller une zone d'un fichier.
`chmod()`, `chown()`, `chgrp()`, `utime()`	Modifier les caractéristiques d'un fichier.
`fsync()`	Synchroniser un fichier.
`chdir()`, `chroot()`	Changer de répertoire courant, changer de répertoire racine.
`mount()`, `umount()`	Montage, démontage de FS.
`readlink()`, `symlink()`	Gestion des liens symboliques.

Les FS

Un FS est une arborescence de fichiers. Typiquement cette arborescence est stockée dans une partition ou dans un volume logique. Il existe également des FS purement réseau comme NFS. Un FS est géré par un pilote du noyau grâce à des tables systèmes. En particulier un Super-bloc et une table d'inodes .Ceux-ci, pour être compatible avec le VFS doivent comporter obligatoirement certains champs. Le pilote doit être aussi capable de traduire un nom de fichier en numéro d'inode.

Par exemple, dans le cas du pilote FAT, un FS d'origine Windows, il n'y a pas sur disque d'inode, le pilote les crée en mémoire. De même, un fichier FAT n'a pas de groupe ou de propriétaire. Ces informations peuvent être fournies au pilote lors du montage.

Lorsque les FS accèdent à des blocs de données sur disque ils sont stockés dans le buffer cache (mémoire tampon). Quand les applications modifient les données, elles sont en fait modifiées en mémoire, dans les tampons. A intervalles réguliers, les tampons modifiés sont recopiés sur disque. Cette stratégie économise des entrées/sorties.

Les types de FS

Notion de type de FS

Grâce au VFS, Linux est capable de gérer simultanément plusieurs FS de type différents. Chaque type de FS, comme on l'a vu est associé à :

- Un pilote
- Des structures de données mémoire et disques.
- Des utilitaires permettant sa création, sa maintenance, sa sauvegarde.

Du point de vue de l'administrateur, ce qui compte en fin de compte ce sont les fonctionnalités. C'est par rapport à elles qu'il choisit tel ou tel FS.

Remarque : Créer un FS, (on dit aussi formater une partition), c'est écrire sur disque les tables systèmes (Super-bloc), tables d'inodes, répertoire racine, ...) associé à son type.

Panorama

- L'ancien standard ext2.
- Les FS à journalisation : ext3, reiserfs, xfs, jfs et vxfs.
- Les FS Microsoft : msdos, vfat, ntfs.
- Le FS des cdrom : iso9660.
- Les FS réseaux : nfs, smbfs, ncpfs.
- Les FS systèmes : proc, sys, selinux.

- Les FS spéciaux : tmpfs, unionfs.

Pour en savoir plus

Les pages de manuel

fs(5)

On peut lire avec profit, les pages de manuel concernant les appels systèmes : creat(2), write(2), …, mkdir(2), chroot(2), …, mount(2), umount(2), …

Howto

Filesystems HOWTO

Internet

Wikipedia – Liste des FS
http://en.wikipedia.org/wiki/List_of_file_systems

Livre

Understanding the Linux Kernel, par D.P.Bovet & M. Cesati, chez O'Reilly.

Les FS à journalisation

La théorie

Les FS à journalisation revêtent une grande importance, car c'est l'un deux que l'on va utiliser majoritairement pour stocker le système et les applications.

La journalisation

Le FS Ext2 est un très bon FS, mais il n'est pas tolérant aux pannes. Pour des raisons de performances, ses tables systèmes sont essentiellement en mémoire et ne sont mises à jour sur disque que de temps en temps. En cas d'arrêt brutal du système, le disque peut être dans un état incohérent.

Dans un FS journalisé, avant de modifier les tables systèmes, le pilote écrit sur disque les détails de la transaction dans un journal. En cas d'arrêt brutal, il suffit de compléter la transaction. Le FS fonctionne alors un peu comme une base de données basée utilisant des transactions avec les opérations COMMIT et ROLL BACK.

Dans un FS journalisé, la vérification du FS est très rapide, car elle peut se résumer à la vérification du journal.

Les différents FS journalisés

Ext3

Ce FS est compatible ascendant descendant avec l'ancien standard ext2. Un FS ext3 est essentiellement un FS ext2 avec un journal.

ReiserFS

Ce FS est très novateur. Il gère très bien les petits fichiers et les répertoires qui contiennent énormément de fichiers.

XFS

Ce FS, crée par SGI pour ses systèmes Unix IRIX est très performant.

Remarque : Linux supporte aussi le FS JFS d'origine IBM, et VxFS de la société Veritas. Pour plusieurs raisons (le prix dans le cas de VxFS), ces FS sont peu utilisés en exploitation.

Les particularités des différentes distributions

RedHat

RedHat ne supporte que le FS Ext3.

SuSE

En standard, SuSE utilise le FS ReiserFS.

Pour en savoir plus

Internet

Softpanorama Filesystems Webliography
http://www.softpanorama.org/Internals/filesystems.shtml

Le FS XFS

La théorie

XFS est un FS à journalisation très performant. Il a été crée par SGI pour ses systèmes UNIX IRIX. Sous Linux, historiquement, c'est le premier qui a supporté les ACL. C'est une des raisons pour laquelle beaucoup de serveurs Samba l'utilisent. Ses grandes capacités et ses performances l'ont fait choisir également pour construire des serveurs Web.

Le savoir concret

Caractéristiques

- FS en Open Source (GPL).
- Adressage 64bits :
  - FS limité à 2 Tera octets en noyau 2.4, et à 16 Tera octets en noyau 2.6 et adressage 32 bits. Le FS est limité à 9 millions de Tera octets en adressage 64 bits.
  - Un fichier à pour taille limite 9 millions de Tera octets.
  - Des millions de fichiers par répertoire.
  - Des millions d'inodes (adressage par défaut inférieur à 32 bits, mais potentiellement sur 64 bits).
- Système de fichiers journalisé (journal interne ou externe).
- Peut être agrandit à chaud.
- Taille des blocs comprise entre 512 octets et 64 Ko (limité à 4Ko sur x86).

Les principales commandes

`mkfs.xfs`	Crée un FS XFS.
`xfs_check`	Vérifie un FS XFS.
`xfs_growfs`	Agrandit un FS XFS monté.
`xfs_info`	Affiche des informations concernant un FS XFS.
`xfs_repair`	Répare un FS XFS.
`xfs_dump`, `xfs_restore`	Sauvegarde et restaure un FS XFS.

Pour en Savoir Plus

Les pages de manuel

xfs(5), xfs_admin(8), xfs_bmap(8), xfs_check(8), xfs_copy(8), xfs_db(8), xfs_freeze(8), xfs_growfs(8), xfs_info(8), xfs_io(8), xfs_logprint(8), xfs_mkfile(8), xfs_ncheck(8), xfs_quota(8), xfs_repair(8), xfs_rtcp(8), mkfs.xfs(8), attr(1)

Internet

XFS Filesystem : Home Page
http://www.sgi.com/products/software/xfs/

Scalability in the XFS File System
http://www.usenix.org/publications/library/proceedings/sd96/sweeney.html

Le FS ReiserFS

La théorie

Le FS ReiserFS a été créé par Hans Reiser. Sous Linux, historiquement, c'est le premier qui a permis le redimensionnement à chaud. Il sait bien gérer les fichiers de nombreux fichiers par répertoire et il est économique pour gérer de petits fichiers. C'est le FS par défaut de la distribution SuSE.

Le savoir concret

Les caractéristiques

- FS en Open Source.
- Limites :
  - 4 Go de fichiers.
  - 2 Go de fichiers par répertoire.
  - Un FS ou un fichier est limité à 16 Tera octets.
- FS à journalisation (journal interne ou externe).
- Redimensionnable à chaud (agrandissement ou rétrécissement).
- Offre plusieurs algorithmes pour la gestion des répertoires (r5, rasparov et tea). Ils permettent (notamment tea), la gestion de répertoires contenant des milliers de fichiers.
- Taille des blocs comprise entre 512 octets et 8Ko.
- Ne dispose pas de commandes de sauvegarde (sauvegarde par tar).

Les commandes

`mkreisrfs`	Crée un FS ReiserFS.
`reiserfstune`	Visualise ou change les caractéristiques d'un FS ReiserFS.
`debugreiserfs`	Règle certains problèmes. Par défaut, affiche le SuperBloc.
`resize_reiserfs`	Retaille un FS ReiserFS à froid (non monté) ou à chaud (monté).
`reiserfsck`	Vérifie ou répare un FS ReiserFS.

Pour en savoir Plus

Les pages de manuel

mkreiserfs(8), reiserfstune(8), debugreiserfs(8), resize_reiserfs(8), reiserfsck(8)

Internet

ReisefFS – Le site officiel
http://www.namesys.com/

Wikipedia – ReiserFS
http://fr.wikipedia.org/wiki/ReiserFS

Le FS Ext3

La théorie

Ext3 est un FS journalisé qui a le grand avantage d'être compatible ascendant/descendant avec l'ancien standard Linux Ext2. Il suffit d'ajouter un journal pour convertir un Ext2 en Ext3, et il suffit d'utiliser le pilote Ext2 pour accéder en Ext2 à un FS Ext3. Le FS Ext3 est le seul FS journalisé supporté par RedHat.

La structure d'un FS Ext3

Un FS Ext3 commence par un Super Bloc qui est suivi de plusieurs groupes de blocs. Chaque groupe de blocs une sorte de mini-FS, il possède sa propre table d'inodes et son propre pool de blocs. Il contient également une copie du Super bloc.

Le journal est présent normalement dans le fichier ./.journal à la racine du FS.

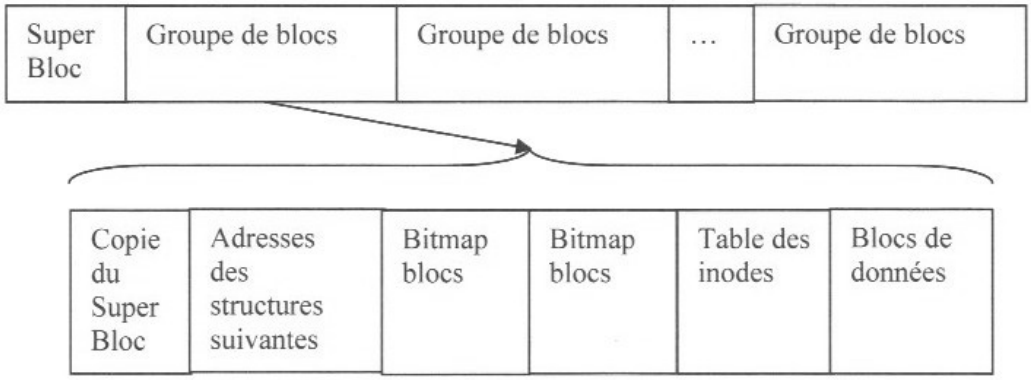


Fig. Structure d'un FS.

Structure du SuperBloc (principaux champs)

- Nombre total d'inodes.
- Nombre total de blocs.
- Nombre de blocs réservés à root.
- Nombre de blocs libres.
- Nombre d'inodes libres
- Taille des blocs.
- Nombre de blocs par groupe de blocs.
- Nombre d'inodes par groupe de blocs.
- Date du dernier montage.
- Date de la dernière écriture.
- Etat du FS.
- Date de la dernière vérification du FS.
- Temps maximal entre 2 vérifications.

Structure d'une Inode (principaux champs)

- Mode (type et droits).
- UID.
- GID.
- Taille en octets.
- Date de dernier accès.
- Date de création.
- Date de dernière modification.
- Date de suppression.
- Nombre de liens matériels.
- Nombre de blocs (512o) alloués à l'inode.
- Attributs.
- Adresses des blocs de données.

Remarque : La date de création correspond en fait à la date de dernière modification de l'inode.

Les adresses de blocs sont composées de 15 adresses :

- Les douze premières entrées correspondent aux adresses des douze premiers blocs.
- La 13ième entrée contient l'adresse d'un bloc qui contient l'adresse des blocs suivants (un niveau d'indirection).
- La 14ième entrée offre deux niveaux d'indirection : elle contient l'adresse d'un bloc qui contient l'adresse de blocs qui contiennent l'adresse des blocs de données suivant.
- La 15ième entrée offre trois niveaux d'indirection.

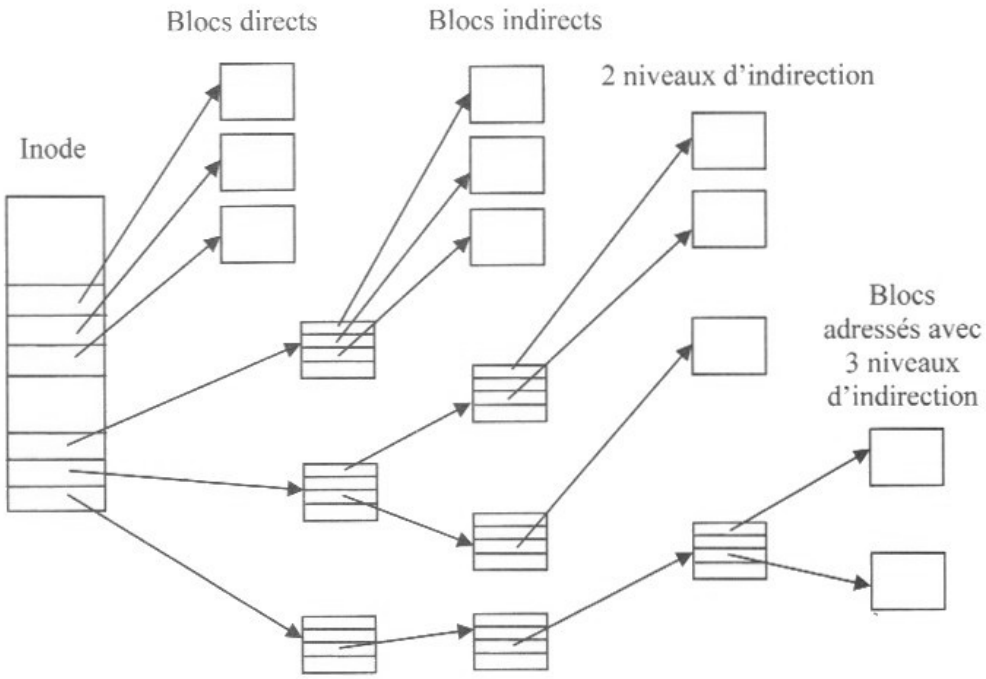


Fig. Adressage des blocs

Structure d'un répertoire

Un répertoire est un fichier particulier. Il est composé de plusieurs entrées, une par fichier contenu dans le répertoire. Les deux premières entrées sont obligatoirement « . » et « .. ». Chaque entrée à la structure suivante :

- Le numéro de l'inode du fichier.
- La taille de l'entrée.
- La taille du nom de fichier.
- Le nom du fichier.

Le savoir concret

Les caractéristiques

- FS en Open Source.
- Limites :
  - FS limité à 16 Tera octets.
  - Fichier limité à 2 Tera octets.
  - Nombre de fichiers limité à la création du FS (dimensionnement de la table d'inodes).
- FS journalisé.

- Peut être agrandit à chaud. Redimensionnable à froid.
- Taille des blocs : 1Ko, 2Ko ou 4Ko.
- Dispose de commandes de sauvegarde.

Les commandes

`mke2fs`	Crée un FS ext2/ext3.
`e2fsck`	Vérifie ou répare un FS ext2/ext3.
`dumpe2fs`	Affiche des informations sur un FS ext2/ext3.
`ext2online`	Agrandit à chaud un FS ext2/ext3.
`resize2fs`	Retaille un FS ext2/ext3 (à froid).
`e2label`	Affiche ou modifie le label d'un FS ext2/ext3.
`dump, restore`	Sauvegarde, restaure un FS ext2/ext3.
`tune2fs`	Modifie les paramètres d'un FS ext2/ext3. Convertie un ext2 en ext3.
`debugfs`	Dépanne un FS ext2/ext3.

Les principales options de création d'un Ext2/Ext3 (mke2fs)

-b taille	Taille du bloc (1024, 2048 ou 4096 octets).
-f taille	Taille des fragments (non supporté).
-g nbbpg	Nombre de blocs par groupe de blocs. On a intérêt à laisser le système choisir cette valeur.
-c	Vérifie la présence de Bad blocs (peu usuel).
-j	Crée un journal (et donc un Ext3).
-F	Force la création du FS.
-q	Travaille en silence.
-i tf	Crée des inodes en supposant que les fichiers font tf octets en moyenne.
-N nb	Crée nb inodes.
-m pourcent	Réserve pourcent % de place pour root. Par défaut 5 %.
-n	Simulation : ne crée par le FS.
-J options	Options de journalisation : size=Taille ou device=Périphérique. Taille indique la taille d'un journal interne et Périphérique indique l'emplacement d'un journal externe.
-O journal_dev <dev>	Crée un journal externe sur dev.

Remarque : Plusieurs paramètres (taille des blocs, nombres d'inodes) sont choisis de manière heuristique par la commande. Il faut vraiment savoir ce que l'on fait quand on propose ses propres valeurs.

Les principales options du montage

`acl`	Le support des ALC est activé.
`errors=<stra>`	La stratégie en cas d'erreur : continue (on continue), remount-ro (on remonte le FS en lecture seul) ou panic (le système s'arrête).
`noload`	Le journal n'est pas chargé.

data=<stra>	La stratégie de gestion du journal : data=ordered (par défaut), les données sont écrites sur disque avant que les données structurelles soient écrites dans le journal. data=writeback, Le mode le plus rapide, les données structurelles sont écrites dans le journal. Des données peuvent être écrites après. Ainsi la structure du FS est garantie mais des incohérences au niveau des fichiers peuvent survenir (utilisation d'anciennes données, …). data=journal, es données structurelles et les données sont écrites dans le journal.

La vérification et la réparation (e2fsck)

-y	Mode automatique : on répare.
-n	Mode automatique : on ne fait que vérifier.
-f	Force une vérification approfondie (pas uniquement le journal).
-j device	Spécifie l'emplacement d'un journal externe.
-b numero	Indique l'emplacement d'un Super Bloc de remplacement. Cet emplacement dépend de la taille en bloc (cf. le man).
-D	Optimise les répertoires.

Paramétrage (tune2fs)

Options –j, -m, -J : voir les options équivalentes de mke2fs.

-c mmc	Spécifie le nombre maximum de montage avant un fsck.
-C nc	Change le nombre de montages déjà comptabilisés.

Pour en savoir plus

Les pages de manuel

debugfs(8), dump(8), dumpe2fs(8), e2fsck(8), e2image(8), e2label(8), ext2online(8), mke2fs(8), resize2fs(8), tune2fs(8)

Les fichiers

/usr/include/ext2fs/*	Les fichiers Include en langage C qui décrivent les tables systèmes d'Ext2/Ext3.

Internet

Wikipedia – Ext3
http://en.wikipedia.org/wiki/Ext3

Linux Ext3 FAQ
http://batleth.sapienti-sat.org/projects/FAQs/ext3-faq.html

Le FS NFS

La théorie

NFS est un service qui offre le partage de fichiers de manière transparente. L'accès aux ressources du serveur passe par un montage de FS au niveau du client. NFS est un standard provenant du monde Unix. Un serveur NFS doit supporter au minimum les deux services NFS et MOUNTD. Ces services sont basés sur le protocole RPC. Ce protocole utilise le service RPC-TCP/IP PORTMAP pour associer un numéro de programme RPC à un numéro de port TCP ou UDP.

L'authentification sur laquelle repose NFS est basée sur les UID et GID. Pour gérer de manière cohérente un ensemble de clients et de serveurs en NFS, il faut qu'ils partagent les mêmes comptes, par exemple via le service NIS ou LDAP. Par défaut l'utilisateur root local, est considéré comme nobody sur le serveur.

Le savoir concret

Les commandes

`rpcinfo`	Affiche les services RPC offerts par un serveur.
`showmount`	Affiche les clients NFS d'un serveur ou les ressources partagés d'un serveur.
`exportfs`	Exporte une arborescence de fichiers. L'option –a lit le fichier exports.
`portmap`	Démon qui traduit un programme RPC en Numéro de ports UDP/TCP.
`mountd`	Démon qui donne au client une référence à la ressource montée.
`nfsd`	Démon qui transfère les données.
`nfs`	RC qui gère le service NFS (active portmap, mount, nfsd, exportfs).

Les fichiers

/etc/rpc	Liste les services RPC de manière symbolique et numérique.
/etc/exports	Liste les ressources exportées.

Focus : Le fichier /etc/exports

```
Serveur# cat /etc/exports
/export/games          venus(rw)
/usr/share/doc         *(ro)
```

Chaque ligne du fichier correspond à une arborescence exportée par le serveur. Le premier champ donne le chemin de cette arborescence, le deuxième indique les clients autorisés à y accéder et entre parenthèses le mode (rw : lecture/écriture, ro : lecture seule). Le caractère * autorise tout client à accéder à la ressource.

Focus : Le montage NFS

```
client#  mount  -t  nfs  venus:/export/games   /mnt
```

On associe l'arborescence /export/games du serveur NFS venus au répertoire local /mnt.

Les protocoles

udp/111,tcp/111, rpc/100000	Le service Portmap.
rpc/100003, udp,tcp/2049	Le service NFS.

rpc/100005	Le service Mountd.

Pour en savoir plus

Les pages de manuel

exportfs(8), exports(5), nfs(5), nfsstat(8), showmount(8), rpcinfo(8), nfsd(8), mountd(8), idmapd(8), idmapd.conf(5), mount(8)

Internet

Linux NFS-HOWTO
http://nfs.sourceforge.net/nfs-howto/

Livres

Managing NFS and NIS, par Hal Stern & all, chez O'Reilly

Administration réseau sous Linux, par Olaf Kirch & Terry Dawson, chez O'Reilly

FS spéciaux

Les FS en mémoire

Disques en RAM (Ramdisk)

Un Ramdisk est une portion de la RAM utilisable comme un disque. Si l'on formate ce disque on obtient un FS en RAM. Les Ramdisks sont supportés naturellement par les noyaux 2.4 et ultérieurs.

Vous pouvez configurer/créer des Ramdisks en configurant le noyau par l'intermédiaire du chargeur. Par exemple voici une ligne de configuration de Grub :

```
kernel /vmlinuz-2.4.20 ro root=LABEL=/  ramdisk_size=16000
```

Potentiellement vous pouvez utiliser plusieurs Ramdisk grâce aux noms de périphériques /dev/ram0, /dev/ram1, etc.

ATTENTION : Une fois la mémoire d'un Ramdisk utilisée, le noyau bloque la mémoire associée. Elle ne pourra plus être utilisée à d'autres fins ultérieurement.

Le FS tmpfs

Le FS tmpfs est en mémoire, mais la mémoire totale (RAM+SWAP). Il est fait notamment pour être monté sur /tmp. On peut spécifier les droits de la racine et sa taille. Par défaut, il utilise la moitié de la mémoire et il a les droits de /tmp.

```
# mount -t tmpfs -o mode=1777,size=512M none /mnt
```

Le FS UnionFS

Le FS UnionFS fusionne plusieurs arborescences en une seule, tout en gardant leur contenu physiquement séparé. UnionFS permet de mélanger des arborescences en lecture seule et en lecture écriture, par exemple un FS d'un Ramdisk et un FS d'un cdrom. Cette technique est utilisée notamment dans les Live-CD.

```
# modprobe unionfs
# mount -t unionfs -o dirs=/export=rw:/cdrom=ro unionfs /mnt/union
```

Le répertoire /mnt/union contient la fusion des arborescences /export et /cdrom.

FS dans un fichier

Grâce au pilote `loop`, l'image d'un FS stockée dans un fichier peut être monté et utilisé.

Cette possibilité offre de multiples utilisations, par exemple, l'accès à plusieurs cdrom simultanément avec un seul lecteur. Il suffit de les copier un par un dans des fichiers disque avec la commande `dd` et ensuite de les monter.

Les cdrom/dvdrom

La création d'un CD-ISO se fait normalement en deux temps :

1) On crée une image iso dans un fichier à partir d'une arborescence.
2) On grave l'image.

Les commandes

`mkisofs`	Créer l'image d'un FS iso9660 d'une arborescence donnée en paramètre.
`cdrecord`	Grave une image (le plus souvent iso).

`growisofs`	Permet de créer une image iso et de la graver sur un DVD. Permet également d'ajouter des fichiers à une image iso.
`eject`	Ejecte le cdrom ou le dvd.

Exemples :

```
# mkisofs -r -o image.iso -J -V docs /usr/share/doc
# cdrecord -scanbus  dev=ATAPI
# cdrecord -v speed=16 dev=ATAPI:0,1,0  image.iso
```

La création de l'image iso image.iso est réalisée à partir de l'arborescence /usr/share/doc avec les extensions Joliet et RockRidge. On lui donne le label docs.

La commande `cdrecord -scanbus` recherche un graveur. Elle donne comme résultat l'adresse SCSI du graveur. Cette adresse est donnée dans la commande suivante qui réalise le gravage.

```
# cdrecord dev=/dev/cdrom image.iso
```

On grave l'image image.iso, mais ici on fait référence directement au périphérique. On utilise les options par défaut.

```
# mkisofs -r /usr/share/doc | cdrecord -v speed=16 dev=ATAPI:0,1,0 -
```

On crée l'image et on la grave en même temps.

```
# growisofs  -Z /dev/dvd  -R -J  /var/ftp
```

Grave sur un DVD au format ISO avec les extensions Joliet et Rock-Ridge l'arborescence /var/ftp.

Pour en savoir plus

Les pages de manuel

Tmpfs : mount(8)

UnionFS : unionctl(8), uniondbg(8), unionfs(4), unionimap(8)

Périphérique Loop : mount(8), losetup(8)

Cdrom/dvdrom : mkisofs(8), cdrecord(1), growisofs(1m)

Howto

CD-Writing HOWTO

Internet

Linux Ramdisk mini-HOWTO
http://www.vanemery.com/Linux/Ramdisk/ramdisk.html

UnionFS – Le site officiel
http://www.filesystems.org/project-unionfs.html

YoLinux Tutorial: Burning a CD or DVD
http://www.yolinux.com/TUTORIALS/LinuxTutorialCDBurn.html

Le montage à la volée

La théorie

Un système de fichiers peut être monté manuellement ou automatiquement au démarrage du système s'il est référencé dans le fichier /etc/fstab. Le montage à la volée, lui, correspond à un montage automatique soit quand le FS est disponible soit quand on y accède.

Panorama des solutions

L'automount

Cette technique est très ancienne et très éprouvée. Elle permet le montage automatique d'un FS lorsque l'on se déplace dans un répertoire associé au FS. Inversement le démontage est réalisé quand on n'utilise plus le FS. Cette technique est essentiellement utilisée pour les montages NFS. On peut aussi l'utiliser pour accéder aux cdroms.

Le supermount

Cette technique, qui fait appel au pilote `supermount` du noyau, n'est utilisée que sur les distributions Mandrake/Mandriva.

Les interfaces graphiques

Aussi bien KDE que Gnome permettent le montage automatique des cdroms ou des clés USB.

Udev

La technologie Udev qui crée automatiquement les entrées de périphériques (/dev/*) peut activer des scripts lorsque l'on connecte un disque hotplug comme par exemple une clé USB.

Le savoir concret

Gnome

On active l'outil `Editeur de configuration GConf` et ensuite on active l'entrée de menu `Desktop -> gnome -> volume_manager`. Enfin on active les clés `automount_media` et `automount_drives`.

KDE

KDE utilise la commande `autorun` du package du même nom.

Pour en savoir plus

Les pages de manuel

Automount : autofs(5), autofs(8), automount(8), auto.master(5)

Supermount : supermount(8)

KDE/autorun : autorun(1)

Howto

Automount mini-Howto

Internet

How to automate auto mounting/unmounting of USB devices in Linux
http://floatingsun.net/articles/howtos/howto-usb-automount.pdf

ATELIERS



Tâche 1 : Créer quatre partitions 5mn
Tâche 2 : Gérer des FS XFS 10mn
Tâche 3: Gérer un FS ReiserFS 10mn
Tâche 4 : Gérer des FS Ext2/Ext3 ..... 20mn
Tâche 5 : Mettre en place un serveur NFS 10mn
Tâche 6 : Créer un disque en RAM 5mn
Tâche 7 : Le FS de type tmpfs 5mn
Tâche 8 : Le FS UnionFS 10mn
Tâche 9 : Créer un FS dans un fichier 5mn
Tâche 10 : Créer une image iso 10mn
Tâche 11 : Monter automatiquement une clé USB avec Udev 10mn

ATTENTION : Les exercices suivants nécessitent un noyau qui supporte les FS ReiserFS, XFS et UnionFS en plus d'Ext2/Ext3 et iso9660.

Tâche 1 : Créer quatre partitions

Remarque : Si les partitions existent déjà (elles ont été créées dans les exercices précédents), il suffit de ne plus y accéder (umount, …) et de modifier leur tag (83).

```
[root@linux1 ~]# fdisk /dev/hda

The number of cylinders for this disk is set to 4111.
There is nothing wrong with that, but this is larger than 1024,
and could in certain setups cause problems with:
1) software that runs at boot time (e.g., old versions of LILO)
2) booting and partitioning software from other OSs
   (e.g., DOS FDISK, OS/2 FDISK)

Command (m for help): p

Disk /dev/hda: 33.8 GB, 33820286976 bytes
255 heads, 63 sectors/track, 4111 cylinders
Units = cylinders of 16065 * 512 = 8225280 bytes

   Device Boot      Start         End      Blocks   Id  System
/dev/hda1   *           1          13      104391   83  Linux
/dev/hda2              14        1288    10241437+  83  Linux
/dev/hda3            1289        1353      522112+  83  Linux
/dev/hda4            1354        4111    22153635    5  Extended
/dev/hda5            1354        1418      522081   82  Linux swap

Command (m for help): n
First cylinder (1419-4111, default 1419):
Using default value 1419
Last cylinder or +size or +sizeM or +sizeK (1419-4111, default 4111): +512M
```

On crée de la même manière trois autres partitions (7, 8 et 9).

```
Command (m for help): w
The partition table has been altered!

Calling ioctl() to re-read partition table.

WARNING: Re-reading the partition table failed with error 16: Device or resource
busy.
The kernel still uses the old table.
The new table will be used at the next reboot.
Syncing disks.
[root@linux1 ~]# partprobe
```

Tâche 2 : Gérer des FS XFS

1. Récupérer les utilitaires XFS (on les télécharge et on les compile).

```
[root@linux1 ~]# wget
'ftp://oss.sgi.com/projects/xfs/download/cmd_tars/xfsprogs_2.8.18-1.tar.gz'
...
    [                              <=>     ] 979,056       26.43K/s

16:11:09 (24.70 KB/s) - `xfsprogs_2.8.18-1.tar.gz' saved [979,056]
[root@linux1 ~]# tar xvzf xfsprogs_2.8.18-1.tar.gz
...
xfsprogs-2.8.18/build/rpm/Makefile
xfsprogs-2.8.18/build/rpm/macros.template
xfsprogs-2.8.18/build/rpm/xfsprogs.spec.in
xfsprogs-2.8.18/build/rpm/rpm-2.rc.template
[root@linux1 ~]# cd xfsprogs-2.8.18
[root@linux1 xfsprogs-2.8.18]# make
=== doc ===
/bin/gzip --best -c < CHANGES > CHANGES.gz
=== po ===
/usr/bin/msgfmt -o pl.mo pl.po
=== debian ===
gmake[2]: Nothing to be done for `default'.
=== build ===
gmake[2]: Nothing to be done for `default'.
make[1]: Leaving directory `/root/xfsprogs-2.8.18'
[root@linux1 xfsprogs-2.8.18]# make install
=== build ===
gmake[1]: Nothing to be done for `install'.
./install-sh -o root -g root -m 755 -d /usr/share/doc/xfsprogs
./install-sh -o root -g root -m 644 README /usr/share/doc/xfsprogs
[root@linux1 xfsprogs-2.8.18]# man -k xfs
xfs: nothing appropriate
[root@linux1 xfsprogs-2.8.18]# makewhatis
[root@linux1 xfsprogs-2.8.18]# man -k xfs
attr                (1)  - extended attributes on XFS filesystem objects
fsck.xfs [fsck]     (8)  - do nothing, successfully
mkfs.xfs [mkfs]     (8)  - construct an XFS filesystem
xfs                 (5)  - layout of the XFS filesystem
xfs_admin           (8)  - change parameters of an XFS filesystem
```

```
xfs_bmap              (8)  - print block mapping for an XFS file
xfs_check             (8)  - check XFS filesystem consistency
xfs_copy              (8)  - copy the contents of an XFS filesystem
xfs_db                (8)  - debug an XFS filesystem
xfs_freeze            (8)  - suspend access to an XFS filesystem
xfs_growfs            (8)  - expand an XFS filesystem
xfs_info [xfs_growfs] (8)  - expand an XFS filesystem
xfs_io                (8)  - debug the I/O path of an XFS filesystem
xfs_logprint          (8)  - print the log of an XFS filesystem
xfs_mkfile            (8)  - create an XFS file
xfs_ncheck            (8)  - generate pathnames from i-numbers for XFS
xfs_quota             (8)  - manage use of quota on XFS filesystems
xfs_repair            (8)  - repair an XFS filesystem
xfs_rtcp              (8)  - XFS realtime copy command
xqmstats              (8)  - Display XFS quota manager statistics from /proc
[root@linux1 xfsprogs-2.8.18]# cd
```

2. Créer un FS XFS.

```
[root@linux1 ~]# mkfs -t xfs /dev/hda6
mkfs.xfs: /dev/hda6 appears to contain an existing filesystem (reiserfs).
mkfs.xfs: Use the -f option to force overwrite.
[root@linux1 ~]# mkfs -t xfs -f /dev/hda6
meta-data=/dev/hda6              isize=256    agcount=8, agsize=15813 blks
         =                       sectsz=512   attr=0
data     =                       bsize=4096   blocks=126504, imaxpct=25
         =                       sunit=0      swidth=0 blks, unwritten=1
naming   =version 2              bsize=4096
log      =internal log           bsize=4096   blocks=1200, version=1
         =                       sectsz=512   sunit=0 blks
realtime =none                   extsz=4096   blocks=0, rtextents=0
```

3. Monter le FS, on liste les FS montés.

```
[root@linux1 ~]# mkdir /mnt/disk
[root@linux1 ~]# mount -t xfs /dev/hda6 /mnt/disk
[root@linux1 ~]# df -Th
Filesystem    Type    Size  Used Avail Use% Mounted on
/dev/hda2     ext3    9.7G  7.9G  1.4G  86% /
/dev/hda1     ext3     99M   21M   74M  22% /boot
none          tmpfs   220M     0  220M   0% /dev/shm
/dev/hda5     ext3    494M   11M  458M   3% /home
/dev/hda6      xfs    490M  144K  490M   1% /mnt/disk
[root@linux1 ~]# mount
/dev/hda2 on / type ext3 (rw)
none on /proc type proc (rw)
none on /sys type sysfs (rw)
none on /dev/pts type devpts (rw,gid=5,mode=620)
usbfs on /proc/bus/usb type usbfs (rw)
/dev/hda1 on /boot type ext3 (rw)
none on /dev/shm type tmpfs (rw)
/dev/hda5 on /home type ext3 (rw)
none on /proc/sys/fs/binfmt_misc type binfmt_misc (rw)
sunrpc on /var/lib/nfs/rpc_pipefs type rpc_pipefs (rw)
/dev/hda6 on /mnt/disk type xfs (rw)
[root@linux1 ~]#
```

4. Afficher des informations sur le FS et on le démonte.

```
[root@linux1 ~]# xfs_info /dev/hda6
meta-data=/dev/hda6              isize=256    agcount=8, agsize=15813 blks
         =                       sectsz=512   attr=0
data     =                       bsize=4096   blocks=126504, imaxpct=25
         =                       sunit=0      swidth=0 blks, unwritten=1
naming   =version 2              bsize=4096
log      =internal               bsize=4096   blocks=1200, version=1
         =                       sectsz=512   sunit=0 blks
realtime =none                   extsz=4096   blocks=0, rtextents=0
[root@linux1 ~]# umount /mnt/disk
[root@linux1 ~]# xfs_info /dev/hda6
xfs_info: /dev/hda6 is not a mounted XFS filesystem
[root@linux1 ~]#
```

Remarque : Beaucoup d'opérations sur XFS nécessitent qu'il soit monté comme par exemple son agrandissement ou sa sauvegarde.

5. Vérifier et réparer le FS.

```
[root@linux1 ~]# fsck -t xfs /dev/hda6
fsck 1.35 (28-Feb-2004)
If you wish to check the consistency of an XFS filesystem or
repair a damaged filesystem, see xfs_check(8) and xfs_repair(8).
[root@linux1 ~]# xfs_check /dev/hda6
[root@linux1 ~]# xfs_repair /dev/hda6
        - creating 2 worker thread(s)
Phase 1 - find and verify superblock...
        - reporting progress in intervals of 15 minutes
Phase 2 - using internal log
        - zero log...
        - scan filesystem freespace and inode maps...
        - 16:25:05: scanning filesystem freespace - 8 of 8 allocation groups
done
        - found root inode chunk
Phase 3 - for each AG...
…
Phase 4 - check for duplicate blocks
...
Phase 5 - rebuild AG headers and trees...
        - 16:25:05: rebuild AG headers and trees - 8 of 8 allocation groups done
        - reset superblock...
Phase 6 - check inode connectivity...
        - resetting contents of realtime bitmap and summary inodes
        - ensuring existence of lost+found directory
        - traversing filesystem starting at / ...
        - 16:25:05: traversing filesystem - 8 of 8 allocation groups done
        - traversal finished ...
        - traversing all unattached subtrees ...
        - traversals finished ...
        - moving disconnected inodes to lost+found ...
Phase 7 - verify and correct link counts...
        - 16:25:05: verify and correct link counts - 64 of 64 inodes done
done
[root@linux1 ~]#
```

Tâche 3 : Gérer un système de fichiers ReiserFS

1. Récupérer et installer les utilitaires ReiserFS.

```
[root@linux1 ~]# wget 'ftp://ftp.namesys.com/pub/reiserfsprogs/reiserfsprogs-
3.6.19.tar.gz'
[root@linux1 ~]# tar xvzf reiserfsprogs-3.6.19.tar.gz
...
reiserfsprogs-3.6.19/mkreiserfs/mkreiserfs.8
reiserfsprogs-3.6.19/mkreiserfs/mkreiserfs.c
[root@linux1 ~]# cd reiserfsprogs-3.6.19
[root@linux1 reiserfsprogs-3.6.19]# ./configure

config.status: creating tune/Makefile
config.status: creating include/config.h
config.status: executing depfiles commands
...
Type 'make' to compile reiserfsprogs
[root@linux1 reiserfsprogs-3.6.19]# make
...
make[1]: Nothing to be done for `all-am'.
make[1]: Leaving directory `/root/reiserfsprogs-3.6.19'
[root@linux1 reiserfsprogs-3.6.19]# make install
...
make[2]: Nothing to be done for `install-data-am'.
make[2]: Leaving directory `/root/reiserfsprogs-3.6.19'
make[1]: Leaving directory `/root/reiserfsprogs-3.6.19'
[root@linux1 reiserfsprogs-3.6.19]# cd

[root@linux1 ~]# man -k reiserfs
reiserfs: nothing appropriate
[root@linux1 ~]# makewhatis-w
[root@linux1 ~]# man -k reiserfs
debugreiserfs         (8)  - The debugging tool for the ReiserFS filesystem
mkreiserfs            (8)  - The create tool for the Linux ReiserFS filesystem
reiserfsck            (8)  - The checking tool for the ReiserFS filesystem
reiserfstune          (8)  - The tunning tool for the ReiserFS filesystem
resize_reiserfs       (8)  - resizer tool for the ReiserFS filesystem
[root@linux1 ~]# man mkreiserfs
MKREISERFS(8)                                                MKREISERFS(8)

NAME
       mkreiserfs - The create tool for the Linux ReiserFS filesystem.

SYNOPSIS
       mkreiserfs  [ -dfV ] [ -b | --block-size N ] [ -h | --hash HASH ] [ -u
       | --uuid UUID ] [ -l | --label LABEL ] [ --format  FORMAT  ]  [  -q  |
       --quiet  ]  [ -j | --journal-device FILE ] [ -s | --journal-size N ] [
       -o | --journal-offset N ] [ -t | --transaction-max-size N  ]  [  -B  |
       --badblocks file ]  device [ filesystem-size ]
...
```

2. Créer un FS ReiserFS, on utilise l'algorithme tea permettant de gérer énormément de fichiers par répertoire.

```
[root@linux1 ~]# mkreiserfs -h tea /dev/hda6
mkreiserfs 3.6.19 (2003 www.namesys.com)
```

```
A pair of credits:
Joshua Macdonald wrote the first draft of the transaction manager. Yuri Rupasov
did testing and benchmarking, plus he invented the r5 hash (also used by the
dcache code). Yura Rupasov, Anatoly Pinchuk, Igor Krasheninnikov, Grigory
Zaigralin, Mikhail Gilula, Igor Zagorovsky, Roman Pozlevich, Konstantin
Shvachko, and Joshua MacDonald are former contributors to the project.

Oleg Drokin was the debugger for V3 during most of the time that V4 was under
development, and was quite skilled and fast at it. He wrote the large write
optimization of V3.


Guessing about desired format.. Kernel 2.6.17 is running.
Format 3.6 with standard journal
Count of blocks on the device: 126496
Number of blocks consumed by mkreiserfs formatting process: 8215
Blocksize: 4096
Hash function used to sort names: "tea"
Journal Size 8193 blocks (first block 18)
Journal Max transaction length 1024
inode generation number: 0
UUID: e6ed3359-8888-40b1-b8ee-138018c6d818
ATTENTION: YOU SHOULD REBOOT AFTER FDISK!
        ALL DATA WILL BE LOST ON '/dev/hda6'!
Continue (y/n):y
Initializing journal - 0%....20%....40%....60%....80%....100%
Syncing..ok

Tell your friends to use a kernel based on 2.4.18 or later, and especially not a
kernel based on 2.4.9, when you use reiserFS. Have fun.

ReiserFS is successfully created on /dev/hda6.
```

3. Visualiser les caractéristiques du FS.

```
[root@linux1 ~]# reiserfstune /dev/hda6
reiserfstune: Journal device has not been specified. Assuming journal is on the
main device (/dev/hda6).

Current parameters:

Filesystem state: consistent

Reiserfs super block in block 16 on 0x306 of format 3.6 with standard journal
Count of blocks on the device: 126496
Number of bitmaps: 4
Blocksize: 4096
Free blocks (count of blocks - used [journal, bitmaps, data, reserved] blocks):
118281
Root block: 8211
Filesystem is clean
Tree height: 2
Hash function used to sort names: "tea"
Objectid map size 2, max 972
```

```
Journal parameters:
        Device [0x0]
        Magic [0x6f4c3e8]
        Size 8193 blocks (including 1 for journal header) (first block 18)
        Max transaction length 1024 blocks
        Max batch size 900 blocks
        Max commit age 30
Blocks reserved by journal: 0
Fs state field: 0x0:
sb_version: 2
inode generation number: 0
UUID: e6ed3359-8888-40b1-b8ee-138018c6d818
LABEL:
Set flags in SB:
        ATTRIBUTES CLEAN
```

4. Monter le FS.

```
[root@linux1 ~]# mount -t reiserfs /dev/hda6 /mnt/disk
[root@linux1 ~]# df -Th
Filesystem    Type    Size  Used Avail Use% Mounted on
/dev/hda2     ext3    9.7G  7.8G  1.4G  86% /
/dev/hda1     ext3     99M   21M   74M  22% /boot
none          tmpfs   220M     0  220M   0% /dev/shm
/dev/hda5     ext3    494M   11M  458M   3% /home
/dev/hda6 reiserfs    495M   33M  463M   7% /mnt/disk
[root@linux1 ~]# mount
/dev/hda2 on / type ext3 (rw)
none on /proc type proc (rw)
none on /sys type sysfs (rw)
none on /dev/pts type devpts (rw,gid=5,mode=620)
usbfs on /proc/bus/usb type usbfs (rw)
/dev/hda1 on /boot type ext3 (rw)
none on /dev/shm type tmpfs (rw)
/dev/hda5 on /home type ext3 (rw)
none on /proc/sys/fs/binfmt_misc type binfmt_misc (rw)
sunrpc on /var/lib/nfs/rpc_pipefs type rpc_pipefs (rw)
/dev/hda6 on /mnt/disk type reiserfs (rw)
```

5. Démonter le FS et le vérifier.

```
[root@linux1 ~]# umount /mnt/disk
[root@linux1 ~]# reiserfsck /dev/hda6
reiserfsck 3.6.19 (2003 www.namesys.com)

************************************************************
** If you are using the latest reiserfsprogs and  it fails **
** please  email bug reports to reiserfs-list@namesys.com, **
** providing  as  much  information  as  possible --  your **
** hardware,  kernel,  patches,  settings,  all reiserfsck **
** messages  (including version),  the reiserfsck logfile, **
** check  the  syslog file  for  any  related information. **
** If you would like advice on using this program, support **
** is available  for $25 at  www.namesys.com/support.html. **
************************************************************

Will read-only check consistency of the filesystem on /dev/hda6
```

```
Will put log info to 'stdout'

Do you want to run this program?[N/Yes] (note need to type Yes if you do):Yes
###########
reiserfsck --check started at Fri Jan  5 15:55:58 2007
###########
Replaying journal..
Reiserfs journal '/dev/hda6' in blocks [18..8211]: 0 transactions replayed
Checking internal tree..finished
Comparing bitmaps..finished
Checking Semantic tree:
finished
No corruptions found
There are on the filesystem:
        Leaves 1
        Internal nodes 0
        Directories 2
        Other files 0
        Data block pointers 0 (0 of them are zero)
        Safe links 0
###########
reiserfsck finished at Fri Jan  5 15:55:59 2007
###########
[root@linux1 ~]#
```

6. Créer un FS ReiserFS sans option, on affiche le Super-Bloc.

```
[root@linux1 ~]# mkreiserfs -q /dev/hda6
mkreiserfs 3.6.19 (2003 www.namesys.com)

A pair of credits:
Alexander Lyamin keeps our hardware running, and was very generous to our
project in many little ways.

Alexander Zarochentcev (zam) wrote the high low priority locking code, online
resizer for V3 and V4, online repacker for V4, block allocation code, and major
parts of the flush code, and maintains the transaction manager code. We give
him the stuff that we know will be hard to debug, or needs to be very cleanly
structured.


[root@linux1 ~]# debugreiserfs /dev/hda6
debugreiserfs 3.6.19 (2003 www.namesys.com)


Filesystem state: consistent

Reiserfs super block in block 16 on 0x309 of format 3.6 with standard journal
Count of blocks on the device: 126496
Number of bitmaps: 4
Blocksize: 4096
Free blocks (count of blocks - used [journal, bitmaps, data, reserved] blocks):
118281
Root block: 8211
Filesystem is clean
```

```
Tree height: 2
Hash function used to sort names: "r5"
Objectid map size 2, max 972
Journal parameters:
 Device [0x0]
 Magic [0x227b1b69]
 Size 8193 blocks (including 1 for journal header) (first block 18)
 Max transaction length 1024 blocks
 Max batch size 900 blocks
 Max commit age 30
Blocks reserved by journal: 0
Fs state field: 0x0:
sb_version: 2
inode generation number: 0
UUID: 6a523d4a-8e38-4b3a-a55a-d80b0a2f7670
LABEL:
Set flags in SB:
 ATTRIBUTES CLEAN
[root@linux1 ~]#
```

Tâche 4 : Gérer des systèmes de fichiers Ext2/Ext3

1. Lister la documentation disponible.

```
[root@linux1 ~]# man -k ext3
debugfs              (8)  - ext2/ext3 file system debugger
dumpe2fs             (8)  - dump ext2/ext3 filesystem information
e2fsck               (8)  - check a Linux ext2/ext3 file system
e2fsck [fsck]        (8)  - check a Linux ext2/ext3 file system
e2image              (8)  - Save critical ext2/ext3 filesystem data to a file
e2label              (8)  - Change the label on an ext2/ext3 filesystem
filesystems [fs]     (5)  - Linux filesystem types: minix, ext, ext2, ext3, xia,
msdos, umsdos, vfat, proc, nfs, iso9660, hpfs, sysv, smb, ncpfs
mke2fs               (8)  - create an ext2/ext3 filesystem
mke2fs [mkfs]        (8)  - create an ext2/ext3 filesystem
resize2fs            (8)  - ext2/ext3 file system resizer
tune2fs              (8)  - adjust tunable filesystem parameters on ext2/ext3
filesystems
```

2. Créer un FS Ext2 avec des blocs de 4ko.

```
[root@linux1 ~]# mkfs -b 4096 /dev/hda6
mke2fs 1.35 (28-Feb-2004)
Filesystem label=
OS type: Linux
Block size=4096 (log=2)
Fragment size=4096 (log=2)
126592 inodes, 126504 blocks
6325 blocks (5.00%) reserved for the super user
First data block=0
Maximum filesystem blocks=130023424
4 block groups
32768 blocks per group, 32768 fragments per group
31648 inodes per group
Superblock backups stored on blocks:
```

```
        32768, 98304

Writing inode tables: done
Writing superblocks and filesystem accounting information: done

This filesystem will be automatically checked every 27 mounts or
180 days, whichever comes first.  Use tune2fs -c or -i to override.
[root@linux1 ~]#
```

3. Créer un journal, le FS devient donc un Ext3.

```
[root@linux1 ~]# tune2fs -j /dev/hda6
tune2fs 1.35 (28-Feb-2004)
Creating journal inode: done
This filesystem will be automatically checked every 27 mounts or
180 days, whichever comes first.  Use tune2fs -c or -i to override.
[root@linux1 ~]#
```

4. Visualiser les caractéristiques du FS.

```
[root@linux1 ~]# tune2fs -l /dev/hda6
tune2fs 1.35 (28-Feb-2004)
Filesystem volume name:   <none>
Last mounted on:          <not available>
Filesystem UUID:          9f9cbcd0-a481-479c-bd93-7d845aa67037
Filesystem magic number:  0xEF53
Filesystem revision #:    1 (dynamic)
Filesystem features:      has_journal resize_inode filetype sparse_super
large_file
Default mount options:    (none)
Filesystem state:         clean
Errors behavior:          Continue
Filesystem OS type:       Linux
Inode count:              126592
Block count:              126504
Reserved block count:     6325
Free blocks:              118336
Free inodes:              126581
First block:              0
Block size:               4096
Fragment size:            4096
Reserved GDT blocks:      30
Blocks per group:         32768
Fragments per group:      32768
Inodes per group:         31648
Inode blocks per group:   989
Filesystem created:       Fri Jan  5 17:42:17 2007
Last mount time:          Fri Jan  5 17:51:25 2007
Last write time:          Fri Jan  5 18:02:41 2007
Mount count:              0
Maximum mount count:      27
Last checked:             Fri Jan  5 18:02:41 2007
Check interval:           15552000 (6 months)
Next check after:         Wed Jul  4 19:02:41 2007
Reserved blocks uid:      0 (user root)
Reserved blocks gid:      0 (group root)
First inode:              11
```

```
Inode size:               128
Journal inode:            8
Default directory hash:   tea
Directory Hash Seed:      a204c21e-fa69-4593-9a9f-201496f4bc7d
Journal backup:           inode blocks
[root@linux1 ~]#
```

5. Monter le FS sans activer le journal.

```
[root@linux1 ~]# mount -t ext2 /dev/hda6 /mnt/disk
[root@linux1 ~]# df -Th /dev/hda6
Filesystem    Type    Size  Used Avail Use% Mounted on
/dev/hda6     ext2    479M   17M  438M   4% /mnt/disk
```

6. Remonter le FS en tant que Ext3, on active donc la journalisation.

```
[root@linux1 ~]# umount /mnt/disk
[root@linux1 ~]# mount -t ext3 /dev/hda6 /mnt/disk/
[root@linux1 ~]# df -Th /dev/hda6
Filesystem    Type    Size  Used Avail Use% Mounted on
/dev/hda6     ext3    479M   17M  438M   4% /mnt/disk
[root@linux1 ~]#
```

7. Vérifier un FS Ext2/Ext3. Il est vivement conseillé qu'il soit démonté.

```
[root@linux1 ~]# umount /mnt/disk/
[root@linux1 ~]# fsck /dev/hda6
fsck 1.35 (28-Feb-2004)
e2fsck 1.35 (28-Feb-2004)
/dev/hda6: clean, 11/126592 files, 8168/126504 blocks
[root@linux1 ~]# fsck -f /dev/hda6    # vérification approfondie
fsck 1.35 (28-Feb-2004)
e2fsck 1.35 (28-Feb-2004)
Pass 1: Checking inodes, blocks, and sizes
Pass 2: Checking directory structure
Pass 3: Checking directory connectivity
Pass 4: Checking reference counts
Pass 5: Checking group summary information
/dev/hda6: 11/126592 files (9.1% non-contiguous), 8168/126504 blocks
[root@linux1 ~]#
```

8. Créer FS Ext3 avec un journal externe (la taille des blocs doit être identique).

```
[root@linux1 ~]# mkfs -b 4096 -q /dev/hda6
[root@linux1 ~]# mke2fs -b 4096 -O journal_dev /dev/hda7
mke2fs 1.35 (28-Feb-2004)
Filesystem label=
OS type: Linux
Block size=4096 (log=2)
Fragment size=4096 (log=2)
0 inodes, 126504 blocks
0 blocks (0.00%) reserved for the super user
First data block=0
0 block group
32768 blocks per group, 32768 fragments per group
0 inodes per group
Superblock backups stored on blocks:
```

```
Zeroing journal device: done
[root@linux1 ~]# tune2fs -j -J device=/dev/hda7 /dev/hda6
tune2fs 1.35 (28-Feb-2004)
Creating journal on device /dev/hda7: done
This filesystem will be automatically checked every 32 mounts or
180 days, whichever comes first.  Use tune2fs -c or -i to override.
[root@linux1 ~]#
```

9. Visualiser la structure du FS.

```
[root@linux1 ~]# dumpe2fs /dev/hda6
dumpe2fs 1.35 (28-Feb-2004)
Filesystem volume name:   <none>
Last mounted on:          <not available>
Filesystem UUID:          89769ec2-dc0c-4002-a47b-ff56bfd14df2
Filesystem magic number:  0xEF53
Filesystem revision #:    1 (dynamic)
Filesystem features:      has_journal resize_inode filetype sparse_super
large_file
Default mount options:    (none)
Filesystem state:         clean
Errors behavior:          Continue
Filesystem OS type:       Linux
Inode count:              126592
Block count:              126504
Reserved block count:     6325
Free blocks:              122438
Free inodes:              126581
First block:              0
Block size:               4096
Fragment size:            4096
Reserved GDT blocks:      30
Blocks per group:         32768
Fragments per group:      32768
Inodes per group:         31648
Inode blocks per group:   989
Filesystem created:       Fri Jan  5 18:13:56 2007
Last mount time:          n/a
Last write time:          Fri Jan  5 18:14:34 2007
Mount count:              0
Maximum mount count:      32
Last checked:             Fri Jan  5 18:13:56 2007
Check interval:           15552000 (6 months)
Next check after:         Wed Jul  4 19:13:56 2007
Reserved blocks uid:      0 (user root)
Reserved blocks gid:      0 (group root)
First inode:              11
Inode size:               128
Journal UUID:             72588764-9da8-4e43-a71a-bd2f01a6c0bf
Journal device:           0x0307
Default directory hash:   tea
Directory Hash Seed:      5ae164f8-ba19-4778-a3f7-385c18dd9d78
```

```
Group 0: (Blocks 0-32767)
  Primary superblock at 0, Group descriptors at 1-1
  Block bitmap at 32 (+32), Inode bitmap at 33 (+33)
  Inode table at 34-1022 (+34)
  31739 free blocks, 31637 free inodes, 2 directories
  Free blocks: 1029-32767
  Free inodes: 12-31648
Group 1: (Blocks 32768-65535)
  Backup superblock at 32768, Group descriptors at 32769-32769
  Block bitmap at 32800 (+32), Inode bitmap at 32801 (+33)
  Inode table at 32802-33790 (+34)
  31745 free blocks, 31648 free inodes, 0 directories
  Free blocks: 33791-65535
  Free inodes: 31649-63296
Group 2: (Blocks 65536-98303)
  Block bitmap at 65536 (+0), Inode bitmap at 65537 (+1)
  Inode table at 65538-66526 (+2)
  31777 free blocks, 31648 free inodes, 0 directories
  Free blocks: 66527-98303
  Free inodes: 63297-94944
Group 3: (Blocks 98304-126503)
  Backup superblock at 98304, Group descriptors at 98305-98305
  Block bitmap at 98336 (+32), Inode bitmap at 98337 (+33)
  Inode table at 98338-99326 (+34)
  27177 free blocks, 31648 free inodes, 0 directories
  Free blocks: 99327-126503
  Free inodes: 94945-126592
[root@linux1 ~]#
```

Remarque : La ligne « Journal device: 0x0307 » indique que le journal est sur /dev/hda7 (Majeur 3, mineur 7)

10. Monter le FS en activant le journal.

```
[root@linux1 ~]# mount -t ext3 /dev/hda6 /mnt/disk
```

11. Remonter le FS en journalisant également les données. On le démonte.

```
[root@linux1 ~]# umount /mnt/disk
[root@linux1 ~]# mount -t ext3 -o data=journal /dev/hda6 /mnt/disk
[root@linux1 ~]# mount | grep /mnt/disk
/dev/hda6 on /mnt/disk type ext3 (rw,data=journal)
[root@linux1 ~]# umount /mnt/disk
```

13. Créer un FS en silence sans zone réservée à l'administrateur.

```
[root@linux1 ~]# mkfs -q -m 0 /dev/hda6
[root@linux1 ~]# tune2fs -l /dev/hda6 | grep -i 'reserved.*count'
Reserved block count:     0
```

14. Modifier le nombre maximum de montages sans vérification (fsck).

```
[root@linux1 ~]# mkfs -q /dev/hda6
[root@linux1 ~]# tune2fs -l /dev/hda6 |grep -i 'mount count'
Mount count:              0
Maximum mount count:      37
[root@linux1 ~]# tune2fs -c 5 /dev/hda6
tune2fs 1.35 (28-Feb-2004)
Setting maximal mount count to 5
```

```
[root@linux1 ~]# tune2fs -l /dev/hda6 |grep -i 'mount count'
Mount count:              0
Maximum mount count:      5
[root@linux1 ~]# mount /dev/hda6 /mnt/disk
[root@linux1 ~]# umount /mnt/disk
[root@linux1 ~]# tune2fs -l /dev/hda6 |grep -i 'mount count'
Mount count:              1
Maximum mount count:      5
```

15. Créer un FS, supprimer par erreur son Super-Bloc, le réparer, le monter et le démonter.

```
[root@linux1 ext2fs]# mke2fs -q /dev/hda6
[root@linux1 ext2fs]# dumpe2fs /dev/hda6 | grep superblock | head -3
dumpe2fs 1.35 (28-Feb-2004)
  Primary superblock at 1, Group descriptors at 2-3
  Backup superblock at 8193, Group descriptors at 8194-8195
  Backup superblock at 24577, Group descriptors at 24578-24579
[root@linux1 ext2fs]# dd if=/dev/zero of=/dev/hda6 bs=4k count=1
1+0 records in
1+0 records out
[root@linux1 ext2fs]# mount -t ext2 /dev/hda6 /mnt
mount: wrong fs type, bad option, bad superblock on /dev/hda6,
       or too many mounted file systems
[root@linux1 ext2fs]# e2fsck -b 8193 /dev/hda6
e2fsck 1.35 (28-Feb-2004)
/dev/hda6 was not cleanly unmounted, check forced.
Pass 1: Checking inodes, blocks, and sizes
Pass 2: Checking directory structure
Pass 3: Checking directory connectivity
Pass 4: Checking reference counts
Pass 5: Checking group summary information

/dev/hda6: ***** FILE SYSTEM WAS MODIFIED *****
/dev/hda6: 11/126976 files (9.1% non-contiguous), 18342/506016 blocks
[root@linux1 ext2fs]# mount -t ext2 /dev/hda6 /mnt
[root@linux1 ext2fs]# umount /mnt
```

16. Explorer/dépanner un FS.

```
[root@linux1 ext2fs]# mount -t ext2 /dev/hda6 /mnt
[root@linux1 ext2fs]# cal > /mnt/toto
[root@linux1 ext2fs]# ls -i /mnt/toto
12 /mnt/toto
[root@linux1 ext2fs]# umount /mnt
[root@linux1 ext2fs]# debugfs /dev/hda6
debugfs 1.35 (28-Feb-2004)
debugfs:  pwd
[pwd]   INODE:      2  PATH: /
[root]  INODE:      2  PATH: /
debugfs:  ls
 2  (12) .    2  (12) ..    11  (20) lost+found    12  (980) toto
debugfs:  stat <12>
Inode: 12   Type: regular    Mode:  0644   Flags: 0x0   Generation: 1199529607
User:     0   Group:     0   Size: 137
File ACL: 0    Directory ACL: 0
Links: 1   Blockcount: 2
```

```
Fragment:  Address: 0    Number: 0    Size: 0
ctime: 0x45d87262 -- Sun Feb 18 16:36:02 2007
atime: 0x45d87262 -- Sun Feb 18 16:36:02 2007
mtime: 0x45d87262 -- Sun Feb 18 16:36:02 2007
BLOCKS:
(0):4609
TOTAL: 1

debugfs:  testb 4609
Block 4609 marked in use
debugfs:  testb 5000
Block 5000 not in use
debugfs:  ncheck
ncheck: Usage: ncheck <inode number> ...
debugfs:  ncheck 12
Inode Pathname
12 /toto
debugfs:  quit
[root@linux1 ext2fs]#
```

Remarques :

1) La sous-commande `mi` permet de modifier une inode. Il faut ouvrir le FS en écriture (option –w). Le manuel et la sous commande `help` liste l'ensemble des opérations qui peuvent être effectuées.
2) Un fichier peut être spécifié par son nom ou son numéro d'inode entre inférieur et supérieur, par exemple : `stat <12>`.
3) La commande standard `stat` affiche les attributs d'un fichier de manière similaires à la sous commande stat de debugfs.

Tâche 5 : Mettre en place un serveur NFS

Remarque : On travaille en binôme ou tout seul.

1. Exporter l'arborescence /var/ftp en lecture seule pour tout le monde.

```
[root@linux1 ~]# vi /etc/exports
/var/ftp *(ro)
```

2. Activer le service NFS, surveiller la présence des démons.

```
[root@linux1 ~]# /etc/init.d/nfs start
Démarrage des services NFS :                               [  OK  ]
Début du quota NFS :                                       [  OK  ]
Démarrage du dÃ©mon NFS :                                  [  OK  ]
Démarrage de NFS mountd :                                  [  OK  ]
[root@linux1 ~]# ps -e |grep -e mountd -e nfsd
26346 ?        00:00:00 nfsd4
26347 ?        00:00:00 nfsd
26350 ?        00:00:00 nfsd
26351 ?        00:00:00 nfsd
26352 ?        00:00:00 nfsd
26353 ?        00:00:00 nfsd
26354 ?        00:00:00 nfsd
26355 ?        00:00:00 nfsd
26356 ?        00:00:00 nfsd
26360 ?        00:00:00 rpc.mountd
[root@linux1 ~]#
```

3. A partir du client, visualiser la présence des services RPC du serveur.

```
[root@linux1 ~]# rpcinfo -p localhost | grep -e mount -e nfs -e portmap
    100000    2   tcp    111  portmapper
    100000    2   udp    111  portmapper
    100003    2   udp   2049  nfs
    100003    3   udp   2049  nfs
    100003    4   udp   2049  nfs
    100003    2   tcp   2049  nfs
    100003    3   tcp   2049  nfs
    100003    4   tcp   2049  nfs
    100005    1   udp    672  mountd
    100005    1   tcp    675  mountd
    100005    2   udp    672  mountd
    100005    2   tcp    675  mountd
    100005    3   udp    672  mountd
    100005    3   tcp    675  mountd
[root@linux1 ~]#
```

4. Lister les ressources exportées du serveur.

```
[root@linux1 ~]# showmount -e localhost
Export list for localhost:
/var/ftp *
[root@linux1 ~]#
```

5. Monter la ressource et y accéder.

```
[root@linux1 ~]# mount -t nfs localhost:/var/ftp /mnt
[root@linux1 ~]# ls /mnt
pub
[root@linux1 ~]# mount |grep -i nfs
sunrpc on /var/lib/nfs/rpc_pipefs type rpc_pipefs (rw)
nfsd on /proc/fs/nfsd type nfsd (rw)
localhost:/var/ftp on /mnt type nfs (rw,addr=127.0.0.1)
[root@linux1 ~]# df -Th
Filesystem    Type    Size  Used Avail Use% Mounted on
…
localhost:/var/ftp
              nfs     9.7G  6.5G  2.7G  71% /mnt
[root@linux1 ~]#
```

6. Visualiser les messages du noyau.

```
[root@linux1 ~]# tail /var/log/messages
Feb 18 19:01:07 linux1 nfs: DÃ©marrage des services NFS : succeeded
Feb 18 19:01:07 linux1 nfs: DÃ©marrage de rpc.rquotad succeeded
Feb 18 19:01:07 linux1 nfsd[26345]: nfssvc_versbits: +2 +3 +4
Feb 18 19:01:07 linux1 kernel: NFSD: Using /var/lib/nfs/v4recovery as the NFSv4
state recovery directory
Feb 18 19:01:07 linux1 kernel: NFSD: unable to find recovery directory
/var/lib/nfs/v4recovery
Feb 18 19:01:07 linux1 kernel: NFSD: starting 90-second grace period
Feb 18 19:01:07 linux1 nfs: DÃ©marrage de rpc.nfsd succeeded
Feb 18 19:01:07 linux1 nfs: DÃ©marrage de rpc.mountd succeeded
Feb 18 19:01:07 linux1 rpcidmapd: rpc.idmapd -SIGHUP succeeded
Feb 18 19:05:33 linux1 mountd[26360]: authenticated mount request from
localhost.localdomain:692 for /var/ftp (/var/ftp)
[root@linux1 ~]#
```

7. Démonter le FS.

```
[root@linux1 ~]# umount /mnt
```

8. Arrêter le service NFS.

```
[root@linux1 ~]# /etc/init.d/nfs stop
ArrÃªt de NFS mountd :                                      [  OK  ]
ArrÃªt du dÃ©mon NFS :                                       [  OK  ]
ArrÃªt des quotas NFS :                                     [  OK  ]
ArrÃªt des services NFS :                                   [  OK  ]
[root@linux1 ~]#
```

Tâche 6 : Création d'un disque en RAM

1. Gestion des ramdisk est-elle assurée ?

```
[root@linux1 ~]# dmesg | grep RAMDISK
RAMDISK driver initialized: 16 RAM disks of 16384K size 1024 blocksize
```

2. Lister les périphériques disponibles.

```
[root@linux1 ~]# ls -l /dev/ram*
lrwxrwxrwx  1 root root     4 Jan  5 12:00 /dev/ram -> ram1
brw-rw----  1 root disk 1,  0 Jan  5 12:00 /dev/ram0
brw-rw----  1 root disk 1,  1 Jan  5 12:00 /dev/ram1
…
brw-rw----  1 root disk 1,  9 Jan  5 12:00 /dev/ram9
lrwxrwxrwx  1 root root     4 Jan  5 12:00 /dev/ramdisk -> ram0
[root@linux1 ~]#
```

3. Formater un des périphériques de type ramdisk.

```
[root@linux1 ~]# mkfs /dev/ram
mke2fs 1.35 (28-Feb-2004)
…
180 days, whichever comes first.  Use tune2fs -c or -i to override.
[root@linux1 ~]# df -h /dev/ram
Filesystem            Size  Used Avail Use% Mounted on
-                     220M  152K  220M   1% /dev
```

4. Monter le FS.

```
[root@linux1 ~]# mount /dev/ram /mnt/disk
[root@linux1 ~]# df -h /mnt/disk
Filesystem            Size  Used Avail Use% Mounted on
/dev/ram1              16M  140K   15M   1% /mnt/disk
```

5. Copier des fichiers sur le FS, le démonter

```
[root@linux1 ~]# cp /bin/* /mnt/disk
[root@linux1 ~]# df -h /mnt/disk
Filesystem            Size  Used Avail Use% Mounted on
/dev/ram1              16M  8.7M  6.1M  59% /mnt/disk
[root@linux1 ~]# umount /mnt/disk
```

Tâche 7 : Le FS de type tmpfs

1. Créer un FS de type tmpfs sans option.

Sa taille maximum prend la moitié de la mémoire.

```
[root@linux1 ~]# mount -t tmpfs tmpfs /mnt/disk
[root@linux1 ~]# free
```

```
             total       used       free     shared    buffers     cached
Mem:        449152     200580     248572          0      25832     144988
-/+ buffers/cache:      29760     419392
Swap:       522104          0     522104
[root@linux1 ~]# df -Th /mnt/disk
Filesystem    Type    Size  Used Avail Use% Mounted on
tmpfs         tmpfs   220M     0  220M   0% /mnt/disk
[root@linux1 ~]# ls -ld /mnt/disk
drwxrwxrwt  2 root root 40 Jan  5 21:45 /mnt/disk
[root@linux1 ~]# umount /mnt/disk/
```

2. Créer un FS de type tmpfs en précisant sa taille maximum et les droits de la racine.

```
[root@linux1 ~]# mount -o size=128M,mode=770 -t tmpfs tmpfs /mnt/disk
[root@linux1 ~]# df -Th /mnt/disk
Filesystem    Type    Size  Used Avail Use% Mounted on
tmpfs         tmpfs   128M     0  128M   0% /mnt/disk
[root@linux1 ~]# ls -ld /mnt/disk
drwxrwx---  2 root root 40 Jan  5 21:42 /mnt/disk
[root@linux1 ~]# umount /mnt/disk
```

Tâche 8 : Le FS Unionfs

1. Charger le module qui gère UnionFS.

```
[root@linux1 ~]# modprobe unionfs
[root@linux1 ~]# lsmod |grep unionfs
unionfs                85152  0
```

2. Créer deux arborescences de fichiers.

```
[root@linux1 ~]# mkdir dir1 dir2
[root@linux1 ~]# mkdir dir1/rep1 dir1/rep2 dir2/rep1 dir2/rep2
[root@linux1 ~]# echo "Bonjour" > dir1/f1
[root@linux1 ~]# echo "Bonjour" > dir1/rep1/f1
[root@linux1 ~]# echo "Salut" > dir2/f2
[root@linux1 ~]# echo "Salut" > dir2/rep1/f13
```

3. Fusionner les deux arborescences.

```
[root@linux1 ~]# mount -t unionfs -o dirs=/root/dir1:/root/dir2=ro unionfs
/mnt/disk
[root@linux1 ~]# df -Th /mnt/disk
Filesystem    Type    Size  Used Avail Use% Mounted on
unionfs    unionfs    9.7G  8.0G  1.3G  87% /mnt/disk
```

4. Lister l'arborescence fusionnée.

```
[root@linux1 ~]# find /mnt/disk
/mnt/disk
/mnt/disk/rep1
/mnt/disk/rep1/f1
/mnt/disk/rep2
/mnt/disk/f1
/mnt/disk/f2
```

5. Lister les branches.

```
[root@linux1 ~]# unionctl /mnt/disk --list
        /root/dir1 (rw-)
        /root/dir2 (r--)
```

6. Modifier un fichier fusionné : c'est celui de la branche de gauche qui est modifié.

```
[root@linux1 ~]# echo "====" >> /mnt/disk/rep1/f1
[root@linux1 ~]# cat /root/dir1/rep1/f1
Bonjour
====
You have new mail in /var/spool/mail/root
[root@linux1 ~]# cat /root/dir2/rep1/f1
Salut
```

7. Démonter le FS.

```
[root@linux1 ~]# umount /mnt/disk
```

Tâche 9 : Créer un FS dans un fichier.

1. Créer un fichier de 100 Mo.

```
[root@linux1 ~]# dd if=/dev/zero of=/usr/GROS_FIC bs=1k count=100000
100000+0 records in
100000+0 records out
[root@linux1 ~]# ls -lh /usr/GROS_FIC
-rw-r--r--  1 root root 98M Jan  5 18:30 /usr/GROS_FIC
```

2. Formater le fichier en tant que FS Ext3.

```
[root@linux1 ~]# mkfs -t ext3 /usr/GROS_FIC
mke2fs 1.35 (28-Feb-2004)
/usr/GROS_FIC is not a block special device.
Proceed anyway? (y,n) n
[root@linux1 ~]# mkfs -t ext3 -F -q /usr/GROS_FIC
```

3. Monter le FS. Utiliser le pilote loop pour voir le fichier comme un périphérique bloc.

```
[root@linux1 ~]# mount -o loop -t ext3 /usr/GROS_FIC  /mnt/disk
[root@linux1 ~]# df -Th /mnt/disk
Filesystem    Type    Size  Used Avail Use% Mounted on
/usr/GROS_FIC ext3     95M  5.6M   85M   7% /mnt/disk
```

4. Créer des fichiers dessus.

```
[root@linux1 ~]# cp /bin/* /mnt/disk
[root@linux1 ~]# df -Th /mnt/disk
Filesystem    Type    Size  Used Avail Use% Mounted on
/usr/GROS_FIC ext3     95M   15M   76M  16% /mnt/disk
[root@linux1 ~]#
```

5. Vérifier le FS.

```
[root@linux1 ~]# umount /mnt/disk
[root@linux1 ~]# fsck /usr/GROS_FIC
fsck 1.35 (28-Feb-2004)
e2fsck 1.35 (28-Feb-2004)
/usr/GROS_FIC: clean, 106/25064 files, 17546/100000 blocks
```

Tâche 10 : Créer une image iso

1. Créer une image iso.

```
[root@linux1 ~]# mkisofs -r -o image.iso -J -V docs /usr/share/doc
Using CYRUS000.19 for  /cyrus-sasl-devel-2.1.19 (cyrus-sasl-2.1.19)
Using LIBXM000.16 for  /libxml2-2.6.16 (libxml2-python-2.6.16)
```

```
Using VALGR000.1 for  /valgrind-3.1.1 (valgrind-callgrind-0.10.1)
Using GNOME000.3 for  /gnome-keyring-manager-0.0.3 (gnome-kerberos-0.3.3)
Using OPENS000.9P1 for  /openssh-3.9p1 (openssh-askpass-3.9p1)
Using LIBXM001.16 for  /libxml2-python-2.6.16 (libxml2-devel-2.6.16)
…
 85.32% done, estimate finish Sun Feb 18 22:22:25 2007
 89.02% done, estimate finish Sun Feb 18 22:22:23 2007
 92.74% done, estimate finish Sun Feb 18 22:22:24 2007
 96.45% done, estimate finish Sun Feb 18 22:22:24 2007
Total translation table size: 0
Total rockridge attributes bytes: 1324207
Total directory bytes: 3780608
Path table size(bytes): 22190
Max brk space used a27000
134800 extents written (263 MB)
```

2. La monter, lister en partie son contenu et la démonter.

```
[root@linux1 ~]# mount -t iso9660 -o loop image.iso /mnt
[root@linux1 ~]# ls /mnt/pam-0.77/
Copyright  html  pdf  ps  rfc86.0.txt  txts
[root@linux1 ~]# umount /mnt
```

3. Ajouter des fichiers à une image iso.

On ajoute à la racine du FS le contenu du répertoire /usr/share/info.

```
[root@linux1 ~]# rpm -q dvd+rw-tools
dvd+rw-tools-5.21.4.10.8-4
[root@linux1 ~]# ls -lh image.iso
-rw-r--r--  1 root root 264M Feb 18 22:00 image.iso
[root@linux1 ~]# growisofs -M image.iso -R -J /usr/share/info
Executing 'mkisofs -C 16,134800 -M /dev/fd/3 -R -J /usr/share/info | builtin_dd
of=image.iso obs=32k seek=8425'
Rock Ridge signatures found
Using LIBC_000.GZ;1 for  /libc.info-7.gz (libc.info-6.gz)
Using LIBC_001.GZ;1 for  /libc.info-6.gz (libc.info-12.gz)
Using LIBC_002.GZ;1 for  /libc.info-12.gz (libc.info-4.gz)
…
Using M4_IN001.GZ;1 for  /m4.info-3.gz (m4.info-2.gz)
 97.28% done, estimate finish Sun Feb 18 22:32:02 2007
Total translation table size: 0
Total rockridge attributes bytes: 1362429
Total directory bytes: 3780608
Path table size(bytes): 22190
Max brk space used 86c000
143914 extents written (281 MB)
builtin_dd: 9120*2KB out @ average 13.2x1385KBps
image.iso: copying volume descriptor(s)
[root@linux1 ~]# ls -lh image.iso
-rw-r--r--  1 root root 282M Feb 18 22:32 image.iso
[root@linux1 ~]# mount -t iso9660 -o loop image.iso /mnt
[root@linux1 ~]# ls /mnt
…
[root@linux1 ~]# umount /mnt
```

4. Rechercher des graveurs.

```
[root@linux1 ~]# cdrecord -scanbus dev=ATAPI
```

```
...
```

5. Supprimer l'image.

```
[root@linux1 ~]# rm -f image.iso
```

6. Copier l'image d'un cdrom dans un fichier, la monter, la lister, la démonter et la détruire.

```
[root@linux1 ~]# dd if=/dev/cdrom  of=image.iso
395744+0 records in
395744+0 records out
[root@linux1 ~]# mount -o loop image.iso /mnt
[root@linux1 ~]# ls /mnt
....
[root@linux1 ~]# umount /mnt
[root@linux1 ~]# rm -f image.iso
[root@linux1 ~]#
```

Tâche 11 : Monter automatiquement une clé USB avec Udev

1. Créer une règle Udev qui crée un nom associé à ma clé.

```
[root@linux1 ~]# vi  /etc/udev/rules.d/20-perso.rules
BUS="scsi", SYSFS{vendor}="Intuix  ",SYMLINK="cleperso"
```

2. Créer un script qui sera déclenché chaque fois que l'on ajoute un disque.

```
[root@linux1 ~]# cd /etc/dev.d/
[root@linux1 dev.d]# mkdir block
[root@linux1 dev.d]# vi block/mount.dev
#!/bin/sh
logger -p user.warn "essai mount cle $DEVNAME , action=$ACTION"
if [ "$ACTION" = "add" -a -e /dev/cleperso ];
then
 mount -t vfat /dev/cleperso /root/cle
fi
[root@linux1 dev.d]# chmod u+x block/mount.dev
[root@linux1 dev.d]# mkdir /root/cle
[root@linux1 dev.d]# cd
[root@linux1 ~]#
```

3. Brancher la clé : elle est montée automatiquement.

```
[root@linux1 block]# tail  /var/log/messages
Feb 19 19:22:00 linux1 kernel: SCSI device sde: 1961695 512-byte hdwr sectors
(1004 MB)
Feb 19 19:22:00 linux1 kernel: sde: Write Protect is off
Feb 19 19:22:00 linux1 kernel: sde: assuming drive cache: write through
Feb 19 19:22:00 linux1 kernel:  sde: sde1
Feb 19 19:22:00 linux1 kernel: sd 10:0:0:0: Attached scsi removable disk sde
Feb 19 19:22:00 linux1 scsi.agent[29120]: disk at
/devices/pci0000:00/0000:00:10.4/usb1/1-5/1-5:1.0/host10/target10:0:0/10:0:0:0
Feb 19 19:22:05 linux1 logger: essai mount cle /dev/sde , action=add
Feb 19 19:22:05 linux1 kernel: FAT: invalid media value (0x01)
Feb 19 19:22:05 linux1 kernel: VFS: Can't find a valid FAT filesystem on dev
sde.
Feb 19 19:22:05 linux1 logger: essai mount cle /dev/sde1 , action=add
^C
[root@linux1 block]# df
Filesystem           1K-blocks      Used Available Use% Mounted on
```

```
...
/dev/sde1              979920    118432    861488  13% /root/cle
```

4. Démonter la clé, la retirer et supprimer les règles ajoutées.

```
[root@linux1 block]# umount /root/cle
[root@linux1 block]# cd
[root@linux1 ~]# rm /etc/udev/rules.d/20-perso.rules
```

- *Quota Hard, Soft*
- *user:pierre:rw-*
- *mask::rwx*
- *chattr +A f1 f2*
- *Page-in, Page-out*

6

Compléments sur les disques et les FS

Objectifs

Ce chapitre complète nos connaissances sur les disques et les FS. On apprend à mettre en œuvre les quotas disques et les ACL. On apprend également à ajouter une zone de swap. On étudie également les attributs spéciaux (Ext2, étendus) qui s'ajoutent aux caractéristiques standards des fichiers.

Contenu

Les quotas

Les ACL

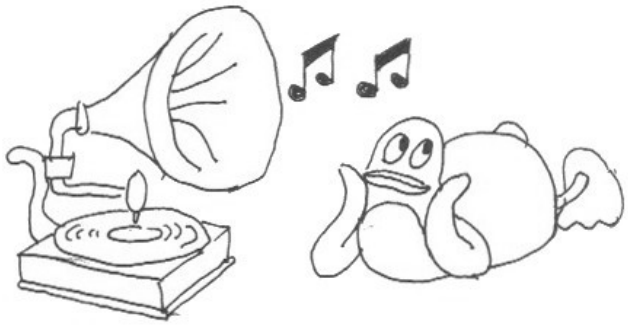
Les attributs Ext2

Les attributs étendus (xattr)

Les disques dédiés

Le Swap

Ateliers



Les quotas

La théorie

Description

La mise en œuvre des quotas permet de limiter l'usage des ressources disques. Par exemple avec les quotas on peut dire :

Pierre ne peut utiliser au maximum que 10 000 blocs dans le FS /home.

Pour un FS donné, les quotas limitent :

- Le nombre de blocs.
- Le nombre d'inodes (et donc le nombre de fichiers que l'on peut créer).

Les limites s'appliquent à

- Un utilisateur.
- Un groupe d'utilisateur.

Limites Hard et Soft

L'administrateur peut mettre en place deux types de limites :

- La limite « Hard » : elle est infranchissable.
- La limite « Soft » : On peut la dépasser pendant une période (grace), mais si on le fait, des messages d'avertissement sont affichés (warning). Logiquement, cette limite doit être inférieure à la limite Hard.

Implémentation

Les quotas sont gérés directement par le noyau au niveau de la VFS et des pilotes des FS. Les FS Unix/Linux les gèrent (Ext2/Ext3, ReiserFS, Xfs, NFS, …), pas les autres (vfat, …).

Les quotas sont stockés dans des fichiers binaires à la racine du FS. L'activation des quotas est une option de montage d'un FS.

Il est possible d'activer/désactiver la prise en compte des quotas pour un FS particulier ou pour l'ensemble des FS. Les scripts de démarrage activent les quotas et les scripts d'arrêt les désactivent.

Remarques :

1) Le noyau vérifie le dépassement des quotas à chaque écriture, ce qui diminue les performances.
2) La prise en compte des quotas est une option de compilation du noyau.

Le savoir concret

Les fichiers

/etc/fstab	Mémorise les FS devant être montés au démarrage ainsi que leurs options de démarrage (quota, …)

Les fichiers suivant sont à la racine d'un FS :

aquota.user	Mémorise les quotas des utilisateurs.
aquota.group	Mémorise les quotas des groupes d'utilisateurs.

Les commandes

`quotaon`	Active la prise en compte des quotas par le noyau.
`quotaoff`	Désactive la prise en compte des quotas par le noyau.
`quotaon -p`	Indique si les quotas sont pris en compte ou non (on/off).
`edquota`	Crée/modifie/supprime les quotas d'un utilisateur ou d'un groupe grâce à un éditeur (vi par défaut).
`setquota`	Idem, mais en mode ligne de commande (scriptable).
`quotacheck`	Vérifie les quotas. Permet également de créer les fichiers quotas.
`quota`	Liste les quotas de l'utilisateur.
`repquota`	Liste les quotas associés à un FS.

En savoir plus

Les pages de manuel

quotaon(8), repquota(8), quota(1), quotacheck(8), setquota(8), mount(8), quotactl(8)

Howto

Quota mini-Howto

Internet

Gérer les quotas
http://www.lea-linux.org/cached/index/Admin-admin_fs-quotas.html

Les ACL

La théorie

Description

Les droits Linux ordinaires (ISO) sont restreints : Il y a les droits qui s'appliquent au propriétaire, ceux qui s'appliquent aux membres du groupe et ceux qui s'appliquent aux autres.

Les ACL Linux, inspirés d'un draft POSIX, vont plus loin : ils permettent de positionner une liste de contrôle d'accès (ACL=Access Control List) associée à un fichier. Chaque élément de cette liste fixant les droits d'un utilisateur ou d'un groupe par rapport au fichier.

Remarque : Les ACL sont prioritaires sur les droits ISO.

Le concept de masque

Le masque ACL fait partie des ACL d'un fichier. Il précise si les ACL doivent être pris en compte en partie (pour r, w ou x), en totalité ou pas du tout.

Les ACL par défaut

La gestion des ACL peut être très lourde si l'on fixe les ACL fichier par fichier.

Les ACL par défaut simplifient les choses : on fixe des ACL par défaut au niveau d'un répertoire : tout fichier créé ensuite dans ce répertoire héritera par défaut de ces ACL.

Remarque : Les ACL par défaut sont une technique qui permet de simuler l'héritage des droits existant sous Windows.

Implémentation

Les ACL sont gérés par le noyau au niveau du VFS et des pilotes de FS. Les ACL sont supportés par les FS Xfs, Ext2/Ext3, ReiserFS et NFS.

Les ACL sont mémorisés sur disque dans les structures de données des FS. Ils sont activés lors du montage du FS. Dans le cas des FS Ext2/Ext3, c'est une option de montage.

Le savoir concret

Les commandes

`getfacl`	Visualise les ACL d'un fichier.
`setfacl`	Crée, modifie, supprime les ACL d'un fichier.
`ls -l`	Liste les attributs d'un fichier, indique la présence d'ACL.
`mount`	Monte un FS. L'option –o acl active l'utilisation des acl.

Focus : la syntaxe de setfacl

```
setfacl  -m  ACL[,...]  fichier ...
```

Syntaxe d'une ACL :

```
<type_d_ACL>:[<valeur>]:<droits>
```

Exemples :

`user:pierre:rw-`	Pierre a les droits de lecture et d'écriture (r et w).

`group:g1:r---`	Le groupe g1 a le droit de lecture (r).
`mask::rwx`	Les ACL sont totalement pris en compte.
`mask::---`	Les ACL sont désactivés.
`mask::r--`	Seul le droit de lecture est pris en compte dans les ACL.
`default:user:paul:rw-`	Exemple d'ACL par défaut (pour un répertoire).

Remarque : La commande `setfacl` peut également modifier les droits ordinaires (ISO) avec les pseudo ACL suivantes : `user::<droits>`, `group::<droits>` et `other::<droits>`.

Sauvegarde des ACL

DANGER : Les ACL ne sont pas sauvegardés par les outils classiques de sauvegarde (`tar`, `dump`, ...). Seule la commande star les prend en compte.

Par contre, il est facile de les sauvegarder avec la commande `getfacl -R` et ensuite de les restaurer avec la commande `setfacl --restore`.

Pour en savoir plus

Les pages de manuel

getfacl(1), setfacl(1), ls(1), star(1)

Internet

La gestion des ACL
http://www.lea-linux.org/cached/index/Gestion_des_ACL.html

Les attributs Ext2

La théorie

Description

Les attributs Ext2 s'ajoutent aux caractéristiques classiques d'un fichier (droits, type, ...).

Ces attributs sont apparus avec Ext2, ils sont supportés par les FS Ext2/Ext3 et ReiserFS.

Le savoir concret

Les principaux attributs

a	Fichier journal : on ne peut qu'ajouter des données au fichier. Il est également impossible de le détruire.
i	Le fichier ne peut ni modifié, ni détruit, ni même déplacé. Cet attribut empêche également la création d'un lien sur le fichier.
s	Le fichier sera physiquement détruit lors de sa suppression.
D	Répertoire synchrone.
S	Fichier synchrone.
A	L'époque (date et l'heure) de dernier accès ne sera pas mise à jour.

Remarques :

1) Quand un fichier est synchrone les modifications qui lui sont apportés sont immédiatement reproduites sur le disque. Normalement les modifications sont réalisées au niveau des tampons systèmes (buffer cache).
2) Des options de la commande `mount` agissent de manière similaire à certains attributs, mais pour l'ensemble des fichiers comme l'option `-o sync` ou l'option `-o noatime` (cf. mount(8).

Les commandes

`chattr`	Modifie les attributs Ext2.
`lsattr`	Visualise la valeur des attributs Ext2 d'un fichier.
`mount`	L'option `-o attrs` est requise sous ReiserFS pour gérer ces attributs.

Focus : La commande chattr

```
# chattr +A f1 f2
```

Ajoute l'attribut A aux fichiers f1 et f2 (-A aurait retiré l'attribut).

Pour en savoir plus

Les pages de manuel

chattr(1), lsattr(1)

Internet

Attributs étendus
http://www.lea-linux.org/cached/index/Attributs_%C3%A9tendus.html

Les attributs étendus (xattr)

La théorie

Description

Les attributs étendus (Extended Attributes ou EA ou xattr) offrent la possibilité d'associer des paires `nom=valeur` à des fichiers.

Ces attributs sont apparus d'abord dans le FS XFS. Ils sont actuellement supportés en noyau 2.6 par les FS Ext2/Ext3, ReiserFS et XFS.

La philosophie des EA est de permettre l'ajout de fonctionnalités aux FS sans changer radicalement leur structure. Ainsi la gestion des ACL ou de la sécurité SELinux peut être gérée grâce aux EA.

Remarque : On vient de le constater, un administrateur n'a pas besoin de gérer directement les EA. Ces derniers sont utilisés par des pilotes ou des applications systèmes.

Structure

Un attribut étendu possède la structure suivante :

<espace_de_nom>.<nomde_l_attribut>

Voici la liste actuelle des espaces de noms (namespace) : user, trusted, system, security.

Le savoir concret

Les commandes

`getfattr`	Visualise les EA d'un fichier.
`getfattr -d`	Visualise les EA d'un fichier et leur valeur.
`setfattr`	Crée, modifie, supprime ou restaure les EA d'un fichier.
`setfattr -x`	Supprime un EA d'un fichier.
`mount`	Le FS Ext3 nécessite l'option –o user_xattr pour utiliser les EA.

DANGER : Les xattr ne sont pas sauvegardés par les outils classiques de sauvegarde (`tar`, `dump`, …). Seule la dernière version de la commande `star` les prend en compte.

Focus : création d'un attribut

```
# setfattr -n user.bidule -v machin un_fichier
```

Associe l'attribut `user.bidule` au fichier un_fichier. L'attribut a la valeur `machin`.

Pour en savoir plus

Les pages de manuel

attr(5), attr(1), getfattr(1), setfattr(1), star(1)

Internet

Linux Extended Attributes and ACLs
http://acl.bestbits.at

Les disques dédiés

La théorie

Un disque (RAM, partition, disque complet, volume logique, ...) peut être dédié à un logiciel d'application.

Dans ce cas, le disque ne comporte pas de fichiers au sens traditionnel du terme, mais des structures de données exploitables uniquement par l'application.

Le plus souvent, les applications proposent cette approche comme une alternative au stockage de leurs données à l'intérieur d'un FS. L'intérêt de l'approche disque dédié est d'offrir de meilleures performances.

Les « raw device »

Les raw device sont des disques dédiés, mais accessibles par les applications grâce à un fichier périphérique en mode caractère (normalement un disque est vu comme un périphérique bloc).

Cette approche est apparue sur les systèmes Unix et apporte de réels gains de performances. Sur un système Linux, l'architecture du noyau fait que la vision caractère offre peu d'intérêt (sauf si c'est requis par le logiciel).

Le savoir concret

Les fichiers

/sys/block/	Répertoire listant les périphériques bloc.
/dev/raw/raw<n>	Périphérique caractère associé à un périphérique bloc.

Les commandes

`raw`	Associe un fichier spécial caractère à un disque.
`sysctl`	Liste les périphériques.

Focus : La commande raw

a) Créer une association fichier spécial caractère – disque.

```
# raw /dev/raw/raw1  /dev/sda
```

b) Lister les associations.

```
# raw  -qa
```

c) Supprimer une association.

```
# raw  /dev/raw/raw1  0 0
```

Pour en savoir plus

Les pages de manuel

raw(8)

Internet

The Linux 2.4 SCSI subsystem HOWTO (Chapter 11. Raw devices)
http://www.linux.com/howtos/SCSI-2.4-HOWTO/rawdev.shtml

SWAP

La théorie

Mémoire virtuelle et SWAP

Une application sous forme binaire contient des instructions machines. Ces instructions référent des registres du processeur et des cases mémoires. Les adresses de ces cases sont dites « virtuelles » car elles n'ont de sens que pour l'application elle-même et sont sans rapport avec les adresses de la mémoire physique (mémoire vive ou RAM).

Exemple : « mov ax,[0x94e0] » Stocke le contenu d'une case mémoire dans le registre ax.

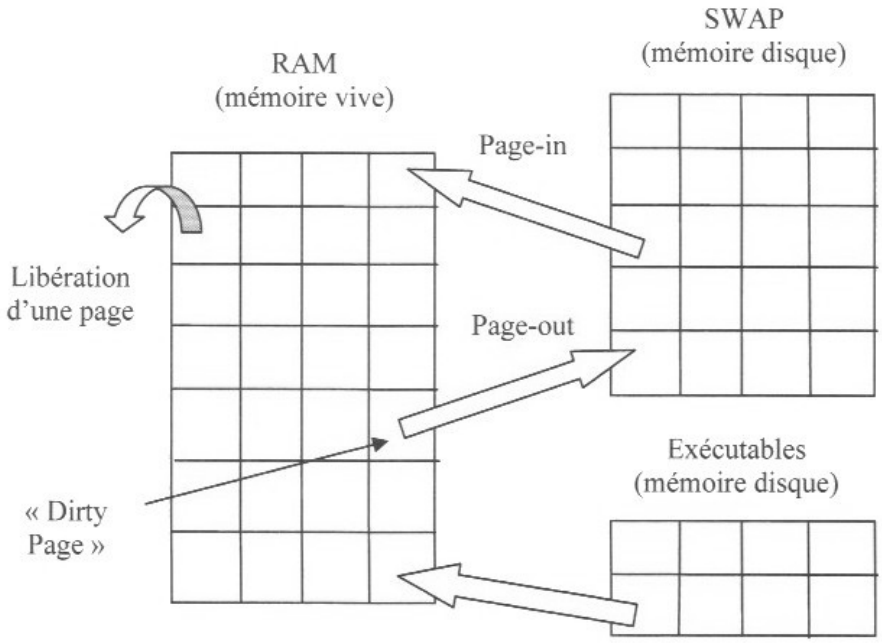


Fig. Page-in, Page-out

L'espace mémoire physique ainsi que l'espace virtuelle des applications sont divisées en page de 4Ko.

Toutes les pages d'un processus n'ont pas besoin d'être en mémoire physique simultanément. Seules les pages référencées à l'instant « T » doivent être en RAM. Les autres peuvent résider sur disque.

L'espace disque associé aux processus se divise en deux :

- Les fichiers exécutables et les bibliothèques dynamiques qui contiennent le code des applications.
- Le swap qui contient les données des applications (chaque processus possède une zone de données statique, une zone d'allocation dynamique et une pile).

Quand l'application référence une page non présente en RAM, le processus est interrompu, on a un « défaut de page » (Page fault). Le processeur (le MMU plus précisément), renseigné par des tables du noyau, amène la page manquante en RAM (Page-in), et l'application peut continuer.

La mémoire est vite pleine. Il faut supprimer de la mémoire des pages qui ne sont plus utilisées (on dit « libérer des pages). Les pages de codes ou les pages de données non modifiées sont supprimées sans autre forme de procès. Par contre, les pages de données

modifiées (Dirty pages) doivent être écrites sur l'espace de swap (Page-out) avant d'être supprimée.

Combien de swap ?

Linux peut fonctionner sans espace de swap. Par contre, si l'espace RAM est trop juste, la présence de swap améliore les performances en évitant la libération de trop de pages de codes.

Il ne faut pas oublier que s'il manque trop de mémoire (RAM + swap), une application ne peut démarrer ou ne peut plus allouer de mémoire (ce qui généralement la fait se terminer).

Si l'espace mémoire est important, la présence de swap va également améliorer les performances en stockant des pages qui sont exceptionnellement utilisées, laissant ainsi la mémoire vive disponible pour des utilisations plus critique.

En conclusion, on configure de l'espace de swap.

Il est difficile à priori de connaître la quantité de swap nécessaire. Le manuel de l'application principale peut vous renseigner sur ses besoins. L'heuristique la plus connue est de prendre deux fois la taille de la mémoire physique.

Où installer le swap ?

- Dans une partition.
- Dans un volume logique.
- Dans un fichier à l'intérieur d'un FS.

Remarques :

1) Pour améliorer les performances, il est préférable d'utiliser plusieurs petits espaces de swap sur des disques différents (SCSI par exemple) plutôt qu'un gros espace de swap.
2) Créer un espace de swap dans un fichier est moins performant que de le créer dans une partition, mais c'est très pratique en dépannage.

Le savoir concret

Les commandes

`mkswap`	Formate un espace de swap.
`swapon`	Active un espace de swap.
`swapon -s`	Liste les espaces de swap, affiche leur usage.
`free`	Visualise l'espace RAM et le swap libre et occupé.
`vmstat`	Affiche des statistiques concernant le CPU, la mémoire et les entrées/sorties.
`fdisk -l`	Liste les partitions, celles ayant le tag « 82 » sont à priori des espaces de swap Linux.

Les fichiers

/etc/fstab	Ce fichier mémorise les FS ainsi que les espaces de swap devant être activés au démarrage.
/proc/swap	La vision « noyau » du swap.

Focus : Le fichier /etc/fstab

Si l'on veut activer automatiquement un espace de swap au démarrage, il doit être mentionné dans le fichier /etc/fstab. Voici une ligne de ce fichier qui décrit un espace de swap :

```
/dev/hda6      swap      swap      defaults   0  0
```

Focus : Créer un espace de swap dans un fichier

1) On crée un gros fichier avec la commande dd.
```
#  dd if=/dev/zero  of=/fichier.swap  bs=1k  count=32768
```

2) On formate le fichier en tant que swap.
```
# mkswap  /fichier.swap
```

3) On active ce nouvel espace de swap.
```
# swapon  /fichier.swap
```

Pour en savoir plus

Les pages de manuel

mkswap(8), swapon(8), fstab(5), vmstat(8)

Howto

Linux Swap Space Mini-Howto

Internet

Wikipedia – Virtual Memory
http://en.wikipedia.org/wiki/Category:Virtual_memory

Red Hat Enterprise Linux 4 - Introduction à l'administration système (chapitres 2 et 4)
http://www.redhat.com/docs/manuals/enterprise/RHEL-4-Manual/fr/admin-guide/index.html

ATELIERS



Tâche 1 : Les quotas 15mn
Tâche 2 : Les ACL 10mn
Tâche 3 : Les attributs Ext2 ..... 5mn
Tâche 4 : Les attributs étendus (xattr) 5mn
Tâche 5 : Les disques dédiés 5mn
Tâche 6 : Le swap 10mn

Tâche 1 : Les quotas

1. Monter un FS avec les options de quota.

```
[root@linux1 ~]# mkfs -t ext3 -q /dev/hda6
[root@linux1 ~]# mount -t ext3 -o usrquota,grpquota /dev/hda6 /mnt/disk
[root@linux1 ~]# mount | grep /mnt
/dev/hda6 on /mnt/disk type ext3 (rw,usrquota,grpquota)
[root@linux1 ~]#
```

2. Créer les fichiers quota à la racine du FS.

```
[root@linux1 ~]# quotacheck -cugm /mnt/disk
[root@linux1 ~]# ls -l /mnt/disk
total 24
-rw-------  1 root root  6144 Feb 20 19:33 aquota.group
-rw-------  1 root root  6144 Feb 20 19:33 aquota.user
drwx------  2 root root 12288 Feb 20 19:31 lost+found
[root@linux1 ~]#
```

3. Vérifier si les quotas sont activés, ensuite les activer.

```
[root@linux1 ~]# quotaon -p /mnt/disk
group quota on /mnt/disk (/dev/hda6) is off
user quota on /mnt/disk (/dev/hda6) is off
[root@linux1 ~]# quotaon /mnt/disk
[root@linux1 ~]# quotaon -p /mnt/disk
group quota on /mnt/disk (/dev/hda6) is on
user quota on /mnt/disk (/dev/hda6) is on
[root@linux1 ~]#
```

4. Modifier les quotas d'un utilisateur et autoriser ce dernier à écrire dans le FS.

On modifie pour l'utilisateur guest la limite Hard des blocs à 1000.

```
[root@linux1 ~]# edquota guest
Disk quotas for user guest (uid 500):
  Filesystem                   blocks       soft       hard     inodes     soft     hard
  /dev/hda6                         0          0       1000          0        0        0
[root@linux1 ~]# chmod a+rwx /mnt/disk
```

5. Se connecter sous ce compte et essayer de dépasser les quotas.

```
[root@linux1 ~]# su - guest
[guest@linux1 ~]$ cd /mnt/disk
[guest@linux1 disk]$ cp /etc/profile .
[guest@linux1 disk]$ quota
Disk quotas for user guest (uid 500):
```

```
     Filesystem  blocks   quota   limit   grace   files   quota   limit   grace
      /dev/hda6       1       0    1000               1       0       0
[guest@linux1 disk]$ dd if=/dev/zero of=toto bs=1k count=1500
hda6: write failed, user block limit reached.
dd: Ã©criture de `toto': DÃ©bordement du quota d'espace disque
995+0 enregistrements lus.
994+0 enregistrements Ã©crits.
[guest@linux1 disk]$ ls -lh toto
-rw-rw-r--  1 guest guest 994K fÃ©v 20 19:42 toto
[guest@linux1 disk]$ quota
Disk quotas for user guest (uid 500):
     Filesystem  blocks   quota   limit   grace   files   quota   limit   grace
      /dev/hda6    1000*      0    1000               2       0       0
[guest@linux1 disk]$ exit
```

6. L'administrateur visualise l'usage des quotas.

```
[root@linux1 ~]# repquota /mnt/disk
*** Report for user quotas on device /dev/hda6
Block grace time: 7days; Inode grace time: 7days
                        Block limits                File limits
User            used    soft    hard  grace    used  soft  hard  grace
----------------------------------------------------------------------
root      --   10544       0       0              4     0     0
guest     --    1000       0    1000              2     0     0
```

7. L'administrateur modifie les quotas de l'utilisateur.

Il supprime la limite sur les blocs, mais il impose une limite sur le nombre de fichiers crées: uniquement 5 fichiers. Dés que l'utilisateur guest créera 4 fichiers, il sera prévenu qu'il est proche de la limite "hard".

```
[root@linux1 ~]# edquota guest
Disk quotas for user guest (uid 500):
  Filesystem          blocks       soft       hard     inodes     soft     hard
  /dev/hda6             1000          0          0          2        4        5
```

8. Se connecter sous ce compte et essayer de dépasser les quotas.

```
[root@linux1 ~]# su - guest
[guest@linux1 ~]$ cd /mnt/disk
[guest@linux1 disk]$ touch f1
[guest@linux1 disk]$ touch f2
[guest@linux1 disk]$ touch f3
hda6: warning, user file quota exceeded.
[guest@linux1 disk]$ ls
aquota.group  aquota.user  f1  f2  f3  lost+found  profile  toto
[guest@linux1 disk]$ touch f4
hda6: write failed, user file limit reached.
touch: ne peut faire un touch sur `f4': DÃ©bordement du quota d'espace disque
[guest@linux1 disk]$ quota
Disk quotas for user guest (uid 500):
     Filesystem  blocks   quota   limit   grace   files   quota   limit   grace
      /dev/hda6    1000       0       0              5*      4       5
[guest@linux1 disk]$ exit
```

9. L'administrateur visualise les quotas.

```
[root@linux1 ~]# repquota /mnt/disk/
*** Report for user quotas on device /dev/hda6
Block grace time: 7days; Inode grace time: 7days
                        Block limits                File limits
User            used    soft    hard  grace    used  soft  hard  grace
----------------------------------------------------------------------
root      --   10544       0       0              4     0     0
guest     -+    1000       0       0              5     4     5  6days
```

10. L'administrateur supprime les quotas en mode ligne de commande.

```
[root@linux1 ~]# setquota -u guest 0 0 0 0 /mnt/disk
[root@linux1 ~]# repquota /mnt/disk
*** Report for user quotas on device /dev/hda6
Block grace time: 7days; Inode grace time: 7days
                        Block limits                File limits
User            used    soft    hard  grace    used  soft  hard  grace
----------------------------------------------------------------------
root      --   10544       0       0              4     0     0
guest     --    1000       0       0              5     0     0
```

Remarque : Syntaxe de la commande `setquota`: setquota –u utilisateur quota_soft_blocs quota_hard_blocs quota_inodes_soft quota_inodes_hard FS

11. Vérifier les quotas (les quotas doivent être désactivés).

```
[root@linux1 ~]# quotacheck /mnt/disk/
quotacheck: Quota for users is enabled on mountpoint /mnt/disk so quotacheck
might damage the file.
Please turn quotas off or use -f to force checking.
[root@linux1 ~]# quotaoff /mnt/disk
[root@linux1 ~]# quotacheck /mnt/disk/
[root@linux1 ~]#
```

12. Démonter le FS.

```
[root@linux1 ~]# umount /mnt/disk
[root@linux1 ~]#
```

Tâche 2 : Les ACL

1. Créer des comptes.

```
[root@linux1 ~]# groupadd paire
[root@linux1 ~]# groupadd impaire
[root@linux1 ~]# useradd -g paire user2
[root@linux1 ~]# useradd -g paire user4
[root@linux1 ~]# useradd -g impaire user1
[root@linux1 ~]# useradd -g impaire user3
[root@linux1 ~]#
```

2. Créer et monter un FS, donner la racine à user1.

```
[root@linux1 ~]# mkfs -t ext3 -q /dev/hda6
[root@linux1 ~]# mount -t ext3 -o acl /dev/hda6 /mnt/disk
[root@linux1 ~]# chown user1 /mnt/disk
[root@linux1 ~]#
```

3. Se connecter sous le compte user1 et créer une arborescence dans le FS.

```
[root@linux1 ~]# su - user1
[user1@linux1 ~]$ cd /mnt/disk
[user1@linux1 disk]$ cal > f1
[user1@linux1 disk]$ date > f2
[user1@linux1 disk]$ mkdir rep
[user1@linux1 disk]$ ls > rep/fic
[user1@linux1 disk]$ chmod go=-  f1 f2 rep/fic
[user1@linux1 disk]$ ls -l f1 f2 rep/fic
-rw-------  1 user1 impaire 137 Feb 20 23:15 f1
-rw-------  1 user1 impaire  30 Feb 20 23:16 f2
-rw-------  1 user1 impaire  21 Feb 20 23:16 rep/fic
[user1@linux1 disk]$
```

4. Positionner des ACL sur les fichiers, les visualiser.

```
[user1@linux1 disk]$ setfacl -m user:user2:r f1
[user1@linux1 disk]$ setfacl -m user:user2:rwx,group:paire:rw- f2
[user1@linux1 disk]$ setfacl -m group:paire:r-- rep/fic
[user1@linux1 disk]$ getfacl f1 f2 rep/fic
# file: f1
# owner: user1
# group: impaire
user::rw-
user:user2:r--
group::---
mask::r--
other::---

# file: f2
# owner: user1
# group: impaire
user::rw-
user:user2:rwx
group::---
group:paire:rw-
mask::rwx
other::---

# file: rep/fic
# owner: user1
# group: impaire
user::rw-
group::---
group:paire:r--
mask::r--
other::---

[user1@linux1 disk]$ ls -l f1 f2 rep/fic
-rw-r-----+ 1 user1 impaire 137 Feb 20 23:15 f1
-rw-rwx---+ 1 user1 impaire  30 Feb 20 23:16 f2
-rw-r-----+ 1 user1 impaire  21 Feb 20 23:16 rep/fic
[user1@linux1 disk]$ exit
```

Remarque : La commande `ls -l` ajoute un + après les droits dans le cas où le fichier possède des ACL.

5. Tester les accès.

a) A partir du compte user2.

```
[root@linux1 ~]# su - user2
[user2@linux1 ~]$ cd /mnt/disk
[user2@linux1 disk]$ id
uid=501(user2) gid=501(paire) groupes=501(paire)
[user2@linux1 disk]$ ls -l
total 17
-rw-r-----+ 1 user1 impaire   137 fÃ©v 20 23:15 f1
-rw-rwx---+ 1 user1 impaire    30 fÃ©v 20 23:16 f2
drwx------  2 root  root    12288 fÃ©v 20 23:12 lost+found
drwxr-xr-x  2 user1 impaire  1024 fÃ©v 20 23:16 rep
[user2@linux1 disk]$ cat f1
    fÃ©vrier 2007
di lu ma me je ve sa
             1  2  3
 4  5  6  7  8  9 10
11 12 13 14 15 16 17
18 19 20 21 22 23 24
25 26 27 28

[user2@linux1 disk]$ cat rep/fic
f1
f2
lost+found
rep
[user2@linux1 disk]$ exit
```

b) A partir du compte user3.

```
[root@linux1 ~]# su - user3
[user3@linux1 ~]$ cd /mnt/disk
[user3@linux1 disk]$ cat f1
cat: f1: Permission non accordÃ©e
[user3@linux1 disk]$ exit
```

6. Désactiver les ACL sur le fichier f1 en modifiant le masque. L'utilisateur user2 n'a plus d'accès au fichier.

```
[root@linux1 ~]# su - user1
[user1@linux1 ~]$ cd /mnt/disk/
[user1@linux1 disk]$ setfacl -m mask::--- f1
[user1@linux1 disk]$ getfacl f1
# file: f1
# owner: user1
# group: impaire
user::rw-
user:user2:r--   #effective:---
group::---
mask::---
other::---
[user1@linux1 disk]$ exit
[root@linux1 ~]# su - user2
[user2@linux1 ~]$ cd /mnt/disk/
[user2@linux1 disk]$ tail -3 f1
tail: Ne peut ouvrir `f1' en lecture: Permission non accordÃ©e
```

```
[user2@linux1 disk]$
[user1@linux1 disk]$ exit
```

7. Mémoriser, supprimer et restaurer les ACLS.

```
[root@linux1 ~]# cd /mnt/disk
[root@linux1 disk]# getfacl -R . > /tmp/acls
[root@linux1 disk]# setfacl -R -b .
[root@linux1 disk]# getfacl f1 f2 rep/fic
# file: f1
# owner: user1
# group: impaire
user::rw-
group::---
other::---

# file: f2
# owner: user1
# group: impaire
user::rw-
group::---
other::---

# file: rep/fic
# owner: user1
# group: impaire
user::rw-
group::---
other::---

[root@linux1 disk]# ls -l f1 f2 rep/fic
-rw-------  1 user1 impaire 137 Feb 20 23:15 f1
-rw-------  1 user1 impaire  30 Feb 20 23:16 f2
-rw-------  1 user1 impaire  21 Feb 20 23:16 rep/fic
[root@linux1 disk]# setfacl --restore=/tmp/acls
[root@linux1 disk]# getfacl f2
# file: f2
# owner: user1
# group: impaire
user::rw-
user:user2:rwx
group::---
group:paire:rw-
mask::rwx
other::---

[root@linux1 disk]#
```

8. Créer des ACL par défaut pour une arborescence.

```
[root@linux1 disk]# su - user1
[user1@linux1 ~]$ cd /mnt/disk/
[user1@linux1 disk]$ setfacl -m default:user2:r  rep
[user1@linux1 disk]$ cal > rep/ficBis
[user1@linux1 disk]$ getfacl rep/ficBis
# file: rep/ficBis
# owner: user1
```

```
# group: impaire
user::rw-
user:user2:r--
group::r-x   #effective:r--
mask::r--
other::r--

[user1@linux1 disk]$ exit
```

9. Démonter le FS.

```
[root@linux1 disk]# cd
[root@linux1 ~]# umount /mnt/disk
[root@linux1 ~]#
```

Tâche 3 : Les attributs Ext2

1. Crée un fichier, lister ses attributs.

```
[root@linux1 ~]# echo "Bonjour" > fichier
[root@linux1 ~]# lsattr fichier
------------- fichier
[root@linux1 ~]#
```

2. Ajouter l'attribut « a » (fichier journal), tester les conséquences.

```
[root@linux1 ~]# chattr +a fichier
[root@linux1 ~]# lsattr fichier
-----a------- fichier
[root@linux1 ~]# echo "Salut" >> fichier
[root@linux1 ~]# cal > fichier
-bash: fichier: Operation not permitted
[root@linux1 ~]# rm -f fichier
rm: cannot remove `fichier': Operation not permitted
[root@linux1 ~]#
```

3. Retirer l'attribut précédent, ajouter l'attribut i (fichier immuable).

```
[root@linux1 ~]# chattr -a fichier
[root@linux1 ~]# chattr +i fichier
[root@linux1 ~]# echo "Bye" >> fichier
-bash: fichier: Permission denied
[root@linux1 ~]# cal > fichier
-bash: fichier: Permission denied
[root@linux1 ~]# rm -f fichier
rm: cannot remove `fichier': Operation not permitted
[root@linux1 ~]# ln fichier ficBis
ln: creating hard link `ficBis' to `fichier': Operation not permitted
[root@linux1 ~]# ln -s fichier ficBis
[root@linux1 ~]# mv fichier /tmp
mv: cannot move `fichier' to `/tmp/fichier': Operation not permitted
[root@linux1 ~]#
```

4. Retirer l'attribut précédent, ajouter l'attribut A (noatime).

```
[root@linux1 ~]# chattr -i fichier
[root@linux1 ~]# chattr +A fichier
[root@linux1 ~]# ls -lu fichier
-rw-r--r--  1 root root 14 Feb 22 19:08 fichier
[root@linux1 ~]# sleep 60; cat fichier
```

```
Bonjour
Salut
[root@linux1 ~]# ls -lu fichier
-rw-r--r--  1 root root 14 Feb 22 19:08 fichier
[root@linux1 ~]# chattr -A fichier
[root@linux1 ~]# ls -lu fichier
-rw-r--r--  1 root root 14 Feb 22 19:08 fichier
[root@linux1 ~]# sleep 60; cat fichier
Bonjour
Salut
[root@linux1 ~]# ls -lu fichier
-rw-r--r--  1 root root 14 Feb 22 19:12 fichier
[root@linux1 ~]#
```

5. Ajouter au fichier les attributs a, i, s, D, S et A au fichier.

```
[root@linux1 ~]# chattr +aisDSA fichier
[root@linux1 ~]# lsattr fichier
s-S-ia-A----- fichier
[root@linux1 ~]#
```

Tâche 4 : Les attributs étendus (xattr)

1. Créer un FS Ext3. Le monter en activant la prise en compte des EA. Créer un fichier.

```
[root@linux1 ~]# mkfs -t ext3 -q /dev/hda6
[root@linux1 ~]# mount -t ext3 -o user_xattr /dev/hda6 /mnt/disk
[root@linux1 ~]# date > /mnt/disk/f1
```

2. Ajouter un des attributs.

On ajoute l'attribut `user.truc` ayant la valeur `chose` et l'attribut `user.bidule` ayant la valeur `machin`.

```
[root@linux1 ~]# setfattr -n user.truc -v chose /mnt/disk/f1
[root@linux1 ~]# setfattr -n user.bidule -v machin /mnt/disk/f1
```

3. Afficher l'ensemble des attributs et leurs valeurs.

```
[root@linux1 ~]# getfattr -d /mnt/disk/f1
getfattr: Removing leading '/' from absolute path names
# file: mnt/disk/f1
user.bidule="machin"
user.truc="chose"
```

4. Afficher juste la liste des attributs d'un fichier.

```
[root@linux1 ~]# getfattr  /mnt/disk/f1
getfattr: Removing leading '/' from absolute path names
# file: mnt/disk/f1
user.bidule
user.truc
```

5. Afficher un attribut et sa valeur.

```
[root@linux1 ~]# getfattr  /mnt/disk/f1 -n user.bidule
getfattr: Removing leading '/' from absolute path names
# file: mnt/disk/f1
user.bidule="machin"
```

6. Sauvegarder les attributs d'une arborescence.

```
[root@linux1 ~]# getfattr -dR  /mnt/disk/ > /tmp/sauve.xattr
getfattr: Removing leading '/' from absolute path names
```

7. Détruire un attribut.

```
[root@linux1 ~]# setfattr -x user.bidule  /mnt/disk/f1
[root@linux1 ~]# getfattr  /mnt/disk/f1
getfattr: Removing leading '/' from absolute path names
# file: mnt/disk/f1
user.truc
```

8. Restaurer les attributs d'une arborescence.

```
[root@linux1 disk]# cd /
[root@linux1 /]# setfattr --restore=/tmp/sauve.xattr
[root@linux1 /]# getfattr /mnt/disk/f1
getfattr: Removing leading '/' from absolute path names
# file: mnt/disk/f1
user.bidule
user.truc

[root@linux1 /]# cd
```

9. Démonter le FS.

```
[root@linux1 ~]# umount /mnt/disk/
[root@linux1 ~]#
```

Tâche 5 : Les disques dédiés

1. Utiliser une partition comme un périphérique d'archivage.

a) On crée un FS dans une partition.

```
[root@linux1 ~]# mkfs -q /dev/hda6
[root@linux1 ~]# tune2fs -l /dev/hda6 | head -2
tune2fs 1.35 (28-Feb-2004)
Filesystem volume name:   <none>
```

b) On sauvegarde des fichiers avec tar dans la partition. On liste l'archive.

```
[root@linux1 ~]# tar cf /dev/hda6 /etc
tar: Removing leading `/' from member names
[root@linux1 ~]# tar tvf /dev/hda6 | tail -3
-rw-r--r-- root/root        83 2006-05-03 23:36:13
etc/security/console.apps/system-config-date
lrwxrwxrwx root/root         0 2007-01-20 10:21:55
etc/security/console.apps/serviceconf -> system-config-services
-rw-r--r-- root/root      2279 2006-08-13 00:31:42 etc/security/access.conf
```

c) On essaye d'accéder à la partition en tant que FS.

```
[root@linux1 ~]# tune2fs -l /dev/hda6 | head -3
tune2fs: Bad magic number in super-block while trying to open /dev/hda6
Couldn't find valid filesystem superblock.
tune2fs 1.35 (28-Feb-2004)
[root@linux1 ~]# mount -t ext2 /dev/hda6 /mnt/disk
mount: wrong fs type, bad option, bad superblock on /dev/hda6,
       or too many mounted file systems
[root@linux1 ~]#
```

Remarque : Le FS a été détruit lorsque l'on a sauvegardé des fichiers par `tar`.

2. Les raw devices.

a) On associe un raw device à une partition.

```
[root@linux1 ~]# raw /dev/raw/raw1 /dev/hda6
/dev/raw/raw1:  bound to major 3, minor 6
[root@linux1 ~]# ls -l /dev/hda6
brw-rw----  1 root disk 3, 6 Feb 23 18:11 /dev/hda6
[root@linux1 ~]# ls -l /dev/raw/raw1
crw-rw----  1 root disk 162, 1 Feb 23 18:19 /dev/raw/raw1
```

b) On liste les associations.

```
[root@linux1 ~]# raw -qa
/dev/raw/raw1:  bound to major 3, minor 6
```

c) On utilise le raw device (on sauvegarde /etc dessus).

```
[root@linux1 ~]# tar cf /dev/raw/raw1  /etc
tar: Removing leading `/' from member names
[root@linux1 ~]# tar tvf /dev/raw/raw1  etc/group
-rw-r--r-- root/root       651 2007-02-20 23:08:40 etc/group
```

d) On supprime l'association.

```
[root@linux1 ~]# raw /dev/raw/raw1  0 0
/dev/raw/raw1:  bound to major 0, minor 0
[root@linux1 ~]# raw -qa
[root@linux1 ~]# ls -l /dev/raw/raw1
ls: /dev/raw/raw1: No such file or directory
```

Tâche 6 : Le Swap

1. Quelle est la taille d'une page mémoire ?

```
[root@linux1 ~]# vi pagesize.c
[root@linux1 ~]# cat pagesize.c
#include <unistd.h>
main() {
        long sz = sysconf(_SC_PAGESIZE);
        printf("page size : %ld bytes\n", sz );
}

[root@linux1 ~]# cc -o pagesize pagesize.c
[root@linux1 ~]# ./pagesize
page size : 4096 bytes
[root@linux1 ~]#
```

Remarque : L'appel système `sysconf()` nous informe de certaines caractéristiques du système (cf. la page de manuel getpagesize(2).

2. Lister les espaces de swap courants et définis au démarrage.

```
[root@linux1 ~]# swapon -s
Filename                        Type            Size    Used    Priority
/dev/hda5                       partition       522072  88      -1
[root@linux1 ~]# grep swap /etc/fstab
LABEL=SWAP-hda5         swap                    swap    defaults        0 0
[root@linux1 ~]#
```

3. Créer une zone de swap dans une partition.

a) On crée une partition de type 82 (ici on modifie juste le tag).

```
[root@linux1 ~]# fdisk /dev/hda

The number of cylinders for this disk is set to 4870.
```

```
There is nothing wrong with that, but this is larger than 1024,
and could in certain setups cause problems with:
1) software that runs at boot time (e.g., old versions of LILO)
2) booting and partitioning software from other OSs
   (e.g., DOS FDISK, OS/2 FDISK)

Command (m for help): t
Partition number (1-9): 6
Hex code (type L to list codes): 82
Changed system type of partition 6 to 82 (Linux swap)

Command (m for help): w
The partition table has been altered!

Calling ioctl() to re-read partition table.

WARNING: Re-reading the partition table failed with error 16: Device or resource
busy.
The kernel still uses the old table.
The new table will be used at the next reboot.
Syncing disks.
[root@linux1 ~]# partprobe
Warning: Unable to open /dev/hdc read-write (Read-only file system).  /dev/hdc
has been opened read-only.
[root@linux1 ~]# fdisk -l | grep hda6
/dev/hda6            1419        1481      506016   82  Linux swap
[root@linux1 ~]#
```

b) On formate l'espace de swap.

```
[root@linux1 ~]# mkswap /dev/hda6
Setting up swapspace version 1, size = 518156 kB
[root@linux1 ~]#
```

c) On active l'espace de swap.

```
[root@linux1 ~]# swapon -v /dev/hda6
swapon on /dev/hda6
[root@linux1 ~]# swapon -s
Filename                                Type            Size    Used    Priority
/dev/hda5                               partition       522072  88      -1
/dev/hda6                               partition       506008  0       -2
[root@linux1 ~]#
```

d) On désactive l'espace de swap et on remet le tag 83 à la partition.

```
[root@linux1 ~]# swapoff /dev/hda6
[root@linux1 ~]# fdisk /dev/hda
...
Command (m for help): t
Partition number (1-9): 6
Hex code (type L to list codes): 83
Changed system type of partition 6 to 83 (Linux)

Command (m for help): w
The partition table has been altered!
...
[root@linux1 ~]# partprobe /dev/hda
[root@linux1 ~]#
```

4. Créer une zone de swap dans un fichier.

a) On crée un fichier ayant la taille désirée (ici 128Mo).

```
[root@linux1 ~]# dd if=/dev/zero of=/fichier.swap bs=1k count=128000
128000+0 records in
128000+0 records out
[root@linux1 ~]# ls -lh /fichier.swap
-rw-r--r--  1 root root 125M Feb 21 22:21 /fichier.swap
```

b) On formate le fichier en tant qu'espace de swap.

```
[root@linux1 ~]# mkswap /fichier.swap
Setting up swapspace version 1, size = 131067 kB
```

c) On rend l'espace inaccessible.

```
[root@linux1 ~]# chmod 0600 /fichier.swap
```

d) On active l'espace de swap, on liste les espaces et leur utilisation.

```
[root@linux1 ~]# swapon -v /fichier.swap
swapon on /fichier.swap
[root@linux1 ~]# swapon -s
Filename                                Type            Size    Used    Priority
/dev/hda5                               partition       522072  88      -1
/fichier.swap                           file            127992  0       -3
[root@linux1 ~]# free
             total       used       free     shared    buffers     cached
Mem:        450160     442124       8036          0      55756     249776
-/+ buffers/cache:     136592     313568
Swap:       650064         88     649976
```

Remarque : L'espace Mem total correspond à la mémoire vive (RAM).

e) On désactive l'espace de swap et on détruit le fichier.

```
[root@linux1 ~]# swapoff /fichier.swap
[root@linux1 ~]# swapon -s
Filename                                Type            Size    Used    Priority
/dev/hda5                               partition       522072  88      -1
[root@linux1 ~]# rm /fichier.swap
rm: remove regular file `/fichier.swap'? y
[root@linux1 ~]#
```

- *RAID 0, 1 et 5*
- *Chunk-size, Spare*
- *DAS, NAS et SAN*
- *Baie de stockage, LUN*
- *mdadm --create /dev/md0*

7

Le RAID

Objectifs

Ce chapitre traite du RAID. Cette technique est indispensable pour obtenir des serveurs performants. Le savoir concret porte essentiellement sur le RAID logiciel qui est offert par le noyau Linux. Le RAID matériel lui, est présenté essentiellement de manière théorique. Pour beaucoup d'administrateurs, il se résume au choix du serveur.

Contenu

La technologie RAID

Le RAID matériel

Les solutions de type SAN

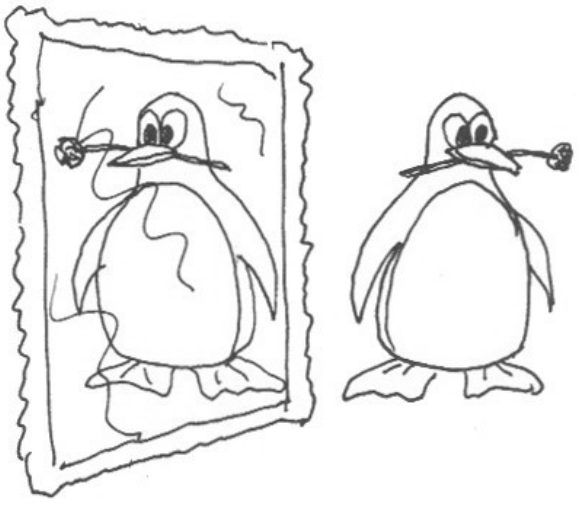
Les solutions de type DAS

Proliant (exemple de solution DAS)

Le RAID logiciel

Le RAID et le LVM

Ateliers



La technologie RAID

La théorie

Description

Un disque RAID (Redundant Array of InDependant Disks ou Disk Array) est constitué d'un ensemble de disques. L'avantage de ce groupement est d'obtenir de la tolérance aux pannes et/ou de meilleures performances.

Les performances sont permises en éclatant les entrées sorties sur plusieurs disques (stripping) SCSI ou sur plusieurs disques IDE gérés chacun par un contrôleur distinct. Dans ces cas-là, les entrées/sorties (I/O) sont exécutées simultanément.

La tolérance aux pannes d'un disque est réalisée soit en dupliquant les mêmes données sur plusieurs disques (on parle de disques miroirs ou mirroring), soit en utilisant des données de parité qui permettent la reconstitution du disque hors service.

Stripping ou concaténation

On peut former un disque de grande capacité en concaténant des disques : quand le 1er est rempli, on continue les ajouts sur un deuxième disque, quand ce dernier est rempli, on utilise un troisième disque, et ainsi de suite.

L'agrégation par bandes (stripping) offre de meilleures performances. Chaque disque est découpé en bandes (strip ou chunk). Les écritures se font alternativement sur les bandes des différents disques.

Par exemple, soit trois disques A, B et C. Si on utilise des bandes de 64ko, lors de l'écriture d'un gros fichier :

- Le 1er bloc (A1) de 64ko est écrit sur la 1re bande du disque A.
- Le 2e bloc (B1) de 64ko est écrit sur la 1re bande du disque B.
- Le 3e bloc (C1) de 64ko est écrit sur la 1re bande du disque C.
- Le 4e bloc (A2) de 64ko est écrit sur la 2e bande du disque A.
- Etc.

Remarque : la taille d'une bande (chunk-size) et le nombre de disques ont un impact très important sur les performances.

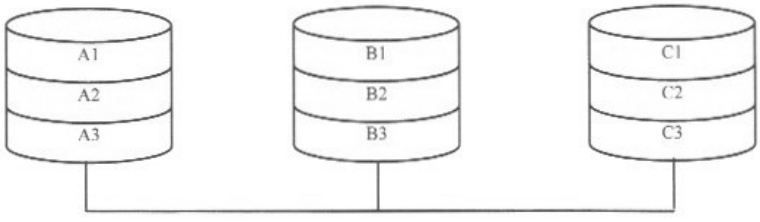


Fig. RAID 0

Les niveaux de RAID

Le niveau de RAID indique le type de technologie RAID, c'est-à-dire le mode de tolérance aux pannes utilisé. Les niveaux les plus courants sont 0, 1 et 5.

RAID 0

Le RAID 0 signifie l'absence de tolérance aux pannes. Cette technologie, par le biais du stripping apporte uniquement de meilleures performances.

RAID 1

Le RAID 1, appelé également MIROIR, consiste à écrire les mêmes informations simultanément sur 2 disques. Si l'un des disques tombe en panne, le disque RAID reste globalement opérationnel.

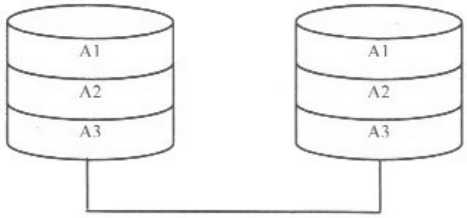


Fig. RAID 1

RAID 5

Le RAID 5 utilise le stripping et la parité. Il nécessite au moins trois disques. Les données de parité sont réparties sur les différents disques. En cas de panne d'un disque, le disque RAID reste opérationnel, le logiciel de gestion recalcule les données manquantes à partir des données utiles et des parités réparties sur les disques restant en fonction.

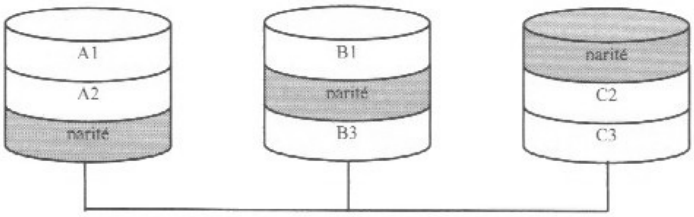


Fig. RAID 5

RAID 4

Le RAID 4 ressemble au RAID 5, mais les données de parité sont centralisées sur un seul disque, ce qui diminue à la fois les performances et la fiabilité. Pour ces raisons, il n'est plus utilisé.

RAID 6

Le RAID 6 est une évolution du RAID 5, mais avec deux bandes de parité au lieu d'une, ce qui accroît la fiabilité : on peut perdre deux disques, mais ce qui accroît également son coût. De ce fait, il est peu utilisé.

RAID 10 (ou 1+0)

Le RAID 10 est un RAID 0 construit sur deux disques RAID 1, chacun étant évidemment constitué de deux disques.

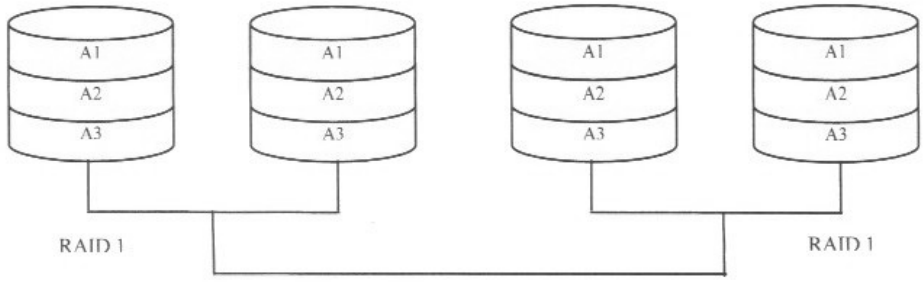


Fig. RAID 1+0

RAID 01 (ou 0+1)

Le RAID 01 est un disque 1 construit sur deux disques RAID 0, chacun étant le plus souvent constitué de deux disques.

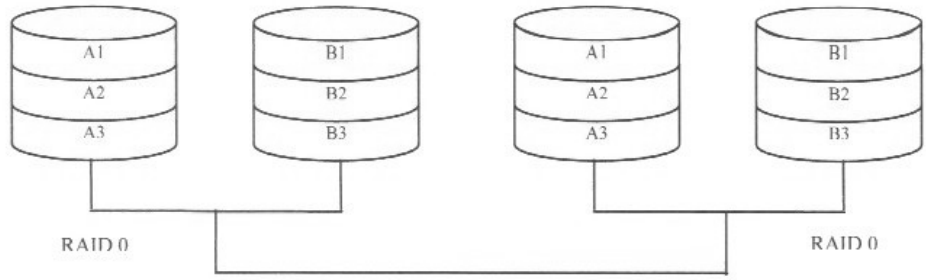


Fig. RAID 0+1

Remarque : Les technologies RAID 10 ou 01 permettent d'avoir un miroir rapide.

Les disques de rechange (Spare)

Quand un disque faisant partie d'un RAID tombe en panne, on est à la merci d'une deuxième panne tant que l'on n'a pas remplacé le disque hors service. Pour palier à ce problème, on peut ajouter des disques Spare qui remplace automatiquement les disques défaillants.

Remarque : Que l'on utilise ou non de disques Spare, le RAID sera défaillant si une deuxième panne survient (sauf en RAID 6) pendant la reconstruction du RAID.

RAID matériel ou logiciel

Le RAID logiciel est complètement géré par le système d'exploitation (l'OS). Dans ce cas, le matériel est quelconque.

Le RAID matériel est géré par un logiciel embarqué au niveau du contrôleur du disque ou même directement au niveau du disque (baie de stockage).

L'indépendance du RAID matériel du point de vue de l'OS le fait préférer par les administrateurs système.

Pour en savoir plus

Internet

RAID (informatique)
http://fr.wikipedia.org/wiki/RAID_(informatique)

Livres

Managing RAID on Linux, par Derek Vadala, chez O'Reilly

Le RAID matériel

La théorie

Les stratégies de stockage d'information et le RAID matériel

Le DAS (Direct Attached Storage)

Dans la technologie DAS, les disques sont internes au système et sont typiquement connectés en SCSI. Si le contrôleur est de type RAID, on peut construire des solutions RAID matériels.

Le NAS (Network Attached Storage)

Dans le NAS, le lieu de stockage est constitué par des disques d'un serveur accessible en réseau, typiquement par des protocoles de type NFS ou SMB. Cette technique est utilisée principalement par les postes clients. Les serveurs stockent leurs données grâce aux techniques DAS ou SAN.

Le SAN (Storage Area Network)

Dans un SAN, les serveurs accèdent à des baies de stockage à travers un réseau dédié, typiquement du Fiber Channel (ou éventuellement du SCSI sur du IP : iSCSI). Une baie offre aux serveurs un ou plusieurs disques RAID.

La baie est gérée grâce à un ordinateur. C'est lui qui définit les disques faisant partie de tel ou tel RAID et le niveau de RAID. En conséquence, une baie permet aux serveurs de voir les disques RAID comme des disques ordinaires.

Les postes clients accèdent aux serveurs à travers un réseau différent, le réseau d'exploitation qui normalement utilise Ethernet et TPC/IP.

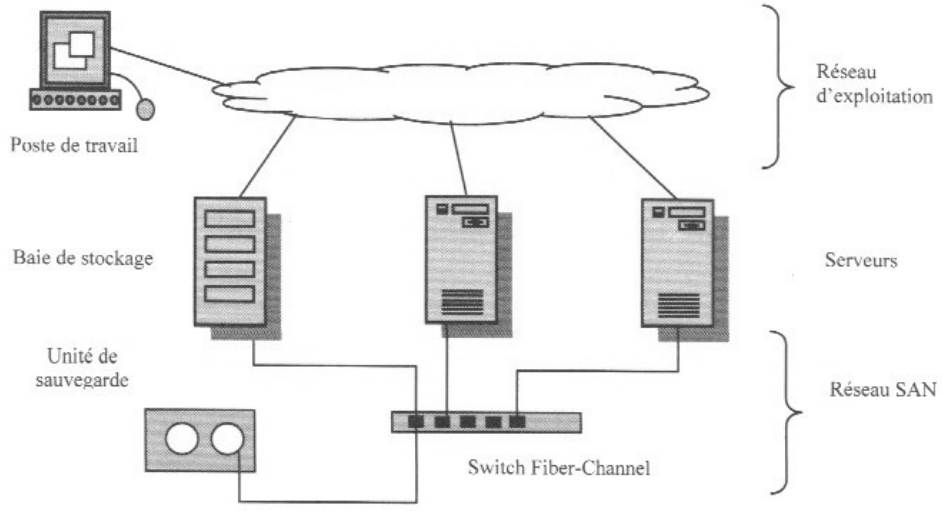


Fig. Réseau SAN

Compléments

Miroring et Duplexing

Un disque RAID 1 peut être géré de deux manières : en miroring ou en duplexing. Dans la technique miroring les deux disques sont reliés au même contrôleur SCSI. Inversement, dans la technique duplexing, chaque disque possède son propre contrôleur.

Multi-path IO

Une baie de disque autorise en théorie l'accès d'un même disque RAID à plusieurs serveurs simultanément. Cette facilité, appelée Multi-path IO, est utilisée en pratique pour construire des clusters.

Pour en savoir plus

Internet

SAN
http://en.wikipedia.org/wiki/Storage_area_network

Linux and HW raid - an administrators summary
http://wiki.debian.org/LinuxRaidForAdmins

Solutions de type SAN

La théorie

Notion de LUN

La baie de stockage offre des disques identifiés par des LUN. Ces disques peuvent être des RAID au niveau de la baie et sont donc constitués de plusieurs disques physiques. C'est complètement transparent au niveau des serveurs.

La notion de HBA

Un HBA (Host Bus Adapter) est une carte contrôleur inclus dans le système qui permet de le relier à un BUS ou à un réseau de stockage (SCSI/Fiber Channel).

Les principaux constructeurs de baie de disques (SUN, IBM, HP) proposent également des HBA. Les sociétés Emulex et QLogic proposent des HBA génériques.

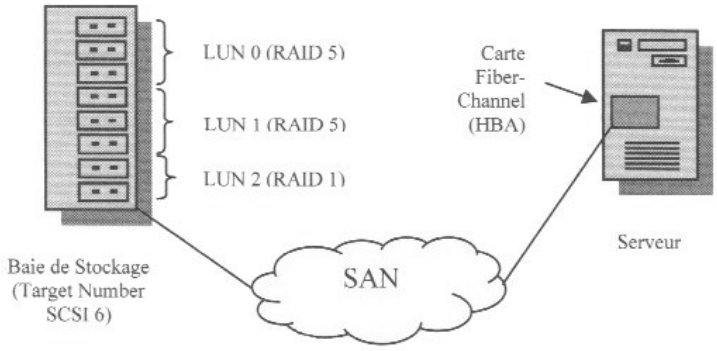


Fig. Réseau SAN et LUN

Remarque : Seules les distributions RedHat et SUSE sont supportées par ces différents fabricants, même si en théorie, si on dispose des sources, la recompilation des pilotes permet de les utiliser sur n'importe quelle distribution.

Le savoir concret

Les commandes

`dmesg`	Visualise les messages du Noyau, notamment les LUNs et les /dev/sd* associés.
`insmod`	Charge un pilote, notamment celui d'un HBA.
`lspci`	Liste les cartes PCI, notamment les cartes Fiber Channel (`lspci |grep fib`).

Remarque : Il existe également des commandes propriétaires.

Pour en savoir plus

Internet

QLogic – Fiber Channel SAN Products
http://www.qlogic.com/products/fc_san.asp

Emulex – Fiber Channel HBA
http://www.emulex.com/products/hba/index.jsp

Notes on getting the initial ramdisk to run udev and provide
http://www.emulex.com/white/hba/wp_linux26udev.pdf

Livre

Using SANs and NAS, Help for Storage Administration, par W. Curtis Preston, chez O'Reilly.

Les solutions de type DAS

La théorie

Carte RAID et serveurs

La plupart des fabricants de cartes SCSI, notamment Adaptec, proposent des cartes RAID SCSI et/ou SATA avec des pilotes pour Linux.

De même, les principaux fabricants de serveurs de type PC offrent des solutions RAID fonctionnant sous Linux.

- Gamme xSeries d'IBM. Ces serveurs utilisent les cartes RAID LSI Logic.
- Gamme PowerEdge de DELL. Ces serveurs utilisent les cartes RAID PERC.
- Gamme Proliant de HP. Ces serveurs utilisent les cartes RAID smart-Array.

Remarque : Comme dans le cas des solutions SAN, les fabricants de cartes RAID ou de serveurs n'offrent de support que pour les distributions RedHat ou SUSE.

Le savoir concret

Les pilotes pour cartes RAID

aic7xxx	Pilote utilisé par les cartes RAID Adaptec.
megaraid	Pilote utilisé pour les serveurs IBM xSeries.
aacraid	Pilote utilisé pour les serveurs DELL PowerEdge.
cciss	Pilote utilisé pour les serveurs HP Proliant.

Pour en savoir plus

Internet

Adaptec
http://linux.adaptec.com/

DELL Linux Engineering – RAID and Storage
http://linux.dell.com/storage.shtml

IBM - About Linux on System x and BladeCenter
http://www-03.ibm.com/systems/x/solutions/os/linux/

Linux for Proliant
http://h18004.www1.hp.com/products/servers/linux/documentation.html

Managing ProLiant servers with Linux HOWTO
http://h20000.www2.hp.com/bc/docs/support/SupportManual/c00223285/c00223285.pdf

Proliant (exemple de solution DAS)

La théorie

Carte Smart-Array

Les serveurs HP Proliant sont livrés avec une carte « Smart-Array ». Cette carte contrôleur RAID gère des disques SCSI connectés à chaud (hot plug).

HP propose le pilote Linux `cciss` pour gérer cette carte.

Remarque : HP ne supporte que les distributions RedHat et SUSE.

Paramétrage

1) On répartit les disques dans un ou plusieurs tableaux de disques (array). Pour chacun, on peut spécifier des disques Spare.

2) Pour chaque tableau de disques, on crée au moins un disque logique (Logical Drive). Ce dernier est vu comme un seul disque par l'OS.

3) Pour chaque disque logique, on choisit le niveau de RAID (0,1,5,ADG ou 1+0) et le chunk-size.

4) A partir de l'OS, on partitionne chaque disque logique comme on veut.

Remarque : le RAID-ADG est une variante du RAID-6.

Le savoir concret

Logiciels de configuration

ORCA

Au démarrage, avant même l'installation de l'OS, on peut activer le logiciel embarqué ORCA pour paramétrer le RAID.

ACU

Lorsque le système est actif, le logiciel ACU permet un paramétrage du RAID via une interface Web via l'URL http://le_serveur:2301 .

Le nom des disques

/dev/cciss/c0d0p1

c#d#	Spécifie le disque logique. c# indique le numéro du contrôleur et d# le numéro du disque logique.
p#	Spécifie le numéro de la partition.

Pour en savoir plus

Internet

HP - Proliant RAID Controler
http://h18004.www1.hp.com/products/servers/proliantstorage/arraycontrollers/

Livre

HP Proliant Servers AIS, par B.H. Weldon et Shawn B. Ruyers, aux éditions HP invent.

Le RAID logiciel

La théorie

Le système Linux offre une gestion RAID logiciel grâce au pilote du noyau MD (Multiple Device). Ce pilote supporte les niveaux de RAID suivants :

- Mode « Linear » : c'est la concaténation de disques : plusieurs disques forment un seul gros disque.
- RAID-0, comme précédemment, plusieurs disques forment un seul gros disque, mais grâce au stripping, les performances sont au rendez-vous.
- RAID-1 ou miroring.
- RAID 4 et 5 : Ces niveaux cumulent le stripping et la redondance des données.
- RAID 6 (une évolution du RAID 5).

Le pilote MD supporte l'utilisation de disques Spare. Il permet également le Multiple-path IO.

La gestion des disques RAID est réalisée par la commande `mdadm` ou bien par un jeu de commandes, appelées « raidtools ». Actuellement, on préfère utiliser la commande `mdadm`.

Les disques physiques qui composent un disque RAID doivent être des partitions (/dev/sda1, ...) de type « fd ».

Le savoir concret

Les commandes

`mdadm`	Commande qui gère un disque RAID (Création, activation, ...).
Les « RaidTools » :	
`mkraid`	Initialise un disque RAID, se base sur le fichier /etc/raidtab.
`lsraid`	Affiche l'état d'un RAID.
`raidstop`	Désactive un disque RAID.
`raidstart`	Active la gestion d'un disque RAID.
`raidsetfaulty`	Simule la panne d'un disque faisant partie d'un RAID.
`raidhotremove`	Retire à chaud un disque en panne.
`raidhotadd`	Ajoute à chaud un disque.

Les principales sous-commandes de la commande mdadm

`--create`	Crée un disque RAID, met à jour les super-blocs des disques spécifiés.
`--assemble`	Active un disque RAID déjà crée précédemment (on ne met pas à jour les super-blocs).
`--build`	Crée un disque RAID sans super-blocs (pour RAID-0 ou Linear).
`--examine`	Affiche le super-bloc d'un disque.
`--add`	Ajoute (à chaud) un disque à un RAID.

`--remove`	Retire (à chaud) un disque d'un RAID.
`--fail`	Positionne un disque en erreur (simule le crash d'un disque).
`--query`	Examine un disque et indique si il membre d'un RAID.
`--detail`	Affiche des informations concernant un RAID.
`--monitor`	Surveille les disques, prévient l'administrateur si besoin (par email, ...). Activé par le RC `mdmonitor`.

Les fichiers

/proc/mdsat	Etat du RAID (publié par le noyau).
/etc/mdadm.conf	Ce fichier n'est pas nécessaire mais il permet de garder une trace d'un disque RAID.
/dev/md#	Fichier spécial représentant un disque RAID logiciel.

Focus : Le super-bloc

Les disques faisant partie d'un RAID (sauf RAID-O et Linear) possèdent en tête un super-bloc qui contient la configuration du RAID. Ces super-blocs permettent également le démarrage à partir d'un disque RAID.

Focus : Créer un disque RAID

```
# mdadm  \
  --create /dev/md0  \
  --leve=1 \
  --raid-devices=2  \
  /dev/hda1  /dev/hdc1
```

Crée un RAID de niveau 1 (un miroir) à partir des partitions hda1 et hdc1.

Remarque : Par défaut, le chunk-size est de 64 Ko.

Focus : Le fichier /etc/mdadm.conf

L'activation d'un disque RAID est simplifiée si on utilise le fichier /etc/mdadm.conf.

```
DEVICE   /dev/sdb1  /dev/sdc1
ARRAY    /dev/md0  devices = /dev/sdb1, /dev/sdc1
```

Remarque : Un RAID est décrit soit par son UUID soit par la liste des disques qui le composent.

Pour en savoir plus

Les pages de manuel

md(4), mdadm(8), mdadm.conf(5), raidtab(5), raid0run(8), raidstop(8), mkraid(8)

Howto

Software-RAID Howto

Boot+Root+Raid+LILO

Internet

Installation d'une Debian en Root RAID-0
http://www.linux-france.org/article/install/debian/rootraid/

Le RAID et le LVM

La théorie

Sur les systèmes UNIX (AIX, HP-UX, ...), la technologie LVM permet la redondance des données et est donc en compétition avec l'approche RAID.

Sur les systèmes Linux, le LVM ne permet pas cette redondance. Si l'on désire la tolérance aux pannes, il faut associer le LVM Linux à la technologie RAID (logiciel ou matériel).

Quelques stratégies

Serveur mono-disque

Sur un petit serveur possédant un seul disque, on peut utiliser les partitions pour stocker les FS systèmes (/boot, /, /usr, /var, /tmp, ...) et le swap. La partie applicative (/home, /usr/local/apache, ...) est associée à des LV inclus dans un VG stocké dans un PV qui utilise le reste du disque. On peut également tout gérer sous forme de LV (sauf /boot). On utilise généralement un RAID 1 (miroir) pour assurer la tolérance aux pannes.

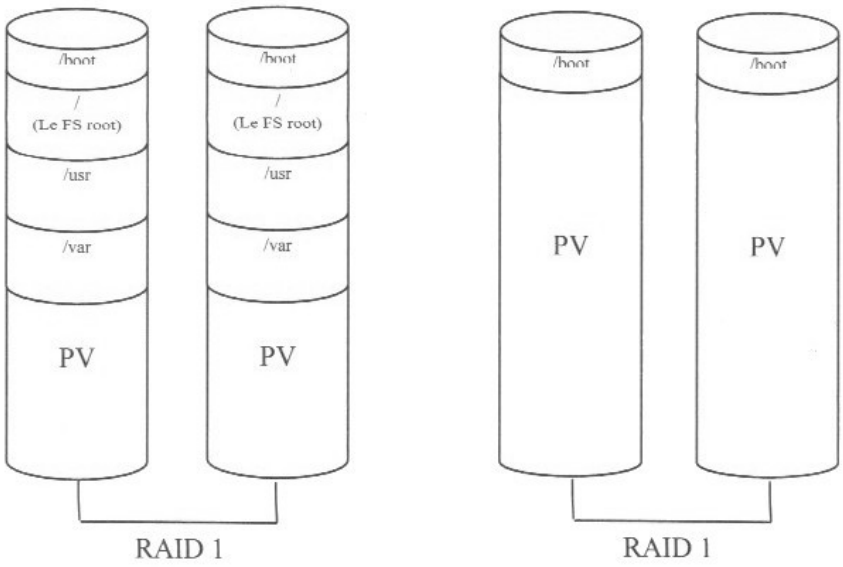


Fig. Deux stratégies de serveurs mono-disque

Gros serveur

Sur un gros serveur, on va gérer les FS systèmes comme précédemment (sous forme de partitions physiques ou de LV d'un VG système), les données applicatives sont stockées sur des disques externes, par exemple faisant partie d'une baie. Dans ce cas, il est logique de les gérer sous forme RAID matériel. Chaque disque RAID est associé à un PV, lui-même associé à un VG. On utilise un VG par grosse application (Apache, MySQL, ...). Le VG est ensuite divisé en LV destinés à abriter les FS de l'application.

Remarque : Selon les cas, /home est considéré comme un FS système ou applicatif.

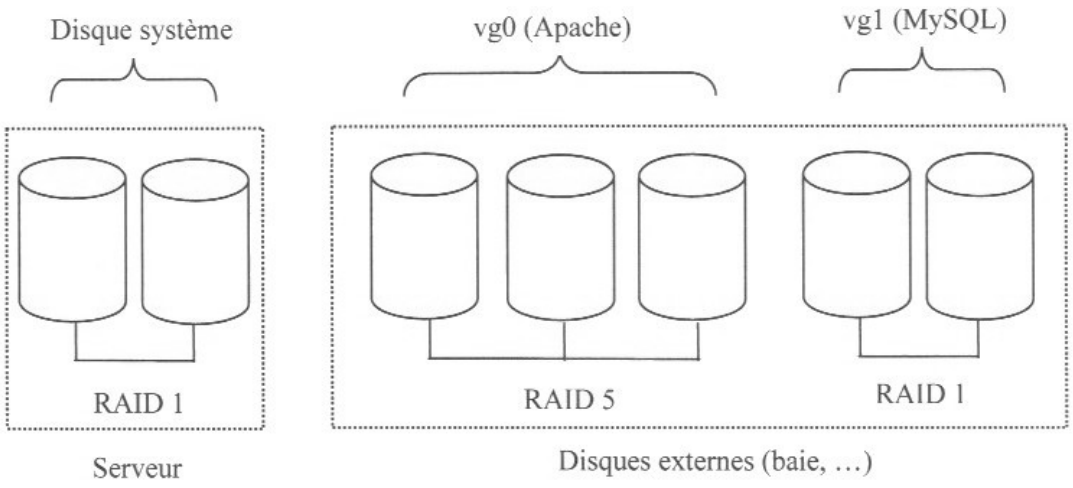


Fig. Un gros serveur : RAID et LVM

ATELIERS



Tâche 1 : Créer un disque RAID 5 logiciel 15mn

Tâche 2 : Créer un VG sur un disque RAID..... 5mn

Tâche 1 : Créer un disque RAID 5 logiciel

1. Créer 4 partitions (6, 7, 8 et 9) de type fd.

On crée 4 partitions. En exploitation, ces partitions doivent correspondre à des partitions appartenant à des disques différents. Dans l'atelier, on simule plusieurs disques physiques avec un seul disque.

Remarque : Si les partitions existent déjà (elles ont été créées dans les exercices précédents), il suffit de ne plus y accéder (`umount`, ...) et de modifier leur tag (fd).

```
[root@linux1 ~]# fdisk /dev/hda

The number of cylinders for this disk is set to 4111.
There is nothing wrong with that, but this is larger than 1024,
and could in certain setups cause problems with:
1) software that runs at boot time (e.g., old versions of LILO)
2) booting and partitioning software from other OSs
   (e.g., DOS FDISK, OS/2 FDISK)

Command (m for help): p

Disk /dev/hda: 33.8 GB, 33820286976 bytes
255 heads, 63 sectors/track, 4111 cylinders
Units = cylinders of 16065 * 512 = 8225280 bytes

   Device Boot      Start         End      Blocks   Id  System
/dev/hda1   *           1          13      104391   83  Linux
/dev/hda2              14        1288    10241437+  83  Linux
/dev/hda3            1289        1353      522112+  83  Linux
/dev/hda4            1354        4111    22153635    5  Extended
/dev/hda5            1354        1418      522081   82  Linux swap

Command (m for help): n
First cylinder (1419-4111, default 1419):
Using default value 1419
Last cylinder or +size or +sizeM or +sizeK (1419-4111, default 4111): +512M

Command (m for help): t
Partition number (1-6): 6
Hex code (type L to list codes): fd
Changed system type of partition 6 to fd (Linux raid autodetect)
```

On crée de la même manière trois autres partitions (7, 8 et 9).

```
Command (m for help): w
```

```
The partition table has been altered!

Calling ioctl() to re-read partition table.

WARNING: Re-reading the partition table failed with error 16: Device or resource
busy.
The kernel still uses the old table.
The new table will be used at the next reboot.
Syncing disks.
[root@linux1 ~]# partprobe
[root@linux1 ~]# fdisk -l

Disk /dev/hda: 33.8 GB, 33820286976 bytes
255 heads, 63 sectors/track, 4111 cylinders
Units = cylinders of 16065 * 512 = 8225280 bytes

   Device Boot      Start         End      Blocks   Id  System
/dev/hda1   *           1          13      104391   83  Linux
/dev/hda2              14        1288    10241437+  83  Linux
/dev/hda3            1289        1353      522112+  83  Linux
/dev/hda4            1354        4111    22153635    5  Extended
/dev/hda5            1354        1418      522081   82  Linux swap
/dev/hda6            1419        1481      506016   fd  Linux raid autodetect
/dev/hda7            1482        1544      506016   fd  Linux raid autodetect
/dev/hda8            1545        1607      506016   fd  Linux raid autodetect
/dev/hda9            1608        1670      506016   fd  Linux raid autodetect
[root@linux1 ~]#
```

2. Créer le disque RAID.

```
[root@linux1 ~]# rpm -q mdadm
mdadm-1.6.0-3
[root@linux1 ~]# vi cree_raid.sh
mdadm  \
        --create /dev/md0  \
        --chunk=4 \
        --level=5 \
        --raid-devices=3 \
        --spare-devices=1 \
        /dev/hda6 /dev/hda7 /dev/hda8 /dev/hda9
[root@linux1 ~]# sh cree_raid.sh
mdadm: array /dev/md0 started.
```

3. Afficher des informations.

```
[root@linux1 ~]# cat /proc/mdstat
Personalities : [raid5]
md0 : active raid5 hda8[3] hda9[4] hda7[1] hda6[0]
      1011840 blocks level 5, 4k chunk, algorithm 2 [3/2] [UU_]
      [=>...................]  recovery =  9.8% (50688/505920) finish=0.8min
speed=8448K/sec
unused devices: <none>
[root@linux1 ~]# cat /proc/mdstat
Personalities : [raid5]
md0 : active raid5 hda8[2] hda9[3] hda7[1] hda6[0]
      1011840 blocks level 5, 4k chunk, algorithm 2 [3/3] [UUU]
```

```
unused devices: <none>
[root@linux1 ~]# mdadm --query /dev/md0
/dev/md0: 988.13MiB raid5 3 devices, 1 spare. Use mdadm --detail for more
detail.
/dev/md0: No md super block found, not an md component.
[root@linux1 ~]# mdadm --query /dev/hda6
/dev/hda6: is not an md array
/dev/hda6: device 0 in 3 device active raid5 md0.  Use mdadm --examine for more
detail.
[root@linux1 ~]# mdadm --detail /dev/md0
/dev/md0:
        Version : 00.90.01
  Creation Time : Mon Feb  5 16:05:53 2007
     Raid Level : raid5
     Array Size : 1011840 (988.13 MiB 1036.12 MB)
    Device Size : 505920 (494.06 MiB 518.06 MB)
   Raid Devices : 3
  Total Devices : 4
Preferred Minor : 0
    Persistence : Superblock is persistent

    Update Time : Mon Feb  5 16:06:53 2007
          State : clean
 Active Devices : 3
Working Devices : 4
 Failed Devices : 0
  Spare Devices : 1

         Layout : left-symmetric
     Chunk Size : 4K

    Number   Major   Minor   RaidDevice State
       0       3        6        0      active sync   /dev/hda6
       1       3        7        1      active sync   /dev/hda7
       2       3        8        2      active sync   /dev/hda8
       3       3        9       -1      spare   /dev/hda9
           UUID : 522d262b:67456653:a355c0cc:8f88c580
         Events : 0.2
[root@linux1 ~]# mdadm --examine /dev/hda6
/dev/hda6:
          Magic : a92b4efc
        Version : 00.90.00
           UUID : 522d262b:67456653:a355c0cc:8f88c580
  Creation Time : Mon Feb  5 16:05:53 2007
     Raid Level : raid5
    Device Size : 505920 (494.06 MiB 518.06 MB)
   Raid Devices : 3
  Total Devices : 4
Preferred Minor : 0

    Update Time : Mon Feb  5 16:06:53 2007
          State : clean
 Active Devices : 3
Working Devices : 4
```

```
 Failed Devices : 0
  Spare Devices : 1
       Checksum : 2112bab3 - correct
         Events : 0.2

         Layout : left-symmetric
     Chunk Size : 4K

      Number   Major   Minor   RaidDevice State
this     0       3       6        0      active sync   /dev/hda6
   0     0       3       6        0      active sync   /dev/hda6
   1     1       3       7        1      active sync   /dev/hda7
   2     2       3       8        2      active sync   /dev/hda8
   3     3       3       9        3      spare   /dev/hda9
[root@linux1 ~]#
```

4. Utiliser le RAID (on crée un FS dessus).

```
[root@linux1 ~]# mkfs -j -b 4096 -m 0 -R stride=1 -q /dev/md0
[root@linux1 ~]# mount -t ext3 /dev/md0 /mnt/disk
[root@linux1 ~]# cp -r /usr/share/doc/mdadm-1.6.0/* /mnt/disk
[root@linux1 ~]# df -h
Filesystem            Size  Used Avail Use% Mounted on
/dev/hda2             9.7G  4.8G  4.4G  53% /
/dev/hda1              99M  8.4M   86M   9% /boot
none                  220M     0  220M   0% /dev/shm
/dev/hda3             494M   11M  458M   3% /home
/dev/md0              973M   18M  956M   2% /mnt/disk
[root@linux1 ~]# ls /mnt/disk
COPYING  ChangeLog  TODO  lost+found  mdadm.conf-example
[root@linux1 ~]#
```

5. Simuler la panne d'un disque.

```
[root@linux1 ~]# mdadm --manage --set-faulty /dev/md0 /dev/hda8
mdadm: set /dev/hda8 faulty in /dev/md0
[root@linux1 ~]# cat /proc/mdstat
Personalities : [raid5]
md0 : active raid5 hda8[3](F) hda9[4] hda7[1] hda6[0]
      1011840 blocks level 5, 4k chunk, algorithm 2 [3/2] [UU_]
      [=>...................]  recovery =  7.6% (39808/505920) finish=0.9min
speed=7961K/sec
unused devices: <none>
```

Remarques :

1) Comme il y a un disque Spare, le RAID est reconstruit immédiatement, sans attendre le remplacement du disque en panne.

2) Le disque en panne est marqué (F).

```
[root@linux1 ~]# cat /proc/mdstat
Personalities : [raid5]
md0 : active raid5 hda8[3](F) hda9[2] hda7[1] hda6[0]
      1011840 blocks level 5, 4k chunk, algorithm 2 [3/3] [UUU]

unused devices: <none>
```

6. Retirer le disque en panne et le remplacer par un disque neuf.

D'abord on retire le disque en panne logiquement et ensuite, si c'était un vrai disque, on pourrait le retirer physiquement.

On remplace le disque par un nouveau disque : on l'ajoute physiquement et ensuite logiquement par la commande `mdadm`.

Si on veut réutiliser un disque faisant partie d'un ancien RAID au lieu d'utiliser un disque neuf, il faut écraser son super-bloc (`mdadm --zero-superblock`).

```
[root@linux1 ~]# mdadm /dev/md0 --remove /dev/hda8
mdadm: hot removed /dev/hda8
[root@linux1 ~]# mdadm --zero-superblock /dev/hda8
[root@linux1 ~]# mdadm /dev/md0 --add /dev/hda8
mdadm: hot added /dev/hda8
[root@linux1 ~]# cat /proc/mdstat
Personalities : [raid5]
md0 : active raid5 hda8[3] hda9[2] hda7[1] hda6[0]
      1011840 blocks level 5, 4k chunk, algorithm 2 [3/3] [UUU]

unused devices: <none>
[root@linux1 ~]# mdadm --detail /dev/md0 |tail
        Layout : left-symmetric
    Chunk Size : 4K

    Number   Major   Minor   RaidDevice State
       0       3        6        0      active sync   /dev/hda6
       1       3        7        1      active sync   /dev/hda7
       2       3        9        2      active sync   /dev/hda9
       3       3        8       -1      spare   /dev/hda8
          UUID : 522d262b:67456653:a355c0cc:8f88c580
        Events : 0.18
```

Remarque : Si l'on n'avait pas eu de disque Spare, la reconstruction du disque aurait commencé. Dans le cas présent, le disque ajouté devient le nouveau disque Spare.

7. Arrêter le RAID.

```
[root@linux1 ~]# umount /mnt/disk
[root@linux1 ~]# mdadm --stop /dev/md0
[root@linux1 ~]# cat /proc/mdstat
Personalities : [raid5]
unused devices: <none>
[root@linux1 ~]#
```

Tâche 2 : Créer un VG sur un disque RAID

1. Activer le disque RAID.

```
[root@linux1 ~]# vi /etc/mdadm.conf
DEVICE  /dev/hda6 /dev/hda7 /dev/hda8 /dev/hda9
ARRAY   /dev/md0  devices=/dev/hda6,/dev/hda7,/dev/hda8,/dev/hda9
[root@linux1 ~]# mdadm --assemble --scan
mdadm: /dev/md0 has been started with 3 drives and 1 spare.
[root@linux1 ~]# cat /proc/mdstat
Personalities : [raid5]
md0 : active raid5 hda6[0] hda8[3] hda9[2] hda7[1]
      1011840 blocks level 5, 4k chunk, algorithm 2 [3/3] [UUU]
```

```
unused devices: <none>
```

2. Créer un PV sur le RAID et ensuite un VG, LV et FS.

```
[root@linux1 ~]# vgscan
  Reading all physical volumes.  This may take a while...
  No volume groups found
[root@linux1 ~]# pvcreate /dev/md0
  Physical volume "/dev/md0" successfully created
[root@linux1 ~]# vgcreate vg0 /dev/md0
  Volume group "vg0" successfully created
[root@linux1 ~]# lvcreate -L100M -n lv0 vg0
  Logical volume "lv0" created
[root@linux1 ~]# mkfs -t ext3 -q /dev/vg0/lv0
[root@linux1 ~]# mount -t ext3 /dev/vg0/lv0 /mnt/disk
[root@linux1 ~]# df /mnt/disk
Filesystem           1K-blocks      Used Available Use% Mounted on
/dev/mapper/vg0-lv0      99150      5664     88366   7% /mnt/disk
```

3. Détruire tout.

```
[root@linux1 ~]# umount /mnt/disk
[root@linux1 ~]# lvremove vg0
Do you really want to remove active logical volume "lv0"? [y/n]: y
  Logical volume "lv0" successfully removed
[root@linux1 ~]# vgremove vg0
  Volume group "vg0" successfully removed
[root@linux1 ~]# pvremove /dev/md0
  Labels on physical volume "/dev/md0" successfully wiped
[root@linux1 ~]# mdadm --stop /dev/md0
[root@linux1 ~]# mdadm --zero-superblock /dev/hda9
[root@linux1 ~]# mdadm --zero-superblock /dev/hda8
[root@linux1 ~]# mdadm --zero-superblock /dev/hda7
[root@linux1 ~]# mdadm --zero-superblock /dev/hda6
[root@linux1 ~]# mdadm --examine /dev/hda6
mdadm: No super block found on /dev/hda6 (Expected magic a92b4efc, got 00000000)
```

4. Détruire (éventuellement) les partitions.

```
[root@linux1 ~]# fdisk /dev/hda
...
Command (m for help): d
Partition number (1-9): 9
```

Idem pour les partitions 8,7 et 6.

```
Command (m for help): w
...
[root@linux1 ~]# partprobe
```

- *VG, PV, LV, PE, LE*
- *ext2online*
- *vgchange -ay*
- *Snapshot*

8

Le LVM

Objectifs

Ce chapitre traite de la gestion des disques avec le LVM. La compréhension de ce chapitre est cruciale pour la gestion d'un gros serveur mais également d'un petit dans la mesure où l'installation par défaut utilise les volumes logiques. A la fin du chapitre on sait créer et agrandir des groupes de volumes et des systèmes de fichiers, mais également on sait accomplir toutes les opérations annexes associées à la gestion LVM.

Contenu

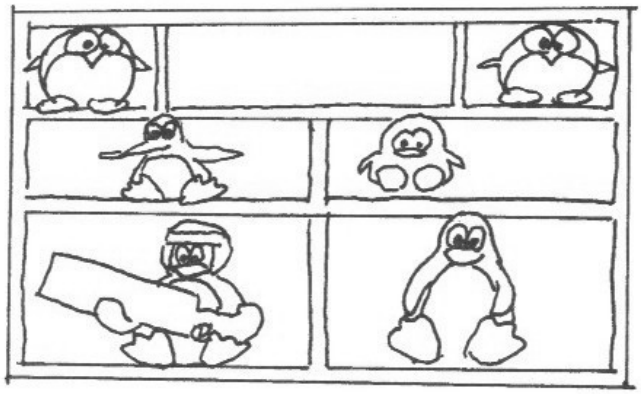
LVM – les éléments clés

L'aspect dynamique des LV et des FS

Les snapshots

LVM – compléments

Ateliers



LVM – les éléments clés

La théorie

Intérêt de l'approche LVM

Un système de fichiers (FS) est installé normalement dans une partition d'un disque dur. En conséquence, un FS ne peut dépasser la taille d'un disque et il est difficile de l'étendre. Il faut sauvegarder les données, repartitionner le disque et enfin restaurer les données. L'approche LVM (Logical Volume Manager) est plus souple. Avec cette approche, un FS est créé dans un volume logique. Ce dernier peut s'étendre sur plusieurs disques et on peut l'agrandir ce qui permet ensuite d'agrandir le FS. Dés que l'on gère de gros serveurs, cette approche est la seule raisonnable.

Les concepts de base : VG, PV, LV

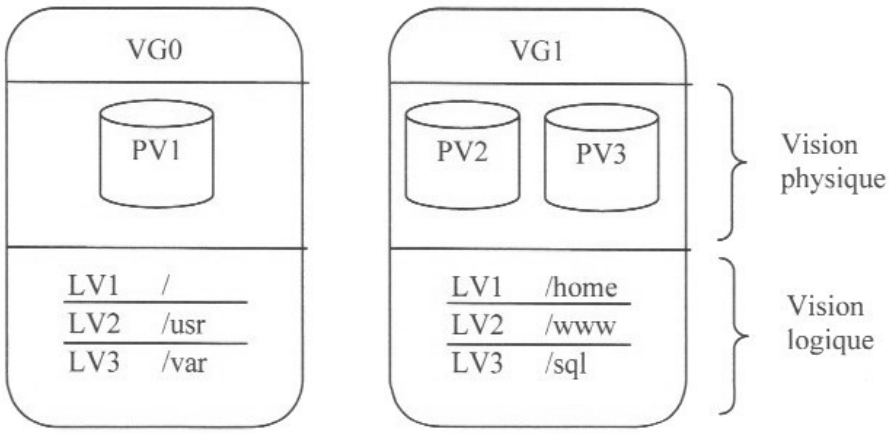


Fig. VG, PV, LV

Un volume logique (Logical Volume ou LV) est l'espace disque abstrait utilisé par l'administrateur à des fins systèmes, principalement en tant que conteneur de FS. Mais un LV peut être également utilisé comme swap ou disque dédié à une application.

Un volume logique est crée à l'intérieur d'un Groupe de volume (Volume Group ou VG). Ce VG est un ensemble de volumes physiques (Physical Volume ou PV). En première approximation un PV est un disque physique.

Ainsi l'espace disque est réparti en VG. Chaque VG a deux aspects :

- Un VG est un ensemble de PV (aspect physique).
- Un VG est un ensemble de LV (aspect logique). La plupart abritant des FS.

Le principal avantage à l'approche LVM est son aspect dynamique :

- Dynamisme des LV : tant qu'il reste de la place dans un VG, on peut étendre ses LV.
- Dynamisme des VG : l'ajout de PV à un VG augmente la place libre du VG.

Remarques :

1) Le fait d'agrandir un LV n'agrandit pas automatiquement le FS associé. Il faut réaliser cette opération séparément par la suite.
2) Rapetisser un FS (et son LV associé) ou un VG est possible. Mais ses opérations sont plus complexes et doivent être évitées. Le plus simple est d'agrandir les LV petit à petit.

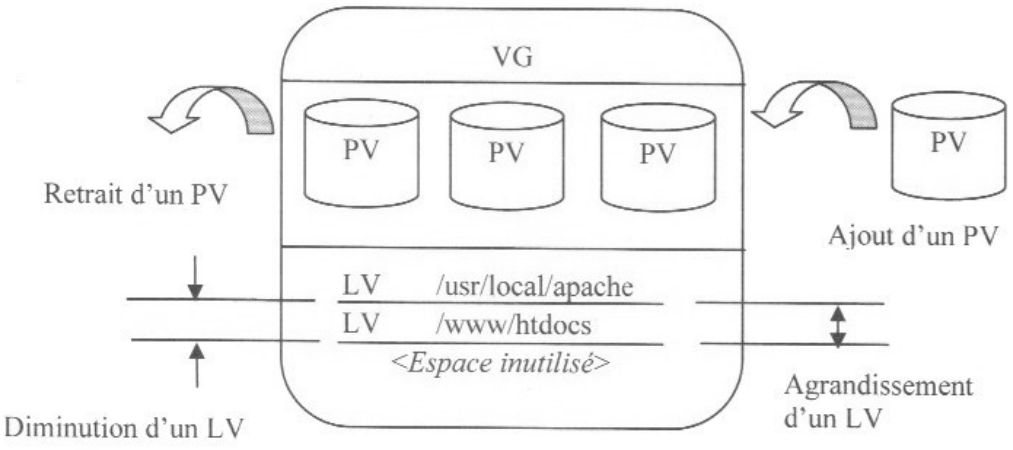


Fig. Aspect dynamique du LVM

Qu'est-ce qu'un PV ?

Typiquement un PV est une partition d'un disque dur. Son type (Tag) doit être 8e pour qu'elle soit reconnue comme abritant un PV par le noyau au démarrage. Le disque système peut contenir les données systèmes dans des partitions classiques (/, /boot, /usr, /var, /tmp, le swap) et le reste de l'espace disque peut correspondre à une partition LVM hébergeant un PV. Par contre pour les disques de données, on a intérêt à ne créer qu'une seule partition de type LVM sur laquelle on installe un PV.

Il est possible de gérer complètement ses FS systèmes (/, /usr, /var, ...) sous forme de LV, sauf /boot. En effet, les chargeurs (Grub, Lilo) ne savent pas gérer actuellement le LVM. En fait l'installation par défaut des systèmes RedHat utilise cette stratégie.

Un PV peut être également un disque complet dont on a effacé le 1er block (le MBR). Ce peut être également un simple fichier grâce au pilote loopback. Enfin, ce peut être un RAID logiciel.

Structure des VG, PV, LV

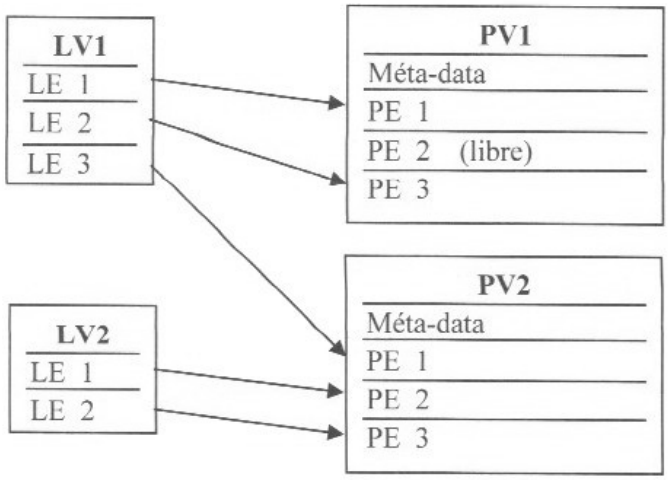


Fig. PV, LV, méta-data, PE et LE

Un VG est un espace disque. Le PE (Physical Extend) constitue sa granularité. La taille du PE est fixée à la création du VG, par défaut elle est de 4Mo. Ainsi un VG est un pool de PE formé par l'ensemble des PE des différents PV qui composent le VG. Certains sont alloués (ils font partie d'un LV), d'autres sont libres.

Un LV est composé de LE (Logical Extend). En fait chaque LE correspond à un PE. Les LE d'un LV peuvent s'étendre sur un ou plusieurs PV d'un même VG.

Un disque ou une partition doivent être « formatés » pour pouvoir jouer le rôle d'un PV. Ce formatage écrit sur le disque des metadatas qui contiendra les tables d'allocations.

VG actif ou non

Pour pouvoir accéder aux LV d'un VG, ce dernier doit être actif. L'activation de l'ensemble des VG est normalement faite au démarrage. Inversement la désactivation des VG est réalisée lors de l'arrêt du système. Pour des raisons de maintenance, on peut être amené à activer/désactiver un VG particulier.

Le savoir concret

Les principales commandes LVM de gestion de PV

`pvscan`	Recherche les PV présents.
`lvmdiskscan`	Recherche les périphériques de type blocs susceptibles d'abriter des PV (on pourra créer des PV dessus).
`pvs`	Affiche des informations sur un PV.
`pvdisplay`	Idem.
`pvcreate`	Crée (formate) un PV, c'est-à-dire écrit une entête sur le PV.

Les principales commandes LVM de gestion de VG

`vgcreate`	Crée un VG. On spécifie un ou plusieurs PV. On peut indiquer la taille du PE.
`vgchange`	Active/Désactive un VG ou tous les VG.
`vgdisplay`	Affiche la liste des VG et leurs caractéristiques. L'option –v permet de visualiser la liste des LV et des PV de chaque VG.
`vgscan`	Recherche les VG.
`vgs`	Affiche les principales caractéristiques des VG.
`vgextend`	Agrandit un VG : on lui ajoute un ou plusieurs PV.
`vgreduce`	Réduit un VG : on luit retire un ou plusieurs PV.
`vgremove`	Supprime un VG.

Les principales commandes LVM de gestion de LV

`lvcreate`	Crée un LV, on spécifie sa taille en LE ou en Mo.
`lvscan`	Recherche les LV.
`lvs`	Affiche les principales caractéristiques d'un LV.
`lvdisplay`	Affiche les caractéristiques d'un LV.
`lvremove`	Supprime un LV.
`lvextend`	Agrandit un LV. Il faut ensuite agrandir le FS.
`lvreduce`	Diminue la taille d'un LV. Il faut au préalable diminuer la taille du FS.

Activation/Désactivation du LVM

Lors du démarrage, un RC doit activer les tables systèmes LVM avant le montage des FS présents dans des VG. Ce RC active principalement les commandes suivantes :

`vgscan` (ou lvm vgscan)

`vgchange -ay` (ou lvm vgchange –ay)

Lors de l'arrêt, Ce RC désactive les VG :

`vgchange -an` (ou lvm vgchange –an)

Remarque : Il est intéressant de connaître ces commandes notamment quand on accède à des LV en mode maintenance. Dans ce cas, il faut activer le LVM manuellement.

Pour en savoir plus

Les pages de manuel

lvcreate(8), lvdisplay(8), lvextend(8), lvmdiskscan(8), lvreduce(8), lvremove(8), lvs(8), lvscan(8), pvcreate(8), pvdisplay(8), pvs(8), pvscan(8), vgchange(8),vgcreate(8), vgdisplay(8), vgextend(8), vgreduce(8), vgremove(8), vgs(8), vgscan(8)

Howto

LVM-Howto

L'aspect dynamique des LV et des FS

La théorie

Le LVM offre un grand dynamisme :

- On peut agrandir ou rétrécir un VG (en lui ajoutant ou en lui retirant des PV).
- On peut agrandir ou rétrécir un LV (en lui ajoutant ou en lui retirant des LE).

Mais ce qui nous importe en final est le dynamisme des données.

Cas général : pas de dynamisme (ou partiel) de la gestion des données

1) On sauvegarde les données.
2) On arrête l'accès aux données (`umount`, ...).
3) On retaille le LV.
4) On réinstalle les données.
5) On redémarre l'exploitation (on réactive les applications, ...).

Remarques :

1) Dans le cas d'un LV de Swap, on n'a pas besoin de faire de sauvegarde.

2) Dans le cas d'un LV abritant un FS, la plupart des FS peuvent au moins être agrandi. Dans ce cas là, il n'est pas nécessaire de réaliser leur sauvegarde. Par contre, si on veut les rétrécir, il faut le plus souvent opérer selon la recette présentée.

Le savoir concret

Les commandes Ext2/Ext3

`resize2fs` Agrandit ou rétrécit un FS Ext2/Ext3 non monté.

`ext2online` Agrandit un FS Ext2/Ext3 monté.

Les commandes ReiserFS

`resize_reiserfs` Agrandit ou rétrécit un FS ReiserFS monté ou non.

Les commandes XFS

`xfs_growfs` Agrandit un FS XFS monté.

Remarques :

1) L'agrandissement d'un FS XFS ne se fait qu'à chaud.

2) Il n'est pas possible de rétrécir un FS XFS.

Pour en savoir plus

Les pages de manuel

resize2fs(8), ext2online(8), resize_reiserfs(8), xfs_growfs(8)

Les snapshots

La théorie

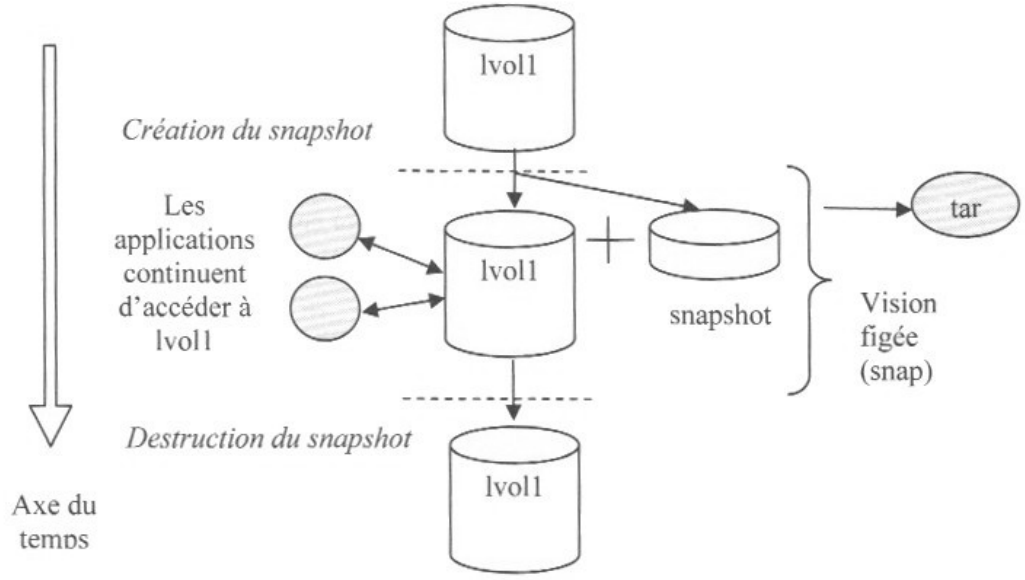


Fig. Snapshot

Les instantanés (Snapshot) LVM permettent une sauvegarde temps réel des données abritées par un LV. Un snapshot contient les pages d'un LV avant leur modification. Ces pages sont écrites dans le snapshot avant que la page correspondante du LV d'origine soit modifiée.

ATTENTION : Le snaphost ne résout pas tous les problèmes. Il faut bien connaître son application pour l'utiliser intelligemment. Par exemple, une sauvegarde d'une base MySQL pourrait être réalisée par le script suivant : verrouillage des tables, démarrage d'un journal des transactions, création du snapshot, déverrouillage des tables, sauvegarde et enfin destruction du snapshot.

Le savoir concret

Focus : La création d'un snapshot

```
# lvcreate --size 100m --snapshot --name snap /dev/vg00/lvol1
```

Cette commande crée le volume logique /dev/vg00/snap. Ce LV joue le rôle du Snapshot. A partir de sa création et jusqu'à sa destruction, il va contenir la vision figée du LV /dev/vg00/lvol1. Si ce LV contient un FS, on peut monter le snaphot et le sauvegarder. On sauvegardera ainsi la vision figée du FS contenu dans lvol1. La taille du snaphot est ici de 100 Mo. Cette taille doit être supérieure à la taille des données modifiées pendant la sauvegarde.

Pour en savoir plus

Les pages de manuel

lvcreate(8)

Internet

Backup with snaphosts (extrait du LVM-HOWTO)
http://tldp.org/HOWTO/LVM-HOWTO/snapshots_backup.html

LVM – Compléments

La théorie

LVM1 et LVM2

Il existe deux versions des pilotes et des commandes LVM : LVM1 et LVM2. LVM2 utilise de nouvelles structures de données (metadata), mais il est compatible avec celles de LVM1 exception faite des snapshots. Par contre, dans un même VG, les metadatas doivent être de même type sur chaque PV. Le LVM2 repose sur le device-mapper du noyau 2.6. Ce pilote gère une indirection entre les blocs d'un LV et leur emplacement réel.

Structure d'un PV

Dans un PV, avant l'espace des PE il y a une zone système (metadata) qui contient les données de gestion LVM : Une zone dite PVRA (PV Reserved Area) qui contient notamment l'identifiant unique d'un PV, son UUID. Ensuite, il y a la VGRA (VG Reserved Area) une table associée à la gestion du VG (tables d'allocation des PE/LE, …). Un PV contient 0, 1 ou 2 zones VGRA. Dans le cas où il y 2 VGRA, la deuxième est en fin du PV. Les VGRA existent selon deux formats lvm1 et lvm2. Le format lvm1 peut être utilisé avec les pilotes LVM2, mais le format lvm2 natif offre plus de fonctionnalités.

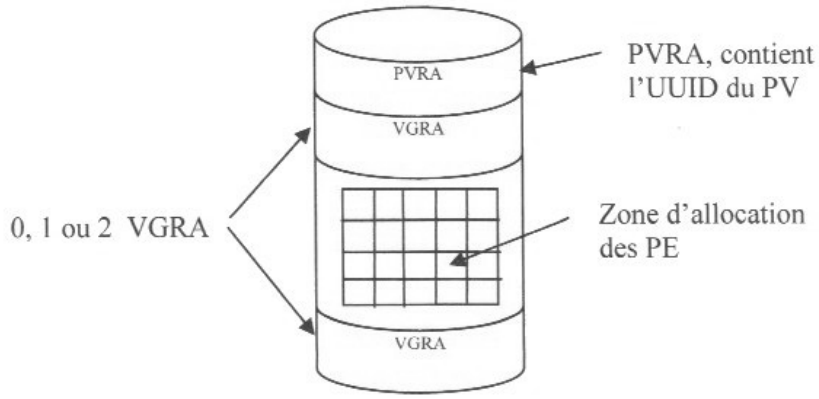


Fig. Structure d'un PV

Sauvegarde d'un VG

La sauvegarde d'un VG se fait en deux temps :

- Sauvegarde des données (FS, …).
- Sauvegarde des metadatas.

La restauration elle, se fait en trois temps :

- Création des PV avec les UUID d'origine.
- Restauration des VGRA. Le VG est alors opérationnel, mais les LV sont vides.
- Restauration des données (FS, …).

Réorganisation des données

Il est possible de réorganiser les données :

- Découper en deux un VG : le nouveau VG contiendra une partie des PV de l'ancien.
- Fusionner deux VG en un seul.

- Déplacer des PE d'un PV à un autre. On peut déplacer tous les PE ou uniquement ceux appartenant à un LV.

Remarque : Un LV ne peut être à cheval sur deux VG.

Stratégie d'allocation des LE, stripping, RAID

Par défaut, les LE d'un LV sont alloués de manière contigus sur un même PV. Si il n'y a plus de place sur un PV, les LE suivants sont alloués sur un autre PV et ainsi de suite.

On peut utiliser la stratégie du « stripping » Dans ce cas, les écritures se feront par « chunk », chaque chunk (ou strip) est écrit sur un PV différent. On choisit la taille des chunk et le nombre de PV impliqués.

La stratégie du stripping améliore les performances dans le cas où les PV correspondent chacun à un disque SCSI.

L'usage d'un disque RAID de type 5 comme support du PV, permet également d'améliorer les performances et en plus de manière transparente.

FS root sous forme de LV

Si l'on veut avoir son FS root (« / ») sous forme de LVM, l'initrd doit activer le LV qui le contient. Si le fichier /etc/fstab indique que le FS root est stocké dans un LV, la commande `mkinitrd` ajoutera les pilotes nécessaires.

Remarque : L'ensemble des FS y compris / peuvent être dans des LV, par contre /boot, qui contient le noyau et l'initrd doivent être dans un FS. En effet les chargeurs comme Grub ou Lilo ne savent pas lire les disques en LVM.

Import/Export, clusters

L'opération d'exportation d'un VG fait qu'il devient inconnu du système. L'intérêt, est que cela permet son importation par un autre système.

Par exemple, les disques qui composent un VG peuvent appartenir à une baie de disques et être reliés à plusieurs ordinateurs. Les commandes d'import/export permettent facilement de basculer leur gestion d'un ordinateur à l'autre. Par contre, contrairement au système GFS, le pilote LVM n'est pas conçu pour une gestion simultanée des données par plusieurs hôtes. Des commandes et des bibliothèques permettent néanmoins une approche cluster limitée (cf. clvmd(8).

Le savoir concret

Autres commandes LVM de gestion de PV

`pvremove`	Supprime l'entête d'un PV (on peut aussi utiliser `dd`).
`pvresize`	Retaille un PV. Par exemple après avoir agrandit une partition avec `fdisk`.
`pvmove`	Déplace les PE d'un PV à un autre. Eventuellement on peut se limiter aux PE d'un LV particulier.

Autres commandes LVM de gestion de VG

`vgimport`	Le VG devient connu du système (il a été au préalable exporté par un autre système).
`vgexport`	Le VG devient inconnu du système (il peut ensuite être importé par un autre système).

`vgck`	Vérifie la cohérence des zones metadatas.
`vgcfgbackup`	Sauvegarde les zones metadatas d'un VG.
`vgcfgrestore`	Restaure les zones metadatas d'un VG.
`vgconvert`	Convertit le format des metadatas d'un VG du format lvm1 à lvm2.
`vgsplit`	Crée un nouveau VG à partir d'un autre VG en lui prenant certains de ses PV. On rappelle qu'un LV ne peut pas être à cheval sur deux VG.
`vgmerge`	Fusionne un VG dans un autre (opération inverse de vgsplit).
`vgrename`	Renomme un VG.
`vgchange`	Permet de changer certaines caractéristiques d'un VG (actif ou non, ...).
`vgmknodes`	Recrée les fichiers spéciaux associés à un LV.

Autres commande LVM de gestion de LV

`lvchange`	Change les caractéristiques d'un LV (actif ou non, accessible en lecture/écriture, stratégie d'allocation, ...).
`lvcreate`	Crée un LV, on peut choisir la stratégie d'allocation.

Les fichiers

/etc/lvm/lvm.conf	Fichier de configuration du LVM (disques à balayer, journaux de bords, sauvegardes des metadatas).
/etc/lvm/backup/	Sauvegarde des metadatas (un fichier par VG).

Pour en savoir plus

Les pages de manuel

clvmd(8), dmsetup(8), lvchange(8), lvcreate(8), lvdisplay(8), lvextend(8), lvmchange(8), lvmdiskscan(8), lvreduce(8), lvremove(8), lvrename(8), lvresize(8), lvs(8), lvscan(8), pvchange(8), pvcreate(8), pvdisplay(8), pvmove(8), pvremove(8), pvresize(8), pvs(8), pvscan(8), vgcfgbackup(8), vgcfgrestore(8), vgchange(8), vgck(8), vgconvert(8), vgcreate(8), vgdisplay(8), vgexport(8), vgextend(8), vgimport(8), vgmerge(8), vgmknodes(8), vgreduce(8), vgremove(8), vgrename(8), vgs(8), vgscan(8), vgsplit(8), lvm.conf(5)

Howto

LVM-Howto : décrit notamment comment convertir un système LVM1 en LVM2.

ATELIERS



Tâche 1 : Créer des PV 10mn
Tâche 2 : Créer des VG et des LV 10mn
Tâche 3 : Détruire des LV, des VG et des PV 5mn
Tâche 4 : Utiliser les LV comme FS, extension d'un FS, d'un LV, d'un VG 10mn
Tâche 4 bis : (avec ReiserFS) 5mn
Tâche 4 ter : (avec XFS) 5mn
Tâche 5 : Réduction d'un FS, d'un LV d'un VG 10mn
Tâche 6 : Utiliser un LV comme swap 10mn
Tâche 7 : Utiliser un LV dédié à une application 5mn
Tâche 8 : Exporter/Importer un VG 5mn
Tâche 9 : Accéder à des FS présents dans des LV en mode maintenance 10mn
Tâche 10 : Sauvegarder un VG 10mn
Tâche 11 : Snapshot 5mn
Tâche 12 : LV « stripped » 5mn
Tâche 13 : Détruire tout (y compris les partitions) 5mn

Tâche 1 : Créer des PV

1. Créer quatre partitions.

Chaque partition (/dev/hda6, /dev/hda7, /dev/hda8, /dev/hda9) simule un disque complet (/dev/sdb, /dev/sdc, /dev/sdd, /dev/sde). Chacune de ces partitions (ou disque) étant destinée à devenir un PV.

```
[root@linux1 ~]# fdisk /dev/hda

The number of cylinders for this disk is set to 4111.
There is nothing wrong with that, but this is larger than 1024,
and could in certain setups cause problems with:
1) software that runs at boot time (e.g., old versions of LILO)
2) booting and partitioning software from other OSs
   (e.g., DOS FDISK, OS/2 FDISK)

Command (m for help): p

Disk /dev/hda: 33.8 GB, 33820286976 bytes
255 heads, 63 sectors/track, 4111 cylinders
Units = cylinders of 16065 * 512 = 8225280 bytes

   Device Boot      Start         End      Blocks   Id  System
/dev/hda1   *           1          13      104391   83  Linux
/dev/hda2              14        1288    10241437+  83  Linux
/dev/hda3            1289        1353      522112+  82  Linux swap
/dev/hda4            1354        4111    22153635    5  Extended
/dev/hda5            1354        1418      522081   83  Linux
```

```
Command (m for help): n
First cylinder (1419-4111, default 1419):
Using default value 1419
Last cylinder or +size or +sizeM or +sizeK (1419-4111, default 4111): +512M

Command (m for help): t
Partition number (1-6): 6
Hex code (type L to list codes): 8e
Changed system type of partition 6 to 8e (Linux LVM)

Command (m for help): p

Disk /dev/hda: 33.8 GB, 33820286976 bytes
255 heads, 63 sectors/track, 4111 cylinders
Units = cylinders of 16065 * 512 = 8225280 bytes

   Device Boot      Start         End      Blocks   Id  System
/dev/hda1   *           1          13      104391   83  Linux
/dev/hda2              14        1288    10241437+  83  Linux
/dev/hda3            1289        1353      522112+  82  Linux swap
/dev/hda4            1354        4111    22153635    5  Extended
/dev/hda5            1354        1418      522081   83  Linux
/dev/hda6            1419        1481      506016   8e  Linux LVM
```

On crée de même trois autres partitions (7,8 et 9) et ensuite on sauvegarde.

Remarque : Depuis la version 2 de LVM, il n'est plus nécessaire que les partitions possède le tag 8e.

```
Command (m for help): w
The partition table has been altered!

Calling ioctl() to re-read partition table.

WARNING: Re-reading the partition table failed with error 16: Device or resource
busy.
The kernel still uses the old table.
The new table will be used at the next reboot.
Syncing disks.
[root@linux1 ~]#
```

2. Écrire la table des partitions sur disque.

```
[root@linux1 ~]# partprobe
[root@linux1 ~]# fdisk -l

Disk /dev/hda: 33.8 GB, 33820286976 bytes
255 heads, 63 sectors/track, 4111 cylinders
Units = cylinders of 16065 * 512 = 8225280 bytes

   Device Boot      Start         End      Blocks   Id  System
/dev/hda1   *           1          13      104391   83  Linux
/dev/hda2              14        1288    10241437+  83  Linux
/dev/hda3            1289        1353      522112+  82  Linux swap
/dev/hda4            1354        4111    22153635    5  Extended
/dev/hda5            1354        1418      522081   83  Linux
```

```
/dev/hda6            1419        1481      506016   8e  Linux LVM
/dev/hda7            1482        1544      506016   8e  Linux LVM
/dev/hda8            1545        1607      506016   8e  Linux LVM
/dev/hda9            1608        1670      506016   8e  Linux LVM
[root@linux1 ~]#
```

3. Créer trois PV pour les trois premières partitions de type 8e.

```
[root@linux1 ~]# pvs
[root@linux1 ~]# pvcreate /dev/hda6
  Physical volume "/dev/hda6" successfully created
[root@linux1 ~]# pvcreate /dev/hda7
  Physical volume "/dev/hda7" successfully created
[root@linux1 ~]# pvcreate /dev/hda8
  Physical volume "/dev/hda8" successfully created
[root@linux1 ~]# pvs
  PV         VG   Fmt  Attr PSize   PFree
  /dev/hda6       lvm2 --   493.97M 493.97M
  /dev/hda7       lvm2 --   493.97M 493.97M
  /dev/hda8       lvm2 --   493.97M 493.97M
[root@linux1 ~]# pvdisplay /dev/hda6
  --- NEW Physical volume ---
  PV Name               /dev/hda6
  VG Name
  PV Size               493.97 MB
  Allocatable           NO
  PE Size (KByte)       0
  Total PE              0
  Free PE               0
  Allocated PE          0
  PV UUID               Qh5DjT-QGfo-midq-2ZIs-dExp-D7TB-P6uKlR

[root@linux1 ~]# pvscan
  PV /dev/hda6         lvm2 [493.97 MB]
  PV /dev/hda7         lvm2 [493.97 MB]
  PV /dev/hda8         lvm2 [493.97 MB]
  Total: 3 [1.45 GB] / in use: 0 [0   ] / in no VG: 3 [1.45 GB]
[root@linux1 ~]# lvmdiskscan
  /dev/hda1 [      101.94 MB]
  /dev/hda2 [        9.77 GB]
  /dev/hda3 [      509.88 MB]
  /dev/hda5 [      509.84 MB]
  /dev/hda6 [      494.16 MB] LVM physical volume
  /dev/hda7 [      494.16 MB] LVM physical volume
  /dev/hda8 [      494.16 MB] LVM physical volume
  /dev/hda9 [      494.16 MB]
  0 disks
  5 partitions
  0 LVM physical volume whole disks
  3 LVM physical volumes
[root@linux1 ~]# lvmdiskscan -l
  WARNING: only considering LVM devices
  /dev/hda6 [      494.16 MB] LVM physical volume
  /dev/hda7 [      494.16 MB] LVM physical volume
  /dev/hda8 [      494.16 MB] LVM physical volume
```

```
  0 LVM physical volume whole disks
  3 LVM physical volumes
[root@linux1 ~]#
```

Tâche 2 : Créer des VG et des LV

1. Lister les VG (il n'y en a pas).

```
[root@linux1 ~]# vgs
[root@linux1 ~]# vgdisplay
[root@linux1 ~]# lvm vgdisplay
[root@linux1 ~]#
```

Remarque : la commande `lvm` peut préfixer toutes les commandes LVM.

2. Créee le VG vg0 en utilisant les deux PV hda6 et hda7.

```
[root@linux1 ~]# vgcreate vg0 /dev/hda6 /dev/hda7
  Volume group "vg0" successfully created
[root@linux1 ~]# pvs
  PV         VG   Fmt  Attr PSize   PFree
  /dev/hda6  vg0  lvm2 a-   488.00M 488.00M
  /dev/hda7  vg0  lvm2 a-   488.00M 488.00M
  /dev/hda8       lvm2 --   493.97M 493.97M
[root@linux1 ~]# vgs
  VG   #PV #LV #SN Attr  VSize   VFree
  vg0    2   0   0 wz--n 976.00M 976.00M
[root@linux1 ~]# vgdisplay
  --- Volume group ---
  VG Name               vg0
  System ID
  Format                lvm2
  Metadata Areas        2
  Metadata Sequence No  1
  VG Access             read/write
  VG Status             resizable
  MAX LV                0
  Cur LV                0
  Open LV               0
  Max PV                0
  Cur PV                2
  Act PV                2
  VG Size               976.00 MB
  PE Size               4.00 MB
  Total PE              244
  Alloc PE / Size       0 / 0
  Free  PE / Size       244 / 976.00 MB
  VG UUID               JU5Db1-DyX1-0PKu-ooUz-8g4e-P4WL-32qiJk

[root@linux1 ~]#
```

Remarque : Le champ Free PE/Size affiche la place restante libre du VG.

3. Créee un LV en spécifiant sa taille en mega-octets.

```
[root@linux1 ~]# lvcreate -L 100M vg0
  Logical volume "lvol0" created
```

Remarque : Le nom par défaut d'un LV est lvl suivit du numéro du LV. Par exemple : /dev/vg0/lvol0, /dev/vg0/lvol1, etc. Le nom du LV est bien sur crée dans l'arborescence associée au VG (ici /dev/vg0).

4. Créer un LV en spécifiant sa taille en LE, lui donner le nom oralog.

```
[root@linux1 ~]# lvcreate -l 30 -n oralog vg0
  Logical volume "oralog" created
[root@linux1 ~]#
```

5. Lister les LV.

a) L'ensemble des LV.

```
[root@linux1 ~]# lvs
  LV     VG   Attr   LSize   Origin Snap%  Move Copy%
  lvol0  vg0  -wi-a- 100.00M
  oralog vg0  -wi-a- 120.00M
```

b) Uniquement les LV d'un VG particulier.

```
[root@linux1 ~]# lvs vg0
  LV     VG   Attr   LSize   Origin Snap%  Move Copy%
  lvol0  vg0  -wi-a- 100.00M
  oralog vg0  -wi-a- 120.00M
[root@linux1 ~]#
```

6. Afficher les caractéristiques d'un LV.

```
[root@linux1 ~]# lvdisplay /dev/vg0/oralog
  --- Logical volume ---
  LV Name                /dev/vg0/oralog
  VG Name                vg0
  LV UUID                pdb2WT-ZhBD-dWQr-JXAX-vu32-V3FV-dGd5kR
  LV Write Access        read/write
  LV Status              available
  # open                 0
  LV Size                120.00 MB
  Current LE             30
  Segments               1
  Allocation             inherit
  Read ahead sectors     0
  Block device           253:1

[root@linux1 ~]#
```

7. Lister tout.

IMPORTANT ! La commande `vgisplay -v` est la commande la plus importante. Elle liste les différents VG, et pour chacun leurs LV et PV. Elle indique la place restante libre par VG (Free PE) et la taille de chacun des LV et PV.

```
[root@linux1 ~]# vgdisplay -v | more
    Finding all volume groups
    Finding volume group "vg0"
  --- Volume group ---
  VG Name               vg0
  System ID
  Format                lvm2
  Metadata Areas        2
  Metadata Sequence No  3
  VG Access             read/write
```

```
  VG Status              resizable
  MAX LV                 0
  Cur LV                 2
  Open LV                0
  Max PV                 0
  Cur PV                 2
  Act PV                 2
  VG Size                976.00 MB
  PE Size                4.00 MB
  Total PE               244
  Alloc PE / Size        55 / 220.00 MB
  Free  PE / Size        189 / 756.00 MB
  VG UUID                JU5Db1-DyX1-0PKu-ooUz-8g4e-P4WL-32qiJk

  --- Logical volume ---
  LV Name                /dev/vg0/lvol0
  VG Name                vg0
  LV UUID                evTOTA-Kle0-aB3j-gs0w-ZXxr-moYL-Zk88Dk
  LV Write Access        read/write
  LV Status              available
  # open                 0
  LV Size                100.00 MB
  Current LE             25
  Segments               1
  Allocation             inherit
  Read ahead sectors     0
  Block device           253:0

  --- Logical volume ---
  LV Name                /dev/vg0/oralog
  VG Name                vg0
  LV UUID                pdb2WT-ZhBD-dWQr-JXAX-vu32-V3FV-dGd5kR
  LV Write Access        read/write
  LV Status              available
# open                 0
  LV Size                120.00 MB
  Current LE             30
  Segments               1
  Allocation             inherit
  Read ahead sectors     0
  Block device           253:1

  --- Physical volumes ---
  PV Name                /dev/hda6
  PV UUID                Qh5DjT-QGfo-midq-2ZIs-dExp-D7TB-P6uKlR
  PV Status              allocatable
  Total PE / Free PE     122 / 67

  PV Name                /dev/hda7
  PV UUID                dCE3KL-2nmq-Iz1J-d5nG-SMT4-ZEkV-xiLYYV
  PV Status              allocatable
  Total PE / Free PE     122 / 122

[root@linux1 ~]#
```

Tâche 3 :
Détruire des LV, des VG et des PV

1. Détruire des LV (avec ou sans demande de confirmation).

```
[root@linux1 ~]# lvremove /dev/vg0/oralog
Do you really want to remove active logical volume "oralog"? [y/n]: y
  Logical volume "oralog" successfully removed
[root@linux1 ~]# lvremove -f /dev/vg0/lvol0
  Logical volume "lvol0" successfully removed
[root@linux1 ~]# lvs
[root@linux1 ~]#
```

2. Détruire le VG.

```
[root@linux1 ~]# vgremove vg0
  Volume group "vg0" successfully removed
[root@linux1 ~]# vgs
[root@linux1 ~]#
```

3. Détruire un PV.

Soit on utilise la commande `pvremove` ou tout simplement on écrit des zéros en tête de la partition avec la commande `dd`.

```
[root@linux1 ~]# pvs
  PV         VG   Fmt  Attr PSize   PFree
  /dev/hda6       lvm2 --   488.00M 488.00M
  /dev/hda7       lvm2 --   488.00M 488.00M
  /dev/hda8       lvm2 --   493.97M 493.97M
[root@linux1 ~]# pvremove /dev/hda6
  Labels on physical volume "/dev/hda6" successfully wiped
[root@linux1 ~]# pvremove /dev/hda7
  Labels on physical volume "/dev/hda7" successfully wiped
 [root@linux1 ~]# dd if=/dev/zero of=/dev/hda8 bs=1k count=100
100+0 records in
100+0 records out
[root@linux1 ~]# pvs
[root@linux1 ~]#
```

Tâche 4 :
Utiliser des LV comme FS, extension d'un FS (ext2/ext3), d'un LV, d'un VG

1. Créer des PV, un VG (avec des PE de 4 mega-octets) et des LV.

```
[root@linux1 ~]# pvs
[root@linux1 ~]# pvcreate /dev/hda6
  Physical volume "/dev/hda6" successfully created
[root@linux1 ~]# pvcreate /dev/hda7
  Physical volume "/dev/hda7" successfully created
[root@linux1 ~]# vgcreate -s 4M vg0 /dev/hda6 /dev/hda7
  Volume group "vg0" successfully created
[root@linux1 ~]# lvcreate -L 100M vg0
  Logical volume "lvol0" created
[root@linux1 ~]# lvcreate -L 100M vg0
  Logical volume "lvol1" created
[root@linux1 ~]# lvcreate -L 100M vg0
  Logical volume "lvol2" created
[root@linux1 ~]# pvs
  PV         VG   Fmt  Attr PSize   PFree
```

```
  /dev/hda6  vg0  lvm2 a-   492.00M 192.00M
  /dev/hda7  vg0  lvm2 a-   492.00M 492.00M
[root@linux1 ~]# vgs
  VG   #PV #LV #SN Attr  VSize   VFree
  vg0    2   3   0 wz--n 984.00M 684.00M
[root@linux1 ~]# lvs
  LV    VG   Attr   LSize   Origin Snap%  Move Copy%
  lvol0 vg0  -wi-a- 100.00M
  lvol1 vg0  -wi-a- 100.00M
  lvol2 vg0  -wi-a- 100.00M
[root@linux1 ~]# lvremove /dev/vg0/lvol2
Do you really want to remove active logical volume "lvol2"? [y/n]: n
  Logical volume "lvol2" not removed
[root@linux1 ~]# lvremove -f /dev/vg0/lvol2
  Logical volume "lvol2" successfully removed
[root@linux1 ~]# lvremove -f /dev/vg0/lvol1
  Logical volume "lvol1" successfully removed
[root@linux1 ~]# lvs
  LV    VG   Attr   LSize   Origin Snap%  Move Log Copy%
  lvol0 vg0  -wi-a- 100,00M
[root@linux1 ~]#
```

2. Créer un FS dans un LV (on formate un des LV en Ext3).

```
[root@linux1 ~]# mkfs -t ext3 -q /dev/vg0/lvol0
 [root@linux1 ~]#
```

3. Monter le FS.

```
[root@linux1 ~]# mkdir /mnt/disk
[root@linux1 ~]# mount -t ext3 /dev/vg0/lvol0 /mnt/disk
[root@linux1 ~]# lvs
  LV    VG   Attr   LSize   Origin Snap%  Move Copy%
  lvol0 vg0  -wi-ao 100.00M
[root@linux1 ~]# df -h
Filesystem            Size  Used Avail Use% Mounted on
/dev/hda2             9.7G  2.3G  6.9G  25% /
/dev/hda1              99M   11M   83M  12% /boot
none                  220M     0  220M   0% /dev/shm
/dev/hda5             494M   11M  458M   3% /home
/dev/mapper/vg0-lvol0
                       97M  5.6M   87M   7% /mnt/disk
[root@linux1 ~]# df -h /mnt/disk
Filesystem            Size  Used Avail Use% Mounted on
/dev/mapper/vg0-lvol0
                       97M  5.6M   87M   7% /mnt/disk
[root@linux1 ~]#
```

4. Étendre la taille du FS à froid (on démonte le FS).

```
[root@linux1 ~]# umount /mnt/disk
[root@linux1 ~]# lvextend -L +300M /dev/vg0/lvol0
  Extending logical volume lvol0 to 400.00 MB
  Logical volume lvol0 successfully resized
[root@linux1 ~]# lvdisplay /dev/vg0/lvol0
  --- Logical volume ---
  LV Name                /dev/vg0/lvol0
  VG Name                vg0
```

```
  LV UUID                1WxY04-3ZYD-3wYZ-26AC-JzKw-wlls-lqQyYt
  LV Write Access        read/write
  LV Status              available
  # open                 0
  LV Size                400.00 MB
  Current LE             100
  Segments               3
  Allocation             inherit
  Read ahead sectors     0
  Block device           253:0

[root@linux1 ~]# fsck -f /dev/vg0/lvol0
fsck 1.35 (28-Feb-2004)
e2fsck 1.35 (28-Feb-2004)
Pass 1: Checking inodes, blocks, and sizes
Pass 2: Checking directory structure
Pass 3: Checking directory connectivity
Pass 4: Checking reference counts
Pass 5: Checking group summary information
/dev/vg0/lvol0: 11/25688 files (9.1% non-contiguous), 8914/102400 blocks
[root@linux1 ~]# resize2fs /dev/vg0/lvol0
resize2fs 1.35 (28-Feb-2004)
Resizing the filesystem on /dev/vg0/lvol0 to 409600 (1k) blocks.
The filesystem on /dev/vg0/lvol0 is now 409600 blocks long.

[root@linux1 ~]# mount -t ext3 /dev/vg0/lvol0 /mnt/disk
[root@linux1 ~]# df -h /mnt/disk
Filesystem            Size  Used Avail Use% Mounted on
/dev/mapper/vg0-lvol0
                      388M  6.3M  362M   2% /mnt/disk
[root@linux1 ~]#
```

5. Étendre la taille du FS à chaud (sans démonter le FS).

```
[root@linux1 ~]# lvextend -L +100M /dev/vg0/lvol0
  Extending logical volume lvol0 to 500.00 MB
  Logical volume lvol0 successfully resized
[root@linux1 ~]# ext2online /dev/vg0/lvol0
ext2online v1.1.18 - 2001/03/18 for EXT2FS 0.5b
[root@linux1 ~]# df -h /mnt/disk
Filesystem            Size  Used Avail Use% Mounted on
/dev/mapper/vg0-lvol0
                      485M  6.3M  454M   2% /mnt/disk
[root@linux1 ~]#
```

6. Étendre le VG.

```
[root@linux1 ~]# vgs
  VG   #PV #LV #SN Attr   VSize   VFree
  vg0    2   1   0 wz--n- 984.00M 484.00M
[root@linux1 ~]# pvs
  PV         VG   Fmt  Attr PSize   PFree
  /dev/hda6  vg0  lvm2 a-   492.00M      0
  /dev/hda7  vg0  lvm2 a-   492.00M 484.00M
[root@linux1 ~]# pvcreate /dev/hda8
  Physical volume "/dev/hda8" successfully created
[root@linux1 ~]# vgextend vg0 /dev/hda8
```

```
  Volume group "vg0" successfully extended
[root@linux1 ~]# vgs
  VG   #PV #LV #SN Attr   VSize VFree
  vg0    3   3   0 wz--n 1.44G 776.00M
[root@linux1 ~]#
```

7. Démonter le FS.

```
[root@linux1 ~]# umount /mnt/disk
[root@linux1 ~]#
```

Tâche 4 bis (avec ReiserFS)

1. Créer des PV, un VG (avec des PE de 4 mega-octets) et des LV (cf. tâche 4).

2. Créer un FS dans un LV (on formate un des LV en ReiserFS).

```
 [root@linux1 ~]# mkreiserfs /dev/vg0/lvol0
mkreiserfs 3.6.19 (2003 www.namesys.com)

A pair of credits:
Elena Gryaznova performed testing and benchmarking.

Nikita Danilov  wrote  most of the core  balancing code, plugin infrastructure,
and directory code. He steadily worked long hours, and is the reason so much of
the Reiser4 plugin infrastructure is well abstracted in its details.  The carry
function, and the use of non-recursive balancing, are his idea.


Guessing about desired format.. Kernel 2.6.17 is running.
Format 3.6 with standard journal
Count of blocks on the device: 25600
Number of blocks consumed by mkreiserfs formatting process: 8212
Blocksize: 4096
Hash function used to sort names: "r5"
Journal Size 8193 blocks (first block 18)
Journal Max transaction length 1024
inode generation number: 0
UUID: 5584e9a9-1013-4cc5-8b5b-7d417b97e772
ATTENTION: YOU SHOULD REBOOT AFTER FDISK!
        ALL DATA WILL BE LOST ON '/dev/vg0/lvol0'!
Continue (y/n):y
Initializing journal - 0%....20%....40%....60%....80%....100%
Syncing..ok

Tell your friends to use a kernel based on 2.4.18 or later, and especially not a
kernel based on 2.4.9, when you use reiserFS. Have fun.

ReiserFS is successfully created on /dev/vg0/lvol0.
```

3. Monter le FS.

```
[root@linux1 ~]# mkdir /mnt/disk
mkdir: Ne peut crÃ©er le rÃ©pertoire `/mnt/disk': Le fichier existe.
[root@linux1 ~]# mount -t reiserfs /dev/vg0/lvol0 /mnt/disk
[root@linux1 ~]# lvs
  LV    VG   Attr   LSize   Origin Snap%  Move Log Copy%
  lvol0 vg0  -wi-ao 100,00M
[root@linux1 ~]# df -Th /mnt/disk
```

```
Sys. de fich. Type     Tail. Occ. Disp. %Occ. MontÃ© sur
/dev/mapper/vg0-lvol0
          reiserfs    100M   33M   68M  33% /mnt/disk
```

4. Étendre la taille du FS à chaud.

```
[root@linux1 ~]# lvextend -L +300M /dev/vg0/lvol0
  Extending logical volume lvol0 to 400,00 MB
  Logical volume lvol0 successfully resized
[root@linux1 ~]# resize_reiserfs /dev/vg0/lvol0
resize_reiserfs 3.6.19 (2003 www.namesys.com)



resize_reiserfs: On-line resizing finished successfully.

[root@linux1 ~]# df -Th /mnt/disk
Sys. de fich. Type     Tail. Occ. Disp. %Occ. MontÃ© sur
/dev/mapper/vg0-lvol0
          reiserfs    400M   33M  368M   9% /mnt/disk
```

5. Diminuer le FS à froid (on démonte le FS).

```
[root@linux1 ~]# umount /mnt/disk
[root@linux1 ~]# resize_reiserfs -s -300M /dev/vg0/lvol0
resize_reiserfs 3.6.19 (2003 www.namesys.com)

You are running BETA version of reiserfs shrinker.
This version is only for testing or VERY CAREFUL use.
Backup of you data is recommended.

Do you want to continue? [y/N]:y
Processing the tree: 0%....20%....40%....60%....80%....100%
left 0, 0 /sec
nodes processed (moved):
int         0 (0),
leaves      1 (0),
unfm        0 (0),
total       1 (0).

check for used blocks in truncated region

ReiserFS report:
blocksize             4096
block count           25600 (102400)
free blocks           17388 (94185)
bitmap block count    1 (4)

Syncing..done


resize_reiserfs: Resizing finished successfully.

[root@linux1 ~]# lvreduce -L -300M /dev/vg0/lvol0
  WARNING: Reducing active and open logical volume to 100,00 MB
  THIS MAY DESTROY YOUR DATA (filesystem etc.)
Do you really want to reduce lvol0? [y/n]: y
  Reducing logical volume lvol0 to 100,00 MB
```

```
  Logical volume lvol0 successfully resized
 [root@linux1 ~]# mount -t reiserfs /dev/vg0/lvol0 /mnt/disk
[root@linux1 ~]# df -Th /mnt/disk/
Sys. de fich. Type     Tail. Occ. Disp. %Occ. MontÃ© sur
/dev/mapper/vg0-lvol0
          reiserfs    100M   33M   68M  33% /mnt/disk
```

6. Démonter le FS.

```
[root@linux1 ~]# umount /mnt/disk
[root@linux1 ~]#
```

Tâche 4 ter (avec XFS)

1. Créer des PV, un VG (avec des PE de 4 mega-octets) et des LV (cf. tâche 4).

2. Créer un FS dans un LV (on formate un des LV en XFS).

```
[root@linux1 ~]# mkfs -t xfs -f /dev/vg0/lvol0
meta-data=/dev/vg0/lvol0         isize=256    agcount=6, agsize=4096 blks
         =                       sectsz=512   attr=0
data     =                       bsize=4096   blocks=24576, imaxpct=25
         =                       sunit=0      swidth=0 blks, unwritten=1
naming   =version 2              bsize=4096
log      =internal log           bsize=4096   blocks=1200, version=1
         =                       sectsz=512   sunit=0 blks
realtime =none                   extsz=4096   blocks=0, rtextents=0
```

3. Monter le FS.

```
[root@linux1 ~]# mkdir /mnt/disk
mkdir: Ne peut créer le répertoire `/mnt/disk': Le fichier existe.
[root@linux1 ~]# mount -t xfs /dev/vg0/lvol0 /mnt/disk/
[root@linux1 ~]# df -Th /mnt/disk/
Sys. de fich. Type     Tail. Occ. Disp. %Occ. MontÃ© sur
/dev/mapper/vg0-lvol0
               xfs     92M  112K   92M   1% /mnt/disk
```

4. Étendre la taille du FS à chaud.

```
[root@linux1 ~]# lvextend -L +300M /dev/vg0/lvol0
  Extending logical volume lvol0 to 400,00 MB
  Logical volume lvol0 successfully resized
 [root@linux1 ~]# xfs_growfs /dev/vg0/lvol0
meta-data=/dev/vg0/lvol0         isize=256    agcount=6, agsize=4096 blks
         =                       sectsz=512   attr=0
data     =                       bsize=4096   blocks=24576, imaxpct=25
         =                       sunit=0      swidth=0 blks, unwritten=1
naming   =version 2              bsize=4096
log      =internal               bsize=4096   blocks=1200, version=1
         =                       sectsz=512   sunit=0 blks
realtime =none                   extsz=4096   blocks=0, rtextents=0
data blocks changed from 24576 to 102400
[root@linux1 ~]# df -Th /mnt/disk/
Sys. de fich. Type     Tail. Occ. Disp. %Occ. MontÃ© sur
/dev/mapper/vg0-lvol0
               xfs    396M  416K  395M   1% /mnt/disk
```

5. Démonter le FS.

```
[root@linux1 ~]# mkfs -t ext3 -q /dev/vg0/lvol0
[root@linux1 ~]#
```

Tâche 5 : Réduire un FS, un LV et un VG

1. Réduire la taille du FS (on indique sa nouvelle taille).

```
[root@linux1 ~]# umount /mnt/disk
[root@linux1 ~]# df -h /dev/vg0/lvol0
Filesystem            Size  Used Avail Use% Mounted on
-                     220M  168K  220M   1% /dev
[root@linux1 ~]# fsck -f /dev/vg0/lvol0
fsck 1.35 (28-Feb-2004)
e2fsck 1.35 (28-Feb-2004)
Pass 1: Checking inodes, blocks, and sizes
Pass 2: Checking directory structure
Pass 3: Checking directory connectivity
Pass 4: Checking reference counts
Pass 5: Checking group summary information
/dev/vg0/lvol0: 11/124488 files (9.1% non-contiguous), 22138/512000 blocks
[root@linux1 ~]# resize2fs /dev/vg0/lvol0 100M
resize2fs 1.35 (28-Feb-2004)
Resizing the filesystem on /dev/vg0/lvol0 to 102400 (1k) blocks.
The filesystem on /dev/vg0/lvol0 is now 102400 blocks long.

[root@linux1 ~]#
```

2. Réduire la taille du LV et remonter le FS.

```
[root@linux1 ~]# lvreduce -L -400M /dev/vg0/lvol0
  WARNING: Reducing active logical volume to 100.00 MB
  THIS MAY DESTROY YOUR DATA (filesystem etc.)
Do you really want to reduce lvol0? [y/n]: y
  Reducing logical volume lvol0 to 100.00 MB
  Logical volume lvol0 successfully resized
[root@linux1 ~]# mount -t ext3 /dev/vg0/lvol0 /mnt/disk
[root@linux1 ~]# df -h /mnt/disk
Filesystem            Size  Used Avail Use% Mounted on
/dev/mapper/vg0-lvol0
                       97M  5.6M   88M   6% /mnt/disk
[root@linux1 ~]#
```

3. Réduire le VG (on retire des PV).

```
[root@linux1 ~]# vgreduce vg0 /dev/hda8
  Removed "/dev/hda8" from volume group "vg0"
[root@linux1 ~]# vgreduce vg0 /dev/hda7
  Removed "/dev/hda7" from volume group "vg0"
[root@linux1 ~]# vgs vg0
  VG   #PV #LV #SN Attr   VSize   VFree
  vg0    1   1   0 wz--n- 492,00M 392,00M
[root@linux1 ~]#
```

4. Créer un LV qui s'étend sur plusieurs PV, essayer de retirer un des PV.

a) On réétend le VG pour qu'il utilise 3 PV.

```
[root@linux1 ~]# vgextend vg0 /dev/hda7
  Volume group "vg0" successfully extended
[root@linux1 ~]# vgextend vg0 /dev/hda8
  Volume group "vg0" successfully extended
[root@linux1 ~]# pvs
```

```
  PV         VG   Fmt  Attr PSize   PFree
  /dev/hda6  vg0  lvm2 a-   492,00M 392,00M
  /dev/hda7  vg0  lvm2 a-   492,00M 492,00M
  /dev/hda8  vg0  lvm2 a-   492,00M 492,00M
```

b) On crée un LV qui s'étend sur 2 PV.

```
[root@linux1 ~]# lvcreate -n grosfs -L 600M vg0
  Logical volume "grosfs" created
[root@linux1 ~]# lvdisplay -m /dev/vg0/grosfs
  --- Logical volume ---
  LV Name                /dev/vg0/grosfs
  VG Name                vg0
  LV UUID                p3WpbT-igq0-3aNU-EHC2-pryD-1D1J-MKcg5a
  LV Write Access        read/write
  LV Status              available
  # open                 0
  LV Size                600,00 MB
  Current LE             150
  Segments               2
  Allocation             inherit
  Read ahead sectors     0
  Block device           253:1

  --- Segments ---
  Logical extent 0 to 122:
    Type                linear
    Physical volume     /dev/hda7
    Physical extents    0 to 122

  Logical extent 123 to 149:
    Type                linear
    Physical volume     /dev/hda8
    Physical extents    0 to 26

[root@linux1 ~]#
```

c) On essaye de retirer le PV hda7.

```
[root@linux1 ~]# vgreduce vg0 /dev/hda7
  Physical volume "/dev/hda7" still in use
```

d) On déplace les LE du PV hda7 vers hda8. D'abord on fait de la place sur hda8.

```
[root@linux1 ~]# pvs
  PV         VG   Fmt  Attr PSize   PFree
  /dev/hda6  vg0  lvm2 a-   492,00M 392,00M
  /dev/hda7  vg0  lvm2 a-   492,00M      0
  /dev/hda8  vg0  lvm2 a-   492,00M 384,00M
[root@linux1 ~]# pvmove -n grosfs /dev/hda8 /dev/hda6
  vg0-pvmove0: device not found
  /dev/hda8: Moved: 100,0%
[root@linux1 ~]# pvs
  PV         VG   Fmt  Attr PSize   PFree
  /dev/hda6  vg0  lvm2 a-   492,00M 284,00M
  /dev/hda7  vg0  lvm2 a-   492,00M      0
  /dev/hda8  vg0  lvm2 a-   492,00M 492,00M
[root@linux1 ~]# pvmove /dev/hda7 /dev/hda8
```

```
  vg0-pvmove0: device not found
  /dev/hda7: Moved: 26,8%
  /dev/hda7: Moved: 49,6%
  /dev/hda7: Moved: 74,0%
  /dev/hda7: Moved: 96,7%
  /dev/hda7: Moved: 100,0%
[root@linux1 ~]# pvs
  PV         VG   Fmt  Attr PSize   PFree
  /dev/hda6  vg0  lvm2 a-   492,00M 284,00M
  /dev/hda7  vg0  lvm2 a-   492,00M 492,00M
  /dev/hda8  vg0  lvm2 a-   492,00M      0
[root@linux1 ~]#
```

e) On peut enfin retirer le PV.

```
[root@linux1 ~]# vgreduce vg0 /dev/hda7
  Removed "/dev/hda7" from volume group "vg0"
[root@linux1 ~]# pvs
  PV         VG   Fmt  Attr PSize   PFree
  /dev/hda6  vg0  lvm2 a-   492,00M 284,00M
  /dev/hda7       lvm2 --   494,16M 494,16M
  /dev/hda8  vg0  lvm2 a-   492,00M      0
[root@linux1 ~]#
```

f) On supprime le LV.

```
[root@linux1 ~]# lvremove /dev/vg0/grosfs
Do you really want to remove active logical volume "grosfs"? [y/n]: y
  Logical volume "grosfs" successfully removed
[root@linux1 ~]#
```

Tâche 6 : Utilisation de LV comme swap

1. Créer un LV et le formater en espace de swap.

```
[root@linux1 ~]# lvcreate -L 100M -n swap1 vg0
  Logical volume "swap1" created
[root@linux1 ~]# mkswap /dev/vg0/swap1
Setting up swapspace version 1, size = 104853 kB
[root@linux1 ~]#
```

2. Activer la nouvelle zone de swap.

```
[root@linux1 ~]# swapon /dev/vg0/swap1
[root@linux1 ~]#
```

Remarque : si l'on veut que la nouvelle zone de swap soit activée automatiquement lors du démarrage, il faut ajouter la ligne suivante au fichier /etc/fstab :

```
/dev/vg0/swap1     swap   swap  defaults   0  0
```

3. Lister les espaces de swap et leur utilisation.

```
[root@linux1 ~]# swapon -s
Filename                                Type            Size    Used    Priority
/dev/hda3                               partition       522104  0       -1
/dev/mapper/vg0-swap1                   partition       102392  0       -2
[root@linux1 ~]#
```

4. Diminuer l'espace de swap.

```
[root@linux1 ~]# swapoff -v /dev/vg0/swap1
swapoff on /dev/vg0/swap1
```

```
[root@linux1 ~]# lvreduce -L -32M /dev/vg0/swap1
  WARNING: Reducing active logical volume to 68.00 MB
  THIS MAY DESTROY YOUR DATA (filesystem etc.)
Do you really want to reduce swap1? [y/n]: y
  Reducing logical volume swap1 to 68.00 MB
  Logical volume swap1 successfully resized
[root@linux1 ~]# mkswap /dev/vg0/swap1
Setting up swapspace version 1, size = 71299 kB
[root@linux1 ~]# swapon /dev/vg0/swap1
[root@linux1 ~]# swapon -s
Filename                                Type            Size    Used    Priority
/dev/hda3                               partition       522104  0       -1
/dev/mapper/vg0-swap1                   partition       69624   0       -3
[root@linux1 ~]#
```

5. Supprimer l'espace de swap.

Remarque : il faut modifier /etc/fstab si l'espace de swap y est inscrit.

```
[root@linux1 ~]# swapoff -v /dev/vg0/swap1
swapoff on /dev/vg0/swap1
[root@linux1 ~]# lvremove /dev/vg0/swap1
Do you really want to remove active logical volume "swap1"? [y/n]: y
  Logical volume "swap1" successfully removed
[root@linux1 ~]#
```

Tâche 7 : Utiliser un LV dédié à une application

1. Sauvegarder des fichiers dans un LV.

```
[root@linux1 ~]# lvcreate -L100M vg0
  Logical volume "lvol1" created
[root@linux1 ~]# tar cvf /dev/vg0/lvol1 /etc
...
/etc/java/jpackage-release
/etc/minicom.users
/etc/inittab
/etc/ld.so.cache
/etc/pear.conf
[root@linux1 ~]#
```

2. Vérifier la présence d'un des fichiers dans l'archive.

```
[root@linux1 ~]# tar tvf /dev/vg0/lvol1 etc/passwd
-rw-r--r-- root/root      1589 2006-11-11 17:56:41 etc/passwd
[root@linux1 ~]#
```

Remarque : Evidemment, si l'on utilise un LV par l'intermédiaire d'une application. Le contenu du LV est détruit. Cette utilisation est incompatible avec une utilisation du LV comme FS ou comme SWAP.

Tâche 8 : Exporter/Importer un VG

1. Détruire tout (pour recommencer sur de bonnes bases).

```
[root@linux1 ~]# umount /mnt/disk
[root@linux1 ~]# lvremove -f vg0
  Logical volume "lvol0" successfully removed
  Logical volume "lvol1" successfully removed
```

```
[root@linux1 ~]# vgremove vg0
  Volume group "vg0" successfully removed
[root@linux1 ~]#
```

2. Créer un VG, un LV et un FS, écrire des données dedans.

```
[root@linux1 ~]# vgcreate vg0 /dev/hda6
  Volume group "vg0" successfully created
[root@linux1 ~]# lvcreate -L 100M vg0
  Logical volume "lvol0" created
[root@linux1 ~]# mkfs -t ext3 -q /dev/vg0/lvol0
[root@linux1 ~]# mount -t ext3 /dev/vg0/lvol0 /mnt/disk
[root@linux1 ~]# cal > /mnt/disk/f1
[root@linux1 ~]# date > /mnt/disk/f2
[root@linux1 ~]# mkdir /mnt/disk/rep
[root@linux1 ~]# uptime > /mnt/disk/rep/fic
[root@linux1 ~]#
```

3. Désactiver le VG.

a) On démonte les FS.

```
[root@linux1 ~]# lvs
  LV    VG   Attr   LSize   Origin Snap%  Move Copy%
  lvol0 vg0  -wi-ao 100.00M
[root@linux1 ~]# umount /mnt/disk
[root@linux1 ~]# lvs
  LV    VG   Attr   LSize   Origin Snap%  Move Copy%
  lvol0 vg0  -wi-a- 100.00M
[root@linux1 ~]#
```

b) On désactive le VG.

```
[root@linux1 ~]# vgchange -a n vg0
  0 logical volume(s) in volume group "vg0" now active
[root@linux1 ~]# lvs
  LV    VG   Attr   LSize   Origin Snap%  Move Copy%
  lvol0 vg0  -wi--- 100.00M
[root@linux1 ~]#
```

c) Quand le VG est désactivé, on ne peut plus accéder aux LV du VG.

```
[root@linux1 ~]# mount -t ext3 /dev/vg0/lvol0 /mnt/disk
mount: special device /dev/vg0/lvol0 does not exist
[root@linux1 ~]#
```

4. Exporter le VG.

```
[root@linux1 ~]# vgexport vg0
  Volume group "vg0" successfully exported
[root@linux1 ~]# vgs
  VG   #PV #LV #SN Attr   VSize   VFree
  vg0    1   1   0 wzx-n 488.00M 388.00M
[root@linux1 ~]#
```

5. Importer le VG, ensuite l'activer et remonte les FS.

a) On importe le VG.

```
[root@linux1 ~]# vgimport vg0
  Volume group "vg0" successfully imported
[root@linux1 ~]# vgs
  VG   #PV #LV #SN Attr   VSize   VFree
  vg0    1   1   0 wz--n 488.00M 388.00M
```

```
[root@linux1 ~]#
```

b) On active le VG.

```
[root@linux1 ~]# vgchange -a y vg0
  1 logical volume(s) in volume group "vg0" now active
[root@linux1 ~]# lvs
  LV    VG   Attr   LSize   Origin Snap%  Move Copy%
  lvol0 vg0  -wi-a- 100.00M
```

c) On monte les FS abrités par les LV du VG.

```
[root@linux1 ~]# mount -t ext3 /dev/vg0/lvol0 /mnt/disk
[root@linux1 ~]# df -h /mnt/disk
Filesystem            Size  Used Avail Use% Mounted on
/dev/mapper/vg0-lvol0
                       97M  5.6M   87M   7% /mnt/disk
[root@linux1 ~]# umount /mnt/disk
```

Tâche 9 : Accéder à des FS présents dans des LV en mode maintenance

1. Rédémarrer à partir du cdrom d'installation.

Au démarrage on précise que l'on utilise le mode maintenance (rescue). On choisit la langue anglaise, le clavier français, on n'active pas l'interface réseau et on ne monte pas automatiquement les FS (skip).

```
boot:   linux rescue
```

2. Explorer les disques et les partitions pour trouver les PV et les VG.

```
-/bin/sh-2.05 # lvm  vgscan
  Reading all physical volumes. This any take a while...
  Found volume group "vg0" using metadata type lmv2.
```

3. Activer le VG vg0.

```
-/bin/sh-2.05 # lvm  vgchange -a y vg0
 1 logical volume(s) in volume group "vg0" now active.
```

4. Monter un FS présent sur un LV, lister son contenu.

```
-/bin/sh-2.05 #  mkdir /mnt/disk
-/bin/sh-2.05 #  lvm  lvs  vg0
  LV    VG   Attr   LSize   Origin Snap%  Move Copy%
  lvol0 vg0  -wi-a- 100.00M
-/bin/sh-2.05 # mount -t ext3 /dev/vg0/lvol0  /mnt/disk
-/bin/sh-2.05 # ls  /mnt/disk
f1  f2  lost+found  rep
```

5. Démonter le FS et redémarrer le système.

```
-/bin/sh-2.05 #  umount  /mnt/disk
CTRL-ALT-DEL
```

Tâche 10 : Sauvegarder un VG

1. Sauvegarder les FS.

```
[root@linux1 ~]# mount -t ext3 /dev/vg0/lvol0 /mnt/disk
[root@linux1 ~]# tar cvf /tmp/disk.tar /mnt/disk
tar: Removing leading `/' from member names
/mnt/disk/
/mnt/disk/lost+found/
```

```
/mnt/disk/f1
/mnt/disk/f2
/mnt/disk/rep/
/mnt/disk/rep/fic
[root@linux1 ~]#
```

2. Sauvegarder la structure du VG.

```
[root@linux1 ~]# vgcfgbackup -f /tmp/vg0.bck vg0
  Volume group "vg0" successfully backed up.
[root@linux1 ~]# more /tmp/vg0.bck
# Generated by LVM2: Sun Nov 12 18:02:21 2006

contents = "Text Format Volume Group"
version = 1

description = "vgcfgbackup -f /tmp/vg0.bck vg0"

creation_host = "linux1.pinguins"       # Linux linux1.pinguins 2.6.9-11.ELsmp #
1 SMP Fri May 20 18:26:27 EDT 2005 i686
creation_time = 1163350941      # Sun Nov 12 18:02:21 2006

vg0 {
        id = "S9L8pS-HqIz-whQD-fRrD-Ah9d-VxZL-SnDKwN"
        seqno = 8
        status = ["RESIZEABLE", "READ", "WRITE"]
        extent_size = 8192              # 4 Megabytes
        max_lv = 0
        max_pv = 0

        physical_volumes {

                pv0 {
                        id = "53GEPD-QF35-fE1w-meNW-D0AG-a4Hj-CXI1O4"
                        device = "/dev/hda6"    # Hint only

                        status = ["ALLOCATABLE"]
                        pe_start = 384
                        pe_count = 122  # 488 Megabytes
                }
        }

        logical_volumes {

                lvol0 {
                        id = "c2FQjr-gpiz-SeK3-waEE-mgyE-tw3J-24XF6r"
                        status = ["READ", "WRITE", "VISIBLE"]
                        segment_count = 1

                        segment1 {
                                start_extent = 0
                                extent_count = 25       # 100 Megabytes

                                type = "striped"
                                stripe_count = 1        # linear

                                stripes = [
```

```
                                        "pv0", 0
                                    ]
                                }
                            }
                        }
}
[root@linux1 ~]#
```

3. Détruire le VG (FS, LV, PV, VG).

```
[root@linux1 ~]# umount /mnt/disk
[root@linux1 ~]# lvremove -f vg0
  Logical volume "lvol0" successfully removed
[root@linux1 ~]# vgremove vg0
  Volume group "vg0" successfully removed
[root@linux1 ~]# pvremove /dev/hda6
  Labels on physical volume "/dev/hda6" successfully wiped
[root@linux1 ~]#
```

4. Restaurer le VG.

a) On restaure les PV. Le problème est qu'ils doivent avoir les mêmes UUID.

```
[root@linux1 ~]# awk '/pv0/ { print; getline; print}' /tmp/vg0.bck
                pv0 {
                        id = "53GEPD-QF35-fE1w-meNW-D0AG-a4Hj-CXI1O4"
                                        "pv0", 0
                                    ]
[root@linux1 ~]# pvcreate --restorefile /tmp/vg0.bck   \
> --uuid 53GEPD-QF35-fE1w-meNW-D0AG-a4Hj-CXI1O4  /dev/hda6
  Couldn't find device with uuid '53GEPD-QF35-fE1w-meNW-D0AG-a4Hj-CXI1O4'.
  Physical volume "/dev/hda6" successfully created
[root@linux1 ~]# pvdisplay /dev/hda6
  --- NEW Physical volume ---
  PV Name               /dev/hda6
  VG Name
  PV Size               493.97 MB
  Allocatable           NO
  PE Size (KByte)       0
  Total PE              0
  Free PE               0
  Allocated PE          0
  PV UUID               53GEPD-QF35-fE1w-meNW-D0AG-a4Hj-CXI1O4

[root@linux1 ~]#
```

b) On restaure la structure du VG.

```
[root@linux1 ~]# vgcfgrestore -f /tmp/vg0.bck vg0
  Restored volume group vg0
[root@linux1 ~]# vgdisplay -v
    Finding all volume groups
    Finding volume group "vg0"
  --- Volume group ---
  VG Name               vg0
  System ID
  Format                lvm2
  Metadata Areas        1
  Metadata Sequence No  9
```

```
  VG Access               read/write
  VG Status               resizable
  MAX LV                  0
  Cur LV                  1
  Open LV                 0
  Max PV                  0
  Cur PV                  1
  Act PV                  1
  VG Size                 488.00 MB
  PE Size                 4.00 MB
  Total PE                122
  Alloc PE / Size         25 / 100.00 MB
  Free  PE / Size         97 / 388.00 MB
  VG UUID                 S9L8pS-HqIz-whQD-fRrD-Ah9d-VxZL-SnDKwN

  --- Logical volume ---
  LV Name                 /dev/vg0/lvol0
  VG Name                 vg0
  LV UUID                 c2FQjr-gpiz-SeK3-waEE-mgyE-tw3J-24XF6r
  LV Write Access         read/write
  LV Status               NOT available
  LV Size                 100.00 MB
  Current LE              25
  Segments                1
  Allocation              inherit
  Read ahead sectors      0

  --- Physical volumes ---
  PV Name                 /dev/hda6
  PV UUID                 53GEPD-QF35-fE1w-meNW-D0AG-a4Hj-CXI1O4
  PV Status               allocatable
  Total PE / Free PE      122 / 97

[root@linux1 ~]#
```

c) On active le VG.

```
[root@linux1 ~]# vgchange -a y vg0
  1 logical volume(s) in volume group "vg0" now active
[root@linux1 ~]#
```

d) On formate les LV.

```
[root@linux1 /]# mkfs -t ext3 /dev/vg0/lvol0
mke2fs 1.35 (28-Feb-2004)
...
```

d) On remonte les FS, et on les restaure.

```
[root@linux1 ~]# mount -t ext3 /dev/vg0/lvol0 /mnt/disk
[root@linux1 ~]# cd /
[root@linux1 /]# tar xvf /tmp/disk.tar
mnt/disk/
mnt/disk/lost+found/
mnt/disk/f1
mnt/disk/f2
mnt/disk/rep/
mnt/disk/rep/fic
[root@linux1 /]# ls /mnt/disk
```

```
f1  f2  lost+found  rep
[root@linux1 /]#
```

Tâche 11 : Snapshot

1. Vérifier que /mnt/disk est monté et contient des fichiers.

```
[root@linux1 /]# df -h /mnt/disk
Filesystem            Size  Used Avail Use% Mounted on
/dev/mapper/vg0-lvol0
                       97M  5.6M   87M   7% /mnt/disk
[root@linux1 ~]# ls /mnt/disk
f1  f2  lost+found  rep
[root@linux1 ~]#
```

2. Créer un snapshot.

```
[root@linux1 ~]# modprobe dm_snapshot
[root@linux1 ~]# lvcreate -L 100M -s -n back /dev/vg0/lvol0
  Logical volume "back" created
[root@linux1 ~]# lvs
  LV    VG   Attr   LSize   Origin Snap%  Move Copy%
  back  vg0  swi-a- 100.00M lvol0    0.02
  lvol0 vg0  owi-ao 100.00M
[root@linux1 ~]#
```

3. Monter le snapshot et le sauvegarder (vision figée du FS).

Pendant ce temps, on crée un fichier dans le FS (il ne sera pas sauvegardé).

```
[root@linux1 ~]# mkdir /mnt/back
[root@linux1 ~]# mount /dev/vg0/back /mnt/back
[root@linux1 ~]# ls -l > /mnt/disk/f3
[root@linux1 ~]# cd /mnt/back
[root@linux1 back]# tar cvf /tmp/sauve.tar .
./
./lost+found/
./f1
./f2
./rep/
./rep/fic
[root@linux1 back]# lvs
  LV    VG   Attr   LSize   Origin Snap%  Move Copy%
  back  vg0  swi-ao 100.00M lvol0    0.07
  lvol0 vg0  owi-ao 100.00M
[root@linux1 back]#
```

4. Supprimer le snapshot.

```
[root@linux1 back]# cd
[root@linux1 ~]# umount /mnt/back
[root@linux1 ~]# lvs
  LV    VG   Attr   LSize   Origin Snap%  Move Copy%
  back  vg0  swi-a- 100.00M lvol0    0.08
  lvol0 vg0  owi-ao 100.00M
[root@linux1 ~]# lvremove /dev/vg0/back
Do you really want to remove active logical volume "back"? [y/n]: y
  Logical volume "back" successfully removed
[root@linux1 ~]# lvs
```

```
  LV    VG   Attr   LSize   Origin Snap%  Move Copy%
  lvol0 vg0  -wi-ao 100.00M
[root@linux1 ~]#
```

Tâche 12 : Les LV « stripped »

1. Étendre le VG sur trois PV et détruire tous les LV.

```
[root@linux1 ~]# vgextend vg0 /dev/hda7 /dev/hda8
  Volume group "vg0" successfully extended
[root@linux1 ~]# pvs
  PV         VG   Fmt  Attr PSize   PFree
  /dev/hda6  vg0  lvm2 a-   488.00M 388.00M
  /dev/hda7  vg0  lvm2 a-   484.00M 484.00M
  /dev/hda8  vg0  lvm2 a-   484.00M 484.00M
[root@linux1 ~]# umount /mnt/disk
[root@linux1 ~]# lvremove -f vg0
  Logical volume "lvol0" successfully removed
[root@linux1 ~]# lvs
[root@linux1 ~]#
```

2. Créer un LV de 600 mega-octets, « stripped » sur les trois PV avec des chunk de 4k.

```
[root@linux1 ~]# lvcreate -i3 -I4 -L600M -n reparti vg0
  Logical volume "reparti" created
[root@linux1 ~]# lvdisplay -m vg0/reparti
  --- Logical volume ---
  LV Name                /dev/vg0/reparti
  VG Name                vg0
  LV UUID                vDErgu-9Hr1-eqcB-TwPo-Xpvb-4Tiy-qH0myp
  LV Write Access        read/write
  LV Status              available
  # open                 0
  LV Size                600.00 MB
  Current LE             150
  Segments               1
  Allocation             inherit
  Read ahead sectors     0
  Block device           253:0

  --- Segments ---
  Logical extent 0 to 149:
    Type                striped
    Stripes             3
    Stripe size         4 KB
    Stripe 0:
      Physical volume   /dev/hda6
      Physical extents  0 to 49
    Stripe 1:
      Physical volume   /dev/hda7
      Physical extents  0 to 49
    Stripe 2:
      Physical volume   /dev/hda8
      Physical extents  0 to 49

[root@linux1 ~]#
```

Tâche 13 :
Détruire tout (y compris les partitions)

```
[root@linux1 ~]# lvremove -f vg0
  Logical volume "reparti" successfully removed
[root@linux1 ~]# vgremove vg0
  Volume group "vg0" successfully removed
[root@linux1 ~]# pvremove /dev/hda8
  Labels on physical volume "/dev/hda8" successfully wiped
[root@linux1 ~]# pvremove /dev/hda7
  Labels on physical volume "/dev/hda7" successfully wiped
[root@linux1 ~]# pvremove /dev/hda6
  Labels on physical volume "/dev/hda6" successfully wiped
[root@linux1 ~]# vgs
  No volume groups found
[root@linux1 ~]# pvs
[root@linux1 ~]# lvs
  No volume groups found
[root@linux1 ~]# fdisk /dev/hda

The number of cylinders for this disk is set to 4111.
There is nothing wrong with that, but this is larger than 1024,
and could in certain setups cause problems with:
1) software that runs at boot time (e.g., old versions of LILO)
2) booting and partitioning software from other OSs
   (e.g., DOS FDISK, OS/2 FDISK)

Command (m for help): d
Partition number (1-9): 9
```

Idem pour les partitions 8, 7 et 6.

```
Command (m for help): w
The partition table has been altered!

Calling ioctl() to re-read partition table.

WARNING: Re-reading the partition table failed with error 16: Device or resource
busy.
The kernel still uses the old table.
The new table will be used at the next reboot.
Syncing disks.
[root@linux1 ~]# partprobe
[root@linux1 ~]# fdisk -l

Disk /dev/hda: 33.8 GB, 33820286976 bytes
255 heads, 63 sectors/track, 4111 cylinders
Units = cylinders of 16065 * 512 = 8225280 bytes

   Device Boot      Start         End      Blocks   Id  System
/dev/hda1   *           1          13      104391   83  Linux
/dev/hda2              14        1288    10241437+  83  Linux
/dev/hda3            1289        1353      522112+  82  Linux swap
/dev/hda4            1354        4111    22153635    5  Extended
/dev/hda5            1354        1418      522081   83  Linux
[root@linux1 ~]#
```

- *Incrémentale, différentielle, complète*
- *dump 0uf /dev/tape /*
- *dd if=/dev/hda | rsh ...*
- *mt –f /dev/nst0 bsf 2*
- *Bare-Metal*

9

La sauvegarde

Objectifs

Ce chapitre traite de la sauvegarde. On désire montrer l'éventail des possibilités de sauvegarde. Mais on veut essentiellement apprendre au lecteur à réaliser des sauvegardes incrémentales. On apprend également à réaliser des sauvegardes complètes du système.

Contenu

Vision générale de la sauvegarde

La sauvegarde incrémentale

Les commandes dump et restore

La commande dd

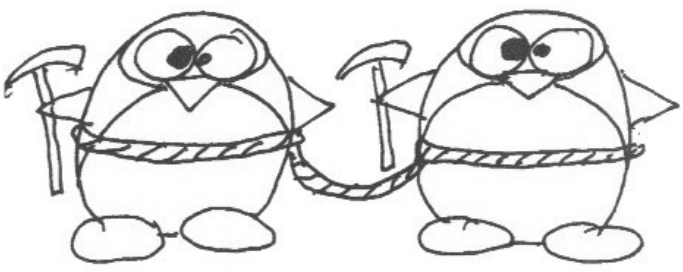
La sauvegarde réseau

La sauvegarde Bare-Metal

Le logiciel Mondo

Les cartouches

Ateliers



Vision générale de la sauvegarde

La théorie

Plan de sauvegarde

IL FAUT DEVELOPPER UN PLAN DE REPRISE D'ACTIVITE. Ce plan sera activé en cas de sinistre. La clé de voûte de ce plan est le plan de sauvegarde et de restauration des données informatiques.

Voici quelques questions qui peuvent vous aider à établir ce plan de sauvegarde.

- Que faut-il sauvegarder, avec quelle fréquence ? (Ou inversement qu'accepter vous de perdre en cas de sinistre) ?
- Corollaire : A combien estimez-vous les conséquences financières d'un sinistre ?
- Corollaire du corollaire : quel sera votre budget de sauvegarde ?
- Combien de temps conservera-t-on les sauvegardes, à quels endroits, en combien d'exemplaires ?
- Quel est le support de sauvegarde approprié ? Quels sont les besoins, en capacité des supports de sauvegarde ? Combien de temps durera la sauvegarde ?

Conseils

- Centraliser vos sauvegardes : utiliser les commandes de sauvegardes réseau ou les logiciels client/serveur.
- Étiqueter vos sauvegardes. On doit indiquer l'hôte, le FS, si c'est une sauvegarde complète ou non, la date, le format de la sauvegarde et même la commande complète qui a permis de sauvegarder. On peut remarquer que les logiciels clients/serveurs imposent un étiquetage avec code à barres.
- Réaliser au minimum une sauvegarde incrémentale chaque jour et une sauvegarde complète le week-end. Ne pas oublier de sauvegarder les fichiers de configuration à chaque modification.
- Structurer vos données notamment en termes de sauvegarde. Créer des FS destinés à être sauvegardés de manière incrémentale. Typiquement, on sépare le système des applications, les données des programmes.
- Limiter la taille de vos FS à la taille des supports de sauvegarde.
- Vous devez avoir une copie de vos sauvegardes sensibles. Cette copie doit être dans un lieu différent de votre site d'exploitation.
- Faites en sorte d'arrêter (ou au moins limiter) l'exploitation durant la sauvegarde. C'est pourquoi on fait les sauvegardes la nuit.
- Protéger vos sauvegardes : il faut les enfermer à clé, elles contiennent des informations sensibles.
- Ne pas oublier que les cartouches de sauvegarde ont un cycle de vie limité. Mémoriser leur utilisation. Les logiciels de sauvegarde client/serveur le font automatiquement.
- Enfin, un conseil majeur : Tester votre plan de reprise d'activité. Ainsi vous devez réaliser de véritables restaurations.

Le savoir concret

Les outils de sauvegarde

Sauvegarde de fichiers : `tar`, `star`, `cpio`, `pax`

Sauvegarde physique : `dd`

Sauvegarde d'images : `partimage`, clonezilla, ghost (produit commerciale)

Sauvegarde système incrémentale de FS :
`dump`/`restore` (Ext2/Ext3), `xfsdump`/`xfsrestore` (xfs)

Sauvegarde complète (Bare Metal) : Mondo

Sauvegarde client/serveur :
Bacula (libre), Amanda (libre), Arkeia, Veritas NetBackup, Networker, Tina, TSM, …

Synchronisation de répertoires : `rsync`

Pour en savoir plus

Les pages de manuel

tar(1), star(1), cpio(1L), pax(1), dd(1), partimage(1), dump(8), restore(8), xfsdump(8), rsync(1), mt(8), rmt(8), st(4), mondoarchive(8), mondorestore(8)

Internet

Backup
http://en.wikipedia.org/wiki/Backup

Préparation à un sinistre (sauvegarde, …)
http://www.redhat.com/docs/manuals/enterprise/RHEL-4-Manual/fr/admin-guide/ch-disaster.html

Le logiciel Arkeia
http://www.arkeia.fr/

Le logiciel Bacula
http://www.bacula.org/

Le logiciel Amanda
http://www.amanda.org

Le logiciel Mondo
http://www.mondorescue.org

Le logiciel Clonezilla
http://clonezilla.sourceforge.net/

Backup Central (site dédié à la sauvegarde)
http://www.backupcentral.com/

Site dédié à la sauvegarde sous Linux
http://www.linux-backup.net/

Panorama des commandes et des logiciels de sauvegarde pour Linux
http://www.linux.org/apps/all/Administration/Backup.html

Livre

Backup & Recovery, par W. Curtis Preston chez O'Reilly.

La sauvegarde incrémentale

La théorie

Sauvegarde complète, incrémentale ou différentielle

Une sauvegarde incrémentale consiste à ne sauvegarder que les modifications effectuées depuis une sauvegarde de référence.

La sauvegarde de référence est une sauvegarde complète des données à sauvegarder (full backup).

On peut sauvegarder de manière incrémentale des arborescences quelconques, mais il est préférable de l'associer à la notion de FS. Dans ce cas, des commandes adaptées existent.

La sauvegarde incrémentale peut être réalisée selon trois stratégies :

Sauvegarde complètement incrémentale

Dans une sauvegarde complètement incrémentale, chaque sauvegarde incrémentale devient la référence de la sauvegarde incrémentale suivante. Dans ce cas, les sauvegardes sont petites et courtes, mais une restauration complète nécessite de restaurer toutes les archives.

Sauvegarde différentielle

Dans la sauvegarde différentielle, on ne sauvegarde que les modifications en se basant systématiquement sur la sauvegarde complète. Dans ce cas, les sauvegardes sont de plus en plus grosses et lentes, mais une restauration complète ne nécessite que la sauvegarde complète et la dernière différentielle.

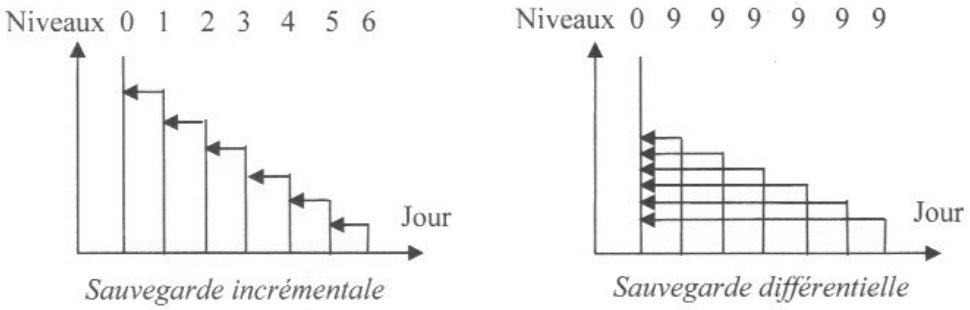


Fig. Sauvegarde incrémentale et différentielle.

Sauvegarde Tour de Hanoi

L'ordonnancement de la Tour de Hanoi est plus complexe. Il est basé sur l'algorithme « Tour de Hanoi ». Ses objectifs sont de minimiser les temps de sauvegarde et de restauration et surtout de faire en sorte que chaque fichier soit sauvegardé au moins deux fois.

Remarque : Le niveau de sauvegarde indique clairement la stratégie : dans une sauvegarde à un niveau donné on ne sauvegarde que les modifications effectuées depuis la sauvegarde de niveau inférieur. Le niveau 0 est réservé aux sauvegardes complètes.

Les cycles de sauvegarde et rétention

A la fin d'un cycle de sauvegarde, on démarre sur une nouvelle sauvegarde complète.

Le cycle de rétention est le nombre de cycles de sauvegarde que l'on conserve.

Le savoir concret

Les commandes

`dump/restore`	Sauvegarde incrémentale des FS Ext2/Ext3.
`xfsdump/xfsrestore`	Sauvegarde incrémentale des FS XFS.

Remarques :

1) ReiserFs ne dispose pas de commande de sauvegarde incrémentale.

2) La commande `tar` permet de réaliser des sauvegardes incrémentales mais elle n'a pas été conçue pour cela. Par exemple, elle ne supprime pas les fichiers détruits.

3) Les logiciels client/serveur réalisent essentiellement des sauvegardes incrémentales.

Focus : Sauvegarde complète, incrémentale ou différentielle avec dump

La sauvegarde complète du FS /home (références des autres sauvegardes) :

```
Dimanche # dump 0uf /dev/st0  /home
```

La sauvegarde incrémentale :

```
Lundi    # dump 1uf /dev/st0  /home
Mardi    # dump 2uf /dev/st0  /home
Mercredi # dump 3uf /dev/st0  /home
...
Samedi   # dump 6uf /dev/st0  /home
```

La sauvegarde différentielle :

```
Lundi    # dump 1uf /dev/st0  /home
Mardi    # dump 9uf /dev/st0  /home
Mercredi # dump 9uf /dev/st0  /home
...
Samedi   # dump 9uf /dev/st0  /home
```

La sauvegarde de la Tour de Hanoi (cycle d'une semaine) :

```
Lundi    # dump 3uf /dev/st0  /home
Mardi    # dump 2uf /dev/st0  /home
Mercredi # dump 5uf /dev/st0  /home
Jeudi    # dump 4uf /dev/st0  /home
Vendredi # dump 7uf /dev/st0  /home
Samedi   # dump 6uf /dev/st0  /home
```

Remarque : Les sauvegardes incrémentales doivent être automatiques. Les commandes présentées sont donc typiquement présentes au sein d'un crontab.

Pour en savoir plus

Les pages de manuel

dump(8), restore(8), xfsdump(8), xfsrestore(8)

Internet

La sauvegarde incrémentale
http://en.wikipedia.org/wiki/Incremental_backup

Les commandes dump et restore

Le savoir concret

Les commandes

`dump`	Sauvegarde incrémentale d'un FS Ext2/Ext3.
`restore`	Restauration d'un FS Ext2/Ext3 à partir d'archives dump.

Les fichiers

/etc/dumpdates Ce fichier contient l'historique des sauvegardes incrémentales.

Focus : Les principales options de la commande dump

-niveau	Valeur numérique indiquant le niveau de la sauvegarde incrémentale. La valeur 0 est réservée à une sauvegarde complète. La valeur par défaut est 9. Les valeurs au-delà de 9 sont autorisées.
-a	Ecrit jusqu'à la fin du média, c'est la valeur par défaut.
-A fichier	Dump stocke la VTOC (Volume Table Of Contents) dans un fichier.
-b bs	Spécifie la taille des blocs, par défaut 10ko.
-F script	Spécifie le script à exécuter à chaque changement de volume.
-f archive	Spécifie le support d'archive, par défaut /dev/st0. Le support peut être distant : hôte:archive.
-u	Mise à jour du fichier /etc/dumpdates. Les sauvegardes incrémentales se basent sur ce fichier pour savoir ce qu'elles doivent sauvegarder.

Focus : Les principales options de la commande restore

-t	On liste la VTOC d'une sauvegarde.
-r	On restaure l'intégralité d'une sauvegarde.
-x	On restaure un ou plusieurs fichiers.
-i	Restauration interactive.
-a	La commande ne demande pas le volume suivant. Elle lit en séquence les différents volumes.

Remarque : cf. la commande `dump` pour les options –f, -b et –F.

Exemples

a) La sauvegarde complète du FS /home (référence des autres sauvegardes).

```
Dimanche # dump 0uf /dev/st0  /home
```

b) La sauvegarde incrémentale effectuée chaque jour. On ne sauvegarde que les modifications effectuées depuis la sauvegarde réalisée au niveau inférieur.

```
Lundi    # dump 1uf /dev/st0  /home
Mardi    # dump 2uf /dev/st0  /home
```

c) Lister le contenu d'une sauvegarde.

```
# restore  -tf  /dev/st0
```

d) Restaurer un fichier.

```
# cd  /home
```

```
# restore -xf  /dev/st0  ./pierre/.bash_profile
```

e) Restaurer tout un FS dans le cas d'une sauvegarde différentielle.

```
# mkfs -t  ext3  /dev/hda6
# mount  /home
# cd /home
# restore -rf  /dev/st0  # on met l'archive de niveau 0
# restore -rf /dev/st0   # on met la dernière archive de niveau 9.
```

Les variables d'environnement

TAPE	Le support d'archive.
RMT	Spécifie la commande `rmt` distante.
RSH	Spécifie la commande à utiliser pour effectuer une sauvegarde réseau.

Les particularités des différentes distributions

Debian

Le fichier dumpdates est dans /var/lib (/var/lib/dumpdates).

Pour en savoir plus

Les pages de manuel

dump(8), restore(8)

La commande dd

La théorie

La commande dd copie un fichier. Son originalité consiste dans le fait que la copie est effectuée bloc par bloc. Elle est donc adaptée à sauvegarder l'image d'un disque ou d'une partition.

Le savoir concret

Les principaux arguments

`if=fichier`	Le fichier source, par défaut l'entrée standard.
`of=fichier`	Le fichier destination, par défaut la sortie standard.
`bs=taille`	La taille des blocs.
`ibs=taille`	La taille des blocs en entrée.
`obs=taille`	La taille des blocs en sortie.
`count=n`	Le nombre de blocs copiés.
`skip=n`	Le nombre de blocs que l'on saute en entrée (on ne les copie pas).
`seek=n`	Le nombre de blocs que l'on saute en sortie (on ne les écrase pas).

Remarque : Il existe aussi de nombreux arguments réalisant des conversions.

Exemples

a) Sauvegarde d'un disque complet.

```
# dd if=/dev/sda  of=/dev/st0
```

b) Restauration d'un disque complet.

```
# dd if=/dev/st0  of=/dev/sda
```

c) Sauvegarde du MBR.

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

d) Connaître le format d'une cartouche.

```
# dd if=/dev/st0  of=/tmp/bloc count=1
# file /tmp/bloc
```

Pour en savoir plus

Les pages de manuel

dd(1)

La sauvegarde réseau

La théorie

Les commandes `rsh` ou `ssh` autorisent l'exécution de commandes à distante. Ce sont elles qui permettent d'effectuer une sauvegarde en réseau de manière explicite ou implicite.

Un serveur de sauvegarde doit donc supporter les services correspondants (`rshd` ou `sshd`). Il doit également posséder une commande de type `rmt` qui contrôle le positionnement sur la cartouche et qui est gérée à distance.

Le savoir concret

Les commandes tar, dd et rsh ou ssh

a) Sauvegarde de l'arborescence /home sur le serveur distant venus.

```
# tar cvzf - /home | rsh venus dd of=/dev/st0
```

b) Restauration des fichiers.

```
# rsh  venus dd if=/dev/st0 | tar xvzf -
```

La commande tar

a) Sauvegarde de l'arborescence /home sur le serveur distant venus.

```
# tar cvzf  venus:/dev/st0  /home
```

b) Restauration des fichiers

```
# tar xvzf  venus:/dev/st0
```

La sauvegarde incrémentale

a) Sauvegarde du FS /home sur le serveur distant venus.

```
# dump  0uf  venus:/dev/st0  /home
```

b) restauration des fichiers.

```
# restore -rf venus:/dev/st0
```

La commande rsync

a) Copier l'arborescence home sur le site distant (ou uniquement les différences si l'arborescence existe sur le site distant). On utilise `rsh`.

```
# rsync -avz  home/   venus:home/
```

b) Idem, mais on utilise `ssh`.

```
# rsync -avz -e ssh home/ venus:home/
```

Remarque : La commande `rsync` utilise le protocole SSH ou RSH, elle peut utiliser également le protocole RSYNC.

Les variables d'environnement

RSH	Le shell utilisé pour les transferts (`rsh` ou `ssh`). Supportée par `tar` et `dump` et `restore`.
RSYNC_RSH	Idem, supportée par `rsync`.
TAPE	Le support d'archive. Supportée par `tar` et `dump`.

Remarques :

1) Les commandes utilisent habituellement `rsh` par défaut.

2) Les commandes peuvent choisir leur mode de transfert par des options de la ligne de commande au lieu d'utiliser l'environnement ;

Les protocoles

22/tcp	Protocole SSH (les échanges sont cryptés et éventuellement authentifiés).
514/tcp	Shell (RSH) (les échanges sont à priori en clair).
873/tcp	Protocole RSYNC.
88, 464, 4444 en udp/tcp et 749/tcp et 754/tcp	Kerberos.

Remarque : La plupart des commandes Unix classiques, comme `telnet` ou `rsh` peuvent utiliser le protocole sécurisé Kerberos au lieu d'utiliser directement TCP ou UDP.

Pour en savoir plus

Les pages de manuel

tar(1), dd(1), rsh(1), ssh(1), dump(8), restore(8), xfsdump(8), xfsrestore(8), rmt(8), rsync(1)

La sauvegarde Bare-Metal

La théorie

Le principe

Une sauvegarde Bare-Metal consiste à sauvegarder un serveur de telle manière que l'on puisse le restaurer y compris en réinstallant le système sur un nouvel ordinateur.

Remarque : Cette sauvegarde est concurrencée par les installations scriptées de type KickStart.

La sauvegarde Bare-Metal manuelle

1) On sauvegarde l'architecture du système : taille et usage des partitions (points de montage, options de montage, options de formatage des partitions,…).
   # fdisk –l
   # cat /etc/fstab
2) On sauvegarde les FS systèmes (/, /usr, /var, /boot) avec les commandes incrémentales.

Remarque : La sauvegarde des autres FS (/home, ..) doit être régulièrement effectuée et ne fait pas stricto sensu partie de la sauvegarde Bare-Metal.

La restauration manuelle

1) On démarre à partir d'un système bootable (cdrom,…).
2) On restaure l'architecture du système (on repartitionne, on reformate).
3) On restaure les FS essentiels.
4) On rend le disque de démarrage bootable.
5) On redémarre le système.

Le savoir concret

Les commandes et les logiciels de sauvegarde Bare-Metal

`dd`	Il est simple de sauvegarder le disque système par cette commande.
`Clonezilla`	Produit réseau qui permet de cloner un système.
`Mondo`	Logiciel de sauvegarde Bare-Metal.
`Bacula`	Logiciel de sauvegarde client/serveur, simplifie la sauvegarde Bare-Metal.

Remarque : Sur un système en RAID miroir, on peut prendre un des disques système, il peut servir de sauvegarde (il faut diminuer, voire arrêter l'exploitation).

Pour en savoir plus

Howto

Linux Complete Backup and Recovery HOWTO

Le logiciel Mondo

La théorie

Le logiciel Mondo Rescue est un logiciel de sauvegarde Bare-Metal en version GPL. Il supporte les principales distributions (RedHat, SuSE, Mandriva, Debian et Gentoo).

Il supporte les cartouches, les disques, le réseau et les cdrom/dvdrom comme support d'archives. Il permet de sauvegarder des systèmes composés de plusieurs FS, utilisant le LVM et le RAID matériel ou logiciel.

Le savoir concret

Sauvegarde - Exemple

```
# mondoarchive -Oi -d /mnt/nfs -E /mnt/nfs
```

Sauvegarde complète du système sous forme d'images ISO dans le répertorie /mnt/nfs.

Remarque : l'option –d précise le lieu de la sauvegarde :

1) C'est un répertoire, par exemple un répertoire de montage NFS ou associé à un disque externe.
2) C'est un périphérique d'archivage comme /dev/st0.
3) C'est un graveur spécifié selon la convention de cdrecord (0,1,0).

Restauration

On démarre à partir du 1er cdrom ou dvdrom généré par la sauvegarde. Il est bootable.

Les principaux choix sont les suivants :

- Nuke : Restauration automatique.
- Interactive : Restauration manuelle, on restaure un fichier, plusieurs fichiers, …
- Expert : Vous obtenez un shell. Vous faîtes ce que vous voulez.

Pour en savoir plus

Les pages de manuel

mondoarchive(8), mondorestore(8)

Internet

Mondo – Le site officiel
http://www.mondorescue.org/

Les cartouches

La théorie

Une cartouche magnétique peut contenir plusieurs fichiers. Chaque fichier est séparé physiquement du suivant par une marque de fin de fichier (EOF). Quand on parle de fichier, c'est une archive généré par une commande tar, dump ou dd.

En argument des commandes de sauvegardes, on peut préciser un fichier spécial qui implique un rembobinage ou non en fin de sauvegarde. Si l'on désire créer plusieurs archives les unes après les autres, il faut utiliser le périphérique sans rembobinage.

Le savoir concret

Les fichiers spéciaux

/dev/st0	Le premier lecteur de cartouche.
/dev/st1	Le deuxième lecteur de cartouche.
/dev/nst0	Le premier lecteur de cartouche, pas de rembobinage.
/dev/tape	Nom générique du lecteur de cartouche par défaut.

Les commandes

`mt`	Permet de se déplacer sur une cartouche.
`mtx`	Gère un robot (demande d'une cartouche particulière, …).

Focus : les arguments de la commande mt

`rewind`	Rembobinage.
`offline`	Rembobinage et éjection.
`fsf [<n>]`	Saute un ou plusieurs fichiers. Par défaut, un fichier.
`bsf [<n>]`	Revient en arrière d'un ou plusieurs fichiers.

Exemple

```
# mt -f /dev/nst0 fsf 1   # saute un fichier
```

Pour en savoir plus

Les pages de manuel

mt(1), st(4), mtx(1), tapeinfo(1), scsitape(1), loaderinfo(1)

Howto

Ftape-HOWTO

Internet

Backup Hardware
http://www.backupcentral.com/component/option,com_mambowiki/Itemid,36/

Archival survival guide: ZDNet Australia: Reviews: Hardware: Storage
http://www.zdnet.com.au/reviews/hardware/storage/soa/Archival_survival_guide/0,139023427,139206969,00.htm

ATELIERS



Tâche 1 : Sauvegarde incrémentale avec dump et restore 15mn
Tâche 2 : La commande dd 5mn
Tâche 3 : Sauvegarde en réseau 10mn
Tâche 4 : Sauvegarde dump en réseau 10mn
Tâche 5 : Sauvegarde Bare-Metal manuelle 40mn
Tâche 6 : Sauvegarde Bare-Metal avec Mondo 60mn

Tâche 1 : Sauvegarde incrémentale avec dump et restore

0. Préliminaire : créer une partition de type Linux.

```
[root@linux1 ~]# fdisk /dev/hda

The number of cylinders for this disk is set to 4870.
There is nothing wrong with that, but this is larger than 1024,
and could in certain setups cause problems with:
1) software that runs at boot time (e.g., old versions of LILO)
2) booting and partitioning software from other OSs
   (e.g., DOS FDISK, OS/2 FDISK)

Command (m for help): n
First cylinder (1419-4111, default 1419):
Using default value 1419
Last cylinder or +size or +sizeM or +sizeK (1419-4111, default 4111): +512M

Command (m for help): w
The partition table has been altered!

Calling ioctl() to re-read partition table.

WARNING: Re-reading the partition table failed with error 16: Device or resource
busy.
The kernel still uses the old table.
The new table will be used at the next reboot.
Syncing disks.
[root@linux1 ~]# partprobe
```

1. Créer un FS et le peupler.

```
[root@linux1 ~]# mkfs -j -q /dev/hda6
[root@linux1 ~]# mount -t ext3 /dev/hda6 /mnt/disk
[root@linux1 ~]# cd /mnt/disk
[root@linux1 disk]# cal > f1
[root@linux1 disk]# date > f2
[root@linux1 disk]# mkdir rep
[root@linux1 disk]# uptime > rep/fic
[root@linux1 disk]# cd
[root@linux1 ~]#
```

2. Effectuer une sauvegarde complète du FS.

```
[root@linux1 ~]# dump 0uf /tmp/sauve_0.dump /dev/hda6
  DUMP: Date of this level 0 dump: Sat Feb 24 19:25:09 2007
  DUMP: Dumping /dev/hda6 (/mnt/disk) to /tmp/sauve_0.dump
  DUMP: Label: none
  DUMP: Writing 10 Kilobyte records
  DUMP: mapping (Pass I) [regular files]
  DUMP: mapping (Pass II) [directories]
  DUMP: estimated 60 blocks.
  DUMP: Volume 1 started with block 1 at: Sat Feb 24 19:25:11 2007
  DUMP: dumping (Pass III) [directories]
  DUMP: dumping (Pass IV) [regular files]
  DUMP: Closing /tmp/sauve_0.dump
  DUMP: Volume 1 completed at: Sat Feb 24 19:25:11 2007
  DUMP: Volume 1 50 blocks (0.05MB)
  DUMP: 50 blocks (0.05MB) on 1 volume(s)
  DUMP: finished in less than a second
  DUMP: Date of this level 0 dump: Sat Feb 24 19:25:09 2007
  DUMP: Date this dump completed:  Sat Feb 24 19:25:11 2007
  DUMP: Average transfer rate: 0 kB/s
  DUMP: DUMP IS DONE
[root@linux1 ~]# cat /etc/dumpdates
/dev/hda6 0 Sat Feb 24 19:25:09 2007 +0100
[root@linux1 ~]#
```

Remarque : On aurait pu indiquer /mnt/disk si le FS avait été inscrit dans /etc/fstab.

3. Lister le contenu de la sauvegarde.

```
[root@linux1 ~]# restore -tvf /tmp/sauve_0.dump
Verify tape and initialize maps
Input is from a local file/pipe
Input block size is 32
Dump   date: Sat Feb 24 19:25:09 2007
Dumped from: the epoch
Level 0 dump of /mnt/disk on linux1.pinguins:/dev/hda6
Label: none
Extract directories from tape
Initialize symbol table.
dir          2  .
dir         11  ./lost+found
leaf      6145  ./f1
leaf      6146  ./f2
dir       4097  ./rep
leaf      4098  ./rep/fic
[root@linux1 ~]#
```

4. Effectuer une sauvegarde incrémentale.

```
[root@linux1 ~]# cp /etc/profile /mnt/disk/rep
[root@linux1 ~]# dump 1uf /tmp/sauve_1.dump  /dev/hda6
  DUMP: Date of this level 1 dump: Sat Feb 24 19:28:05 2007
  DUMP: Date of last level 0 dump: Sat Feb 24 19:25:09 2007
  DUMP: Dumping /dev/hda6 (/mnt/disk) to /tmp/sauve_1.dump
  DUMP: Label: none
  DUMP: Writing 10 Kilobyte records
  DUMP: mapping (Pass I) [regular files]
```

```
  DUMP: mapping (Pass II) [directories]
  DUMP: estimated 48 blocks.
  DUMP: Volume 1 started with block 1 at: Sat Feb 24 19:28:07 2007
  DUMP: dumping (Pass III) [directories]
  DUMP: dumping (Pass IV) [regular files]
  DUMP: Closing /tmp/sauve_1.dump
  DUMP: Volume 1 completed at: Sat Feb 24 19:28:07 2007
  DUMP: Volume 1 50 blocks (0.05MB)
  DUMP: 50 blocks (0.05MB) on 1 volume(s)
  DUMP: finished in less than a second
  DUMP: Date of this level 1 dump: Sat Feb 24 19:28:05 2007
  DUMP: Date this dump completed:  Sat Feb 24 19:28:07 2007
  DUMP: Average transfer rate: 0 kB/s
  DUMP: DUMP IS DONE
[root@linux1 ~]# cat /etc/dumpdates
/dev/hda6 0 Sat Feb 24 19:25:09 2007 +0100
/dev/hda6 1 Sat Feb 24 19:28:05 2007 +0100
[root@linux1 ~]#
```

5. Restaurer l'intégralité du FS.

```
[root@linux1 ~]# umount /mnt/disk
[root@linux1 ~]# mkfs -j -q /dev/hda6
```

Remarque : On simule un gros problème en reformatant la partition.

```
[root@linux1 ~]# mount -t ext3 /dev/hda6 /mnt/disk/
[root@linux1 ~]# cd /mnt/disk/
[root@linux1 disk]# restore -rvf /tmp/sauve_0.dump
Verify tape and initialize maps
Input is from a local file/pipe
Input block size is 32
Dump   date: Sat Feb 24 19:25:09 2007
Dumped from: the epoch
Level 0 dump of /mnt/disk on linux1.pinguins:/dev/hda6
Label: none
Begin level 0 restore
Initialize symbol table.
Extract directories from tape
Calculate extraction list.
restore: ./lost+found: File exists
Make node ./rep
Extract new leaves.
Check pointing the restore
extract file ./rep/fic
extract file ./f1
extract file ./f2
Add links
Set directory mode, owner, and times.
Check the symbol table.
Check pointing the restore
[root@linux1 disk]# restore -rvf /tmp/sauve_1.dump
Verify tape and initialize maps
Input is from a local file/pipe
Input block size is 32
Dump   date: Sat Feb 24 19:28:05 2007
```

```
Dumped from: Sat Feb 24 19:25:09 2007
Level 1 dump of /mnt/disk on linux1.pinguins:/dev/hda6
Label: none
Begin incremental restore
Initialize symbol table.
Extract directories from tape
Mark entries to be removed.
Calculate node updates.
Find unreferenced names.
Remove old nodes (directories).
Extract new leaves.
Check pointing the restore
extract file ./rep/profile
Add links
Set directory mode, owner, and times.
Check the symbol table.
Check pointing the restore
[root@linux1 disk]# find
.
./lost+found
./rep
./rep/fic
./rep/profile
./restoresymtable
./f1
./f2
[root@linux1 disk]# cd
```

Remarque : On doit restaurer l'archive complète (de niveau 0) et ensuite l'archive incrémentale (de niveau 1).

6. Restaurer un fichier.

a) On détruit un fichier par erreur.

```
[root@linux1 ~]# rm -f /mnt/disk/f1
```

b) On le restaure.

```
[root@linux1 ~]# cd /mnt/disk
[root@linux1 disk]# restore -xvaof /tmp/sauve_0.dump  ./f1
Verify tape and initialize maps
Input is from a local file/pipe
Input block size is 32
Dump   date: Sat Feb 24 19:25:09 2007
Dumped from: the epoch
Level 0 dump of /mnt/disk on linux1.pinguins:/dev/hda6
Label: none
Extract directories from tape
Initialize symbol table.
Extract requested files
extract file ./f1
Add links
Set directory mode, owner, and times.
[root@linux1 disk]# ls -l f1
-rw-r--r--  1 root root 137 Feb 24 19:20 f1
[root@linux1 disk]# cd
```

7. Restaurer un fichier de manière intéractive.

a) On détruit un fichier par erreur.

```
[root@linux1 ~]# rm -f /mnt/disk/f1
```

b) On le restaure.

```
[root@linux1 ~]# cd /mnt/disk
[root@linux1 disk]# restore -if /tmp/sauve_0.dump
restore > help
Available commands are:
        ls [arg] - list directory
        cd arg - change directory
        pwd - print current directory
        add [arg] - add `arg' to list of files to be extracted
        delete [arg] - delete `arg' from list of files to be extracted
        extract - extract requested files
        setmodes - set modes of requested directories
        quit - immediately exit program
        what - list dump header information
        verbose - toggle verbose flag (useful with ``ls'')
        prompt - toggle the prompt display
        help or `?' - print this list
If no `arg' is supplied, the current directory is used
restore > what
Dump   date: Sat Feb 24 19:25:09 2007
Dumped from: the epoch
Level 0 dump of /mnt/disk on linux1.pinguins:/dev/hda6
Label: none
restore > ls
.:
f1          f2          lost+found/ rep/

restore > cd rep
restore > pwd
/rep
restore > ls
./rep:
fic

restore > add fic
restore: ./rep: File exists
restore > ls
./rep:
*fic

restore > del fic
restore > ls
./rep:
fic

restore > cd ..
restore > pwd
/
restore > add f1
```

```
restore > ls
.:
*f1            f2            lost+found/ *rep/
restore > extract
You have not read any volumes yet.
Unless you know which volume your file(s) are on you should start
with the last volume and work towards the first.
Specify next volume # (none if no more volumes): 1
set owner/mode for '.'? [yn] n
restore > quit
[root@linux1 disk]# ls -l f1
-rw-r--r--  1 root root 137 Feb 24 19:20 f1
[root@linux1 disk]# cd
[root@linux1 ~]# umount /mnt/disk/
```

Tâche 2 : La commande dd

1. Sauvegarder la partition hda6 dans un fichier.

```
[root@linux1 ~]# dd if=/dev/hda6 of=/usr/image.hda6
1012032+0 records in
1012032+0 records out
```

2. Détruire le formatage de cette partition.

```
[root@linux1 ~]# dd if=/dev/zero of=/dev/hda6 bs=1k count=100
100+0 records in
100+0 records out
[root@linux1 ~]# tune2fs -l /dev/hda6
tune2fs 1.35 (28-Feb-2004)
tune2fs: Bad magic number in super-block while trying to open /dev/hda6
Couldn't find valid filesystem superblock.
[root@linux1 ~]#
```

3. Restaurer la partition.

```
[root@linux1 ~]# dd of=/dev/hda6 if=/usr/image.hda6
1012032+0 records in
1012032+0 records out
[root@linux1 ~]# tune2fs -l /dev/hda6 | head -3
tune2fs 1.35 (28-Feb-2004)
Filesystem volume name:   <none>
Last mounted on:          <not available>
[root@linux1 ~]# mount -t ext3 /dev/hda6 /mnt/disk
[root@linux1 ~]# ls /mnt/disk
f1  f2  lost+found  rep  restoresymtable
[root@linux1 ~]# umount /mnt/disk/
[root@linux1 ~]#
```

4. Sauvegarder le MBR.

```
[root@linux1 ~]# dd if=/dev/hda of=/root/MBR bs=512 count=1
1+0 records in
1+0 records out
[root@linux1 ~]# file /root/MBR
/root/MBR: x86 boot sector, code offset 0x48
[root@linux1 ~]#
```

Tâche 3 : Sauvegarde en réseau

1. Configurer un serveur RSH.

a) On active le service RSH sur le serveur.

```
[root@dracaufeu ~]# rpm -q rsh-server
rsh-server-0.17-29
root@dracaufeu ~]# echo linux1.pinguins >> /etc/hosts.equiv
[root@dracaufeu ~]# echo linux1.pinguins >> /root/.rhosts
[root@dracaufeu ~]# vi /etc/pam.d/rsh
#%PAM-1.0
# For root login to succeed here with pam_securetty, "rsh" must be
# listed in /etc/securetty.
auth       required     pam_nologin.so
#auth       required     pam_securetty.so
auth       required     pam_env.so
auth       required     pam_rhosts_auth.so
account    required     pam_stack.so service=system-auth
session    required     pam_stack.so service=system-auth
```

Remarque : On met un commentaire sur la ligne `pam_securetty.so`.

```
[root@dracaufeu ~]# chkconfig rsh on
[root@dracaufeu ~]# netstat -an |grep 514
tcp        0      0 0.0.0.0:514                 0.0.0.0:*
LISTEN
[root@dracaufeu ~]#
```

b) On teste le service à partir du client.

```
[root@linux1 ~]# rsh instructeur.pinguins date
connect to address 192.168.0.200: Connection refused
Trying krb4 rsh...
connect to address 192.168.0.200: Connection refused
trying normal rsh (/usr/bin/rsh)
Sat Feb 24 21:13:43 CET 2007
[root@linux1 ~]#
```

c) On crée sur le serveur le fichier d'archive.

```
[root@dracaufeu ~]# touch /root/backup.dev
```

Remarque : Le fichier qui simule le périphérique de sauvegarde doit exister préalablement à la sauvegarde.

2. Réaliser une sauvegarde avec tar.

```
[root@linux1 ~]# mount -t ext3 /dev/hda6 /mnt/disk
[root@linux1 ~]# tar cvf instructeur:/root/backup.dev /mnt/disk/
tar: Removing leading `/' from member names
/mnt/disk/
/mnt/disk/lost+found/
/mnt/disk/rep/
/mnt/disk/rep/fic
/mnt/disk/rep/profile
/mnt/disk/restoresymtable
/mnt/disk/f1
/mnt/disk/f2
[root@linux1 ~]# tar tvf instructeur:/root/backup.dev
drwxr-xr-x root/root         0 2007-02-24 20:13:51 mnt/disk/
```

```
drwx------ root/root         0 2007-02-24 19:19:59 mnt/disk/lost+found/
drwxr-xr-x root/root         0 2007-02-24 19:27:50 mnt/disk/rep/
-rw-r--r-- root/root        72 2007-02-24 19:20:54 mnt/disk/rep/fic
-rw-r--r-- root/root       842 2007-02-24 19:27:50 mnt/disk/rep/profile
-rw------- root/root    117472 2007-02-24 19:35:35 mnt/disk/restoresymtable
-rw-r--r-- root/root       137 2007-02-24 19:20:28 mnt/disk/f1
-rw-r--r-- root/root        29 2007-02-24 19:20:30 mnt/disk/f2
[root@linux1 ~]# rm -rf /mnt/disk/*
[root@linux1 ~]# cd /
[root@linux1 /]# tar xvf instructeur:/root/backup.dev
mnt/disk/
mnt/disk/lost+found/
mnt/disk/rep/
mnt/disk/rep/fic
mnt/disk/rep/profile
mnt/disk/restoresymtable
mnt/disk/f1
mnt/disk/f2
[root@linux1 /]# ls /mnt/disk
f1  f2  lost+found  rep  restoresymtable
[root@linux1 /]# cd
```

3. Sauvegarder en utilisant tar, dd et rsh.

```
[root@linux1 ~]# tar cvzf - /mnt/disk | rsh instructeur dd of=/root/backup.dev
tar: Removing leading `/' from member names
/mnt/disk/
/mnt/disk/lost+found/
/mnt/disk/rep/
/mnt/disk/rep/fic
/mnt/disk/rep/profile
/mnt/disk/restoresymtable
connect to address 192.168.0.200: Connection refused
Trying krb4 rsh...
connect to address 192.168.0.200: Connection refused
trying normal rsh (/usr/bin/rsh)
/mnt/disk/f1
/mnt/disk/f2
20+0 records in
20+0 records out
[root@linux1 ~]# rsh instructeur dd if=/root/backup.dev | tar tvzf -
connect to address 192.168.0.200: Connection refused
Trying krb4 rsh...
connect to address 192.168.0.200: Connection refused
trying normal rsh (/usr/bin/rsh)
20+0 records in
20+0 records out
drwxr-xr-x root/root         0 2007-02-24 20:13:51 mnt/disk/
drwx------ root/root         0 2007-02-24 19:19:59 mnt/disk/lost+found/
drwxr-xr-x root/root         0 2007-02-24 19:27:50 mnt/disk/rep/
-rw-r--r-- root/root        72 2007-02-24 19:20:54 mnt/disk/rep/fic
-rw-r--r-- root/root       842 2007-02-24 19:27:50 mnt/disk/rep/profile
-rw------- root/root    117472 2007-02-24 19:35:35 mnt/disk/restoresymtable
-rw-r--r-- root/root       137 2007-02-24 19:20:28 mnt/disk/f1
-rw-r--r-- root/root        29 2007-02-24 19:20:30 mnt/disk/f2
```

```
[root@linux1 ~]# rm -rf /mnt/disk/*
[root@linux1 ~]# cd /
[root@linux1 /]# rsh instructeur dd if=/root/backup.dev | tar xvzf -
connect to address 192.168.0.200: Connection refused
Trying krb4 rsh...
connect to address 192.168.0.200: Connection refused
trying normal rsh (/usr/bin/rsh)
20+0 records in
20+0 records out
mnt/disk/
mnt/disk/lost+found/
mnt/disk/rep/
mnt/disk/rep/fic
mnt/disk/rep/profile
mnt/disk/restoresymtable
mnt/disk/f1
mnt/disk/f2
[root@linux1 /]# cd
[root@linux1 ~]#
```

4. Sauvegarder avec rsync.

```
[root@linux1 ~]# cd /mnt
[root@linux1 mnt]# rsync -avz -e rsh disk/  instructeur:disk/
connect to address 192.168.0.200: Connection refused
Trying krb4 rsh...
connect to address 192.168.0.200: Connection refused
trying normal rsh (/usr/bin/rsh)
building file list ... done
created directory disk
./
f1
f2
lost+found/
rep/
rep/fic
rep/profile
restoresymtable

sent 1417 bytes  received 120 bytes  1024.67 bytes/sec
total size is 118552  speedup is 77.13
[root@linux1 mnt]# cd
```

Remarques :

1) Selon la version, le transfert se fait par défaut en RSH ou SSH. Dans cette version, le transfert par défaut est réalisé en SSH. On peut aussi utiliser le protocole RSYNC.

2) Si l'on active périodiquement cette commande (par un crontab par exemple), la commande ne transfert que les modifications apportées à l'arborescence.

3) La sauvegarde a eu lieu à partir du répertoire de connexion de l'utilisateur distant. Dans le cas présent /root. Ainsi on a créé l'arborescence /root/disk.

Tâche 4 : Sauvegarde dump en réseau

1. Créer un serveur RSH (cf. tâche 3).

2. Créer un fichier crontab pour automatiser la sauvegarde.

On réalise une sauvegarde complète tout les dimanches et une sauvegarde différentielle par jour à 23h.

```
[root@linux1 ~]# crontab -e
0 23 * * 0   /sbin/dump 0uf instructeur:/root/backup.dev /dev/hda6 >>
/var/log/dump.log 2>&1
0 23 * * 1-5 /sbin/dump 9uf instructeur:/root/backup.dev /dev/hda6 >>
/var/log/dump.log 2>&1
```

3. Modifier temporairement le crontab pour tester son fonctionnement.

On déclenche la sauvegarde complète dans deux minutes environ (il faut modifier l'heure, les minutes et le jour de la semaine !).

```
[root@linux1 ~]# crontab -l > crontab.bck
[root@linux1 ~]# crontab -e
```

On attend le déclenchement automatique...

```
[root@linux1 ~]# touch /var/log/dump.log
[root@linux1 ~]# tail -f /var/log/dump.log
...
  DUMP: dumping (Pass IV) [regular files]
  DUMP: Closing /root/backup.dev
  DUMP: Volume 1 completed at: Sat Feb 24 22:27:04 2007
  DUMP: Volume 1 170 blocks (0.17MB)
  DUMP: 170 blocks (0.17MB) on 1 volume(s)
  DUMP: finished in less than a second
  DUMP: Date of this level 0 dump: Sat Feb 24 22:27:01 2007
  DUMP: Date this dump completed:  Sat Feb 24 22:27:04 2007
  DUMP: Average transfer rate: 0 kB/s
  DUMP: DUMP IS DONE
^C
[root@linux1 ~]# tail -2 /var/log/cron
Feb 24 22:26:01 linux1 crond[3544]: (root) RELOAD (cron/root)
Feb 24 22:27:01 linux1 crond[12184]: (root) CMD (/sbin/dump 0uf
instructeur:/root/backup.dev /dev/hda6 >> /var/log/dump.log 2>&1)
```

4. Essayer de restaurer un fichier.

```
[root@linux1 ~]# rm /mnt/disk/f1
rm: remove regular file `/mnt/disk/f1'? y
[root@linux1 ~]# restore -tvf instructeur:/root/backup.dev
Connection to instructeur established.
Verify tape and initialize maps
connect to address 192.168.0.200: Connection refused
Trying krb4 rsh...
connect to address 192.168.0.200: Connection refused
trying normal rsh (/usr/bin/rsh)
Input is from a remote tape
Tape block size is 32
Dump   date: Sat Feb 24 22:27:01 2007
Dumped from: the epoch
Level 0 dump of /mnt/disk on linux1.pinguins:/dev/hda6
Label: none
Extract directories from tape
```

```
Initialize symbol table.
dir          2  .
dir      28673  ./lost+found
dir     118785  ./rep
leaf    118786  ./rep/fic
leaf    118787  ./rep/profile
leaf         11  ./restoresymtable
leaf         12  ./f1
leaf         13  ./f2
[root@linux1 ~]# cd /mnt/disk
[root@linux1 disk]# restore -axvof instructeur:/root/backup.dev ./f1 2>/dev/null
Connection to instructeur established.
Verify tape and initialize maps
Input is from a remote tape
Tape block size is 32
Dump   date: Sat Feb 24 22:27:01 2007
Dumped from: the epoch
Level 0 dump of /mnt/disk on linux1.pinguins:/dev/hda6
Label: none
Extract directories from tape
Initialize symbol table.
Extract requested files
extract file ./f1
Add links
Set directory mode, owner, and times.
[root@linux1 disk]# ls /mnt/disk/f1
/mnt/disk/f1
[root@linux1 disk]# cd
[root@linux1 ~]# umount /mnt/disk
[root@linux1 ~]#
```

Tâche 5 : Sauvegarde Bare-Metal manuelle

1. Créer un serveur RSH (cf. tâche 3).

2. Effectuer la sauvegarde.

a) On crée les répertoires et les fichiers nécessaires sur le serveur.

```
instructeur# mkdir /usr/BCK
instructeur# touch /usr/BCK/root.dump
instructeur# touch /usr/BCK/boot.dump
instructeur# touch /usr/BCK/home.dump
```

b) On mémorise l'architecture du client.

```
venus# df -h
Filesystem            Size  Used Avail Use% Mounted on
/dev/hda2             6.8G  1.6G  4.8G  25% /
/dev/hda1             198M   17M  172M   9% /boot
/dev/hda3             244M  4.2M  227M   2% /home
none                   31M     0   31M   0% /dev/shm
venus# cat /etc/fstab
LABEL=/                 /           ext3        defaults        1 1
LABEL=/boot             /boot       ext3        defaults        1 2
LABEL=/home             /home       ext3        defaults        1 2
LABEL=SWAP-hda5         swap         swap       defaults        0 0
```

```
...
venus# fdisk  -l  /dev/hda
...
venus# cat /boot/grub/grub.conf
...
```

c) On sauvegarde la table des partitions.

```
venus# export LANG=C
venus# sfdisk -d /dev/hda > /tmp/hda.tbl
venus# rcp /tmp/hda.tbl  instructeur:/usr/BCK/
```

c-bis) On sauvegarde la table des partitions.

Une alternative (plus tolérante aux pannes) est de créer un script pour fdisk.

```
Instructeur # cat > /usr/BCK/fdisk.script
o
n
p
1

+100M
n
p
2

+10000M
n
e
3


n
l

+512M
t
5
82
w
^D
```

d) On sauvegarder les FS.

```
venus# dump 0f instructeur:/usr/BCK/root.dump /
venus# dump 0f instructeur:/usr/BCK/boot.dump /boot
venus# dump 0f instructeur:/usr/BCK/home.dump /home
```

3. Effectuer la restauration.

Suite à un crash disque, votre système est HS, il faut le reconstituer sur une autre machine.

```
venus# dd if=/dev/zero of=/dev/hda bs=512 count=1     # simule un crash disque
```

a) On démarre sur un système autonome (par exemple le cd d'installation).

```
boot : linux rescue
# ifconfig eth0 192.168.0.1
```

b) On recrée la table des partitions.

```
# rcp 192.168.0.200:/usr/BCK/hda.tbl /tmp
# sfdisk /dev/hda < /tmp/hda.tbl
```

```
# fdisk -l /dev/hda
```

b-bis) On recrée la table des partitions à partir du script.

```
# rcp 192.168.0.200:/usr/BCK/fdisk.script /tmp
# fdisk /dev/hda < /tmp/fdisk.script
# fdisk -l /dev/hda
```

c) On crée les FS, on les étiquette, on formate et on étiquette le swap.

```
# mke2fs -j /dev/hda1
# mke2fs -j /dev/hda2
# mke2fs -j /dev/hda3
# e2label  /dev/hda1  /boot
# e2label  /dev/hda2  /
# e2label  /dev/hda3  /home
# mkswap -L SWAP-hda5  /dev/hda5
```

d) On restaure les FS.

```
# mkdir /mnt/images
# mount /dev/hda2 /mnt/images
# cd /mnt/images
# restore -rvf 192.168.0.200:/usr/BCK/root.dump
# mount /dev/hda1 /mnt/images/boot
# cd boot
# restore -rvf 192.168.0.200:/usr/BCK/boot.dump
# mount /dev/hda3 /mnt/images/home
# cd ../home
# restore -rvf 192.168.0.200:/usr/BCK/home.dump
```

e) On réinstalle le chargeur (au préalable on installe les fichiers périphériques).

```
# cd  /mnt/images
# chroot .
# mount  /proc
# mount  /sys
# udevstart   #  recrée les entrées de périphériques
# grub-install /dev/hda
```

f) On redémarre le système.

```
CTRL-ALT-SUPPR
```

Tâche 6 : La sauvegarde Bare-Metal avec Mondo

1. Télécharger le logiciel.

```
# wget 'ftp://ftp.mondorescue.org/rhel/4/afio-2.4.7-1.i586.rpm'
# wget 'ftp://ftp.mondorescue.org/rhel/4/mindi-1.2.1-1.rhel4.i586.rpm'
# wget 'ftp://ftp.mondorescue.org/rhel/4/mindi-busybox-1.2.2-1.rhel4.i586.rpm'
# wget 'ftp://ftp.mondorescue.org/rhel/4/mondo-2.2.1-1.rhel4.i586.rpm'
# wget 'ftp://ftp.mondorescue.org/rhel/4/mondo-doc-2.2.1-1.rhel4.noarch.rpm'
# wget 'ftp://ftp.mondorescue.org/rhel/4/buffer-1.19-1.i386.rpm'
```

2. Installer le logiciel.

```
# rpm -Uvh afio-2.4.7-1.i586.rpm
# rpm -Uvh mindi-busybox-1.2.2-1.rhel4.i586.rpm
# rpm -Uvh mindi-1.2.1-1.rhel4.i586.rpm
# rpm -Uvh mondo-2.2.1-1.rhel4.i586.rpm
# rpm -Uvh mondo-doc-2.2.1-1.rhel4.noarch.rpm
# rpm -Uvh buffer-1.19-1.i386.rpm
```

3. Sauvegarder le système sur des cdroms grâce à un graveur local.

On sauvegarde sur CD (limité à 700 mega chacun) à la vitesse x16. Le lecteur a été trouvé à l'adresse 1,0,0 par cdrecord. On ne sauvegarde par ~ftp.

```
# mondoarchive -O -d ATAPI:1,0,0 -c 16 -s 700m -E ~ftp
```

4. Sauvegarder le système en client/serveur avec NFS.

a) On crée un serveur.

```
SRV# mkdir ~ftp/MONDO
SRV# chmod 777 ~ftp/MONDO/
SRV# vi /etc/exports
/var/ftp/MONDO  *(rw)
SRV# /etc/init.d/nfs restart
SRV# showmount -e localhost
```

b) Sur le client, on accède à la ressource NFS et on active la sauvegarde.

```
CLI# mkdir /mnt/nfs
CLI# mount -t nfs 192.168.218.15:/var/ftp/MONDO /mnt/nfs
CLI# mondoarchive -OV -n 192.168.218.15:/var/ftp/MONDO -F -E /var/ftp
```

c) On crée le système bootable.

```
# umount /mnt/nfs
# cdrecord dev=ATAPI:1,0,0 speed=16 -eject -v /root/images/mindi/mondorescue.iso
```

Remarque : On peut copier sur le serveur ce fichier et le graver si l'on ne dispose pas de graveur local.

5. Restaurer le système.

On démarre à partir du premier cdrom et on suit les instructions.

- *Chargeur primaire*
- *boot : linux rescue*
- *append=*
- *Grub : stage1, stage2*
- *initdefault*

10

Le démarrage

Objectifs

Ce chapitre traite du démarrage. Il a pour premier objectif de comprendre dans le détail le démarrage d'un système Linux non seulement pour le paramétrer mais aussi pour savoir agir dans le cas où le système ne démarre pas. Ce chapitre aborde également la création d'un cdrom ou d'une clé USB bootable.

Contenu

Les grandes étapes du démarrage

Dépannage du démarrage

Init

Le script rc.sysinit

Démarrer avec LILO

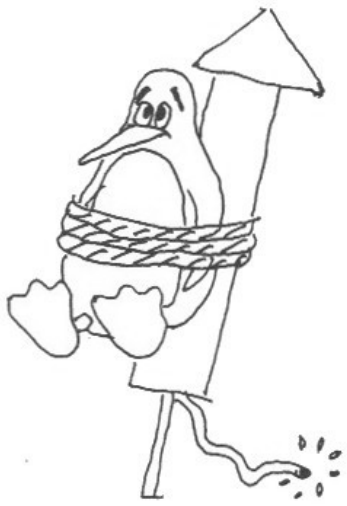
Démarrer avec GRUB

Démarrer avec Syslinux

Initrd

Créer un live-cd, une live-key

Ateliers



Les grandes étapes du démarrage

La théorie

Les grandes étapes du démarrage

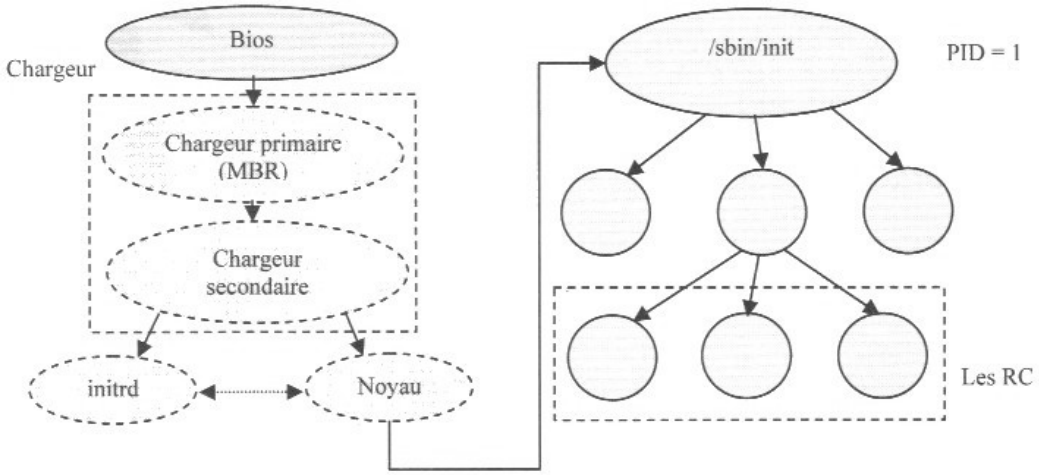


Fig. Les grandes étapes du démarrage.

Bios

Quand on allume l'ordinateur, le BIOS (Basic Input Output System) qui est en mémoire morte accomplit la 1ère phase de démarrage.

Chargeur

Le chargeur (loader) est activé par le BIOS, il a pour rôle d'activer le système d'exploitation, en l'occurrence le noyau Linux.

Noyau

Le noyau Linux (Kernel) à son tour active le 1er programme : /sbin/init, dont le PID est égal à 1.

Initrd

L'initrd est un système de fichiers racine installé directement en mémoire par le chargeur. L'utilisation d'un initrd est optionnelle. Son intérêt principal est de charger des modules du noyau nécessaires durant les premiers stades du démarrage, par exemple le module ext3. Si le noyau possède ces modules, l'initrd est inutile.

Init

Init est l'ancêtre de tous les processus.

RC

Les RC (Run Command) sont les scripts de démarrage des services.

Le chargeur

Le chargeur (loader) a pour rôle d'activer le système d'exploitation, en l'occurrence le noyau Linux. Le chargeur est composé de deux parties :

- Le chargeur primaire
- Le chargeur secondaire

Le chargeur primaire est activé par le BIOS. Son code est minuscule car il est contenu dans le 1er secteur d'un disque, le MBR (Master Boot Record). Il ne sert qu'à activer le chargeur secondaire. Celui-ci est le véritable chargeur. Son rôle est d'activer un système d'exploitation (Windows, Linux, ...). Dans le cas de Linux, il active un noyau Linux et lui passe des paramètres. Il peut être utile de configurer un chargeur pour activer plusieurs noyaux Linux. Par exemple, on peut offrir le choix de différentes versions du noyau ou d'une même version avec des paramètres différents.

Il existe plusieurs chargeurs libres :

- Lilo
- Grub
- Syslinux

Lilo est le chargeur le plus ancien. Il est très simple, on lui préfère actuellement Grub pour démarrer le système à partir du disque dur. Syslinux est utilisé pour démarrer en réseau (pxelinux) ou à partir d'un cdrom/dvdrom (isolinux) ou d'une clé USB (syslinux).

Remarque : Il est également possible de démarrer Linux à partir du chargeur Windows.

Les paramètres du noyau

Le chargeur non seulement active le noyau mais il lui transmet également des paramètres. Ceux-ci sont destinés pour la plupart à des modules du noyau. Si le noyau ne sait pas les interpréter il les transmet à `init`.

Le savoir concret

Les commandes

`dmesg`	Affiche les messages provenant du noyau, essentiellement les messages du démarrage.

Les fichiers

/proc/cmdline	Les paramètres du noyau actuel.

Les principaux paramètres du noyau

`root=`	Le FS root, par exemple /dev/hda2.
`ro`	Le FS root est monté en lecture seul (il sera monté en lecture/écriture après la vérification effectuée par `fsck`).
`rw`	Le FS root est monté en lecture/écriture. Ce choix n'est pas usuel.
`single`	On démarre en mode simple utilisateur.
`mem=`	Définit la taille maximale vue par le noyau, par exemple `mem=128m` pour limiter la mémoire à 128 Mo. Permet de tester le comportement du système avec une quantité particulière de mémoire (inférieure à la mémoire installée).
`nosmp`	Désactive la gestion multiprocesseurs.
`quiet`	Affiche peu de messages à l'écran.
`debug`	Affiche beaucoup de messages à l'écran.
`loglevel=`	Niveau des messages d'erreur affiché (de 0 à 7).
`init=`	Spécifie le programme init, par défaut /sbin/init.

`console=`	Définit un port série comme port de la console maîtresse, par exemple : `console=ttyS0,19200n81` pour une liaison sur COM1 : à 19200 bauds, sans parité, 8 bits de données et un bit stop.
`ramdisk_size=`	Taille maximale d'un disque en mémoire vive.
`acpi=`	Permet de désactiver l'ACPI (`acpi=off`).
`nolapic`	N'active pas l'APIC local.

Pour en savoir plus

Les pages de manuel

bootparam(7), boot(7), dmesg(1), lilo(8), grub(8), initrd(4)

Fichiers

kernel-parameters.txt	Les paramètres du noyau (fait partie de la documentation du noyau).

Howto

From-PowerUp-to-bash-prompt-HOWTO

BootPrompt-HOWTO (les paramètres du noyau)

Loadlin+Win95-98-ME

Linux i386 Boot Code HOWTO

Internet

Syslinux
http://syslinux.zytor.com/

Activer Linux à partir de boot.ini (Windows 2000, ...)
http://www.laboratoire-microsoft.org/articles/win/boot/

Dépannage du démarrage

La théorie

Si le système ne démarre pas, l'administrateur doit pouvoir démarrer autrement. Le support de démarrage le plus usuel est le cdrom. Typiquement on utilisera le 1er cdrom de la distribution ou un live-cd (Knoppix, SystemRescueCd, …). Il est bootable et permet le démarrage en mode maintenance.

Précautions

Il est important, en prévision d'un problème de démarrage de conserver une sauvegarde de la configuration (/etc) et particulièrement les fichiers /etc/inittab, /etc/fstab, le partitionnement (`fdisk -l`) et le fichier de configuration du noyau (grub.conf). Une sauvegarde complète du système résout évidemment tous les problèmes.

Le savoir concret

Démarrer le système à partir du cdrom

Sur un système RedHat, lorsque l'on démarre à partir du cdrom, on peut démarrer en mode maintenance en entrant la commande suivante (le FS root est en mémoire) :

```
boot: linux rescue
```

On peut aussi démarrer en mode maintenance, mais à partir du disque dur :

```
boot: linux single
```

On peut même démarrer sur un disque différent :

```
boot: linux single root=/dev/hdb1 initrd=
```

Les commandes utiles

`mount`	Monte un FS appartenant au disque dur.
`fdisk`	Gère les partitions.
`parted`	Gère les partitions sans perte de données.
`swapon`	Active un espace de swap.
`lvm`	Commande générique LVM (ex : lvm vgchange).
`vgchange`	Permet d'activer le VG présent sur le disque dur.
`lvextend`	Agrandit un LV, par exemple celui abritant le FS root.
`chroot`	Change de répertoire racine. Permet de basculer sur le disque dur.
`udevstart`	Démarre le système Udev, crée les entrées de périphériques.
`grub-install, lilo`	Réinstalle le chargeur.
`ifconfig, dhclient`	Fixent l'adresse réseau (statique ou dynamique).

Pour en savoir plus

Internet

Main Page – SystemRescueCd
http://www.sysresccd.org/

Init

La théorie

Le concept de niveau d'init

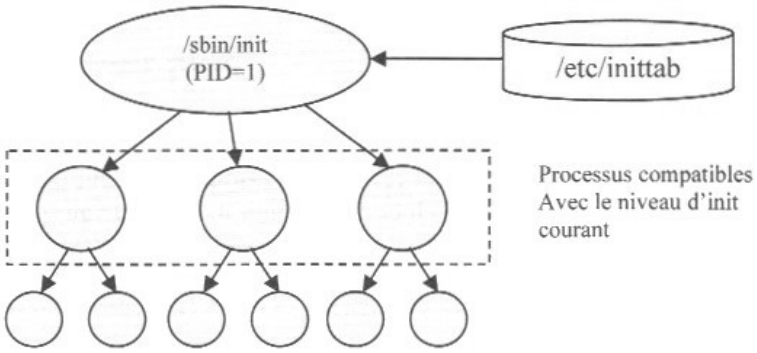


Fig. Le concept de niveau d'init.

Le système Linux offre plusieurs modes d'exploitation appelés « niveau d'init ». A chaque niveau d'init correspond un ensemble d'applications formant un tout cohérent. A l'instant « T », le système se trouve dans un niveau déterminé. L'administrateur système paramètre le niveau du démarrage. Ultérieurement il peut changer de niveau. L'arrêt du système est également associé à un niveau d'init de même que le mode maintenance.

Les différents niveaux

0	Le système va s'arrêter.
S,1	Modes maintenance (ou mono-utilisateur).
2,3,4,5	Modes normaux d'exploitation (ou multi-utilisateur).
6	Redémarrage.

Le savoir concret

Les fichiers

/etc/inittab	Ce fichier décrit les applications activées par `init` en fonction du niveau choisi. Le niveau de démarrage est spécifié également.

Les commandes

`runlevel, who -r`	Indiquent le niveau courant ainsi que le niveau précédent.
telinit, init	Permet à l'administrateur de changer le niveau d'init.

La structure du fichier /etc/inittab

Chaque ligne du fichier /etc/inittab décrit une application activée par init.

Syntaxe d'une ligne

```
étiquette:[niveau...]:action:commande
```

L'étiquette identifie la ligne.

Le champ niveau indique dans quels niveaux la commande est active.

Le champ action est décrit plus loin.

La commande activée par `init` via un shell.

Action

`respawn`	La commande est relancée si elle se termine.
`wait`	Les commandes suivantes sont activées après la fin de la commande.
`once`	La commande est activée à chaque fois que l'on entre dans le niveau.
`boot`	La commande est activée au démarrage. Le niveau est ignoré.
`bootwait`	Idem, mais les commandes suivantes sont activées après la fin de la commande.
`off`	Ne fait rien (présent pour la compatibilité ascendante).
`ondemand`	La commande est activée si l'on va dans un des pseudo-niveaux a, b ou c (le vrai niveau n'est pas modifié).
`initdefault`	Spécifie le niveau au démarrage.
`sysinit`	Activé au démarrage avant les lignes `boot` ou `bootwait`. Le niveau est ignoré.
`powerfail`	Activé quand le système va s'éteindre (l'alimentation secourue a prévenu `init` d'un problème de courant électrique).
`powerwait`	Idem, mais les commandes qui suivent sont bloquées.
`ctraltdel`	`init` a reçu le signal SIGINT. Quelqu'un a tapé Ctrl-Alt-De à la console.

Exemples

a) Un fichier standard.

```
# more  /etc/inittab
id:3:initdefault:
si::sysinit:/etc/rc.d/rc.sysinit
l0:0:wait:/etc/rc.d/rc 0
l1:1:wait:/etc/rc.d/rc 1
l2:2:wait:/etc/rc.d/rc 2
l3:3:wait:/etc/rc.d/rc 3
l4:4:wait:/etc/rc.d/rc 4
l5:5:wait:/etc/rc.d/rc 5
l6:6:wait:/etc/rc.d/rc 6
ca::ctrlaltdel:/sbin/shutdown -t3 -r now
pf::powerfail:/sbin/shutdown -f -h +2 "Power Failure; System
Shutting Down"
pr:12345:powerokwait:/sbin/shutdown -c "Power Restored; Shutdown
Cancelled"
1:2345:respawn:/sbin/mingetty tty1
2:2345:respawn:/sbin/mingetty tty2
3:2345:respawn:/sbin/mingetty tty3
4:2345:respawn:/sbin/mingetty tty4
5:2345:respawn:/sbin/mingetty tty5
6:2345:respawn:/sbin/mingetty tty6
x:5:respawn:/etc/X11/prefdm -nodaemon
```

b) Une ligne décrivant un terminal relié par une liaison série.

```
S0:3:respawn:/sbin/getty -L 9600 ttyS0 vt320
```

Particularités des distributions

RedHat

Les niveaux d'init

2 Mode multi-utilisateur restreint.

3 Mode multi-utilisateur, les applications serveurs NFS sont actives.

4 Mode non-défini, il doit être paramétré par l'administrateur.

5 Mode graphique.

Le fichier /etc/rc.sysinit

Ce script de démarrage, activé par init, initialise les éléments clés du système.

SuSE

Utilise les mêmes niveaux d'init que RedHat.

Debian

Par défaut, on travaille dans le niveau 2 d'init.

Pour aller plus loin

Les pages de Manuel

init(8), telinit(8), inittab(5), runlevel(8), initscript(5)

Le script rc.sysinit

La théorie

Description

Le fichier rc.sysinit est un script de démarrage propre aux distributions RedHat. Il regroupe plusieurs étapes essentielles du démarrage du système. Il est activé par `init` via inittab quelque soit le niveau de démarrage.

Les grandes étapes du script

- Le nom du système est fixé.
- Des opérations associées à SELinux sont accomplies si ce dernier est actif.
- Les FS /proc et /sys sont montés.
- Crée les TTY définies dans /etc/inittab (en écrivant dessus).
- Fixe le niveau des messages du noyau envoyés sur la console maîtresse.
- Crée les entrées de périphériques par le système Udev.
- Initialisation du matériel (chargement des pilotes par modprobe,…).
- Active le RAID logiciel.
- Configure les paramètres du noyau.
- Configure l'horloge.
- Configure le clavier (charge le bon « Keymaps » : fr,…).
- Initialise l'ACPI.
- Provoque ou non la vérification des FS (présence de /fastboot,…).
- Vérifie le FS root (/) en lecture seule et si tout est correct, le remonte en lecture/écriture.
- Active les Quota.
- Initialise le LVM.
- Monte les autres FS.
- Active le swap.

Pour en savoir plus

Les pages de manuels

initlog(8), mount(8), sysctl(8), hwclock(8), loadkeys(1)

Fichiers

/etc/rc.d/rc.sysinit	Le script rc.sysinit.

Internet

Linux rc.sysinit script
http://www.comptechdoc.org/os/linux/startupman/linux_surcsysinit.html

Démarrer avec LILO

La théorie

Le chargeur LILO (LInux LOader) permet le démarrage du noyau Linux.

LILO est installé par la commande `lilo` selon les directives d'un fichier de configuration. Si l'on veut modifier le démarrage on est obligé de réinstaller LILO.

Le savoir concret

Les commandes

lilo	Installe le chargeur LILO.

Les fichiers

/etc/lilo.conf	Les instructions d'installation de LILO
/boot/boot.b	Le chargeur LILO installé par la commande lilo.
/boot/map	Ce fichier mémorise l'emplacement des blocs qui composent le chargeur.
/boot/boot.0300	Sauvegarde de l'ancien MBR pour un disque IDE.
/boot/boot.0800	Idem, pour un disque SCSI.
/etc/lilo.conf.anaconda	Fichier produit par l'installateur RedHat dans le cas où l'on utilise Grub.

Focus : Un fichier lilo.conf

```
# more /etc/lilo.conf
boot=/dev/hda
delay=100
image=/boot/vmlinuz
  root=/dev/hda1
  initrd=/boot/initrd.img
  append="single"
  label=linux
```

Le paramètre `boot` indique à LILO où installer le chargeur. Ici il est installé dans le MBR. Il sert donc de chargeur primaire. Si l'on spécifie une partition, par exemple /dev/hda2, LILO sera installé en tant que chargeur secondaire. Le paramètre `delay` spécifie un délai de 100 dixièmes de secondes avant le démarrage automatique.

Il y a un paragraphe image par noyau ou OS que l'on désire activer. Par défaut, LILO active la première image. A l'écran le choix se fait selon les chaînes présentes dans le paramètre `label`. Le paramètre `root` désigne le FS root. Le paramètre `initrd` désigne un fichier initrd. Le paramètre `append` spécifie les autres paramètres du noyau.

Focus : Installer LILO

```
# lilo -C /etc/lilo.conf
```

Cette commande installe LILO comme chargeur en se basant sur le fichier de configuration donné en paramètre.

```
# lilo
```

Idem, car le fichier /etc/lilo.conf est le fichier par défaut.

Astuce : Enlever LILO du MBR

On veut remettre l'ancien chargeur :

- Sous Linux : `# lilo -U`
- Sous Windows : `C:\> fdisk /MBR`

Pour en savoir plus

Les pages de manuel

lilo(8), lilo.conf(5)

Howto

LILO

Multiboot-with-LILO

Internet

Booter avec LILO
http://www.linux-france.org/article/sys/chargeurs/ix86/lilo/boot-lilo.html

Démarrer avec GRUB

La théorie

Le chargeur GRUB

GRUB (GRand Unified Bootloader) est un chargeur qui permet d'activer un système d'exploitation (Linux, Unix BSD, Windows, Solaris, …).

Les composants du chargeur GRUB

Stage2

Stage2 contient le shell GRUB sous forme de chargeur secondaire, il constitue donc l'essentiel du code du chargeur. Il est installé sous forme d'un fichier dans un FS.

Il interprète les instructions présentes dans le fichier de configuration de GRUB ou saisies au clavier lors du démarrage.

Stage1

Stage1 est le chargeur primaire installé dans le MBR. Son seul rôle est d'activer Stage2.

Stage1_5

Un module Stage1_5 est un pilote de FS. Il permet à l'interpréteur Grub de rechercher un fichier dans la partition active : le fichier de configuration, le noyau Linux, le fichier initrd, … Il existe autant de modules Stage1_5 que de FS.

Comparaison entre LILO et GRUB

A chaque modification de LILO il faut réinstaller LILO, car il est constitué de blocs chaînés. Inversement, comme GRUB est capable de lire des fichiers, on n'a pas besoin de réinstaller GRUB, il suffit de modifier son fichier de configuration.

Il est possible également de démarrer le système en activant un shell GRUB interactif, ce qui impossible avec LILO.

Le savoir Concret

Utiliser l'interface de démarrage GRUB

Quand on démarre un système configuré avec GRUB on voit, après les messages du BIOS, un écran où le mot GRUB apparaît ainsi qu'un décompte en secondes avant le démarrage automatique. On peut l'interrompre en appuyant sur une touche.

On a alors les possibilités suivantes :

- Choisir (avec les flèches) l'OS que l'on veut activer.
- Valider le choix par défaut.
- Activer le shell GRUB interactif (touche 'C').
- Modifier les arguments du noyau courant (touche 'A').
- Modifier les commandes actuellement chargées (touche 'E').

Les principales commandes de l'interpréteur GRUB

`root`	Active la partition de démarrage. Par exemple root (hd0,1) active la deuxième partition (on compte à partir de zéro) du premier disque dur (hd).

`kernel`	Charge en mémoire un noyau (Linux, BSD, ...) à partir d'un fichier. La commande est suivie des arguments du noyau.
`initrd`	Le fichier qui contient l'initrd.
`title`	Débute une section décrivant un OS. Cette commande n'existe que dans un fichier de configuration.
`boot`	Active l'OS actuellement en mémoire. Cette commande est implicite dans un fichier de configuration.
`timeout`	Indique le délai en secondes avant le démarrage automatique. En l'absence de cette commande, on a un démarrage manuel.

IMPORTANT ! Les chemins de fichiers (pour le noyau, l'initrd) sont des chemins absolus par rapport à la partition activée par la commande `root`. Par exemple, si le noyau est à la racine de la partition /boot, cela donne : `kernel /vmlinuz`. Inversement, si le noyau est dans le répertoire /boot qui fait partie de la partition racine, cela donne : `kernel /boot/vmlinuz`.

Les autres commandes

`help`	Liste les commandes.
`rootnoverify`	Active la partition de démarrage dans le cas de Windows.
`makeactive`	Positionne le drapeau « amorce » sur la partition que l'on vient d'activer par `rootnoverify`.
`chainloader`	Charge en mémoire un fichier ou une suite de blocs qui font office de chargeur secondaire. Par exemple pour activer Windows NT, XP et 2000 : `chainloader +1` et pour activer Windows 9X : `chainloader /bootsect.dos`
`default`	Spécifie l'OS à activer par défaut, on compte à partir de zéro. Par défaut c'est le premier OS (`default=0`).
`configfile`	Charge un fichier de configuration.
`pause`	Affiche un message et provoque une pause.
`hiddenmenu`	Cache le menu dans la phase automatique.
`splashimage`	Charge une image en fond d'écran.
`password`	Dans une section décrivant un OS, cette commande provoque la demande d'un mot de passe pour continuer.
`reboot`	Redémarre le système.

Les commandes d'installation de GRUB

`grub`	Active un pseudo shell GRUB. La sous-commande `root` active une partition. La sous-commande `setup` installe GRUB. Par exemple `setup (hd0)` installe GRUB dans le MBR et `setup (hd0,0)` installe GRUB comme chargeur secondaire en tête de la première partition du 1er disque.
`grub-install`	Installe GRUB de manière simplifiée. C'est un script shell qui sert d'interface à la commande `grub`.

Focus : Un exemple de fichier de configuration

```
# more /boot/grub/grub.conf
timeout = 10
title Fedora Core (2.6.15-1.2054_FC5)
        root (hd0,0)
        kernel /vmlinuz-2.6.15-1.2054_FC5 ro root=LABEL=/12 quiet
        initrd /initrd-2.6.15-1.2054_FC5.img
```

Le paramètre `timeout` est global. Ensuite, il y a un paragraphe `title` par OS que l'on désire démarrer.

Les particularités des distributions

RedHat

/boot/grub/grub.conf — Le fichier de configuration de GRUB.

SuSE, Debian

/boot/grub/menu.lst — Le fichier de configuration de GRUB.

Pour en savoir plus

Les pages de manuel

grub(8)

Howto

Multiboot-with-GRUB

Linux+Win9x+Grub-HOWTO

Internet

Le site officiel du logiciel GRUB (téléchargement, documentation, …)
http://www.gnu.org/software/grub/

Démarrer avec Syslinux

La théorie

Syslinux est un package contenant un ensemble de chargeurs pour Linux développés par H.Peter Anvin. Il comprend essentiellement les chargeurs suivants :

- isolinux : Chargeur permettant le démarrage à partir d'un cdrom ISO.
- Pxelinux : Chargeur permettant le démarrage à partir d'un serveur PXE.
- syslinux : Chargeur permettant le démarrage à partir d'une clé USB.

isolinux - Le savoir concret

Démarche globale

1) On crée une arborescence de fichiers destinée à être gravée.
2) On crée un sous répertoire nommé isolinux. On copie à l'intérieur le noyau Linux, l'initrd et le chargeur isolinux.bin. En un mot tout ce qui est nécessaire au boot.
3) On crée le fichier de configuration du chargeur : isolinux/isolinux.cfg.
4) On crée l'image bootable du cdrom et on la grave.

Focus : Créer le fichier de configuration

```
default Distro
prompt 1
timeout 100
label Distro
  kernel vmlinuz
  append initrd=initrd.gz ramdisk_size=5000 rw root=/dev/ram0
label hd
localboot 0x80
```

La syntaxe ressemble beaucoup à Lilo. L'utilisateur a le choix entre deux images : Distro et hd. L'image hd est associée au premier disque dur. La commande `prompt 1` affiche l'invite « boot : ». La commande `timeout 100` provoque le démarrage automatique (de l'image Distro) au bout de 10 secondes.

Focus : Créer l'image bootable du cdrom

```
# mkisofs -o output.iso -b isolinux/isolinux.bin -c \
isolinux/boot.cat -no-emul-boot -boot-load-size 4 -boot-info-table \
root-of-iso-tree
```

Cette commande crée une image ISO bootable. Elle installe le chargeur `isolinux.bin`. L'arborescence dont on crée l'image est donnée en dernier argument (root-of-iso-tree).

Pour en savoir plus

Fichiers

Documentation du package : isolinux.doc, pxelinux.doc, syslinux.doc, …

Internet

SYSLINUX - The Easy-to-use Linux Bootloader
http://syslinux.zytor.com/

Initrd

La théorie

Le concept d'initrd

Si un pilote est nécessaire pour accéder au disque dur, il ne peut être chargé par la commande `insmod` qui nécessite que le noyau accède déjà au disque. Dans ce cas, l'utilisation d'un initrd est astucieuse.

Un fichier initrd contient les pilotes nécessaires au démarrage (EXT3, SCSI, RAID, ...), mais il est chargé par le chargeur ! Il devient temporairement le FS racine (/). Dans ce FS, un script activé automatiquement charge les pilotes et bascule ensuite sur le disque dur à partir duquel le démarrage classique a lieu.

Le démarrage avec initrd

1. Le chargeur (lilo, grub) charge le noyau et l'initrd.
2. Le noyau convertit l'initrd en un disque en mémoire et libère l'espace initiale.
3. L'initrd est monté comme FS racine (/).
4. Le script /linuxrc est exécuté.
5. /linuxrc monte le FS racine définitif (celui du disque dur).
6. Dans /linuxrc, la commande `pivot_root` bascule sur le FS racine du disque dur.
7. Le programme /sbin/init du disque dur est exécuté (celui du nouveau FS racine).
8. Le FS initrd est supprimé.

Remarques :

1) Un noyau 2.6 exécute le programme /sbin/init et non /linuxrc.
2) Si l'on veut créer un live-cd, on ne bascule pas sur le disque dur (étape 6).

Le savoir concret

Les commandes

`mkinitrd`	Crée automatiquement un fichier initrd qui contient les pilotes nécessaires au démarrage.

Les formats d'un initrd

Le format traditionnel	Le fichier initrd est un FS (ext2, ...) dans un fichier compressé.
Le format cpio	Le fichier initrd est une archive CPIO compressée utilisant des entêtes ascii (à partir du noyau 2.6).

Pour en savoir plus

Les pages de manuel

mkinitrd(8), initrd(4)

Fichier

Le fichier initrd.txt de la documentation du noyau.

Créer un live-cd, une live-key

La théorie

Qu'est-ce qu'un live-cd, une live-key ?

Un live-cd est un cd bootable qui se suffit à lui-même. Il installe un FS racine (/) en mémoire et ensuite active une application, au minimum un shell. Si l'on veut accéder aux partitions du disque dur par exemple, il faut les monter.

Une livekey, c'est pareil, c'est un OS autonome (Linux par exemple) stocké dans une clé USB.

La démarche pour créer un live-cd

1. On crée une arborescence de fichiers qui sera destinée à être gravée sur le cdrom.
2. On ajoute dans cette arborescence le chargeur (isolinux) et on le configure.
3. On copie le noyau. On peut en copier plusieurs si on propose un multi-boot.
4. On crée le fichier initrd et on le copie dans l'arborescence. C'est l'étape clé.
5. On copie les fichiers souhaitées dans l'arborescence (logiciels, fichiers de données, archive de sauvegarde, ...). Ces fichiers seront accessibles via un montage du cdrom à partir du live-cd.
6. On crée l'image ISO renfermant l'arborescence. On précise où se trouve le chargeur pour qu'il soit installé en tête de l'image (et donc du cdrom).
7. On grave l'image.

Créer l'initrd d'un live-cd

1. On crée l'arborescence constituant l'initrd. C'est-à-dire une arborescence racine (/bin, /sbin, /dev, /etc/, ...).

Remarque : Si l'on utilise l'ancien format de `initrd`, il faut créer cette arborescence dans un système de fichiers stocké dans un fichier.

2. On peuple l'arborescence avec les commandes, les bibliothèques nécessaires à notre objectif.
3. On installe les fichiers périphériques ou bien on met en place le système Udev.
4. On configure les fichiers et les scripts de démarrage (/etc/fstab, /etc/inittab ou /linuxrc, /etc/sysinit, ...).
5. On fabrique le fichier initrd.

Remarque : Si l'on utilise l'ancien format de `initrd`, il faut démonter le FS avant de le compresser.

Le savoir concret

Astuces de réalisation

- Utiliser des exécutables statiques, c'est-à-dire qui ne font pas référence à des bibliothèques dynamiques, par exemple le shell ash.static.
- Utiliser l'exécutable BusyBox qui remplace la plupart des principales commandes (ls, cp, rm, ...).

- Modifier un système live-cd au lieu de partir de zéro.

Astuces de mise au point

- On peut utiliser la commande `ldd` pour connaître les bibliothèques utilisées par un exécutable.
- L'utilisation de la commande `chroot` permet de vérifier l'indépendance d'une arborescence car celle-ci devient la nouvelle racine. On peut ainsi détecter l'absence de commandes ou de bibliothèques.
- Tester un initrd en l'activant à partir du disque dur.
- Utiliser des cd réinscriptibles au lieu de graver définitivement chaque essai.

Créer une clé USB bootable

Globalement, la création d'une clé USB bootable est similaire à la création d'un live-CD.

Les principales différences sont les suivantes :

- On formate la clé en FAT (12, 16 ou 32).
- On utilise le chargeur syslinux au lieu d'isolinux (la configuration est identique).
- Il faut installer un chargeur primaire qui active syslinux (le chargeur secondaire).
- Il est conseillé d'installer les fichiers de démarrage directement sous la racine.

Pour en savoir plus

Howto

Linux Bootdisk HOWTO

From-PowerUp-To-Bash-Prompt-HOWTO

Internet

Making Bootable Linux CDs
http://www.geocities.com/potato.geo/bootlinuxcd.html

Linux Live CD/USB scripts
http://www.linux-live.org/

Knoppix Remastering Howto
http://www.knoppix.net/wiki/Knoppix_Remastering_Howto

How to create a live boot CD containing your favorite Linux Distro!
http://www.babytux.org/articles/howto/how2livecd.php

BusyBox
http://www.busybox.net

Linux From Scratch
http://lfs.traduc.org/
http://www.linuxfromscratch.org/

ibuild – logiciel pour créer un Live-CD
http://ibuild.livecd.net/

Create a Custom Live Linux CD - Leveraging BusyBox and OpenSSH
http://souptonuts.sourceforge.net/cdrom.htm

Internet – Quelques live-cd

Liste des live-cd de FrozenTech
http://www.frozentech.com/content/livecd.php

Knoppix
http://www.knopper.net/knoppix/

Internet – Quelques livekey

Your Pocket OS: MCNLive On A USB Flashdrive
http://www.mcnlive.org/usblive.htm

Damn Small Linux (DSL)
http://www.damnsmalllinux.org/

Linux on a stick
http://www.dcs.napier.ac.uk/~05012779/dsl.html

Slax
http://www.slax.org/

Livres

Knoppix à 200%, par Kyle Rankin, cher O'Reilly. Le chapitre 9 explique comment créer une version Knoppix personnalisée.

ATELIERS



Tâche 1 : Dépanner avec un CD bootable 10mn
Tâche 2 : Init 5mn
Tâche 3 : Démarrer avec LILO 10mn
Tâche 4 : Démarrer avec GRUB 5mn
Tâche 5 : Initrd 10mn
Tâche 6 : Créer un cdrom bootable 40mn
Tâche 7 : Créer une clé USB bootable 10mn
Tâche 8 : Installer sur la clé l'image de maintenance 5mn
Tâche 9 : Tester la commande busybox 5mn

Tâche 1 : Dépanner avec un CD bootable

1. Anticiper les problèmes.

a) On sauvegarde la configuration.

```
[root@linux1 ~]# tar czf /root/etc.tar.gz /etc
tar: Removing leading `/' from member names
```

b) On sauvegarde le MBR (on peut également le sauvegarder sur un support annexe).

```
[root@linux1 ~]# dd if=/dev/hda of=/root/MBR count=1 bs=512
1+0 records in
1+0 records out
```

c) On imprime l'architecture du système (ou on la copie sur un support annexe).

```
[root@linux1 ~]# fdisk -l /dev/hda | lp
[root@linux1 ~]# lp  /etc/fstab
[root@linux1 ~]# lp /boot/grub/grub.conf
[root@linux1 ~]#
```

d) On fait une sauvegarde Bare-Metal (avec Mondo par exemple).

2. Dépanner un système.

Exemple de problème : perte du mot de passe de root.

a) On démarre sur un CD bootable (Knoppix, le cd d'installation,…).

On met le CD et CTRL-ALT-SUPPR

- Sous Knoppix V3.3 (http://www.knoppix.de/) :

```
boot :  knoppix  2  lang=fr
```

- Sous Redhat/Fedora :

```
boot:   linux  rescue
```

b) Eventuellement, on configure la carte réseau si l'on ne dispose pas du DHCP.

```
# ifconfig eth0 192.168.0.1
# ping  -c3 192.168.0.200
# route add default gw 192.168.0.254
# echo  "nameserver    192.168.0.200"    >  /etc/resolv.conf
```

c) On fait la réparation.

```
# fdisk -l /dev/hda
# mkdir /mnt/hda2   # inutile sous Knoppix
# mount  /dev/hda2 /mnt/hda2   #  le FS root (« / »)
# cd /mnt/hda2
# ls
# cd etc
# cp  shadow  shadow.bak
# vi shadow   #  on supprime le mot de passe de root
root::13468:0:99999:7:::...
```

Remarque : Selon les distributions il faut plutôt supprimer le champ mot de passe du fichier /etc/passwd.

```
# cd
# umount /mnt/hda2
# reboot  # ou Ctrl-Alt-Suppr ou exit, ATTENTION! Retirer le cd-rom bootable !
```

d) Après le redémarrage on remet un mot de passe pour root.

```
login: root
# echo secret | passwd --stdin root
```

Tâche 2 : Init

1. Sauvegarder inittab et le modifier.

On supprime la possiblité de redémarrer le système avec Ctrl+Alt+Suppr.

On permet à une personne ayant accès physiquement à la console maitresse de travailler sous le compte de root.

```
[root@linux1 ~]# cp /etc/inittab /etc/inittab.$(date '+%m%d%H%M')
[root@linux1 ~]# vi /etc/inittab
...
# Trap CTRL-ALT-DELETE
#ca::ctrlaltdel:/sbin/shutdown -t3 -r now
...
#2:2345:respawn:/sbin/mingetty tty2
2:2345:respawn:/bin/bash < /dev/tty2 > /dev/tty2 2>&1
...
```

2. Tester.

a) On redémarre le système.

```
[root@linux1 ~]# reboot
```

b) On bascule sur le terminale 2.

```
Ctrl-Alt-F2
bash-3.00# id
uid=0(root) gid=0(root)
```

c) On essaye de rebooter le système par [Ctrl]+[Alt]+[Suppr].

```
Ctrl-Alt-Suppr
```

Il ne se passe rien !

Tâche 3 : Démarrer avec LILO

1. Afficher le fichier de configuration de LILO généré par Anaconda.

```
[root@linux1 ~]# cat /etc/lilo.conf.anaconda
prompt
timeout=20
default=linux
boot=/dev/hda
map=/boot/map
install=/boot/boot.b
message=/boot/message
linear

image=/boot/vmlinuz-2.6.9-42.EL
        label=linux
        initrd=/boot/initrd-2.6.9-42.EL.img
        read-only
        append="rhgb quiet root=LABEL=/"
[root@linux1 ~]#
```

2. Installer LILO (détruire grub!).

```
[root@linux1 ~]# rpm -q lilo
lilo-21.4.4-26.1
[root@linux1 ~]# lilo -C /etc/lilo.conf.anaconda
Added linux *
[root@linux1 ~]#
```

3. Tester : essayer de redémarrer le système.

```
[root@linux1 ~]# reboot
```

4. Réinstaller grub. Quel est le chargeur installé ?

```
[root@linux1 ~]# dd if=/dev/hda of=/tmp/MBR bs=512 count=1
1+0 enregistrements lus.
1+0 enregistrements Ã©crits.
[root@linux1 ~]# strings /tmp/MBR
|lbaLILO
0<:r
VVWQ
D[YSRWQ
[root@linux1 ~]# grub-install /dev/hda
Installation finished. No error reported.
This is the contents of the device map /boot/grub/device.map.
Check if this is correct or not. If any of the lines is incorrect,
fix it and re-run the script `grub-install'.

# this device map was generated by anaconda
(fd0)     /dev/fd0
(hd0)     /dev/hda
[root@linux1 ~]# dd if=/dev/hda of=/tmp/MBR bs=512 count=1
1+0 enregistrements lus.
1+0 enregistrements Ã©crits.
[root@linux1 ~]# strings /tmp/MBR
lbaLILO
ZRrI
D|f1
```

```
GRUB
Geom
Hard Disk
Read
 Error
[root@linux1 ~]#
```

Tâche 4 : Démarrer avec grub

1. Modifier le fichier de configuration de grub.

On ajoute une entrée pour démarrer en mode maintenance.

On supprime la temporisation : on restera bloqué sur l'écran de démarrage.

On démarre par défaut sur le noyau 2.6.17.

On supprime l'argument quiet du noyau.

```
[root@linux1 ~]# cd /boot/grub
[root@linux1 grub]# cp grub.conf grub.conf.$(date '+%m%d%H%M')
[root@linux1 grub]# vi grub.conf
default=1
#timeout=5
title CentOS-4 i386 (2.6.9-42.EL)
        root (hd0,0)
        kernel /vmlinuz-2.6.9-42.EL ro root=LABEL=/ rhgb quiet
        initrd /initrd-2.6.9-42.EL.img
title New Kernel (2.6.17)
        root (hd0,0)
        kernel /linux-2.6.17 ro root=LABEL=/ rhgb
        initrd /initrd-2.6.17.img
title CentOS-4 i386 single
        root (hd0,0)
        kernel /vmlinuz-2.6.9-42.EL ro root=LABEL=/ rhgb single
        initrd /initrd-2.6.9-42.EL.img
[root@linux1 grub]# cd
[root@linux1 ~]#
```

2. Tester.

```
[root@linux1 ~]# reboot
```

a) On démarre en mode maintenance.

b) On démarre avec le choix par défaut.

3. Remettre la configuration précédente.

```
[root@linux1 ~]# cd /boot/grub/
[root@linux1 grub]# ls grub.conf.*
grub.conf.000  grub.conf.02252133
[root@linux1 grub]# cp grub.conf.02252133 grub.conf
cp: Ã©craser `grub.conf'?y
[root@linux1 grub]# cd -
[root@linux1 ~]#
```

Tâche 5 :
Initrd

1. Copier l'initrd courant dans un fichier et déterminer son type.

```
[root@linux1 ~]# cp /boot/initrd-2.6.17.img .
[root@linux1 ~]# file initrd-2.6.17.img
initrd-2.6.17.img: gzip compressed data, from Unix, max compression
[root@linux1 ~]# gunzip < initrd-2.6.17.img > initrd.unzip
[root@linux1 ~]# file initrd.unzip
initrd.unzip: ASCII cpio archive (SVR4 with no CRC)
```

2. Lister son contenu.

```
[root@linux1 ~]# cpio -itcv < initrd.unzip
drwx------  10 root     root            0 Feb  6 18:19 .
lrwxrwxrwx   1 root     root            3 Feb  6 18:19 sbin -> bin
drwxr-xr-x   2 root     root            0 Feb  6 18:19 sys
-rwxr-xr-x   1 root     root          780 Feb  6 18:19 init
drwxr-xr-x   2 root     root            0 Feb  6 18:19 sysroot
drwxr-xr-x   2 root     root            0 Feb  6 18:19 bin
lrwxrwxrwx   1 root     root           10 Feb  6 18:19 bin/modprobe ->
/sbin/nash
-rwxr-xr-x   1 root     root        12904 Feb  6 18:19 bin/insmod
lrwxrwxrwx   1 root     root           10 Feb  6 18:19 bin/hotplug -> /sbin/nash
lrwxrwxrwx   1 root     root            4 Feb  6 18:19 bin/udevstart -> udev
-rwxr-xr-x   1 root     root       542500 Feb  6 18:19 bin/udev
-rwxr-xr-x   1 root     root        37960 Feb  6 18:19 bin/nash
drwxr-xr-x   3 root     root            0 Feb  6 18:19 etc
drwxr-xr-x   2 root     root            0 Feb  6 18:19 etc/udev
-rw-r--r--   1 root     root         1128 Feb  6 18:19 etc/udev/udev.conf
drwxr-xr-x   2 root     root            0 Feb  6 18:19 lib
-rw-r--r--   1 root     root        67784 Feb  6 18:19 lib/jbd.ko
-rw-r--r--   1 root     root       140448 Feb  6 18:19 lib/ext3.ko
-rw-r--r--   1 root     root        64308 Feb  6 18:19 lib/dm-mod.ko
drwxr-xr-x   2 root     root            0 Feb  6 18:19 dev
crw-r--r--   1 root     root       4,   0 Feb  6 18:19 dev/systty
crw-r--r--   1 root     root       4,   3 Feb  6 18:19 dev/tty3
crw-r--r--   1 root     root       1,   3 Feb  6 18:19 dev/null
crw-r--r--   1 root     root       4,   1 Feb  6 18:19 dev/tty1
brw-r--r--   1 root     root       1,   1 Feb  6 18:19 dev/ram
crw-r--r--   1 root     root       4,   4 Feb  6 18:19 dev/tty4
crw-r--r--   1 root     root       5,   1 Feb  6 18:19 dev/console
crw-r--r--   1 root     root       4,   2 Feb  6 18:19 dev/tty2
drwxr-xr-x   2 root     root            0 Feb  6 18:19 proc
drwxr-xr-x   2 root     root            0 Feb  6 18:19 loopfs
1703 blocks
[root@linux1 ~]#
```

Tâche 6 :
Créer un cdrom bootable

1. Installer et configurer syslinux.

a) On vérifie la présence des logiciels syslinux, mkisofs et cdrecord (on les installe si besoin).

```
# rpm -q syslinux  mkisofs  cdrecord
```

b) On crée le répertoire qui contiendra l'arborescence à installer sur le CD.

```
# mkdir ~/cd
```

c) On crée l'arborescence isolinux et on la peuple (on y met isolinux, le noyau, …).

```
# cd ~/cd ;  mkdir isolinux
# cp /usr/lib/syslinux/isolinux.bin  isolinux/
# mount /media/cdrom      #  on monte le 1er cd d'installation
# cp /media/cdrom/isolinux/vmlinuz  isolinux/linux
```

d) On crée le fichier de configuration de isolinux.

```
# cd isolinux/
# vi isolinux.cfg
default linux
timeout 600
prompt 1
F1 f1.txt
label linux
        kernel linux
        append initrd=initrd.img ramdisk_size=32768 root=/dev/ram0

# echo "Noyau standard, 2.6.11-1" > f1.txt
```

2. Créer le fichier initrd (étape la plus longue et la plus complexe).

a) On crée une arborescence de fichiers dans un système de fichiers virtuels.

```
# dd if=/dev/zero of=/root/fichier  bs=1k count=32768
# mkfs  /root/fichier
# mount -o loop /root/fichier /mnt/disk
# cd /mnt/disk
# mkdir  bin  dev  etc  lib  proc  sys  usr  var
# ln -s bin sbin
```

b) On ajoute des commandes, des bibliothèques et des fichiers périphériques.

```
# cd /bin
# cp bash sh cp mv ln rm ls df cat pwd chmod chgrp chown loadkeys /mnt/disk/bin
# cp mkdir rmdir grep gzip ed vi dd tar ping mount umount ps mknod /mnt/disk/bin
# cd /sbin
# cp ifconfig restore dump e2fsck nash fdisk mke2fs lsmod mingetty /mnt/disk/bin
# cp insmod  /mnt/disk/bin
# cd /usr/bin
# cp less id find /mnt/disk/bin
# cd /lib
# cp libc.so.6 ld-linux.so.2 libtermcap.so.2 libdl.so.2 libacl.so.1 libattr.so.1
/mnt/disk/lib
# cp libresolv.so.2  libpcre.so.0 libext2fs.so.2  libuuid.so.1  libe2p.so.2
/mnt/disk/lib
# cp librt.so.1 libselinux.so.1 libpthread.so.0  /mnt/disk/lib
# cp libcom_err.so.2 libproc-3.2.5.so libsepol.so.1  /mnt/disk/lib
# mkdir -p  /mnt/disk/usr/lib
# cp /usr/lib/libncurses.so.5 /mnt/disk/usr/lib
```

c) On installe UDEV.

```
# for i in console null zero; do /sbin/MAKEDEV -d /mnt/disk/dev -x $i; done
# cp /sbin/udevd /mnt/disk/sbin/
# cp /sbin/udev.static /mnt/disk/sbin/udev
# cp /sbin/udevstart.static /mnt/disk/sbin/udevstart
```

```
# cp -r /etc/udev /mnt/disk/etc
# cp -r /dev/.udevdb/ /mnt/disk/dev
```

d) On configure le démarrage.

```
# cp /sbin/init  /mnt/disk/bin
# cd /mnt/disk
# vi etc/fstab
/dev/ram0     /         ext2         defaults         1   1
none             /dev/pts devpts       gid=5,mode=620   0   0
none             /proc    proc         defaults         0   0
none             /sys     sysfs        defaults         0   0

# vi etc/inittab
in:2:initdefault:
si::sysinit:/etc/sysinit
# Run gettys in standard runlevels
1:2345:respawn:/bin/bash < /dev/console > /dev/console 2>&1
2:2345:respawn:/bin/bash < /dev/tty2 > /dev/tty2 2>&1

# vi  etc/sysinit
#!/bin/sh
/bin/insmod /lib/jbd.ko
/bin/insmod /lib/ext3.ko
/bin/mount -n -t proc proc /proc
/bin/mount -n -t sysfs sysfs /sys
/bin/mount -n -t tmpfs tmpfs /dev
mkdir /dev/pts
mkdir /dev/shm
/sbin/udevstart
/sbin/udevd &
/bin/mount -t devpts devpts /dev/pts
/bin/e2fsck -f /dev/ram0
/bin/mount -n -o remount,rw /
rm -f /etc/mntab*
cat /proc/mounts > /etc/mtab
/bin/mount  -a  -t noproc,nosysfs,notmpfs,nodevpts
echo "" > /dev/tty2
echo "" > /dev/tty3
echo "" > /dev/tty4
/bin/loadkeys /etc/fr.map.gz

# chmod u+x  etc/sysinit
# cp  /lib/kbd/keymaps/i386/azerty/fr.map.gz  etc
# cp  /lib/modules/2.6.11-1.1369_FC4/kernel/fs/jbd/jbd.ko  lib
# cp  /lib/modules/2.6.11-1.1369_FC4/kernel/fs/ext3/ext3.ko  lib
```

e) On compresse l'image et on l'installe dans l'arborescence du CD.

```
# cd ;  umount /mnt/disk
# gzip -9 < /root/fichier  > /root/cd/isolinux/initrd.img
```

3. Tester votre initrd

a) On copie l'initrd dans /boot et on crée une nouvelle entrée à grub.conf pour un noyau utilisant cet initrd.

```
# cp /root/cd/isolinux/initrd.img /boot
```

```
# vi /boot/grub/grub.conf
...
title RedHat INITRD essai
        root (hd0,0)
        kernel /vmlinuz-2.6.9-42.EL ramdisk_size=32768 root=/dev/ram0
        initrd /boot/initrd.img
```

b) On redémarre à partir de cette nouvelle entrée et on teste notre image.

4. Créer et graver l'image de l'arborescence.

a) On crée l'image ISO.

```
# cd /root/cd
# mkisofs -R -b isolinux/isolinux.bin -c isolinux/boot.cat -no-emul-boot -boot-
load-size 4 -boot-info-table /root/cd > /tmp/linux.iso
# ls -lh /tmp/linux.iso
```

b) On grave l'image ISO (si on possède un graveur).

```
# cdrecord -scanbus       # ou drecord -scanbus dev=ATAPI
...
        0,0,0     0) 'IDE-CD  ' 'R/RW 4x4x32     ' '1.3B' Removable CD-ROM
...
# cd /tmp
# cdrecord -v speed=4 dev=0,0,0 linux.iso  # ou cdrecord dev=ATAPI:0,1,0 ...
# eject cdrom
```

c) On teste : on démarre à partir du cdrom bootable ...

Tâche 7 : Créer une clé USB bootable

1. Télécharger l'image ISO de la distribution (ici une slax-popcorn).

```
# wget 'ftp://ftp.linux.cz/pub/linux/slax/SLAX-5.x/special-editions/slax-
popcorn-*.iso'
```

2. Monter l'image.

```
# mount -o loop  slax*.iso  /mnt/disk
```

3. Partitionner la clé (elle doit avoir une partition).

Dans la suite, on suppose que la clé se nomme /dev/sda. On crée une partition : /dev/sda1.

```
# dmesg
# fdisk  /dev/sda
Command (m for help): o
Command (m for help): n
p
Partition number (1-4): 1
First cylinder (1-1020, default 1):
Last cylinder or +size or +sizeM or +sizeK (1-1020, default 1020):
Using default value 1020
Command (m for help): t
Hex code (type L to list codes): 6
Command (m for help): w
```

4. Formater la partition que l'on vient de créer en FAT-16.

```
# mkdosfs -F 16 /dev/sda1
```

5. Copier les fichiers provenant de l'image ISO.

```
# cd /mnt/disk
# ./make_disk.sh   /dev/sda1
```

```
# cd
```

6. Installer le chargeur primaire.

On peut utiliser le chargeur mbr.bin provenant du logiciel officiel syslinux.

```
# wget 'ftp://rpmfind.net/linux/dag/redhat/el4/en/i386/dag/RPMS/syslinux-3.35-
1.el4.rf.i386.rpm'
# rpm2cpio syslinux-3.35-1.el4.rf.i386.rpm > syslinux.cpio
# cpio -iv mbr.bin < syslinux.cpio
# dd if=mbr.bin  of=/dev/sda
```

7. Configurer le chargeur secondaire (le noyau doit être à la racine et s'appeler linux).

```
# mkdir /mnt/cle
# mount  /dev/sda1  /mnt/cle
# cd /mnt/cle/boot
# mv  vmlinuz  ../linux
# mv  initrd.gz  ..
# cd ..
# mv  isolinux.cfg  syslinux.cfg
# vi   syslinux.cfg
display boot/splash.cfg
default slax
prompt 1
timeout 40
F1 boot/splash.txt
F2 boot/splash2.txt
F3 boot/splash.cfg

label slax
kernel linux
append vga=769 changes=slaxsave.dat max_loop=255 initrd=initrd.gz init=linuxrc
load_ramdisk=1 prompt_ramdisk=0 ramdisk_size=4444 root=/dev/ram0 rw

label linux
kernel linux
append vga=769 changes=slaxsave.dat max_loop=255 initrd=initrd.gz init=linuxrc
load_ramdisk=1 prompt_ramdisk=0 ramdisk_size=4444 root=/dev/ram0 rw

label memtest
kernel boot/memtest
```

Remarque : On modifie les chemins (on supprime /boot) et le nom du noyau (linux).

```
# cd
# umount /mnt/cle
```

8. Installer le chargeur secondaire.

```
# syslinux  /dev/sda1
```

9. Tester sur une machine dont le BIOS accepte de démarrer avec une clé.

Tâche 8 : Installer sur la clé l'image de maintenance

1. Récupérer l'image de maintenance présente sur le 1er cdrom de la distribution.

```
[root@linux1 ~]# mount /media/cdrecorder/
mount: block device /dev/hdc is write-protected, mounting read-only
[root@linux1 ~]# more /media/cdrecorder/images/README
This directory contains image files that can be used to create media
```

```
capable of starting the CentOS-4 i386 installation process.

The boot.iso file is an ISO 9660 image of a bootable CD-ROM.  It is useful
in cases where the CD-ROM installation method is not desired, but the
CD-ROM's boot speed would be an advantage.

To use this image file, burn the file onto CD-R (or CD-RW) media as you
normally would.

The diskboot.img file is a VFAT filesystem image that can be written to a
USB pendrive or other bootable media larger than a floppy.  Note that
booting via USB is dependent on your BIOS supporting this.  It should
be written to the device using dd.
[root@linux1 ~]# cp /media/cdrecorder/images/diskboot.img .
```

2. Sauvegarder l'image de sa clé.

```
[root@linux1 ~]# dd if=/dev/sda of=/usr/cle.save
1961695+0 records in
1961695+0 records out
```

3. Installer l'image de maintenance sur la clé.

```
[root@linux1 ~]# dd if=/root/diskboot.img of=/dev/sda
12288+0 records in
12288+0 records out
[root@linux1 ~]#
```

4. Tester (démarrer à partir de la clé).

Tâche 9 : Tester la commande busybox

```
[root@linux1 ~]# rpm -q busybox
busybox-1.00.rc1-5
[root@linux1 ~]# whereis busybox
busybox: /sbin/busybox /usr/share/man/man1/busybox.1.gz
[root@linux1 ~]# ldd /sbin/busybox
        not a dynamic executable
[root@linux1 ~]# whereis ash
ash: /bin/ash.static /bin/ash /usr/share/man/man1/ash.1.gz
[root@linux1 ~]# ldd /bin/ash.static
        not a dynamic executable
[root@linux1 ~]# mkdir -p /tmp/chroot/bin
[root@linux1 ~]# cp /sbin/busybox /tmp/chroot/bin
[root@linux1 ~]# cp /bin/ash.static /tmp/chroot/bin/sh
[root@linux1 ~]# chroot /tmp/chroot/ bin/sh
# ls
ls: not found
# /bin/busybox ls
bin
# /bin/busybox mkdir rep
# /bin/busybox ls -l
drwxr-xr-x    2 0        0            4096 Mar 17 09:22 bin
drwxr-xr-x    2 0        0            4096 Mar 17 09:25 rep
# exit
[root@linux1 ~]#
```

- *RAM, SWAP, CPU, IO*
- *vmstat, sar*
- */proc/**
- */var/log/sa/sa12*
- *ulimit –a, chattr +A*

11

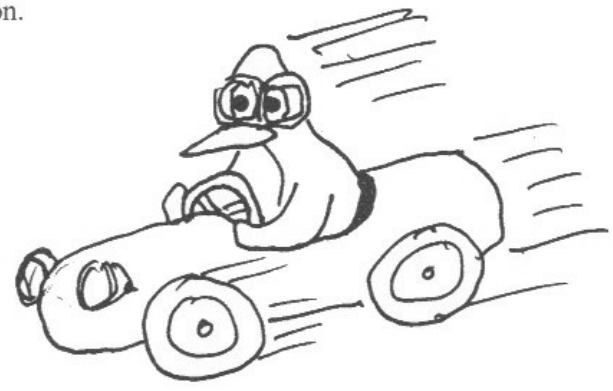
La gestion des performances

Objectifs

Ce chapitre traite de la gestion des performances. Globalement, l'administrateur va surveiller l'usage des ressources et agir en conséquence. Le chapitre présente les principales commandes de surveillance, comme vmstat ou sar. On essaye également de présenter différentes stratégies pour améliorer les performances.

Contenu

Améliorer les performances, introduction.
Surveiller le système
La commande vmstat
La commande sar
La comptabilité
Augmenter les ressources
Diminuer l'usage des ressources
Ateliers



Améliorer les performances, introduction

La théorie

Avoir un système performant

L'administrateur doit veiller à ce que son système soit le plus performant possible ou tout au moins qu'il s'exécute avec les performances à priori escomptées.

Méthodologie

Pour obtenir un système performant, l'administrateur peut suivre plusieurs stratégies :

Mettre en place la configuration recommandée

Il faut suivre les recommandations de l'éditeur du système d'exploitation et des éditeurs des logiciels utilisées. Par exemple, le serveur WebSphere v5 d'IBM nécessite un minimum de 256 Mo de mémoire. Mais il est recommandé 512Mo de mémoire pour obtenir de bonnes performances. Le système RedHat Enterprise 4 nécessite au minimum 256 Mo de mémoire pour s'installer (même si ensuite il peut fonctionner avec uniquement 128 Mo).

Réagir à une baisse de performances

Dés la mise en exploitation, on met en place la surveillance automatique de l'usage des ressources, par exemple, l'usage du CPU. Dés que l'on constate une baisse des performances, on essaye de la mettre en relation avec les mesures. Les paramètres dont les valeurs ont changées sont à prendre en compte en priorité. On établit un diagnostic qui établit quelles sont la ou les ressources devenues insuffisantes. Ensuite, on établit un plan d'action consistant à augmenter les ressources insuffisantes ou à diminuer l'usage des ces ressources. Enfin, on vérifie si les mesures prises ont eu l'incidence escomptée.

En résumé :

- Surveiller l'usage des ressources.
- Diagnostiquer l'origine des goulots d'étranglements (bottle-neck)
- Agir : augmenter les ressources ou diminuer leur usage.

Paramétrer finement le système a priori

Avant la mise en exploitation, on essaye de construire des benchmarks représentant le plus fidèlement possible le comportement de l'activité de l'application. Ensuite, on paramètre le système et l'application pour améliorer les performances. Evidemment, on ne modifie qu'un seul paramètre à la fois.

Remarques :

1) Les stratégies présentées ne sont pas du tout exclusives, ont peut les suivre toutes les trois.

2) La recherche des raisons d'une baisse de performance est assimilable à du dépannage (Troubleshooting).

Les principales ressources du système

Les principales ressources d'un serveur se résument à :

- La puissance de calcul : puissance (vitesse, taille des registres) et nombre de processeurs.
- La mémoire vive (RAM) et disque (SWAP).
- Les entrées/sorties (I/O) disques et réseaux.

Ce sont essentiellement ces éléments qu'il faut surveiller et adapter en cas de problème.

Pour en savoir plus

Internet

Red Hat Enterprise Linux - Performance Tuning Guide
http://people.redhat.com/mbehm/RHEL_Tuning_Guide.pdf

Oracle9iR2 on Linux: Performance, Reliability and Manageability
http://www.redhat.com/whitepapers/rhel/OracleonLinux.pdf

Tuning SuSE LINUX Enterprise Server on IBM Eserver xSeries Servers
(paramétrer l'OS, Apache, Oracle, DB2, LDAP, Samba, Lotus Domino)
http://www.redbooks.ibm.com/redpapers/pdfs/redp3862.pdf

Livres

Optimizing Linux Performance, par Esolt, Philip G aux éditions Prentice Hall

Permformance Tuning for Linux Servers, par Johnson S. & all, aux éditions IBM Press.

Système Performance Tuning, par Mike Loukides & all, aux éditions O'Reilly

Surveiller le système

Le savoir concret

Les commandes de surveillance généralistes

`vmstat`	Affiche des statistiques sur l'utilisation des ressources systèmes (CPU, mémoire et entrées/sorties).
`procinfo`	Affiche des statistiques sur l'utilisation des ressources systèmes et les IRQ.
`sar`	Commande générale de surveillance des ressources systèmes. Possède de nombreuses options. La récolte des données est facilement programmable via un crontab.

Les autres commandes de surveillance

`uptime`	Affiche la charge globale du système.
`ps`	Liste les processus actifs. L'argument `aux` visualise l'utilisation des ressources mémoire et CPU par processus.
`free`	Affiche l'utilisation de la mémoire
`top`	Affiche les processus les plus consommateurs de ressources.
`w`	Affiche les processus associés aux utilisateurs connectés.
`iostat`	Affiche l'utilisation du CPU et des disques et partitions.
`mpstat`	Affiche des statistiques en liaison avec les processeurs.
`netstat`	Affiche des statistiques sur l'utilisation des cartes réseaux.
`time`	Active une application et ensuite affiche des statistiques sur l'utilisation du CPU. Cette commande existe en deux versions : interne et externe (GNU).
`opcontrol`	Configure le système OProfile de contrôle des performances qui utilise le matériel de contrôle du processeur.

Commande en mode graphique

Gnome-system-monitor	Outil de surveillance du système en mode graphique.

Les fichiers

/proc/*	Origine des informations présentées par les commandes précédentes.
/proc/meminfo	Contient les données associées à l'utilisation de la mémoire.

Focus : La commande uptime

```
# uptime
 14:34:33 up 12 days, 16:01,  2 users,  load average: 0.52, 0.15, 0.05
```

La commande `uptime` affiche l'heure, le nombre de jours depuis le dernier redémarrage, le nombre d'utilisateurs connectés et la charge système dans la dernière minute, les cinq dernières minutes et le dernier quart d'heure. Ainsi, on voit du premier coup d'œil si la charge est en train d'augmenter ou de diminuer.

Les particularités des différentes distributions

RedHat

La commande `sar` est recommandée par RedHat pour surveiller le système.

La commande `procinfo` n'est pas disponible.

Pour en savoir plus

Les pages de manuel

vmstat(8), procinfo(8), sar(1), uptime(1), ps(1), free(1), top(1), w(1), iostat(1), mpstat(1), netstat(8), time(1), opcontrol(1)

Internet

Guide RedHat : Contrôle des ressources, Puissance de traitement et largeur de bande
http://www.redhat.com/docs/manuals/enterprise/RHEL-4-Manual/fr/admin-guide/ch-resource.html

http://www.redhat.com/docs/manuals/enterprise/RHEL-4-Manual/fr/admin-guide/ch-bandwidth.html

OProfile
http://www.redhat.com/docs/manuals/enterprise/RHEL-4-Manual/fr/admin-guide/s1-resource-rhlspec.html

System Tuning Info for Linux Servers (dommage, un peu ancien)
http://people.redhat.com/alikins/system_tuning.html

La commande vmstat

La théorie

`vmstat` est la commande native pour visualiser l'usage des ressources (CPU, mémoire et entrée/sortie) d'un système Linux.

Le savoir concret

Syntaxe

```
vmstat [option...] [délai][n]
```

La commande peut afficher `n` mesures à `délai` secondes d'intervalle.

Options

-a	Affiche la mémoire active/inactive.
-f	Affiche le nombre de création de processus (`fork()` depuis le démarrage.
-m	Affiche les données slabinfo.
-s	Affiche différentes valeurs liées à la mémoire, le CPU, …

Remarque : les objets du noyau utilisés fréquemment (tampons, inodes, …) ont leur propre cache. Les statistiques s'y référant sont regroupées sous le vocable slabinfo.

Fichiers

/proc/meminfo, /proc/stat, /proc/*/stat, /proc/slabinfo
Les données de base d'où la commande `vmstat` tire ses informations.

Focus : un exemple commenté

```
# vmstat  5 3
procs -----------memory---------- ---swap-- -----io---- --system-- ----cpu----
 r  b   swpd   free   buff  cache   si   so    bi    bo   in    cs us sy id wa
 2  0      0  74244  58668 240596    0    0     1     1   34    25  0  0 100  0
 1  0      0  74244  58668 240596    0    0     0    12 1042   114 100  0  0  0
 2  0      0  74244  58668 240596    0    0     0     8 1062   160 100  0  0  0
```

On mesure l'activité trois fois à cinq secondes d'intervalle.

Processus

r	Le nombre de processus en attente d'un processeur libre.
b	Le nombre de processus en attente de fin d'entrées/sorties.

Mémoire

swpd	La quantité de mémoire virtuelle utilisée.
free	La quantité de mémoire vive libre.
buff	La quantité de mémoire utilisée comme tampon (cache).

Swap

si	La quantité de mémoire lue à partir du disque (page-in).
so	La quantité de mémoire écrite en mémoire de swap (page-out).

Entrées/sorties (IO)

bi	Le nombre de blocs lus par seconde.
bo	Le nombre de blocs écrits par seconde.

Système

in	Le nombre d'interruption par seconde.
cs	Le nombre de commutation de processus par seconde.

CPU

us	Le pourcentage de temps CPU consommé en mode utilisateur (incluant le `nice time`).
sy	Le pourcentage de temps CPU consommé en mode noyau.
id	Le pourcentage de temps CPU non utilisé (idle).
wd	Le pourcentage de temps CPU non utilisé, mais des processus sont en attente de fin d'entrées/sorites.

Pour en savoir plus

Les pages de manuel

vmstat(8)

La commande sar

La théorie

La commande `sar` offre une vision fine de l'usage des ressources. Elle est prévue nativement pour récolter de manière automatique des données grâce à `crontab`.

Le savoir concret

Les commandes

`sar`	La commande qui visualise les ressources systèmes. Soit on l'utilise en direct, soit on analyse des données déjà récoltées.
`sadc`	Cette commande enregistre l'activité du système dans un fichier binaire ayant le quantième du jour comme suffixe.
`sa1`	Cette commande est une interface à la commande `sadc`. Elle est typiquement activée par un crontab.
`sa2`	Cette commande, non obligatoire, provoque un rapport journalier. Elle est typiquement activée par un crontab.

Les principales options de sar

-u	Utilisation du CPU (option par défaut).
-q	Nombre de processus en attente d'un processeur libre.
-r	Mémoire vive et swap.
-R	Mémoire (compléments).
-W	Swapping.
-b	Statistiques sur les entrées/sorties.
-d	Activité de chaque périphérique bloc.
-r	Les tables du noyau (inodes, …)
-x arg	Etat d'un processus (arg peut être un PID).
-X arg	Comme précédent, mais prend en compte les enfants.
-n DEV	Statistiques associées aux cartes réseaux.
-n EDEV	Statistiques associées aux erreurs réseaux.
-n SOCK	Statistiques associées aux sockets réseaux.
-w	Nombre de permutation de contextes par seconde.
-y	Activité des terminaux.

Remarques :

1) Avec l'option `-P numero_cpu`, on obtient des statistiques pour un CPU. Avec l'option `-P ALL`, on obtient les statistiques pour chaque CPU et globalement.

2) Par défaut, on liste les valeurs de la veille.

3) Avec l'option `-f fichier`, on explore les données d'un fichier particulier. Par exemple, -f /var/log/sa/sa12 accède aux données du 12ième jour du mois.

Focus : La charge CPU (sar –u)

```
# sar -u 10 3
Linux 2.6.9-42.EL (linux1.pinguins)     02/04/07

12:21:33          CPU     %user     %nice   %system   %iowait     %idle
12:21:43          all    100.00      0.00      0.00      0.00      0.00
12:21:53          all     76.45      0.00      0.20      0.00     23.35
12:22:03          all      0.00      0.00      0.00      0.00    100.00
Average:          all     58.83      0.00      0.07      0.00     41.10
```

`CPU`	Le numéro du CPU.
`%user`	Le pourcentage de temps passé en mode utilisateur.
`%nice`	Le pourcentage de temps passé pour exécuter des processus à faible priorité (nice).
`%system`	Le pourcentage de temps passé en mode noyau.
`%iowait`	Le pourcentage de temps pendant lequel le processeur est inoccupé. En parallèle, des processus sont bloqués en attente de fin d'entrées/sorties.
`%idle`	Le pourcentage de temps pendant lequel le processeur ne fait rien.

Remarques :

1) Une valeur `%user` élevée indique que le système fait globalement beaucoup de calculs. Inversement, une valeur `%system` importante indique que le système fait beaucoup d'entrées/sorties.

2) Une valeur `%iowait` élevée montre que le système des entrées/sorties est sous dimensionné par rapport à la puissance de calcul.

3) Une charge système importante n'est pas forcément dramatique dans la mesure où les processus n'attendent pas. Le paramètre `runq-sz` donnée par la commande `sar -q` indique justement le nombre de processus en attente d'un processeur libre.

Pour en savoir plus

Les pages de manuel

sar(8), sa1(8), sa2(8), sadc(8)

La comptabilité

La théorie

Description

Le paquetage de comptabilité (accounting) est composé de commandes qui établissent des statistiques concernant l'usage des ressources CPU par utilisateur.

Ce paquetage est très ancien, il a été crée sur les premiers systèmes Unix Berkeley (BSD). L'objectif à l'époque, était d'établir des factures adressées aux utilisateurs en fonction de leur temps de connexion et de leur utilisation du CPU.

Les enregistrements comptables sont générés par le noyau lors de l'exécution de chaque commande. En conséquence, la comptabilité consomme beaucoup de ressources systèmes. On peut l'activer à des fins de sécurité ou pour mesurer l'utilisation des ressources par utilisateur. Dans ce dernier cas, il est conseillé de l'utiliser par sondage. C'est-à-dire, épisodiquement et non pas en permanence.

Le savoir concret

Les commandes

`psacct`	Le script de démarrage (RC) qui gère le service. Il active la commande `accton`.
`accton`	Cette commande active ou arrête la comptabilité. Sans argument, la commande arrête la comptabilité. Avec un nom de fichier en argument elle l'active. Ce fichier contiendra les événements comptables. Ce fichier doit exister.
`login`	Enregistre les connexions.
`ac`	Affiche les temps de connexions.
`sa`	Affiche les totaux d'utilisation du CPU (par commande, par utilisateur).
`lastcomm`	Affiche en clair le contenu du fichier comptable (ce dernier est en binaire).

Les particularités des différentes distributions

RedHat

Les commandes comptables appartiennent au paquetage psacct.

SuSE, Debian

Les commandes comptables appartiennent au paquetage acct.

Pour en savoir plus

Les pages de manuel

acct(5), ac(8), login(1), init(8), wtmp(5), sa(8), lastcomm(1), last(1)

Howto

Process-Accounting HOWTO

Augmenter les ressources

La théorie

Augmenter les ressources matérielles

CPU

Il est possible de changer de CPU et d'en prendre une plus rapide (3Ghz au lieu 1.7), ou d'utiliser des processeurs gérant des registres plus grand (64 bits au lieu de 32 bits) ou plus nombreux (un système bicore au lieu d'un monoprocesseur).

Mémoire vive

L'augmentation de la mémoire vive est sans doute l'élément qui a le plus d'impact sur l'amélioration des performances.

Remarques :

1) On va être limité par les possibilités de la carte mère pour l'augmentation de la mémoire. Généralement seuls les PC serveur haut de gamme supportent des quantités importantes de mémoire.

2) Par défaut, les noyaux Linux ne gèrent qu'un Go de mémoire. Pour gérer des tailles mémoire supérieures il faut des noyaux ad hoc (kernel-hugemem).

Les entrées/sorties

Voici les principaux éléments qui influencent positivement les entrées/sorties :

- Utiliser des canaux d'entrées sorties puissants (64 bits, 66Mhz, ...)
- Utiliser des disques RAID.
- Avoir des cartes réseaux puissantes en plusieurs exemplaires et en équilibrage de charge (load balancing).

Augmenter les ressources logicielles

Le paramétrage de l'OS

Les paramètres du noyau

Le noyau est optimisé pour un usage standard. Si l'on veut paramétrer finement son système, il est possible de jouer sur certains paramètres.

Mémoire de swap

Même si un système Linux peut fonctionner sans mémoire de swap, on peut utiliser entre une et trois fois la quantité de mémoire vive comme mémoire de swap. Si cette mémoire de swap est installée et/ou répartie sur plusieurs disques rapides on améliore d'autant les performances. Il faut éviter d'utiliser des fichiers de swap.

Les paramètres de configuration des structures disques

Le choix de la taille des PE (physical extend) lors de la création d'un LV ou le choix de la taille des blocs lors de la création d'un FS sont des éléments qui influent beaucoup sur les performances. Mais il ne faut pas oublier que d'utiliser des unités d'allocation importante à pour conséquence une perte d'espace disque.

Les paramètres de l'application

Un logiciel comme Apache possèdent beaucoup de paramètres qui ont une incidence sur les performances. Par exemple MaxClients, MinSpareServers, ...

Conclusion

Les éléments clés de la performance

L'expérience nous apprend qu'en surdimensionnement la mémoire vive et en utilisant des technologies RAID on résout globalement les problèmes de performances.

Il faut des machines de type serveur

Le choix d'un PC de type serveur a pour conséquence l'usage d'un ordinateur possédant d'emblée des ressources importantes. De plus il offre une redondance de ses ressources. En pratique, on choisira des xSeries d'IBM, des PowerEdge de DELL ou des Proliant de HP. On peut également faire confiance à des assembleurs.

En final, ne pas oublier le CPU

Sauf dans le cas où le système est utilisé pour des applications de calcul, le nombre ou la puissance des CPU n'est pas d'emblée le facteur critique. Par contre, à toute chose égale, quand tous les autres paramètres sont optimaux, la puissance de calcul fait la différence.

Le savoir concret

Les commandes

`hdparm` Paramètre les performances des disques IDE (et SATA si on utilise des pilotes IDE).

Quelques paramètres du noyau

`file-max` Le nombre maximal de fichiers ouverts.

`inode-max` Le nombre maximal d'inodes ouvertes.

`dirty_background_ratio`
Spécifie à quel pourcentage de la mémoire, le démon `pdflush` commence à écrire les pages modifiées (dirty_data) en mémoire de swap.

`dirty_writeback_centisecs`
Périodicité du réveil du démon `pdflush` en 1/100 de secondes.

`dirty_expire_centisecs`
Age (en 1/100 de secondes) à partir duquel une page est vieille.

`page-cluster` Nombre de pages écrites en mémoire de swap en une écriture.

Pour en savoir plus

Les pages de manuel

hdparm(8)

Paquetage

filesystems/proc.txt Ce fichier de la documentation du noyau, décrit l'arborescence /proc et les paramètres modifiables à chaud du noyau.

Internet

Exemple de paramétrage de la commande hdparm.
http://www.laukas.com/knbase/linux_special/

Diminuer l'usage des ressources

La théorie

Les applications

- Supprimer ou désactiver les services inutiles. Si vous utilisez votre système en tant que serveur Web, est-il nécessaire de garder ou d'activer Cups ?
- Faire fonctionner les procédures de maintenance (sauvegarde, purge de /tmp, ...) à des heures creuses.
- Limiter les ressources maximales utilisées par une application ou un utilisateur. Par exemple le nombre de processus ou la taille mémoire maximale utilisée.
- Ne pas hésiter à activer à faible priorité les tâches de calcul.

Les aspects disques

- Mettre en place une politique des quotas disques ou enfermer un utilisateur ou une application dans un système de fichier.
- Mettre dans des FS différents les arborescences qui sont modifiées très fréquemment et celles qui sont stables.
- Dans le cas d'application lisant principalement des fichiers (Web), désactiver la mise à jour de la date de dernier accès.

Divers

Désactiver l'interface graphique. Cette dernière, indispensable sur une station peut utiliser de manière abusive les ressources d'un serveur.

Le savoir concret

Les commandes

`chattr +A`	Désactive la mise à jour des dates de derniers accès. On peut également configurer ce comportement au montage du FS.
`ulimit`	Cette commande interne du shell, limite les ressources des applications futures.
`nice`, `renice`	Diminuent la priorité (valeur NICE) d'un processus et de ses descendants.
PAM	Cette technologie, principalement ayant pour objectif la sécurité, peut être utilisée pour limiter les ressources des utilisateurs (comme `ulimit`).

Particularités des différentes distributions

RedHat

Le mode serveur non graphique est associé au niveau d'init 3 configuré dans la ligne `initdefault` du fichier /etc/inittab. Par défaut, on travaille dans le mode graphique (niveau 5).

ATELIERS



Tâche 1 : Surveiller le système 10mn
Tâche 2 : Surveiller une commande avec time 10mn
Tâche 3 : Surveiller le système avec la commande sar 10mn
Tâche 4 : La comptabilité 10mn
Tâche 5 : Surveiller le système en pénurie de ressources 5mn
Tâche 6 : Limiter l'usage des ressources 10mn
Tâche 7 : Modifier les paramètres du noyau 5mn
Tâche 8 : Simuler une charge avec un benchmark, l'influence de la mémoire 15mn

Tâche 1 : Surveiller le système

1. Activer les principales commandes de surveillance natives et observer les types d'informations recueillies.

```
[root@linux1 ~]# uptime
 12:46:29 up 13 days, 14:13,  2 users,  load average: 0.00, 0.00, 0.00
[root@linux1 ~]# free
             total       used       free     shared    buffers     cached
Mem:        449908     444876       5032          0      62376     309356
-/+ buffers/cache:      73144     376764
Swap:       522072          0     522072
[root@linux1 ~]# vmstat
procs -----------memory---------- ---swap-- -----io---- --system-- ----cpu----
 r  b   swpd   free   buff  cache   si   so    bi    bo   in    cs us sy id wa
 0  0      0   5032  62380 309356    0    0     1     1   27    33  0  0 100  0
[root@linux1 ~]# vmstat 5 3
procs -----------memory---------- ---swap-- -----io---- --system-- ----cpu----
 r  b   swpd   free   buff  cache   si   so    bi    bo   in    cs us sy id wa
 0  0      0   5032  62384 309356    0    0     1     1   27    33  0  0 100  0
 0  0      0   5032  62384 309356    0    0     0     6 1042   120  0  0 100  0
 0  0      0   5032  62384 309356    0    0     0     0 1061   167  0  0 100  0
[root@linux1 ~]# w
 12:46:56 up 13 days, 14:13,  2 users,  load average: 0.00, 0.00, 0.00
USER     TTY      FROM              LOGIN@   IDLE   JCPU   PCPU WHAT
root     tty1     -                22Jan07 12days  0.01s  0.01s -bash
root     pts/1    raichu.pokemon   Tue15    0.00s  0.27s  0.00s w
[root@linux1 ~]# ps aux | head
USER       PID %CPU %MEM    VSZ   RSS TTY      STAT START   TIME COMMAND
root         1  0.0  0.1   3084   552 ?        S    Jan21   0:00 init [5]
root         2  0.0  0.0      0     0 ?        SN   Jan21   0:14 [ksoftirqd/0]
root         3  0.0  0.0      0     0 ?        S<   Jan21   0:00 [events/0]
root         4  0.0  0.0      0     0 ?        S<   Jan21   0:00 [khelper]
root         5  0.0  0.0      0     0 ?        S<   Jan21   0:00 [kacpid]
root        29  0.0  0.0      0     0 ?        S<   Jan21   0:00 [kblockd/0]
root        50  0.0  0.0      0     0 ?        S<   Jan21   0:00 [aio/0]
root        30  0.0  0.0      0     0 ?        S    Jan21   0:00 [khubd]
root        49  0.0  0.0      0     0 ?        S    Jan21   0:00 [kswapd0]
[root@linux1 ~]# netstat -i
```

```
Kernel Interface table
Iface       MTU Met    RX-OK RX-ERR RX-DRP RX-OVR    TX-OK TX-ERR TX-DRP TX-OVR
Flg
eth0       1500   0   222799      0      0      0    18335      0      0      0
BMRU
eth0:1     1500   0      - no statistics available -
BMRU
lo        16436   0     3179      0      0      0     3179      0      0      0
LRU
[root@linux1 ~]# top
top - 12:50:59 up 13 days, 14:17,  2 users,  load average: 0.51, 0.12, 0.04
Tasks:  54 total,   2 running,  52 sleeping,   0 stopped,   0 zombie
Cpu(s):  0.0% us,  0.0% sy,  0.0% ni, 100.0% id,  0.0% wa,  0.0% hi,  0.0% si
Mem:    449908k total,   445132k used,     4776k free,    62392k buffers
Swap:   522072k total,        0k used,   522072k free,   309496k cached

  PID USER      PR  NI  VIRT  RES  SHR S %CPU %MEM    TIME+  COMMAND
    1 root      16   0  3084  552  472 S  0.0  0.1   0:00.66 init
    2 root      34  19     0    0    0 S  0.0  0.0   0:14.28 ksoftirqd/0
    3 root       5 -10     0    0    0 S  0.0  0.0   0:00.00 events/0
    4 root      11 -10     0    0    0 S  0.0  0.0   0:00.01 khelper
    5 root      15 -10     0    0    0 S  0.0  0.0   0:00.00 kacpid
   29 root       5 -10     0    0    0 S  0.0  0.0   0:00.00 kblockd/0
   50 root      12 -10     0    0    0 S  0.0  0.0   0:00.00 aio/0
   30 root      15   0     0    0    0 S  0.0  0.0   0:00.06 khubd
   49 root      15   0     0    0    0 S  0.0  0.0   0:00.19 kswapd0
  196 root      22   0     0    0    0 S  0.0  0.0   0:00.00 kseriod
  318 root      15   0     0    0    0 S  0.0  0.0   0:00.22 kjournald
 1473 root      10 -10  2176  468  380 S  0.0  0.1   0:00.01 udevd
 2115 root      20   0     0    0    0 S  0.0  0.0   0:00.00 scsi_eh_0
 2116 root      15   0     0    0    0 S  0.0  0.0   1:18.07 usb-storage
 2609 root       6 -10     0    0    0 S  0.0  0.0   0:00.00 kauditd
 2650 root       6 -10     0    0    0 S  0.0  0.0   0:00.00 kmirrord
 2681 root      15   0     0    0    0 S  0.0  0.0   0:00.00 kjournald
```

Remarque : La commande `top` affiche la charge système, l'utilisation de la mémoire et les processus les plus consommateurs de ressources (%CPU, %MEM). La commande est rafraîchie périodiquement. On arrête la commande par Ctrl-C.

```
[root@linux1 ~]# top -b -d 10 -n 4 >> fic
[root@linux1 ~]# head fic
top - 13:03:10 up 13 days, 14:29,  2 users,  load average: 0.02, 0.37, 0.32
Tasks:  54 total,   1 running,  53 sleeping,   0 stopped,   0 zombie
Cpu(s):  0.1% us,  0.0% sy,  0.0% ni, 99.8% id,  0.0% wa,  0.0% hi,  0.0% si
Mem:    449908k total,   445260k used,     4648k free,    62464k buffers
Swap:   522072k total,        0k used,   522072k free,   309520k cached

  PID USER      PR  NI  VIRT  RES  SHR S %CPU %MEM    TIME+  COMMAND
    1 root      16   0  3084  552  472 S  0.0  0.1   0:00.66 init
    2 root      34  19     0    0    0 S  0.0  0.0   0:14.28 ksoftirqd/0
    3 root       5 -10     0    0    0 S  0.0  0.0   0:00.00 events/0
[root@linux1 ~]#
```

Remarque : On vient d'utiliser top en mode « batch ». On a fait 4 mesures à 10 secondes d'intervalle.

```
[root@linux1 ~]# watch free
Every 2.0s: free                                Sun Feb  4 13:04:50 2007
```

```
             total       used       free     shared    buffers     cached
Mem:        449908     445452       4456          0      62472     309584
-/+ buffers/cache:      73396     376512
Swap:       522072          0     522072
```

Remarque : On arrête la commande par Ctrl-C.

2. On utilise les commandes du package systat (sauf sar).

```
[root@linux1 ~]# rpm -q sysstat
sysstat-5.0.5-11.rhel4
[root@linux1 ~]# rpm -ql sysstat | grep bin
/usr/bin/iostat
/usr/bin/mpstat
/usr/bin/sar
[root@linux1 ~]# yum provides iostat
...
sysstat.i386                               5.0.5-11.rhel4         installed
...
[root@linux1 ~]# mpstat
Linux 2.6.9-42.EL (linux1.pinguins)     02/04/07

13:53:28     CPU   %user   %nice %system %iowait    %irq   %soft   %idle
intr/s
13:53:28     all    0.07    0.00    0.04    0.04    0.02    0.00   99.83
1050.55
[root@linux1 ~]# iostat
Linux 2.6.9-42.EL (linux1.pinguins)     02/04/07

avg-cpu:  %user   %nice    %sys %iowait   %idle
           0.07    0.00    0.05    0.04   99.83

Device:            tps   Blk_read/s   Blk_wrtn/s   Blk_read   Blk_wrtn
hda               0.12         1.52         1.74    1787853    2048820
hda1              0.00         0.00         0.00       1370         30
hda2              0.26         1.49         1.71    1755778    2015048
hda3              0.00         0.00         0.00       1411         76
hda5              0.00         0.00         0.00       1074          0
hda10             0.01         0.02         0.03      20970      33650
hdc               0.00         0.01         0.00      12636          0
```

3. Installer et utiliser procinfo.

```
[root@linux1 ~]# wget 'http://downloads.sourceforge.net/mcj/procinfo-18.tar.bz2'
[root@linux1 ~]# tar xvjf procinfo-18.tar.bz2
procinfo-18/
procinfo-18/lsdev.8
procinfo-18/procinfo.8
procinfo-18/procinfo.c
procinfo-18/procinfo.h
procinfo-18/routines.c
procinfo-18/socklist.8
procinfo-18/lsdev.pl
procinfo-18/socklist.pl
procinfo-18/procinfo.lsm
procinfo-18/README
```

```
procinfo-18/CHANGES
procinfo-18/Makefile
[root@linux1 ~]# cd procinfo-18
[root@linux1 procinfo-18]# ls
CHANGES   README   lsdev.pl    procinfo.c  procinfo.lsm  socklist.8
Makefile  lsdev.8  procinfo.8  procinfo.h  routines.c    socklist.pl
[root@linux1 procinfo-18]# make
gcc -Wall -Wstrict-prototypes -O2   -c -o procinfo.o procinfo.c
procinfo.c:19: warning: 'rcsid' defined but not used
gcc -Wall -Wstrict-prototypes -O2   -c -o routines.o routines.c
routines.c:19: warning: 'rcsid' defined but not used
gcc -Wall -Wstrict-prototypes -s  procinfo.o routines.o  -ltermcap -o procinfo
[root@linux1 procinfo-18]# make install
mkdir -p /usr/bin
install procinfo /usr/bin/procinfo
install lsdev.pl /usr/bin/lsdev
install socklist.pl /usr/bin/socklist
mkdir -p /usr/man/man8
install -m 644  procinfo.8 /usr/man/man8/procinfo.8
install -m 644  lsdev.8 /usr/man/man8/lsdev.8
install -m 644  socklist.8 /usr/man/man8/socklist.8
[root@linux1 procinfo-18]# procinfo
Linux 2.6.9-42.EL (buildcentos@build-i386) (gcc 3.4.6 20060404 ) #1 1CPU
[linux1]

Memory:      Total        Used        Free      Shared     Buffers
Mem:        449908      371636       78272           0       55740
Swap:       522072           0      522072

Bootup: Sun Jan 21 22:33:28 2007    Load average: 0.00 0.00 0.00 1/54 15738

user  :       0:08:47.63   0.0%  page in :         0
nice  :       0:00:00.32   0.0%  page out:         0
system:       0:06:00.92   0.0%  swap in :         0
idle  :   2d 14:40:16.99  99.9%  swap out:         0
uptime:   2d 15:04:59.20         context :154378682

irq  0:1086255246 timer                 irq 12:        85 i8042
irq  1:       145 i8042                 irq 14:    115924 ide0
irq  4:         6                       irq 15:   9773947 ide1
irq  6:         2                       irq169:    214956 eth0
irq  7:         4                       irq185:  44604067 ehci_hcd, uhci_hcd,
irq  8:         1 rtc                   irq193:         0 VIA8233
irq  9:         0 acpi

[root@linux1 procinfo-18]# cd
```

4. Créer un script de surveillance activé par un crontab.

```
[root@linux1 ~]# vi surveille.sh
[root@linux1 ~]# cat surveille.sh
#!/bin/sh
uptime
free
ps aux |sort -rk +3 | head -10
```

```
echo "============================================================"
[root@linux1 ~]# chmod u+x surveille.sh
[root@linux1 ~]# crontab -l > /tmp/new_cron
no crontab for root
[root@linux1 ~]# echo "* * * * * /root/surveille.sh > /var/log/perf.log 2>&1" >>
/tmp/new_cron
[root@linux1 ~]# crontab /tmp/new_cron
[root@linux1 ~]# head /var/log/perf.log
 14:06:01 up 13 days, 15:32,  2 users,  load average: 0.00, 0.00, 0.00
             total       used       free     shared    buffers     cached
Mem:        449908     403816      46092          0      61948     268180
-/+ buffers/cache:      73688     376220
Swap:       522072          0     522072
USER       PID %CPU %MEM   VSZ  RSS TTY      STAT START   TIME COMMAND
gdm       4498  0.0  2.3 21264 10348 ?        Ss   Jan21   0:00
/usr/bin/gdmgreeter
root      4155  0.0  1.9 11944 8600 ?         S    Jan21   0:00 /usr/X11R6/bin/X
:0 -audit 0 -auth /var/gdm/:0.Xauth -nolisten tcp vt7
root      3428  0.0  1.7 11364 7952 ?         Ss   Jan21   7:55 hald
root      4133  0.0  0.5 12892 2432 ?         S    Jan21   0:00 /usr/bin/gdm-
binary -nodaemon
```

Tâche 2 :
Surveiller une commande avec time

1. Créer un programme consommateur de CPU (calcul de la constante mathématique e).

```
[root@linux1 ~]# vi e.py
# calcul de e
import sys
un_ch ="1"
for i in range(int(sys.argv[1])):
        un_ch = un_ch + "0"

un = int(un_ch)
e = un + un
n = 1
factorielle = 1

while 1:
        n = n + 1
        factorielle = factorielle * n
        e_old = e
        e = e + un/ factorielle
        if e == e_old : break

print "essai n : ", n, " e = ", e
```

2. Utiliser la commande time interne.

```
[root@linux1 ~]# time python e.py 10
essai n :  14  e =  27182818277

real    0m0.015s
user    0m0.011s
sys     0m0.005s
```

Remarques :

1) Le script de calcul de e demande en argument le nombre de décimal voulu.

2) Le temps réel est le temps pour l'être humain. Le temps CPU consommé est user+sys.

3. Utiliser la commande time externe (GNU).

```
[root@linux1 ~]# /usr/bin/time python e.py 100
essai n :  70  e =
271828182845904523536028747135266249775724709369995957496696762772407663035354759457138217852516642 38
0.00user 0.00system 0:00.01elapsed 86%CPU (0avgtext+0avgdata 0maxresident)k
0inputs+0outputs (0major+601minor)pagefaults 0swaps
```

4. Idem, mais en affichant plus d'informations.

```
[root@linux1 ~]# /usr/bin/time -v python e.py 100
essai n :  70  e =
271828182845904523536028747135266249775724709369995957496696762772407663035354759457138217852516642 38
        Command being timed: "python e.py 100"
        User time (seconds): 0.01
        System time (seconds): 0.00
        Percent of CPU this job got: 80%
        Elapsed (wall clock) time (h:mm:ss or m:ss): 0:00.01
        Average shared text size (kbytes): 0
        Average unshared data size (kbytes): 0
        Average stack size (kbytes): 0
        Average total size (kbytes): 0
        Maximum resident set size (kbytes): 0
        Average resident set size (kbytes): 0
        Major (requiring I/O) page faults: 0
        Minor (reclaiming a frame) page faults: 581
        Voluntary context switches: 1
        Involuntary context switches: 3
        Swaps: 0
        File system inputs: 0
        File system outputs: 0
        Socket messages sent: 0
        Socket messages received: 0
        Signals delivered: 0
        Page size (bytes): 4096
        Exit status: 0
[root@linux1 ~]#
```

Tâche 3 : Surveiller le système avec sar

1. Utiliser en direct la commande sar.

Dans les exemples suivants, on fait trois mesures à cinq secondes d'intervalle.

```
[root@linux1 ~]# sar -u 5 3
Linux 2.6.9-42.EL (linux1.pinguins)     02/04/07

14:30:03          CPU     %user     %nice   %system   %iowait     %idle
14:30:08          all      0.00      0.00      0.00      0.00    100.00
14:30:13          all      0.00      0.00      0.00      0.00    100.00
14:30:18          all      0.00      0.00      0.00      0.00    100.00
Average:          all      0.00      0.00      0.00      0.00    100.00
```

Remarques :

1) La somme `%user+%nice+%system` donne le pourcentage d'utilisation des processeurs.

2) La somme `%iowait+%idle` donne le pourcentage d'inactivité des processeurs.

3) Le rapport `%user/%system` indique si le système fait globalement des calculs ou des IO.

```
[root@linux1 ~]# sar -q 5 3
Linux 2.6.9-42.EL (linux1.pinguins)     02/04/07

14:30:30      runq-sz  plist-sz   ldavg-1   ldavg-5  ldavg-15
14:30:35            0        55      0.00      0.00      0.00
14:30:40            0        55      0.00      0.00      0.00
14:30:45            0        55      0.00      0.00      0.00
Average:            0        55      0.00      0.00      0.00
```

Remarque : `plist-sz` indique le nombre total de processus. `runq-sz` Indique le nombre de processus en attente d'un processeur.

```
[root@linux1 ~]# sar -r 5 3
Linux 2.6.9-42.EL (linux1.pinguins)     02/04/07

14:31:49    kbmemfree kbmemused  %memused kbbuffers  kbcached kbswpfree
kbswpused  %swpused  kbswpcad
14:31:54        46476    403432     89.67     61980    268232    522072
0      0.00         0
14:31:59        46476    403432     89.67     61980    268232    522072
0      0.00         0
14:32:04        46220    403688     89.73     61980    268232    522072
0      0.00         0
Average:        46391    403517     89.69     61980    268232    522072
0      0.00         0
[root@linux1 ~]# sar -b 5 3
Linux 2.6.9-42.EL (linux1.pinguins)     02/04/07

14:32:30          tps      rtps      wtps   bread/s   bwrtn/s
14:32:35         0.00      0.00      0.00      0.00      0.00
14:32:40         0.00      0.00      0.00      0.00      0.00
14:32:45         0.00      0.00      0.00      0.00      0.00
Average:         0.00      0.00      0.00      0.00      0.00
```

Remarque : `tps` indique le nombre de transfert entre la mémoire et les disques physiques. `rtps` indique le nombre de lectures et `wtps` le nombre d'écriture. `bread/s` indique le nombre de blocs de 512 octets lus par seconde et `wrtn/s` le nombre de blocs écrits par seconde.

```
[root@linux1 ~]# sar -B 5 3
Linux 2.6.9-42.EL (linux1.pinguins)     02/04/07

15:00:36     pgpgin/s pgpgout/s   fault/s  majflt/s
15:00:41         0.00      7.19      9.98      0.00
15:00:46         0.00      0.00      6.20      0.00
15:00:51         0.00      0.00      2.00      0.00
Average:         0.00      2.40      6.07      0.00
```

Remarque : `pgpgin/s` et `pgpgout/s` indiquent les transferts en Ko entre la mémoire et le disque. `fault/s` indique le nombre de défaut de page par seconde. `majflt/s` indique le nombre de défauts de page qui entraînent une lecture disque.

```
[root@linux1 ~]# sar -R 5 3
Linux 2.6.9-42.EL (linux1.pinguins)     02/04/07

15:08:07      frmpg/s   bufpg/s   campg/s
15:08:12         0.00      0.00      0.00
15:08:17         0.00      0.00      0.00
15:08:22         0.00      0.00      0.00
Average:         0.00      0.00      0.00
```

Remarque : `frmpg/s` indique le nombre de pages libérées par seconde. Une valeur négative que le système alloue des pages. `bugfpg/s` indique l'augmentation d'allocation de pages en tant que tampon disque (buffer). Une valeur négative montre une diminution de ces tampons disques.

```
[root@linux1 ~]# sar -W 5 3
Linux 2.6.9-42.EL (linux1.pinguins)     02/04/07

15:08:49     pswpin/s pswpout/s
15:08:54         0.00      0.00
15:08:59         0.00      0.00
15:09:04         0.00      0.00
Average:         0.00      0.00
```

Remarque : Nombre de page-in par seconde (`pswin/s`) et de page-out par seconde (`pswpout/s`).

```
[root@linux1 ~]# ps -e |grep cupsd
16810 ?        00:00:00 cupsd
[root@linux1 ~]# sar -x 16810 5 3
Linux 2.6.9-42.EL (linux1.pinguins)     02/04/07

15:19:54          PID  minflt/s  majflt/s     %user   %system   nswap/s   CPU
15:19:59        16810      0.00      0.00      0.00      0.00      0.00     0
15:20:04        16810      0.00      0.00      0.00      0.00      0.00     0
15:20:09        16810      0.00      0.00      0.00      0.00      0.00     0
Average:        16810      0.00      0.00      0.00      0.00      0.00
[root@linux1 ~]# sar -v 5 3
Linux 2.6.9-42.EL (linux1.pinguins)     02/04/07

15:20:44    dentunusd   file-sz  inode-sz  super-sz %super-sz  dquot-sz %dquot-
sz  rtsig-sz %rtsig-sz
15:20:49        26708       780     26625         0      0.00         0
0.00         0      0.00
15:20:54        26708       780     26625         0      0.00         0
0.00         0      0.00
15:20:59        26708       780     26625         0      0.00         0
0.00         0      0.00
Average:        26708       780     26625         0      0.00         0
0.00         0      0.00
[root@linux1 ~]# sar -c 5 3
Linux 2.6.9-42.EL (linux1.pinguins)     02/04/07

15:22:28       proc/s
15:22:33         0.00
15:22:38         0.00
15:22:43         0.00
Average:         0.00
```

Remarque : `proc/s` indique le nombre de processus créés par seconde.

```
[root@linux1 ~]# sar -n DEV 5 3
Linux 2.6.9-42.EL (linux1.pinguins)     02/04/07
...
Average:        IFACE   rxpck/s   txpck/s   rxbyt/s   txbyt/s   rxcmp/s
txcmp/s  rxmcst/s
Average:           lo      0.00      0.00      0.00      0.00      0.00
0.00      0.00
Average:         eth0      0.60      0.33     60.57     75.12      0.00
0.00      0.00
Average:         sit0      0.00      0.00      0.00      0.00      0.00
0.00      0.00
```

2. Collecter les données de sar.

```
[root@linux1 ~]# crontab -e
crontab: installing new crontab
[root@linux1 ~]# crontab -l
*/20    17-20   *       *       *       /usr/lib/sa/sa1 300 4 &
5       23      *       *       *       /usr/lib/sa2  -A  &
[root@linux1 ~]#
```

Remarque : En argument de `sa1` on indique un intervalle et un nombre de mesures. Dans l'exemple `sa1` est activé toute les 20 minutes. A chaque fois il fait 4 mesures à 300 secondes (5 minutes) d'intervalle, ce qui donne un total de 20 minutes.

3. Visualiser les données récoltées (il faut attendre au moins 5 minutes).

```
[root@linux1 ~]# sar
Linux 2.6.9-42.EL (linux1.pinguins)     02/04/07

00:00:01          CPU     %user     %nice   %system   %iowait     %idle
00:10:01          all      0.03      0.00      0.03      0.00     99.94
00:20:01          all      0.03      0.00      0.03      0.00     99.94
```

Remarque : Par défaut, `sar` utilise l'option `-u` et visualise les données récoltés dans la journée.

```
[root@linux1 ~]# ls /var/log/sa/
sa04
[root@linux1 ~]# sar -u -f /var/log/sa/sa04
Linux 2.6.9-42.EL (linux1.pinguins)     02/04/07

00:00:01          CPU     %user     %nice   %system   %iowait     %idle
00:10:01          all      0.03      0.00      0.03      0.00     99.94
00:20:01          all      0.03      0.00      0.03      0.00     99.94
```

Remarque : Avec l'option `-f`, on peut voir les données récoltés un jour quelconque. Dans l'exemple, les données du 4ième jour du mois.

Tâche 4 : La comptabilité

1. Est-ce que le paquetage de comptabilité est installé ?

```
[root@linux1 ~]# rpm -q psacct
psacct-6.3.2-38.rhel4
```

2. Quel est le fichier comptable ?

```
[root@linux1 ~]# more /etc/init.d/psacct
...
# The location of the accounting file
ACCTFILE=/var/account/pacct
```

```
start() {
        [ ! -r $ACCTFILE ] && touch $ACCTFILE && chmod 600 $ACCTFILE
        action $"Starting process accounting: " /sbin/accton $ACCTFILE
        touch /var/lock/subsys/psacct
}

stop() {
        action $"Shutting down process accounting: " /sbin/accton
        rm -f /var/lock/subsys/psacct
}
...
```

3. Démarrer le service.

```
[root@linux1 ~]# /etc/init.d/psacct start
Activation de la gestion des processus :                   [  OK  ]
[root@linux1 ~]#
```

4. Dans une autre session, se connecter en tant que guest pour générer des enregistrements comptables.

```
login as: guest
Sent username "guest"
guest@192.168.218.99's password: wwii1945
Last login: Sun Feb  4 08:57:53 2007 from raichu.pokemon
[guest@linux1 ~]$ date
dim fÃ©v  4 09:06:48 CET 2007
[guest@linux1 ~]$ who
root     tty1         Jan 22 20:30
root     pts/1        Jan 30 15:20 (raichu.pokemon)
guest    pts/2        Feb  4 09:06 (raichu.pokemon)
[guest@linux1 ~]$ uptime
 09:06:54 up 13 days, 10:33,  3 users,  load average: 0.00, 0.00, 0.00
[guest@linux1 ~]$ exit
```

5. Lister les totaux de connexions.

```
[root@linux1 ~]# ac
        total        0.01
```

6. Lister les totaux de connexions par jour.

```
[root@linux1 ~]# ac -d
Today   total        0.01
```

7. Lister les totaux de connexions par utilisateur.

```
[root@linux1 ~]# ac -p
        guest                                   0.01
        total        0.01
```

8. Lister les totaux d'utilisation du CPU par commande. C'est l'option par défaut de la commande sa.

```
[root@linux1 ~]# sa
     42       2.43re       0.00cp     1054k
     12       1.53re       0.00cp     1005k   ***other
     12       0.00re       0.00cp     1186k   bash*
      6       0.00re       0.00cp     1001k   id
      4       0.00re       0.00cp      710k   ac
      2       0.90re       0.00cp     1882k   sshd*
      2       0.00re       0.00cp     1296k   initlog
```

```
      2        0.00re        0.00cp       796k   grep
      2        0.00re        0.00cp       592k   egrep
```

9. Afficher les totaux d'utilisation du CPU par utilisateur.

```
[root@linux1 ~]# sa -m
                                    43       2.43re       0.00cp     1041k
guest                               31       1.25re       0.00cp     1039k
root                                11       0.91re       0.00cp      986k
sshd                                 1       0.28re       0.00cp     1723k
```

10. Afficher l'utilisation du CPU pour toutes les commandes enregistrées, l'utilisateur est mentionné.

```
[root@linux1 ~]# sa -u
root       0.00 cpu       331k mem accton
root       0.00 cpu      1334k mem initlog
root       0.00 cpu      1259k mem initlog
root       0.00 cpu      1263k mem touch
root       0.00 cpu      1195k mem psacct
root       0.00 cpu       700k mem ac
sshd       0.00 cpu      1723k mem sshd              *
guest      0.00 cpu       532k mem id
...
```

11. Afficher les données brutes.

```
[root@linux1 ~]# lastcomm
sa                S     root     pts/1      0.00 secs Sun Feb  4 09:10
crond             SF    root     __         0.00 secs Sun Feb  4 09:10
sadc              S     root     __         0.00 secs Sun Feb  4 09:10
date              S     root     __         0.00 secs Sun Feb  4 09:10
sa                S     root     pts/1      0.00 secs Sun Feb  4 09:09
sa                S     root     pts/1      0.00 secs Sun Feb  4 09:09
```

12. Idem, mais concernant uniquement un utilisateur.

```
[root@linux1 ~]# lastcomm guest
sshd              SF    guest    __         0.00 secs Sun Feb  4 09:06
bash                    guest    pts/2      0.01 secs Sun Feb  4 09:06
clear                   guest    pts/2      0.00 secs Sun Feb  4 09:07
uptime                  guest    pts/2      0.00 secs Sun Feb  4 09:06
who                     guest    pts/2      0.00 secs Sun Feb  4 09:06
date                    guest    pts/2      0.00 secs Sun Feb  4 09:06
```

13. Arrêter le service.

```
[root@linux1 ~]# /etc/init.d/psacct stop
ArrÃªt de la gestion des processus :                        [  OK  ]
[root@linux1 ~]#
```

Tâche 5 : Surveiller le système en pénurie de ressources

1. Créer des commandes pour consommer des ressources.

```
[root@linux1 ~]# vi charge_cpu.sh
{ perl -e '$var=2.5;while(1){$var*=3;$var/=3;}' & sleep 10;kill $!; } &
[root@linux1 ~]# vi charge_mem.sh
{ perl -e 'for($i=0;$i<100000000;$i++){$tb[$i]=$i;}' & sleep 10;kill $!; } &
[root@linux1 ~]# vi charge_io.sh
{ find / -type f -print -exec grep -i Linux {} \; >/dev/null 2>&1 & sleep
10;kill $!; } &
```

ATTENTION : NE PAS UTILISER CES COMMANDES SUR UN SYSTEME EN EXPLOITATION !

2. Activer les commandes de simulation de charge et de surveillance en même temps.

```
[root@linux1 ~]# { sleep 10; sh charge_mem.sh & sh charge_cpu.sh & sh
charge_io.sh & } &
[1] 30019
[root@linux1 ~]# vmstat 10 5
procs -----------memory---------- ---swap-- -----io---- --system-- ----cpu----
 r  b   swpd   free   buff  cache   si   so    bi    bo   in   cs us sy id wa
 0  0      0 150292  55744 182292    0    0     1     1    0   34  0  0 100  0
 4  0      0 116692  55744 182340    0    0     2     4 1052  248 19  3 78  0
 0  0      0 196236  55632 140596    0    0   131    37 1057  448 69  9 21  1
 0  0      0 196364  55632 140596    0    0     0     3 1051  143  0  0 100  0
 0  0      0 196428  55632 140596    0    0     0    17 1053  143  0  0 100  0
[1]+  Done                    { sleep 10; sh charge_mem.sh & sh charge_cpu.sh &
sh charge_io.sh & }
```

Remarque : On peut utiliser d'autres commandes de surveillance au lieu de `vmstat`.

Tâche 6 : Limiter l'usage des ressources

1. Limiter les ressources avec ulimit.

```
[root@linux1 ~]# ssh -l guest localhost
guest@localhost's password: wwii1945
Last login: Sun Feb  4 09:06:46 2007 from raichu.pokemon
[guest@linux1 ~]$ ulimit -a
core file size          (blocks, -c) 0
data seg size           (kbytes, -d) unlimited
file size               (blocks, -f) unlimited
pending signals                 (-i) 1024
max locked memory       (kbytes, -l) 32
max memory size         (kbytes, -m) unlimited
open files                      (-n) 1024
pipe size            (512 bytes, -p) 8
POSIX message queues     (bytes, -q) 819200
stack size              (kbytes, -s) 10240
cpu time               (seconds, -t) unlimited
max user processes              (-u) 7151
virtual memory          (kbytes, -v) unlimited
file locks                      (-x) unlimited
[guest@linux1 ~]$ ulimit -S -m 32
```

Remarque : On limite la limite soft de la taille mémoire à 32 Mo.

```
[guest@linux1 ~]$ ulimit -u -m
max user processes              (-u) 7151
max memory size         (kbytes, -m) 32
[guest@linux1 ~]$ ulimit -m
32
[guest@linux1 ~]$ perl -e 'for($i=0;$i<1000000000;$i++){$tb[$i]=$i;}'
Out of memory!
[guest@linux1 ~]$ free
             total       used       free     shared    buffers     cached
Mem:        449908      25020     424888          0        252       7240
```

```
-/+ buffers/cache:       17528     432380
Swap:       522072       24180     497892
[guest@linux1 ~]$ ulimit -S -u 5
```

Remarque : On limite à 5, le nombre de processus simultané pour guest.

```
[guest@linux1 ~]$ ulimit -u -m
max user processes              (-u) 5
max memory size         (kbytes, -m) 32
[guest@linux1 ~]$ ulimit -u
5
[guest@linux1 ~]$ sleep 10 & sleep 10 & sleep 10 &
[1] 32055
[2] 32056
[3] 32057
[guest@linux1 ~]$ sleep 10 &
-bash: fork: Ressource temporairement non disponible
```

Remarque : l'activation d'un cinquième processus (ne pas oublier le shell), échoue.

```
[guest@linux1 ~]$ ulimit -u unlimited
-bash: ulimit: max user processes: cannot modify limit: OpÃ©ration non permise
[guest@linux1 ~]$ ulimit -u 751
[guest@linux1 ~]$ ulimit -m unlimited
[guest@linux1 ~]$ ulimit -u -m
max user processes              (-u) 751
max memory size         (kbytes, -m) unlimited
[guest@linux1 ~]$ ulimit -m 32
[guest@linux1 ~]$ export LANG=C
[guest@linux1 ~]$ ulimit -m unlimited
-bash: ulimit: max memory size: cannot modify limit: Operation not permitted
[guest@linux1 ~]$ ulimit -m 64
-bash: ulimit: max memory size: cannot modify limit: Operation not permitted
[guest@linux1 ~]$ exit
```

Remarque : Quand on modifie la limite hard, on ne pas revenir en arrière (pour cette session).

2. Limiter les ressources de guest à chaque connexion.

```
[root@linux1 ~]# cp /etc/security/limits.conf /etc/security/limits.conf.000
[root@linux1 ~]# vi /etc/security/limits.conf
...
#@student        -       maxlogins       4
guest            -       maxlogins       1
guest            hard    nproc           5
# End of file
```

Remarque : On ajoute les lignes en italique.

```
[root@linux1 ~]# ssh -l guest localhost
guest@localhost's password: wwii1945
Last login: Sun Feb  4 17:02:10 2007 from localhost.localdomain
[guest@linux1 ~]$ sleep 30 & sleep 30 & sleep 30 &
[1] 32104
[2] 32105
[3] 32106
[guest@linux1 ~]$ sleep 10 &
-bash: fork: Ressource temporairement non disponible
[guest@linux1 ~]$
```

A partir, d'un autre terminal, on essaye de se connecter, cela échoue.

```
[root@linux1 ~]# ssh -l guest localhost
guest@localhost's password: wwii1945
Read from remote host localhost: Connection reset by peer
Connection to localhost closed.
```

On remet la configuration d'origine.

```
[root@linux1 ~]# cp /etc/security/limits.conf.000 /etc/security/limits.conf
cp: overwrite `/etc/security/limits.conf'? y
[root@linux1 ~]#
```

3. Supprimer la mise à jour des dates de derniers accès.

```
[root@linux1 ~]# dd if=/dev/zero of=/root/GROS_FIC bs=1k count=100000
100000+0 records in
100000+0 records out
[root@linux1 ~]# mkfs -q -F /root/GROS_FIC
[root@linux1 ~]# mount -o loop /root/GROS_FIC /mnt
[root@linux1 ~]# cal > /mnt/f1
```

On attend un peu (deux minutes environ).

```
[root@linux1 ~]# cat /mnt/f1 > /dev/null
[root@linux1 ~]# ls -l /mnt/f1
-rw-r--r--  1 root root 137 Feb  4 17:40 /mnt/f1
[root@linux1 ~]# ls -lu /mnt/f1
-rw-r--r--  1 root root 137 Feb  4 17:41 /mnt/f1
```

Remarque : La lecture du fichier à provoqué la mise à jour de la date de dernier accès et donc à provoqué UNE ECRITURE SUR LE DISQUE !

On recommence l'exercice mais en désactivant la mise à jour des dates de dernier accès.

```
[root@linux1 ~]# umount /mnt
[root@linux1 ~]# mount -o loop,noatime /root/GROS_FIC /mnt
[root@linux1 ~]# date
Sun Feb  4 17:46:52 CET 2007
[root@linux1 ~]# cat /mnt/f1 > /dev/null
[root@linux1 ~]# ls -lu /mnt/f1
-rw-r--r--  1 root root 137 Feb  4 17:41 /mnt/f1
[root@linux1 ~]# umount /mnt
[root@linux1 ~]#
```

Tâche 7 : Modifier les paramètres du noyau.

1. Créer des scripts

a) Un script qui simule une application qui écrit sur disque.

```
[root@linux1 ~]# vi load_output.sh
#!/bin/sh
GROSFIC=/usr/grosfic.dat
for i in 1 2 3 4 5 6 7
do
        dd if=/dev/zero of=$GROSFIC bs=1k count=100000
        rm $GROSFIC
done
```

b) Un script de surveillance de la mémoire.

```
[root@linux1 ~]# vi visu_charge_memoire.sh
```

```
#!/bin/sh
for i in 1 2 3 4 5 6 7
do
        clear
        date
        head -20 /proc/meminfo
        sleep 3
done
```

2. Activer les écritures disque et surveiller l'activité mémoire.

a) Avant de changer un paramètre du noyau.

```
[root@linux1 ~]# cat /proc/sys/vm/page-cluster
3
[root@linux1 ~]# time sh load_output.sh > /tmp/result 2>&1 &
[1] 32416
[root@linux1 ~]# sh visu_charge_memoire.sh
Sun Feb  4 18:17:36 CET 2007
MemTotal:       449908 kB
MemFree:        279088 kB
Buffers:          4076 kB
Cached:         130568 kB
SwapCached:       6032 kB
...
```

b) Après (on prend l'exemple du paramètre page-cluster).

```
[root@linux1 ~]# echo 20 > /proc/sys/vm/page-cluster
[root@linux1 ~]# cat /proc/sys/vm/page-cluster
20
```

On refait l'exercice précédent.

Tâche 8 : Simuler une charge avec un benchmark, l'influence de la mémoire

On simule une charge Web avec l'outil `ab` (Apache doit être actif).

```
[root@linux1 ~]# /etc/init.d/httpd restart
ArrÃªt de httpd :                                           [ÃCHOUÃ]
DÃ©marrage de httpd :                                       [  OK  ]
[root@linux1 ~]# ab -c 20 -n 10000 'http://localhost/'
This is ApacheBench, Version 2.0.41-dev <$Revision: 1.141 $> apache-2.0
Copyright (c) 1996 Adam Twiss, Zeus Technology Ltd, http://www.zeustech.net/
Copyright (c) 1998-2002 The Apache Software Foundation, http://www.apache.org/

Benchmarking localhost (be patient)
Completed 1000 requests
Completed 2000 requests
Completed 3000 requests
Completed 4000 requests
Completed 5000 requests
Completed 6000 requests
Completed 7000 requests
Completed 8000 requests
Completed 9000 requests
Finished 10000 requests
```

```
Server Software:        Apache/2.0.52
Server Hostname:        localhost
Server Port:            80

Document Path:          /
Document Length:        5044 bytes

Concurrency Level:      20
Time taken for tests:   6.656780 seconds
Complete requests:      10000
Failed requests:        0
Write errors:           0
Non-2xx responses:      10010
Total transferred:      52472420 bytes
HTML transferred:       50490440 bytes
Requests per second:    1502.23 [#/sec] (mean)
Time per request:       13.314 [ms] (mean)
Time per request:       0.666 [ms] (mean, across all concurrent requests)
Transfer rate:          7697.72 [Kbytes/sec] received

Connection Times (ms)
              min  mean[+/-sd] median   max
Connect:        0    4   2.1      5      23
Processing:     4    7   2.6      7      22
Waiting:        1    5   2.2      6      21
Total:          9   12   1.4     12      30

Percentage of the requests served within a certain time (ms)
  50%     12
  66%     13
  75%     13
  80%     13
  90%     13
  95%     13
  98%     14
  99%     18
 100%     30 (longest request)
[root@linux1 ~]#
```

Remarques :

1) Dans l'exemple, la commande `ab` active 10000 requêtes HTTP, on active 20 requêtes simultatnément (cf. ab(8).

2) On peut activer la commande ab à partir du poste en binôme.

On réalise le test :

a) En mode graphique avec 128 Mo de mémoire.

b) En mode texte avec 129 Mo de mémoire.

c) En mode graphique dans l'état par défaut du serveur.

Remarque : Pour indiquer le niveau d'init et la mémoire, on édite la configuration Grub lors du démarrage, par exemple :

kernel /vmlinuz-2.6.9-11.EL.smp ro root=LABEL=/ rhgb quiet ***3 mem=128M***

- *Troubleshooting*
- */var/log/messages*
- *Google site:apache.org*
- *lsof, strace, ping*
- *Bugzilla*

12

Le dépannage

Objectifs

Ce chapitre traite du dépannage d'un système informatique fonctionnant sous Linux. On commence par expliquer la méthode générale de résolution d'incident. On présente également de manière plus concrète les principales commandes de diagnostics.

Contenu

L'approche générale du dépannage

Étape n°1 : Faire une description claire du problème

Étape n°2 : Imaginer les causes possibles

Étape n°3 : Rassembler des informations

Étape n°4 : Analyser les faits rassemblés, éliminer des causes

Les précautions

Régler les problèmes sans les résoudre

Ateliers



L'approche générale du dépannage

La théorie

Pour trouver la raison d'un mauvais fonctionnement d'un système informatique, l'approche générale du dépannage (Troubleshooting) consiste à isoler le problème parmi la multitude des causes possibles. Il faut au préalable avoir une connaissance la plus fine possible du fonctionnement du système pour pouvoir émettre des hypothèses.

Le savoir pratique

L'algorithme général de résolution de problème :

Etape n°1 : faire une description claire du problème.

Etape n°2 : Imaginer les causes possibles.

Etape n°3 : Rassembler des informations. On utilise des méthodes non destructives.

Etape n°4 : Analyser les faits rassemblés. On élimine des causes.

Etape n°5 : On crée un plan d'action basé sur les causes restantes. On prévoit de ne changer qu'un seul paramètre à la fois.

Etape n°6 : On met en œuvre le plan.

Etape n°7 : On récolte les résultats.

Etape n°8 : On analyse les résultats.

Etape n°9 : Si le problème n'a pas été résolu, on boucle (on revient à l'étape 5).

Etape n°10 : Si le problème a été résolu, on met à jour le système, les sauvegardes, la documentation, etc. pour éviter qu'il se produise à nouveau.

Pour en savoir plus

Internet

Troubleshooting (Dépannage)
http://en.wikipedia.org/wiki/Troubleshooting

Troubleshooting -- the HP Way
http://redmondmag.com/features/article.asp?EditorialsID=555

Livre

Linux Troubleshooting for System Administrators and Power Users, par James Kirkland , David Carmichael , Christopher L. Tinker , Gregory L. Tinker, chez HP invent

Etape n° 1 : Faire une description claire du problème

La théorie

Quand le problème survient, il est important de le décrire clairement par écrit. L'idéal est de mémoriser dans une base de données informatique des « fiches incidents ».

Si possible, il faut reproduire le problème pour visualiser notamment un message d'erreur éventuel.

Le savoir pratique

Une fiche incident type

- N° de pièce.
- Date/Heure.
- Brève description du problème (en une phrase).
- Degré de gravité, degré d'urgence.
- Qui a rempli ce document.
- Qui a signalé le problème.
- Environnement concerné (Matériel, OS, logiciel, personne).
- Description complète du problème (enregistrer notamment les messages d'erreur).
- Cause du problème.
- Actions réalisées (quand, qui, matériels remplacés, …).
- Etat (résolu/non résolu).

Pour en savoir plus

Internet

Bugzilla – Logiciel pour enregistrer et rechercher des bugs.
http://www.bugzilla.org/

Etape n°2 : Imaginer les causes possibles

La théorie

L'expérience nous enseigne rapidement les causes les plus vraisemblables.

Malheureusement, dans plusieurs cas, seule la connaissance théorique des systèmes informatiques et de leur interaction nous permet d'envisager toutes les causes possibles.

L'approche générale du dépannage à pour but d'écarter le maximum possible de causes. Il faut pouvoir isoler le ou les sous-systèmes responsables. Ensuite il faut trouver la cause élémentaire du problème.

Le savoir pratique

Les sous-systèmes

- Le matériel et le logiciel (Hardware/Software).
- Au niveau logiciel : le système d'exploitation et les logiciels applicatifs.
- Au niveau logiciel : le logiciel concerné ou le paquetage concerné.
- Au niveau logiciel : le cœur de l'applicatif (core) ou ses modules (bibliothèques)
- Au niveau logiciel : problème local ou problème réseau.
- Au niveau réseau : la couche réseau concernée. Le client ou le serveur.

Causes possibles du problème pour un sous-système donné

- MAUVAISE CONFIGURATION ! Un fichier de configuration peut être tout simplement absent. Sa syntaxe peut être mauvaise. Un ou plusieurs paramètres sont mal configurés. La configuration par défaut est inadéquate ou différente de l'ancienne version. La configuration est incompatible avec la sécurité.
- BUG ! Hélas, la plupart du matériel ou des logiciels utilisés présentent des défauts : erreurs de conception, de fabrication ou de programmation.
- Panne d'un sous-système du sous-système. Par exemple un processus mort.
- Prérequis ou ressources insuffisantes (mémoire, disque, …).
- Le piratage.

Indices

- Les sous-systèmes directement concernés par le problème et qui sont à l'origine du message d'erreur ou mentionné dans le message d'erreur.
- Les faits récents : mise à jour du système, changement de configuration, mise en place d'un nouveau matériel, …

Etape n°3 : Rassembler des informations

La théorie

Après avoir imaginé un ensemble de causes, il faut rassembler des informations. Mais pour le moment on utilise des méthodes non destructives. Il est important de ne pas détruire le système ou d'effacer des indices avant d'avoir établi un diagnostic complet.

Différentes sources d'information

- Les messages d'erreurs apparaissant sur le poste.
- Les journaux de bords.
- Les commandes de diagnostics.

Normalement, les journaux de bords constituent la meilleure source d'information (à condition qu'ils aient été activés et que leur niveau de verbiage soit suffisant).

Remarque : La gestion des journaux de bords passe le plus souvent par le système Syslog qu'il faut maîtriser.

L'idéal est de travailler de manière systématique en utilisant des « check-list ». Pour les diagnostics de chaque sous-système il existe souvent une check-list. Si besoin, on a tout intérêt à les amender ou à les créer de toutes pièces. Elles permettent de travailler plus vite et d'être exhaustif.

Le savoir concret

Commandes d'informations concernant les applications

`ps`	Liste des processus.
`ldd`	Liste les bibliothèques partagées d'un exécutable.
`strace`	Trace les appels système d'une application.
`ltrace`	Trace les appels à une bibliothèque.
`lslk`	Liste les verrous.
`who, w`	Qui est connecté
`fuser`	Liste les applications accédants à un fichier.
`lsof`	Liste les fichiers ouverts.
`ipcs`	Affiche des informations sur les IPC.
`strings`	Liste les chaînes ASCII d'un binaires (les messages d'erreurs, …).
`rpm`	Liste des paquetages installés, les fichiers d'un paquetage, vérifie un paquetage.
`chkconfig --list`	Liste les services actifs.

Commandes d'informations concernant l'état du système

`uptime`	Affiche la charge système.
`free`	Affiche l'utilisation de la mémoire et de la mémoire de swap.
`top`	Affiche les applications le plus consommatrices de ressources.

`vmstat`	Affiche des statistiques sur l'utilisation des ressources systèmes.
`df`	Affiche la place disque disponible.
`mount`	Liste les FS montés et leurs options de montage.
`vgdisplay -v`	Affiche la configuration LVM (VG, LV, PV).
`runlevel`	Affiche l'état d'init.
`repquota`	Affiche l'utilisation des quota.

Commandes d'informations concernant la sécurité

`getenforce`	Affiche l'état de SELinux.
`iptables`	Affiche les règles pare-feu présentes dans le noyau.
`ldd`	Liste les bibliothèques d'un exécutable (PAM, WRAP, ...)
`last`	Affiche les dernières connexions.
`rpm -V`	Vérifie un paquetage (droits, sommes de contrôles, ...)
`ls -l`	Affiche les droits d'un fichier.
`lsattr`	Liste les attributs Linux d'un fichier.
`getfacl`	Liste les ACL d'un fichier.

Commandes d'informations concernant les versions des logiciels

`rpm -q <pkg>`	Affiche la version d'un paquetage RPM
`<logiciel> --version`	Affiche la version d'un logiciel GNU
`uname -a`	Affiche la version du noyau.
`cat /etc/issue`	Affiche la version de la distribution.

Les commandes d'informations réseau

`ifconfig`	Affiche les cartes réseau. Affiche leur paramétrage : adresse IP, masque, adresse de diffusion.
`netstat`	Affiche des statistiques. Affiche les tables de routage (dont la passerelle par défaut). Affiche les sockets actives.
`ping`	Teste de connectivité.
`ethtool`	Informe de la connexion entre une carte réseau et le switch.
`nslookup`	Réalise une recherche DNS ou reverse-DNS.
`tcpdump, ethereal`	Analyse les échanges réseaux.
`telnet`	Client universel (en mode texte).
`nc`	Client ou serveur universel (NetCat).
`arp`	Affiche les tables ARP.
`nmap`	Affiche les ports ouverts d'un système distant
`rpcinfo`	Affiche les services RPC actifs d'un système distant.
`uname, hostname`	Affiche le nom réseau du système.
`getent`	Visualise un enregistrement ou un annuaire réseau complet.

Les commandes d'informations sur le matériel

`dmesg`	Affiche les messages provenant du noyau, notamment au démarrage.
`lspci`	Visualise les bus PCI.
`cat /proc/cpuinfo`	Information sur le ou les CPU (vitesse, ...).
`cat /proc/mdstat`	Affiche la configuration RAID software.
`cat /proc/mounts`	Les FS montés (vision du noyau).
`free`	Affiche la quantité de mémoire.
`lsusb`	Visualise les bus .
`lsmod`	Visualise les modules du noyau en mémoire.
`sfdisk -s`	Liste les disques.
`sdisk -l`	Liste les partitions d'un disque.

Divers

Memtest	Logiciel standalone de test de la mémoire vive. Il est présent notamment sur le live-cd « Knoppix ». On peut l'activer au démarrage au lieu de d'activer Linux.

Les principaux fichiers de configuration

/etc/passwd	Les comptes utilisateurs.
/etc/group	Les comptes groupes.
/etc/shadow	Les mots de passe des utilisateurs.
/etc/inittab	Les premières applications activées au démarrage.
/etc/fstab	Les FS et le swap activés au démarrage.
/boot/grub/grub.conf	La configuration du chargeur Grub.
/etc/modprobe.conf	La configuration des pilotes dynamiques.
/etc/nsswitch.conf	La configuration des annuaires.
/etc/resolv.conf	La configuration du DNS.
/etc/syslog.conf	La configuration du service Syslog.

Focus : La commande strace

La commande `strace` est une des commandes les plus puissantes dans le domaine de la résolution de problèmes. Elle trace les appels au noyau d'une application.

Quelques options

-f	Trace les processus fils.
-p pid	S'attache à un processus dont on indique le PID.
-t	Préfixe chaque ligne de l'heure.

Exemple

```
# strace ps 2> err
  PID TTY          TIME CMD
 5763 pts/0    00:00:00 bash
 7072 pts/0    00:00:00 strace
```

```
 7073 pts/0    00:00:00 ps
# head -3 err
execve("/bin/ps", ["ps"], [/* 22 vars */]) = 0
uname({sys="Linux", node="linux1.pinguins", ...}) = 0
brk(0)                                  = 0x9b98000
[root@linux1 ~]# head -5 err
execve("/bin/ps", ["ps"], [/* 22 vars */]) = 0
uname({sys="Linux", node="linux1.pinguins", ...}) = 0
brk(0)                                  = 0x9b98000
access("/etc/ld.so.preload", R_OK)      = -1 ENOENT (No such file or
directory)
open("/etc/ld.so.cache", O_RDONLY)      = 3
```

Focus : La commandes lsof

La commande `lsof` est une des commandes les plus puissantes dans le domaine de la résolution de problèmes. Elle liste l'ensemble des fichiers ouverts et indique pour chacun l'application qui l'a ouverte. Les sockets réseaux et les périphériques sont considérés également comme des fichiers.

```
[root@linux1 ~]# lsof |more
COMMAND    PID  USER   FD   TYPE     DEVICE     SIZE  NODE NAME
init         1  root  cwd   DIR        3,2     4096     2 /
udevd     1495  root    3r  FIFO         0,7           3396 pipe
udevd     1495  root    4w  FIFO         0,7           3396 pipe
udevd     1495  root    5u  unix 0xdb17bb80           3397 socket
portmap   3021   rpc    2u  CHR        1,3           1811 /dev/null
portmap   3021   rpc    3u  IPv4       6579            UDP *:sunrpc
portmap   3021   rpc    4u  IPv4       6582            TCP *:sunrpc
(LISTEN)
syslogd   2988  root    0u    unix 0xdb1864c0          6528 /dev/log
syslogd   2988  root    1w    REG        3,2   199606 294099
/var/log/messages
```

Focus : un exemple de Check-list

La checklist suivante :
/usr/share/doc/samba-3.*/htmldocs/Samba-HOWTO-Collection/diagnosis.html
qui est le chapitre 33 du « The Official Samba-3 HOWTO and Reference Guide » décrit les opérations à effectuer dans le cas où l'on ne peut pas faire communiquer un serveur Samba et un poste Windows. A chaque étape, une liste des causes possibles d'échecs est mentionnée. Voici un résumé des principales étapes.

1) Est-ce que la syntaxe du fichier de configuration de Samba est correcte ?
   # testparm

2) Est-ce que la connectivité réseau est assurée (et la résolution DNS) ?
   a) A partir du client :
   C:\> ping LE_SERVEUR
   b) A partir du serveur :
   # ping LE_CLIENT

3) Est-ce que les ressources partagées du serveur sont visibles ?
   # smbclient –L –v //LE_SERVEUR

4) Est-ce que le service Netbios fonctionne correctement côté serveur ?
   # nmblookup –B LE_SERVEUR __SAMBA__

5) Est-ce que le service Netbios fonctionne correctement côté client ?
nmblookup –B LE_CLIENT '*'

6) Idem, mais on utilise la diffusion et un niveau de debugging plus élevé :
nmblookup –d 2 '*'

7) Est-ce que l'on peut ouvrir une session cliente à partir du serveur ?
smbclient //LE_SERVEUR/TMP -Upierre%secret
(On utilise un partage et un compte déjà initialisés)

8) Est-ce que les partages sont visibles du client ?
C:\> net view \\LE_SERVEUR

9) Est-ce que l'on peut ouvrir une session cliente à partir du client ?
C:\> net use X: \\LE_SERVEUR\TMP secret /user:pierre

10) Est-ce que le service explorateur fonctionne correctement ? Est-ce qu'il y a un explorateur maître du groupe de travail ?
nmblookup –M workgroup

11) Essayer de se connecter à partir du poste client avec le File Manager.

Astuces

Utiliser la langue anglaise lorsqu'on reproduit un problème !

Si on utilise la langue anglaise pour un logiciel, les messages d'erreurs seront en anglais. Une recherche avec Google sera beaucoup fructueuse qu'en utilisant le message français.

Exemple :

```
root@herbizarre:~# date -ererer
date: option invalide -- e
Pour en savoir davantage, faites: « date --help ».
root@herbizarre:~# export LANG=C
root@herbizarre:~# date -ererer
date: invalid option -- e
Try `date --help' for more information.
```

Créer des scripts

Les commandes de diagnostics peuvent être regroupées dans des scripts. On peut ainsi réaliser les diagnostics très rapidement. Ces scripts peuvent également récolter les messages récents provenant des journaux de bords.

Comparer

Si vous avez un système de secours ou un serveur similaire (sur certains points ou totalement) au système défaillant, vous pouvez les comparer (les fichiers de configuration, les processus actifs, …).

Pour en savoir plus

Les pages de manuel

lsof(8), strace(1)

Internet

Check-list de diagnostic de Samba
http://samba.org/samba/docs/man/Samba-HOWTO-Collection/diagnosis.html

Red Hat Enterprise Linux Step By Step Guide (III. Troubleshooting Your System)
http://www.redhat.com/docs/manuals/enterprise/RHEL-4-Manual/step-guide/

SuSE Linux 10 Start-Up Guide. (Chap IV : Troubleshooting)
www.novell.com/documentation/suse10/startguide/data/bookinfo_book_startguide.html

Debian Tutorial - Chapter 15 – Troubleshooting
http://www.debian.org/doc/manuals/debian-tutorial/ch-troubleshooting.html

HP ProLiant Servers Troubleshooting Guide
http://h20000.www2.hp.com/bc/docs/support/SupportManual/c00300504/c00300504.pdf

Troubleshooting - IBM System x
http://www-1.ibm.com/support/docview.wss?uid=psg1MIGR-50556

Livre

Network Troubleshooting Tools, par Joseph D. Sloan, chez O'Reilly

Etape n°4 : Analyser les faits rassemblés, éliminer des causes

Le savoir concret

Stratégies

- Effectuer des recherches avec Google (par exemple faire un couper/coller du message d'erreur).
- Rechercher/interroger les FAQ, les forums, les mailling list.
- Rechercher les symptômes dans les rapports de bugs officiels.
- Compléter ses connaissances théoriques générales ou ponctuelles.
- Rechercher les modifications récentes des logiciels dans les Changelog des RPM.
- Lire le code source du logiciel (méthode normalement réservée à l'expert). On peut par exemple y rechercher l'origine exacte du message d'erreur.

Remarque : Il faut aussi rapidement prendre en ligne de compte ses propres limites : possède-t-on les connaissances, les outils et le temps pour résoudre le problème ? Ne faut-il pas chercher de l'aide (spécialiste des sous-systèmes) ?

Rechercher avec Google

- Ajouter ou retirer des mots clés pour restreindre ou élargir la recherche.
- Mettre la phrase entre guillemets (") pour rechercher l'expression exacte (sinon la phrase donnée est considérée comme une suite de mots clés).
- Débuter les mots que l'on veut exclure par moins (-).
- Séparer les mots clés par OR si l'on recherche des mots alternatifs.
- Rechercher sur un site particulier (site:www.redhat.com).
- Utiliser les autres sites de Google (groups.google.fr pour rechercher dans les forums ou www.google.fr/dirhp pour rechercher dans l'annuaire).

Pour en savoir plus

Internet

Bugzilla (Red Hat bug-tracking system)
https://bugzilla.redhat.com

Novell/SUSE Bugzilla overview
https://bugs.launchpad.net/bugs/bugtrackers/novell-bugs

Système de suivi des bogues Debian
http://www.debian.org/Bugs/

Google – Recherche avancée
http://www.google.fr/advanced_search

Les FAQ
http://www.faqs.org

Un exemple de dépannage

Le problème

Paul, qui travaille sur une station Linux m'a signalé qu'il ne peut plus imprimer à partir d'Open Office.

Dépannage

Etape n°1 : Faire une description claire du problème

Je remplis la fiche suivante :

```
N° de pièce :  243
Date/Heure :  10/12/06 à 10h30
Brève description du problème (en une phrase) : L'impression locale
à partir du poste cao23 ne fonctionne plus à partir d'Open Office.
Degré de gravité, degré d'urgence : 5/10, 5/10
Qui a rempli ce document : Jean Grandgourou
Qui a signalé le problème : Paul Hochon
Environnement concerné (Matériel, OS, logiciel, personne) : Dell
C521, RedHat 4.4, OpenOffice 1.1.5, Paul Hochon
Description complète du problème (enregistrer notamment les messages
d'erreur) :
<Pas de message d'erreur !>
Cause de l'erreur :
Actions réalisées (quand, qui, matériels remplacés, …):
Etat (résolu/non résolu):
```

Etape n°2 : Imaginer les causes possibles

Les sous-systèmes a priori impliqués

Le matériel : l'imprimante, l'ordinateur (carte mère, …).

Logiciels : Open Office, le système d'impression Cups.

Indices

La semaine dernière Paul s'est fait aidé par un jeune stagiaire féru en Sécurité sous Linux. Il lui a donné le mot de passe de root. Ce jeune stagiaire a terminé son stage et Paul n'a pas ses coordonnées.

On peut donc ajouter aux sous-systèmes à explorer :

Les modules systèmes liés à la sécurité (pare-feu, SELinux, …).

Etape n°3 : Rassembler des informations

Le système Cups

```
# lp  /etc/issue
^C
# lpstat -t
^C
```

Les commandes se bloquent, je suis obligé de les arrêter par Ctrl+C.

Je recommence, mais j'insiste : j'attends 5 minutes : et je vois un message d'erreur !

```
# lpstat -t
lp: error -scheduler not responding
```

```
# ps -e | grep cupsd
2195 ?  00:00:00 cupsd
# netstat -an | grep 631  # le port d'écoute de cupsd
tcp        0      0 127.0.0.1:631    0.0.0.0:*           LISTEN
udp        0      0 0.0.0.0:631      0.0.0.0:*
```

Le matériel

```
# cat  un_fichier.ps  >  /dev/lp0
```

J'écris directement un fichier PS sur l'imprimante PostScript. Il s'imprime.

La sécurité SELinux

```
# getenforce
Enforcing
```

La sécurité pare-feu

```
# iptables -L
Chain FORWARD (policy ACCEPT)
target     prot opt source               destination

Chain INPUT (policy ACCEPT)
target     prot opt source               destination
DROP       tcp  --  192.168.218.17      anywhere      tcp dpt:smtp
DROP       tcp  --  anywhere            anywhere      tcp dpt:ipp

Chain OUTPUT (policy ACCEPT)
target     prot opt source               destination
```

Etape n°4 : Analyser les faits rassemblés

Éliminer des causes

A la lumière des essais précédents, on peut éliminer deux causes d'erreur :

- Le matériel : L'imprimante est fonctionnelle ainsi que l'ordinateur (carte mère, …) : on a pu imprimer directement par une redirection.
- Open Office : En essayant d'imprimer directement par la commande `lp`, on n'a pas pu imprimer. Il semble que le logiciel Open Office ne soit pas en cause.

Rechercher sur Google

On fait un copier/coller du message d'erreur (`lp: error -scheduler not responding`) dans Google. Je trouve plusieurs personnes qui semblent obtenir le même message.

- L'un des dépanneurs suggère de regarder le journal /var/log/cups/error_log.
- Un autre prétend que le problème vient d'une mauvaise configuration du fichier client /etc/cups/client.conf.

Je regarde le fichier journal de Cups (j'aurai pu y penser à l'étape précédente !) :

```
# tail /var/log/cups/error_log
I [05/Mar/2007:21:44:55 +0100] Added remote printer "laser"...
I [05/Mar/2007:21:44:55 +0100] Added remote printer "laser@cao23"...
I [05/Mar/2007:21:44:55 +0100] Added implicit class "laser"...
```

Je ne vois rien qui a un rapport avec mon problème à l'époque où il s'est produit.

A priori je n'ai pas modifié le fichier client.conf. Je télécharge un exemplaire de ce fichier provenant du poste cao15 et je le compare au fichier présent sur le poste ayant un problème :

```
#  diff  /etc/cups/client.conf  /tmp/client.conf
#
```

Il n'y a pas de différences.

Etape n°5 : On crée un plan d'action

Je crée un plan d'action basé sur les causes restantes. Je prévois de ne changer qu'un seul paramètre à la fois.

Voici les causes que j'ai retenues :

- Le système d'impression Cups.
- La sécurité SELinux.
- La sécurité de type pare-feu

Etapes n°6, n°7, n°8 et n°9

On met en œuvre le plan, on récolte les résultats, on analyse les résultats et on boucle si l'on n'a pas résolu le problème.

a) Le système d'impression Cups

On fait un arrêt/redémarrage du service, on vérifie si le redémarrage s'est bien passé. Ensuite on essaye d'imprimer.

```
# /etc/init.d/cups stop
# /etc/init.d/cups start
# ps -e |grep cupsd
# netstat -an |grep 631
# tail /var/log/cups/error_log
# lp  /etc/issue
^C
```

Conclusion : le système Cups a bien redémarré, mais l'impression ne fonctionne toujours pas.

b) La sécurité SELinux

Je désactive la sécurité SELinux, on redémarre le système. Ensuite on essaye d'imprimer.

```
# vi /etc/selinux/config
SELINUX=disabled
# reboot
...
# lp  /etc/issue
^C
```

Conclusion : la sécurité SELinux ne semble pas être en cause.

Je reconfigure SELinux avant de continuer.

```
# vi /etc/selinux/config
SELINUX=enforcing
# reboot
```

c) Le pare-feu interne

Je supprime les règles pare-feu et j'essaie d'imprimer.

```
# /etc/init.d/iptables stop
# lp  /etc/issue
request id is imp-323 (1 file(s)
```

Conclusion : l'impression fonctionne ! Le jeune stagiaire avait ajouté une règle pare-feu interdisant le dialogue entre `lp` et `cupsd`.

Etape n° 10 : Je règle le problème, j'évite qu'il se reproduise

Régler définitivement le problème

La commande `iptables stop` a temporairement supprimée les règles pare-feu. Je veux maintenant supprimer définitivement la règle concernant IPP :

```
# /etc/init.d/iptables start
# iptables -L --line-numbers  # affiche le numéro des règles
# iptables -D INPUT 2
# /etc/init.d/iptables save
```

Je mets à jour la fiche incident

```
Cause du problème : un stagiaire à modifier les règles pare-feu du
poste. Le dialogue entre un client Cups (lp, lpstat, OpenOffice, …)
et le serveur Cupsd était rendu impossible.
Actions réalisées (quand, qui, matériels remplacés, …) : suppression
d'une règle pare-feu
Etat (résolu/non résolu) : résolu.
```

Eviter qu'il se reproduise

Je change ma politique : Je vais retirer l'accès à root aux utilisateurs des stations de CAO. Ces derniers pourront continuer à accomplir certaines opérations (comme les sauvegardes) grâce à `sudo` et Webmin. Je dois lister les commandes d'administration dont ils ont absolument besoin pour configurer ces produits.

Remarques :

1) En fonction de mes connaissances systèmes, mon diagnostic peut être établi plus ou moins rapidement. Par exemple, si j'avais utilisé la commande `strace` et si j'avais eu des compétences réseaux, j'aurai constaté que la connexion avec le serveur ne pouvait être réalisée :

strace lp /etc/issue 2> err

more err

…

connect(3, {sa_family=AF_INET, sin_port=htons(631), sin_addr=inet_addr("127.0.0.

1")}, 16 <unfinished …>

2) Si je dispose d'un historique des configurations je peux utiliser une ancienne configuration opérationnelle.

Précautions

La théorie

La meilleure solution est d'anticiper les problèmes. Soit pour faire en sorte qu'ils n'apparaissent pas, soit pour limiter leur impact, soit encore pour aider à les diagnostiquer quand ils surviennent.

Cette anticipation se base principalement sur les causes possibles des problèmes et sur leur résolution.

Le savoir concret

Les précautions à prendre

- IL FAUT FAIRE DES SAUVEGARDES !
  - Des sauvegardes totales.
  - Des sauvegardes journalières (incrémentales).
  - Des sauvegardes de la configuration (/etc, ….).
- IL FAUT ACTIVER LES JOURNAUX DE BORD !
- IL FAUT FAIRE LES MISES A JOUR DU SYSTEME.
- IL FAUT CONNAITRE LE FONCTIONNEMENT DU SYSTEME.
- IL FAUT COMPARTIMENTER VOTRE SYSTEME. Il faut diviser le système en disque système et disques applicatifs. Il faut diviser le système et les applications en FS : FS stables et FS fréquemment modifiés. D'un strict point de vue du dépannage, il faut installer chaque service sur des ordinateurs différents. L'idée sous-jacente de la compartimentation est de pouvoir plus facilement isoler un problème et remplacer une unité défaillante.
- Il faut tracer, dater, commenter, mémoriser les modifications des configurations. En clair, il faut tenir un journal de bord d'administration. Il faut aussi garder la configuration d'origine.
- Vérifier la syntaxe de fichiers de configurations avant de les mettre en production. Par exemple avec Apache, on peut utiliser la commande `httpd -t` pour vérifier la syntaxe du fichier de configuration (httpd.conf).
- Il faut limiter les prérogatives des administrateurs et tracer leurs actions. Il faut aussi un responsable en chef qui coordonne les actions. Tout le monde ne doit pas pouvoir faire n'importe quoi. L'usage d'outils comme Webmin ou de sudo résout beaucoup de problèmes.
- Il faut posséder un système annexe de démarrage.
- Il faut mémoriser périodiquement l'état du système.
- Il faut avoir une copie des mots de passe.
- Il faut sécuriser son système.

Régler le problème sans le résoudre

La théorie

Quelquefois on a besoin de résoudre le problème même si l'on ne peut pas le comprendre. Nécessité fait loi.

Le savoir concret

Heuristiques

Recommencer l'opération

L'opération qui a échoué l'a été peut-être pour des raisons fortuites (panne temporaire d'un sous-système, charge réseau importante, timeout un peu court, ...). Il ne faut pas hésiter à recommencer l'opération (si cela ne présente pas de danger ou d'effets de bord) avant de conclure à une panne persistante.

Redémarrez l'application

Par exemple, suite à des fuites de mémoire, le redémarrage de l'application règle le problème.

Si un processus est mort, le redémarrage réactive l'ensemble des processus.

C'est souvent au démarrage de l'application que les sockets sont ouvertes, le redémarrage permet leur réinitialisation.

Redémarrez le système

Par exemple, si des tables IPC sont pleines, le redémarrage du système résout le problème. Les problèmes liés à la mort d'un processus sont également résolus de même.

Mettre à jour le système

Le problème que vous avez a déjà été résolu. Il suffit de passer les correctifs.

Réinstaller l'application, réinstaller le système

S'il manque un composant la réinstallation du logiciel résout le problème. Si le logiciel est buggé l'installation d'une version plus ancienne (plus stable) ou plus récente (contenant le correctif) est la solution.

Réinstaller le système et les applications sur un autre ordinateur

Par exemple, si c'est un problème lié au matériel, l'utilisation d'un autre ordinateur va régler le problème.

Utiliser le système de secours pour continuer l'exploitation

Reprendre l'exploitation à partir des sauvegardes

ATELIERS



Tâche 1 : Réaliser un script de diagnostic et de sauvegarde de la configuration... 15mn

Tâche 2 : Se renseigner sur un logiciel 10mn

Tâche 3 : Diagnostiquer et réparer un système 120mn

Tâche 1 :
Réaliser un script de diagnostic et de sauvegarde de la configuration

1. Écrire le script.

```
# more diagnostic.sh
#!/bin/sh
# $Id: diagnostic.sh,v 1.3 2007/01/04 11:23:49 root Exp root $
# $Log: diagnostic.sh,v $
# auteur: Jean-Francois BOUCHAUDY
echo "....il faut attendre un peu"
exec > /tmp/bidule$$ 2>&1

# ==== entete
echo "=== Diagnostics ===="
echo "Date: $(date)"
set -x
uname -a

# ==== etat du systeme, processus, services
runlevel
ps -ef
ps aux
free
uptime
lsof
lslk
ipcs
vmstat

# === FS et LVM
mount
df -h
df -i
vgdisplay -v

# ==== hardware & kernel
cat /proc/cpuinfo
lspci
lsusb
dmesg
lsmod
fdisk -l
```

```
# ==== reseau
ifconfig
netstat -nr
netstat -anpe

# ==== securite
getenforce
iptables -t nat    -L
iptables -t filter -L
iptables -t mangle -L
last

# ---- cloture
set -
exec > /dev/tty  2> /dev/tty
cp /tmp/bidule$$ /etc/.diag.txt
tar czf /root/etc_$(date '+%m%d%H%M').tar.gz /etc /var/log/messages
mv /tmp/bidule$$ /root/diag.txt.$(date '+%m%d%H%M')

# chmod +x  diagnostic.sh
# ./diagnostic.sh
```

Tâche 2 :
Se renseigner sur un logiciel : les commandes strace, strings,...

1. Activer une commande sous le contrôle de strace, visualiser les appels système.

```
# strace ls -l 2> /tmp/ls_strace.log
-rw-r--r--   1 root  root       1286 Jan 20 10:36 anaconda-ks.cfg
-rw-r--r--   1 root  root         72 Feb  4 15:52 charge_cpu.sh
...
# more /tmp/ls_strace.log
[root@linux1 ~]# more /tmp/ls_strace.log
execve("/bin/ls", ["ls", "-l"], [/* 22 vars */]) = 0
uname({sys="Linux", node="linux1.pinguins", ...}) = 0
brk(0)                                  = 0x8738000
access("/etc/ld.so.preload", R_OK)      = -1 ENOENT (No such file or directory)
open("/etc/ld.so.cache", O_RDONLY)      = 3
...
```

2. Visualiser les fichiers accédés par Apache au démarrage. On ne tient pas compte des bibliothèques.

```
# /etc/init.d/httpd stop
# strace /usr/sbin/httpd 2> /tmp/httpd_strace.log
# grep '^open' /tmp/httpd_strace.log  |grep -v '\.so' |more
open("/etc/httpd/conf/httpd.conf", O_RDONLY) = 3
open("/etc/httpd/conf.d", O_RDONLY|O_NONBLOCK|O_LARGEFILE|O_DIRECTORY) = 4
open("/etc/httpd/conf.d/manual.conf", O_RDONLY) = 4
open("/etc/httpd/conf.d/perl.conf", O_RDONLY) = 4
open("/etc/httpd/conf.d/php.conf", O_RDONLY) = 4
open("/etc/httpd/conf.d/python.conf", O_RDONLY) = 4
open("/etc/httpd/conf.d/ssl.conf", O_RDONLY) = 4
open("/etc/httpd/conf.d/webalizer.conf", O_RDONLY) = 4
open("/etc/httpd/conf.d/welcome.conf", O_RDONLY) = 4
open("/dev/urandom", O_RDONLY)          = 3
open("/dev/urandom", O_RDONLY)          = 3
```

```
open("/etc/nsswitch.conf", O_RDONLY)     = 5
open("/etc/passwd", O_RDONLY)            = 5
open("/etc/group", O_RDONLY)             = 5
open("/etc/resolv.conf", O_RDONLY)       = 5
open("/etc/host.conf", O_RDONLY)         = 5
open("/etc/hosts", O_RDONLY)             = 5
open("/dev/null", O_RDONLY|O_LARGEFILE) = 5
open("/usr/lib/perl5/5.8.5/i386-linux-thread-multi/DynaLoader.pm", O_RDONLY|O_LA
RGEFILE) = 5
open("/usr/lib/perl5/5.8.5/i386-linux-thread-multi/vars.pm", O_RDONLY|O_LARGEFIL
E) = -1 ENOENT (No such file or directory)
# /etc/init.d/httpd stop
ArrÃªt de httpd :                                          [  OK  ]
```

3. Surveiller un démon et ses enfants.

```
# ps -e |grep cupsd
3245 ?        00:00:00 cupsd
# strace -p 3245 -f > /tmp/cups_strace.log  2>&1 &
[1] 7173
# more /tmp/cups_strace.log
Process 3245 attached - interrupt to quit
select(1024, [0 2 3], [], NULL, {210, 792000}) = 1 (in [2], left {205, 996000})
recvfrom(2, "3056 3 ipp://192.168.0.2/printer"..., 1540, 0, {sa_family=AF_INET,
sin_port=htons(631), sin_addr=inet_addr("192.168.0.2")}, [16]) = 112
...
# pkill strace
[1]+  Done                    strace -p 3245 -f >/tmp/cups_strace.log 2>&1
```

4. Afficher les chaînes ASCII d'un binaire.

```
# strings /usr/sbin/httpd
/lib/ld-linux.so.2
libpcre.so.0
_DYNAMIC
_GLOBAL_OFFSET_TABLE_
_Jv_RegisterClasses
__gmon_start__
libpcreposix.so.0
libaprutil-0.so.0
apr_brigade_puts
apr_brigade_split
apr_brigade_partition
apr_hook_sort_all
...
```

5. Afficher les commentaires RCS/CVS d'un binaire.

```
# ident /usr/bin/ab
/usr/bin/ab:
     $Revision: 1.141 $
     $Revision: 1.141 $
```

6. Afficher les bibliothèques dynamiques référencées dans un exécutable.

```
# ldd /usr/sbin/httpd
        libpcre.so.0 => /lib/libpcre.so.0 (0x0088c000)
        libpcreposix.so.0 => /usr/lib/libpcreposix.so.0 (0x0024d000)
        libaprutil-0.so.0 => /usr/lib/libaprutil-0.so.0 (0x00a78000)
```

```
        libldap-2.2.so.7 => /usr/lib/libldap-2.2.so.7 (0x009a9000)
        liblber-2.2.so.7 => /usr/lib/liblber-2.2.so.7 (0x006a0000)
...
```

7. Afficher les en-têtes.

```
# objdump -x /usr/sbin/httpd |more

/usr/sbin/httpd:     file format elf32-i386
/usr/sbin/httpd
architecture: i386, flags 0x00000150:
HAS_SYMS, DYNAMIC, D_PAGED
start address 0x0000f060

Program Header:
    PHDR off    0x00000034 vaddr 0x00000034 paddr 0x00000034 align 2**2
         filesz 0x00000100 memsz 0x00000100 flags r-x
  INTERP off    0x00000134 vaddr 0x00000134 paddr 0x00000134 align 2**0
         filesz 0x00000013 memsz 0x00000013 flags r--
    LOAD off    0x00000000 vaddr 0x00000000 paddr 0x00000000 align 2**12
         filesz 0x0003d6b4 memsz 0x0003d6b4 flags r-x
    LOAD off    0x0003d6b4 vaddr 0x0003e6b4 paddr 0x0003e6b4 align 2**12
         filesz 0x00001994 memsz 0x00004774 flags rw-
 DYNAMIC off    0x0003e160 vaddr 0x0003f160 paddr 0x0003f160 align 2**2
         filesz 0x00000130 memsz 0x00000130 flags rw-
    NOTE off    0x00000148 vaddr 0x00000148 paddr 0x00000148 align 2**2
         filesz 0x00000020 memsz 0x00000020 flags r--
EH_FRAME off    0x0003d63c vaddr 0x0003d63c paddr 0x0003d63c align 2**2
         filesz 0x0000001c memsz 0x0000001c flags r--
```

Tâche 3 : Diagnostiquer et réparer un système

1. Récupérer le logiciel breaksys et le compiler.

Le logiciel `breaksys` essaye de provoquer des pannes sur votre système. C'est à vous de le réparer. Vous pouvez le récupérer avec son source à partir du site Tsoft (cf. Avant-Propos). Le source contient les commandes qui ont provoqués le problème ainsi que les commandes qui permettent de le diagnostiquer et de réparer.

ATTENTION ! Ne pas exécuter ce programme sur une machine en exploitation.

```
[root@linux1 ~]# cc -o breaksys breaksys.c
[root@linux1 ~]# cc -o securise securise.c
```

2. Activer le logiciel et provoquer une panne.

```
[root@linux1 ~]# ./breaksys


                Nature des problemes

 1 - crontab
 2 - crontab
 3 - crontab
 4 - utilisateurs & droits
 5 - utilisateurs & droits
 6 - utilisateurs & droits
 7 - utilisateurs & droits
```

```
 8 - syslog
 9 - FS
10 - FS
11 - FS
12 - Demarrage
13 - Demarrage
14 - X window
15 - X window
16 - Impression
17 - Reseau

Votre choix (99:sortie) ? 1
============================================================


l'utilisateur guest ne peut pas creer de crontab

============================================================
[Le systeme sera endomage apres avoir frappe sur validation]
...Vous pouvez encore interrompre le programme par Ctrl-C!
^C
[root@linux1 ~]#
```

- *RAM, SWAP, CPU, IO*
- *vmstat, sar*
- */proc/**
- */var/log/sa/sa12*
- *ulimit –a, chattr +A*

13

Compléments

Objectifs

Ce chapitre traite de sujets divers qui complémentent les autres chapitres. En particulier tout ce qui concerne les particularités associées à un pays (langues, heure, clavier) et les tâches périodiques (rotation des journaux, anacron, …). On présente enfin les terminaux séries.

Contenu

L'environnement

La localisation

L'heure

Le clavier

Chkconfig

Logrotate

Les travaux périodiques

Anacron

Les terminaux et la liaison série

Ateliers



L'environnement

La théorie

Quelque soit l'application (une simple commande comme `ls` ou un service réseau comme Apache), elle possède un espace mémoire dit d'environnement. Cet espace est fait de variables d'environnement qui ont un nom et une valeur sous forme d'une chaîne de caractères, par exemple : `TAPE=/dev/st0`. Par convention, le nom d'une variable d'environnement est en majuscule.

La particularité de cet espace est d'être hérité de processus père en processus fils : une application reçoit en héritage une copie de l'environnement de l'application qui l'a activée (via les appels systèmes `fork`() et exec()).

Le shell permet facilement de modifier l'environnement.

Le savoir concret

Les commandes

`export`	Cette commande interne du shell met une variable dans l'environnement.
`env`	Liste les variables d'environnement et leur valeur.
`unset`	Cette commande interne du shell détruit une variable.

Quelques variables d'environnement

USER, LOGNAME	Le nom de l'utilisateur connecté.
HOME	Le répertoire de connexion.
LANG, LC_ALL, LC_*	Les variables de localisation.
PATH	Les chemins de répertoires où sont recherchés les commandes.
TERM	Le type du terminal.
SHELL	Le shell utilisé à la connexion.
TZ	Spécifie le décalage horaire. Permet de calculer l'heure locale.
PRINTER, LPDEST	L'imprimante par défaut.
LD_LIBRARY_PATH	Dans quels répertoires une application doit rechercher ses bibliothèques dynamiques.
DISPLAY	L'adresse du terminal X (doit être connu des clients X).

Initialisation de l'environnement

`init`	Le programme `init` fixe notamment la variable PATH.
`login`	Le programme `login` initialise une session. Le plus souvent il active un shell.
`pam_env.so`	Cette bibliothèque fixe l'environnement de plusieurs applications (comme login ou crond) par l'intermédiaire du fichier /etc/security/pam_env.conf.
`bash`	Le shell est le moyen le plus simple de changer l'environnement, soit en positionnant une variable dans la ligne de commande soit en modifiant l'environnement du shell grâce à la commande interne `export`.

Les fichiers de démarrage du shell Bash

En bref, pour un shell de connexion (cf. man du bash pour une vision plus complète) : Les scripts de l'administrateur /etc/profile et /etc/profile.d/* sont exécutés. Ensuite le script de l'utilisateur ~/.bash_profile est exécuté. Ce dernier active ~/.bashrc qui active lui-même /etc/bashrc. Le script $ENV est également exécuté. Si le fichier ~/.bash_profile n'existe pas, c'est le fichier ~/.profile qui est exécuté à sa place.

Remarque : Le répertoire /etc/skel/ contient les modèles des fichiers de démarrage : ceux-ci sont recopiés dans le répertoire de connexion d'un nouvel utilisateur.

Pour en savoir plus

Les pages de manuel

bash(1), environ(5), login(1), exec(P), exec(3), execve(2), env(1), init(8), useradd(8)

Remarque : Ce sont les applications qui exploitent les variables d'environnement. C'est donc dans leurs pages de manuel qu'est spécifiée leur signification. Par exemple, la variable TAPE est décrite dans le paragraphe ENVIRONMENT du manuel de la commande dump.

dump(8), man(1), bash(1), ssh(1), …

Internet

Wikipedia – Environment variable
http://en.wikipedia.org/wiki/Environment_variable

La localisation

La théorie

La localisation correspond à la configuration des applications pour qu'elles tiennent compte de la langue de l'utilisateur. Cette localisation passe principalement par le paramétrage de variables d'environnement.

La valeur d'une variable de localisation

Une variable de localisation a souvent la structure `ab_CD[.norme][@variante]` où ab désigne la langue (spécifié par la norme ISO-639) et CD désigne le pays de l'utilisateur (spécifié par la norme ISO-3166).

Exemples simples : fr_FR, fr_CH, de_DE, de_CH, pt_BR respectivement le français de France, le français de Suisse, l'allemand d'Allemagne, l'allemand de Suisse et le portugais du Brésil.

Exemple complexe : fr_FR.ISO-8859-15@euro.

La valeur « C » ou POSIX correspond à un paramétrage POSIX ou ISO (dit aussi Langage C). La langue associée est l'anglais. Son utilisation garantie une totale portabilité de l'application.

L'unicode

L'Unicode est un standard qui permet de représenter chaque caractère et chaque symbole des différents langages qui existent de par le monde. Les codes actuellement disponibles s'expriment sur 16 bits et les 256 premières entrées sont identiques à la norme ISO-8859 (latin-1).

L'Unicode permet dans un même document de mélanger les codes provenant de différentes langues.

Il existe l'Unicode UTF-8 et UTF-16 (UTF=Unicode Transfer Format). L'UTF-16 s'exprime sur 16 bits. L'UTF-8 lui, est composé d'octets (8 bits). Ainsi les 127 premiers codes correspondent à l'ASCII. Pour exprimer les caractères non ASCII, on doit utiliser plusieurs octets. Les premiers bits du premier octet indiquent sur combien d'octets est défini le caractère.

Evidement, l'Unicode est adapté à l'environnement graphique.

Le savoir concret

Les variables d'environnement

LANG	Définit par défaut toutes les variables de localisation.
LC_ALL	Idem.

Il existe des variables spécifiques POSIX, globalement référencées par LC_* :

LC_COLLATE	Définit l'ordre alphabétique pour les chaînes de caractères.
LC_CTYPE	Définit les types de caractères : les lettres, les numériques, … Définit également le jeu de caractères.
LC_MESSAGES	Définit la localisation des messages du logiciel.
LC_MONETARY	Définit le symbole monétaire.
LC_TIME	Définit le format des dates et heures.
LC_NUMERIC	Définit le symbole décimal et le séparateur des milliers.

Il existe aussi d'autres variables non-Posix :

LANGUAGE	Liste de langues séparées par deux-points (:).
LINGUAS	Liste de langues séparées par des espaces. Utilisée par KDE et OpenOffice.

Remarque : Si l'on définit à la fois l'ensemble de ces variables, LC_ALL l'emporte sur les variables LC_*. LC_* et LC_ALL l'emportent sur la variable LANG.

Les commandes

`locale`	Affiche les locales et leur valeur.
`locale -a`	Affiche les choix possibles.

Particularité des distributions

RedHat

/etc/sysconfig/i18n	Ce fichier contient la définition de la variable LANG.
system-config-language	Modifie la langue du système.

SuSE

/etc/sysconfig/language	Ce fichier contient la définition de la variable LANG.

Debian

On définit les paramètres régionaux et la variable LANG grâce à la commande `dpkg-reconfigure locales`.

Le fichier /etc/environment contient la définition de la variable LANG.

Pour en savoir plus

Les pages de manuel

bash(1), locale(1), setlocale(1), locale(5), locale(7), localdef(1), ascii(7), utf-8(7), unicode(7), loadunimap(8), unicode_start(1), unicode_stop(1), psfxtable(1)

Howto

German-Howto, Hebrew-Howto, Helenic-Howto, Turkish-Howto, Spanish-Howto, Thai-Howto, Helenic-Howto, Italian-Howto, Serbian-Howto, Portuguese-Howto.

Unicode-Howto

Internet

Francophones-Howto
http://www.linux.com/howtos/Francophones-HOWTO.shtml

Utiliser et configurer Debian GNU/Linux pour le français
http://www.debian.org/doc/manuals/fr/debian-fr-howto/index.html

SUSE : COMMENT RESOUDRE LES PROBLEMES D'ACCENTS FRANÇAIS
http://frenchsuse.free.fr/aide-suse_accents.html

L'heure

La théorie

Horloge interne et heure système

Un ordinateur possède une horloge interne (un périphérique) connectée à une batterie qui lui permet de rester en fonction même quand l'ordinateur est arrêté.

Le système Linux, par l'intermédiaire du noyau possède une horloge qui sert de base de temps pour toutes les opérations effectuées : les dates de mise à jour des fichiers, les dates des actions consignées dans les journaux de bords, … L'horloge du noyau, qui donne l'heure système est initialisée par rapport à l'heure de l'horloge interne au démarrage. Il faut ensuite que ces horloges restent synchronisées.

Des ordinateurs en réseau

Des que l'on utilise des applications client/serveur sensibles (liées à la sécurité, gérant des bases de données, …), il faut impérativement que les horloges systèmes soient synchronisées. Le protocole standard NTP est fait pour cela.

Il existe sur Internet plusieurs serveurs de temps accessible librement qui donne l'heure exacte (l'heure atomique). Ils récupèrent par radio l'heure officielle.

Heure locale et heure GMT

Le noyau Linux utilise en interne l'heure universelle appelée UTC ou GMT (Universal Time Coordinated, Greenwich Mean Time). Ainsi les dates de modifications ou d'accès à un fichier sont mémorisées en temps universel. Mais pour le bien être des utilisateurs, les heures sont affichées en heure locale, qui tient compte du décalage horaire.

Le savoir concret

Les commandes

`hwclock`	Récupère ou modifie l'heure de l'horloge interne. Permet également de mettre à jour l'heure système ou inversement de se synchroniser sur l'heure système.
`date`	Affiche la date et heure système.
`ntpdate`	Synchronise l'heure système par rapport à un serveur NTP. Cette action est typiquement réalisée au démarrage.
`ntpd`	Démon NTP, il sert à garder synchronisé des ordinateurs. Il peut être client et/ou serveur.

Les fichiers

/etc/ntpd.conf	Fichier de configuration du démon ntpd.
/etc/ntp.drift	Fichier « driftfile » contenant les données d'ajustement de l'heure (les dérives des horloges). Le nom réel du fichier est indiqué dans /etc/ntpd.conf.
/etc/localtime	Indique le décalage horaire par défaut. Lien sur un fichier de l'arborescence /usr/share/zoneinfo. La variable TZ est prioritaire.
/usr/share/zoneinfo	La base de données contenant les données de décalage horaire.

L'environnement

TZ	Cette variable mémorise soit les règles permettant le calcul de l'heure locale soit une clé donnant accès à une base de données renfermant ces règles.

Protocoles

123/udp	NTP (Network Time Protocol).

Focus : La variable d'environnement TZ

La variable d'environnement TZ permet à une application de calculer l'heure locale. Elle a deux syntaxes :

La syntaxe Posix

On a d'abord un code qui indique la zone géographique ou le pays et ensuite le décalage horaire. Ce dernier est positif vers l'ouest et négatif vers l'est. On peut avoir ensuite des expressions qui permettent le calcul de l'heure d'été. Exemples :

AST4	Décalage horaire au Québec (-4h, AST=Atlantic Standard Time).
JST-9	Décalage horaire du Japon (+9h, JST=Japan Standard Time).

La syntaxe moderne

On a tout simplement un continent suivi d'une ville. Les bibliothèques de gestion du temps vont permettre d'accéder à une base de données stockées dans l'arborescence /usr/share/zoneinfo/. Exemples :

Europe/Paris	Décalage pour la France, L'heure d'été est prise en compte.
America/Martinique	Décalage horaire à la Martinique.

Les particularités des distributions

Redhat

system-config-time ou system-config-date Configure l'heure.

/etc/sysconfig/ntpd	Options de configurations du service ntpd.

Pour en savoir plus

Les pages de manuel

hwclock(8), date(1), ntpdate(1), ntpd(1), timeconfig(8), tzselect(1), tzfile(5)

Howto

TimePrecision-HOWTO

Clock

Internet

Wikipedia – Time Zone
http://en.wikipedia.org/wiki/Time_zone

Abréviation (codes) des fuseaux horaires
http://docs.postgresqlfr.org/pgsql-7.4.15-fr/datetime-keywords.html

Wikipedia – NTP
http://fr.wikipedia.org/wiki/Network_Time_Protocol

Le clavier

La théorie

Quand on presse ou quand on relâche une touche du clavier, le périphérique envoie un scancode au pilote du noyau qui gère le clavier.

Quand on est en mode graphique, c'est directement le scancode qui est envoyé à l'application cliente X.

En mode texte, sur la console maîtresse, le scancode est traduit en keycode (on a un seul keycode quand on appuie et que l'on relâche une touche). Les kermaps traduisent les scancodes en keycodes. Ils dépendent du pays (clavier AZERTY, QWERTY, ...).

Le savoir concret

Les commandes

`loadkeys`	Charge un fichier keymap.
`dumpkeys`	Affiche la keymap courante.
`stty`	Affiche ou modifie l'interprétation de caractères spéciaux (backspace, ...).
`xmodmap`	Configure le clavier du terminal X à partir d'un fichier spécifiant la traduction des keycodes en keysyms.

Les fichiers

/etc/X11/xorg.conf	Le fichier de configuration du serveur X (clavier, souris, ...).
/lib/kbd/keymaps/i386/	Répertoire contenant les keymaps pour PC.

Focus : configurer le retour arrière (backspace)

```
# stty   erase   ^H      # ou ^?
```

Focus : modifier son environnement graphique

1) On enregistre la traduction actuelle keycode/keysym.

```
# xmodmap -pke > ~/.Xmodmap
```

2) On la modifie.

```
# vi ~/.Xmodmap
```

3) On modifie son fichier .xinitrc.

```
# vi ~/.xinitrc
...
xmodmap ~/.Xmodmap
```

Les particularités des distributions

RedHat

Le clavier est initialisé par /etc/rc.sysinit au démarrage. Il utilise un des fichiers suivants : /etc/sysconfig/console/default.kmap, /etc/sysconfig/keyboard.

system-config-keyboard	Configure le clavier.

SuSE

/etc/sysconfig/keyboad	Configure le clavier.

Debian

Le clavier est initialisé par /etc/init.d/keymap.sh au démarrage. Il charge le fichier suivant : /etc/console/boottime.kmap.gz.

dpkg-reconfigure console-data Configure le clavier.

Pour en savoir plus

Les pages de manuel

stty(1), loadkeys(1), dumpkeys(1), keymaps(5), showkey(1), xmodmap(1), X(7x)

Howto

BackspaceDelete

Internet

Utiliser et configurer Debian pour le français - Chapitre 4 - Configurer l'entrée …
http://www.debian.org/doc/manuals/fr/debian-fr-howto/ch4.html

Chkconfig

La théorie

La commande `chkconfig` est une solution simple pour paramétrer le démarrage des services (appelés scripts RC=Run Command). Normalement les services sont activés ou arrêtés en fonction de l'existence de scripts commençant par S (start) ou K (kill) dans les répertoires /etc/rc#.d/ où # indique le niveau de démarrage (niveau d'init).

Normalement ces scripts sont des liens vers des fichiers présents dans le répertoire /etc/init.d/. Si l'administrateur veut organiser différemment le système : désactiver ou activer un service, il doit créer ou supprimer des liens. La commande `chkconfig` réalise ces opérations à sa place.

La commande `chkconfig` est utilisée en standard dans les distributions RedHat et SuSE.

Remarque : La commande `chkconfig` gère non seulement les RC mais également le démarrage des services contrôlés par Xinetd.

Le savoir concret

Syntaxe de la commande chkconfig

a) Lister l'état des services ou d'un service

```
chkconfig --list [service]
```

b) Prendre en compte un nouveau service (il doit posséder l'entête had-hoc et être dans /etc/init.d)

```
chkconfig --add service
```

c) Ne plus prendre en compte un service

```
chkconfig --del service
```

d) Activer ou désactiver un service pour le prochain démarrage

```
chkconfig [--level niveaux] service <on|off|reset>
```

e) Lister l'état des services pour un ou plusieurs niveau

```
chkconfig [--level niveaux] service
```

Un script RC géré par chkconfig

```
# more  /etc/init.d/un_service
#!/bin/sh
# chkconfig: -  60 20
# description: ce service permet …
```

La ligne « chkconfig » (il y a un blanc devant chkconfig) indique les niveaux pour lesquels le service est actif («-» signifie les niveaux normaux : 2345) suivit de l'ordre pour le démarrage et le niveau pour l'arrêt.

Pour en savoir plus

Les pages de manuel

chkconfig(8)

Logrotate

La théorie

Le logiciel `logrotate` effectue la rotation des journaux de bords. Il offre également la possibilité de compresser les archives de ces journaux et de supprimer les anciennes archives ou de les transférer par courrier électronique.

Chaque journal peut être archivé périodiquement (chaque jour, chaque semaine ou chaque mois) ou quand il atteint une certaine taille.

Le savoir concret

Commande et fichier

`logrotate`	La commande qui réalise la rotation des journaux. Elle est typiquement activée journellement par un crontab. On donne en argument de la commande le chemin de son fichier de configuration.
/etc/logrotate.conf	Le fichier de configuration de la commande `logrotate`. Précise les journaux à gérer et comment les gérer.

Les principales directives du fichier de configuration

`daily`	La rotation des journaux est effectuée journellement.
`weekly`	La rotation des journaux est effectuée de manière hebdomadaire.
`montly`	La rotation des journaux est effectuée mensuellement.
`size=t`	La rotation est effectuée quand le fichier atteint la taille t. Exemples : 100K, 10M.
`compress`	On compresse les archives.
`nocompress`	On ne compresse pas les archives.
`create m u g`	Juste après l'archivage (rotation), le fichier journal est recrée vide. On indique ses droits (m), son propriétaire (u) et son groupe (g).
`rotate n`	On conserve n archives du journal (dernière, avant-dernière, ...).
`include c`	On charge des fichiers de configuration (un fichier ou les fichiers d'un répertoire). On indique son chemin (c).
`postrotate`	Débute un script qui est exécuté après la rotation.
`prerotate`	Débute un script qui est exécuté avant la rotation.
`endscript`	Termine un script `prerotate` ou `postrotate`.
`sharedscripts`	Les scripts (pre- ou post-) sont exécutés une seule fois. Cette directive est utilisée dans le cas où l'on gère plusieurs journaux.
`mail a`	Au lieu de détruire l'archive elle est envoyée à l'adresse email indiquée (a).
`missingok`	Ne génère pas d'erreur dans la cas où le journal n'existe pas.
`notifempty`	N'archive pas le journal si il est vide.

Focus : un exemple de fichier de configuration

```
# more /etc/logrotate.conf
```

```
weekly
rotate 4
compress

include /etc/logrotate.d

/var/log/wtmp {
    montly
    nocompress
    create 0664 root utmp
    rotate 1
}

/var/log/messages /var/log/mailog {
    postrotate
        /bin/kill -HUP `cat /var/run/syslogd.conf`
    endscript
}
```

Les directives en début de fichier spécifient les valeurs par défaut. Ensuite, on a une section par groupe de journaux gérés de la même manière. Une section débute par la liste de journaux (on peut utiliser les jockers) qui est suivie d'un bloc de directives délimité par des accolades ({ … }). Les directives du bloc ne s'appliquent qu'aux fichiers qui précédent le bloc, elles invalident les directives globales. La directive `include` charge tous les fichiers de rotation de journaux présents dans le répertoire /etc/logrotate.d.

Remarque : Le répertoire /ec/logrotate.d/ est utilisé par les applications packagées pour y déposer leur stratégie de rotation de journaux.

Pour en savoir plus

Les pages de manuel

logrotate(8)

Les travaux périodiques

La théorie

Sous Linux, il existe plusieurs commandes qui permettent l'exécution de travaux périodiques ou différés.

Le savoir concret

Les commandes

`crontab`	Demande de travaux périodiques.
`at`	Demande de travaux différés.
`batch`	Demande de travaux exécutés à faible charge.
`cron` ou `crond`	Démon gérant le service CRON.
`atd`	Démon gérant les services AT et BATCH.
`anacron`	Exécute des tâches périodiques sur un système fréquemment arrêté.
`fcron`	Logiciel libre qui prétend vouloir remplacer CRON et anacron.

Les fichiers

/etc/cron.allow, /etc/cron.deny	Autorisent ou interdisent l'accès au service CRON.
/etc/at.allow, /etc/at.deny	Autorisent ou interdisent l'accès au service AT.
/etc/crontab	Crontab géré par l'administrateur mais dont l'uid sous lequel s'exécute les commandes peut être paramétré.
/etc/cron.d/	Répertoire contenant les crontabs provenant des packages ;
/etc/anacrontab	Spécifie les travaux périodiques gérés par anacron.

Focus : La commande at

Soumettre un travail en différé

```
# at  1015  < un_script.sh
```

Lister les travaux en attente

```
# at -l        # ou atq
```

Détruire un travail

```
# at -r   1    # ou atrm
```

Pour en savoir plus

Les pages de manuel

cron(8), atd(8), crontab(1), at(1), anacron(8)

Internet

fcron – La page officiel
http://fcron.free.fr/

Anacron

La théorie

La commande `anacron` exécute périodiquement des tâches. Mais à la différence de `CRON`, elle ne considère pas que le système fonctionne en permanence. Elle est donc tout à fait adaptée pour automatiser des tâches périodiques d'un poste de travail.

Pour mémoriser le fait qu'il a (ou que CRON a) déjà accompli un travail (job), Anacron crée un fichier repère (timestamp).

Le savoir concret

Les commandes

`anacron -u job`	Met à jour les fichiers repères. Ne prend en compte que les travaux dont le nom est donné en argument. Typiquement cette commande est activée journellement par un crontab. Les travaux n'ont pas besoin d'être exécuté, crontab s'en charge.
`anacron -s "*"`	Active tous les travaux les uns après les autres (ils sont sérialisés). Typiquement cette commande est lancée par un script RC au démarrage.

Les fichiers

/etc/anacrontab	Spécifie les travaux périodiques.
/var/spool/anacron/	Répertoire qui contient les fichiers repères.
/var/log/cron	Journal d'Anacron (il journalise dans la facilité cron de Syslog).

Focus : Le fichier de configuration

Structure

Chaque ligne du fichier correspond à un travail périodique.

```
n_jours      x_minutes      identifiant      commande
```

Pour chaque travail (identifié par le 3ième champ), `anacron` vérifie s'il a été déjà exécuté (grâce aux fichiers repères) dans les n derniers jours (premier champ). Sinon `anacron` exécute la commande associée (dernier champ) après avoir attendu x minutes (deuxième champ). Après son exécution, `anacron` enregistre la date dans un fichier repère afin de ne pas l'exécuter de nouveau.

Exemple

```
# more /etc/anacrontab
SHELL=/bin/sh
PATH=/usr/local/sbin:/usr/local/bin:/sbin:/bin:/usr/sbin:/usr/bin
MAILTO=root
1       65      cron.daily          run-parts /etc/cron.daily
7       70      cron.weekly         run-parts /etc/cron.weekly
30      75      cron.monthly        run-parts /etc/cron.monthly
```

Pour en savoir plus

Les pages de manuel

anacron(8), anacrontab(5)

Les terminaux et la liaison série

La théorie

Liaison Rs-232

La liaison série est sans doute le moyen le plus ancien pour faire communiquer deux ordinateurs. Même si de nos jours l'USB est privilégié, la liaison série reste encore utilisée.

La liaison série est normalisée sous l'appellation RS-232. Ce protocole réalise le dialogue entre deux ordinateurs (DTE=Data Terminal Equipment) sur une ligne téléphonique par l'intermédiaire de couple de modem (DCE=Data Communication Equipment). Des protocoles (Avis V32, V90, ...) règlent les échanges entre modems.

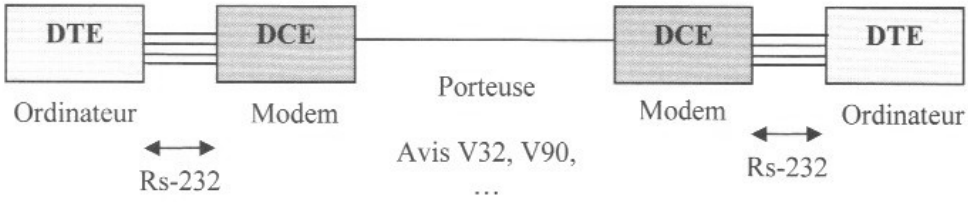


Fig. Rs-232

Les connecteurs sont 25 broches ou 9 broches. Maintenant on utilise le plus souvent des connecteurs téléphoniques (RJ-45). Le câble de liaison DTE/DCE est un câble plat. Si l'on utilise un câble croisé, on peut relier directement les deux ordinateurs sans modem (câble NULL-Modem). Les codes Hayes permettent de donner des ordres au modem (numéroter : ATDT0102030405, ou raccrocher : ATH, ...).

Utilisations de la liaison série

- Couche physique réseau. Associé au protocole PPP, la liaison Rs-232 supporte TCP/IP.
- Support de la connexion (login).
- Couche physique des protocoles Kermit et UUCP.
- Réception de FAX

La gestion plein écran des terminaux texte

Une application texte plein écran comme `vi` envoie des codes de contrôle à l'écran pour déplacer le curseur (Escape-sequence). Ces codes dépendent de la nature du terminal. Une application utilise au choix une des deux bases TERMCAP ou TERMINFO pour obtenir les bonnes séquences. Ces bases sont indexées selon le type du terminal.

Le savoir concret

Les commandes

`stty`	Paramètre la liaison série (vitesse, parité, bits stop, ...).
`mgetty`	Configure une liaison série pour accepter des connexions (login) ou pour recevoir des Fax.
`minicom`	Logiciel de gestion d'un terminal. Grâce à lui on peut se connecter à un système distant Unix/Linux.
`tset`	Fixe le type de terminale (TERM).

Les fichiers

/dev/ttyS0	Le port COM1:
/dev/ttyS1	Le port COM2:
/etc/inittab	Active les premières applications, par exemple les gestionnaires de connexions sur liaison série (`getty`, `agetty`, `mgetty`).
/etc/gettydefs	Contient des informations de paramétrages des liaisons utilisées par `getty` ou `mgetty`.
/etc/termcap	Base de données TERMCAP.
/usr/share/terminfo/*	Base de données TERMINFO.

Focus : accepter des connexions

```
# grep  mgetty /etc/inittab
a2:345:respawn:/sbin/mgetty  -rs  38400  /dev/ttyS0
```

Les variables d'environnement

TERM	Le type de terminal (vt100, xterm, ...)

Pour en savoir plus

Les pages de manuel

stty(1), setserial(8), getty(1), mgetty(8), ttys(4), terminfo(5), tset(1), sendfax(8), gettydefs(5), mgettydefs(5), termios(3), termcap(5), terminfo(5)

Howto

Remote-Serial-Console-HOWTO

Serial-HOWTO

Serial-Laplink-HOWTO

Text-Terminal-HOWTO

Fax-Server

Internet

Wikipedia – Rs-232, modem
http://en.wikipedia.org/wiki/RS232
http://fr.wikipedia.org/wiki/Modem

Linux Serial Console HOWTO
http://www.vanemery.com/Linux/Serial/serial-console.html

Wikipedia – UUCP
http://en.wikipedia.org/wiki/UUCP

Kermit
http://www.columbia.edu/kermit/

Livre

PPP, configuration et mise en œuvre, par A. Sun, chez O'Reilly

ATELIERS



Tâche 1 : L'environnement 5mn
Tâche 2 : La localisation 5mn
Tâche 3 : L'heure 15mn
Tâche 4 : Le clavier 5mn
Tâche 5 : Chkconfig 5mn
Tâche 6 : Logrotate 10mn
Tâche 7 : At et batch 5mn
Tâche 8 : Anacron 10mn

Tâche 1 : L'environnement

1. Visualiser l'environnement du shell courant.

```
[root@linux1 ~]# env
...
HOSTNAME=linux1.pinguins
TERM=xterm
SHELL=/bin/bash
HISTSIZE=1000
USER=root
MAIL=/var/spool/mail/root
PATH=/usr/kerberos/sbin:/usr/kerberos/bin:/usr/local/sbin:/usr/local/bin:/sbin:/
bin:/usr/sbin:/usr/bin:/usr/X11R6/bin:/root/bin
INPUTRC=/etc/inputrc
PWD=/root
LANG=C
HOME=/root
LOGNAME=root
_=/bin/env
[root@linux1 ~]#
```

2. Afficher la valeur d'une variable d'environnement.

```
[root@linux1 ~]# env | grep PATH
PATH=/usr/kerberos/sbin:/usr/kerberos/bin:/usr/local/sbin:/usr/local/bin:/sbin:/
bin:/usr/sbin:/usr/bin:/usr/X11R6/bin:/root/bin
[root@linux1 ~]#
```

3. Afficher la signification des variables d'environnement gérées par une application.

```
[root@linux1 ~]# man dump |col -b | sed  -n '/^ENVIRONMENT/,+3 p'
ENVIRONMENT
       TAPE   If  no -f option was specified, dump will use the device speci-
              fied via TAPE as the dump device.  TAPE  may  be  of  the  form
              tapename, host:tapename, or user@host:tapename.
[root@linux1 ~]#
```

4. Activer une application en précisant l'environnement.

a) On modifie l'environnement uniquement pour la commande.

```
[root@linux1 ~]# TAPE=/root/etc.tar.gz   tar cz /etc
tar: Removing leading `/' from member names
[root@linux1 ~]#
```

b) On modifie l'environnement du shell.

```
[root@linux1 ~]# export TAPE=/root/archive
[root@linux1 ~]# env | grep TAPE
TAPE=/root/archive
[root@linux1 ~]# tar c /etc
tar: Removing leading `/' from member names
[root@linux1 ~]#
```

Tâche 2 : La localisation

1. Visualiser la configuration de la langue pour le système.

```
[root@linux1 ~]# cat /etc/sysconfig/i18n
LANG="fr_FR.UTF-8"
SUPPORTED="en_US.UTF-8:en_US:en:fr_FR.UTF-8:fr_FR:fr"
SYSFONT="latarcyrheb-sun16"
[root@linux1 ~]#
```

2. Utiliser le français.

a) On utilise la variable LANG définie à l'installation.

```
[root@linux1 ~]# echo $LANG
fr_FR.UTF-8
[root@linux1 ~]# man cal | col -b | head -20
CAL(1)                  Manuel de lâutilisateur Linux                  CAL(1)



NOM
       cal - Afficher un calendrier.

SYNOPSIS
       cal [-jy] [[mois] [annee]]

DESCRIPTION
       Cal  affiche  un  calendrier.  Si  les  arguments  sont omis, le mois
       courant est affichÃ©.  Les options sont les suivantes :


       -j     Affiche les numÃ©ros des jours dans lâannÃ©e (le numÃ©ro 1  corre-
              spond au 1er Janvier).

       -y     Affiche le calendrier de lâannÃ©e courante.
```

b) On essaye de régler le problème des accents.

```
[root@linux1 ~]# export LANG=fr_FR@euro
[root@linux1 ~]# man cal | col -b | head -20
CAL(1)                  Manuel de l'utilisateur Linux                  CAL(1)



NOM
       cal - Afficher un calendrier.
```

```
SYNOPSIS
       cal [-jy] [[mois] [annee]]

DESCRIPTION
       Cal affiche un calendrier. Si les arguments sont omis, le mois
       courant est affiché. Les options sont les suivantes :


       -j     Affiche les numéros des jours dans l'année (le numéro 1 corre-
              spond au 1er Janvier).

       -y     Affiche le calendrier de l'année courante.

[root@linux1 ~]#
```

Remarques :

1) Si l'on se connecte au système à partir de Windows avec le programme `telnet` dans une session cmd, les accents seront mal interprétés de toute façon car le logiciel cmd ne respecte pas les standards. Vous pouvez utiliser en lieu et place PUTTY.EXE qui permet d'utiliser les protocoles SSH, RLOGIN et TELNET.

2) En environnement graphique, on a le contraire de l'exemple (on était connecté au système en mode texte). Il faut utiliser la valeur fr_FR.UTF-8 pour avoir les accents.

3. Lister les valeurs supportées pour le français.

```
[root@linux1 ~]# locale -a | grep fr
français
fr_BE
fr_BE@euro
fr_BE.iso88591
fr_BE.iso885915@euro
fr_BE.utf8
fr_CA
fr_CA.iso88591
fr_CA.utf8
fr_CH
fr_CH.iso88591
fr_CH.utf8
french
fr_FR
fr_FR@euro
fr_FR.iso88591
fr_FR.iso885915@euro
fr_FR.utf8
fr_LU
fr_LU@euro
fr_LU.iso88591
fr_LU.iso885915@euro
fr_LU.utf8
[root@linux1 ~]#
```

4. Afficher la valeur des locales

```
[root@linux1 ~]# locale
LANG=fr_FR@euro
LC_CTYPE="fr_FR@euro"
```

```
LC_NUMERIC="fr_FR@euro"
LC_TIME="fr_FR@euro"
LC_COLLATE="fr_FR@euro"
LC_MONETARY="fr_FR@euro"
LC_MESSAGES="fr_FR@euro"
LC_PAPER="fr_FR@euro"
LC_NAME="fr_FR@euro"
LC_ADDRESS="fr_FR@euro"
LC_TELEPHONE="fr_FR@euro"
LC_MEASUREMENT="fr_FR@euro"
LC_IDENTIFICATION="fr_FR@euro"
LC_ALL=
[root@linux1 ~]#
```

5. Jouer avec les locales.

a) On vérifie que la variable LC_ALL est prioritaire et que LC_* est prioritaire sur LANG.

```
[root@linux1 ~]# export LANG=fr_FR
[root@linux1 ~]# export LC_ALL=en_US
[root@linux1 ~]# export LC_TIME=es_ES
[root@linux1 ~]# date
Thu Mar  1 16:30:09 CET 2007
[root@linux1 ~]# unset LC_ALL
[root@linux1 ~]# date
jue mar  1 16:30:30 CET 2007
[root@linux1 ~]# unset LC_TIME
[root@linux1 ~]# date
jeu mar  1 16:31:01 CET 2007
```

b) On essaye quelques langues.

```
[root@linux1 ~]# LANG=fr_FR  date +%A
jeudi
[root@linux1 ~]# LANG=en_US  date +%A
Thursday
[root@linux1 ~]# LANG=de_DE  date +%A
Donnerstag
[root@linux1 ~]# LANG=it_IT  date +%A
giovedì
[root@linux1 ~]# LANG=es_ES  date +%A
jueves
[root@linux1 ~]# LANG=POSIX  date +%A
Thursday
[root@linux1 ~]# LANG=C  date +%A
Thursday
```

6. Installer les pages de manuel en allemand.

a) Quelles sont les langues actuellement supportées ?

```
[root@linux1 ~]# rpm -qa |grep man-pages
man-pages-fr-0.9.7-11
man-pages-1.67-9.EL4
```

b) On installe les pages de manuel en allemand.

```
[root@linux1 ~]# yum -y install man-pages-de
...
```

c) On teste.

```
[root@linux1 ~]# LANG=de_DE man ls | col -b | grep -v '^$' | head
VDIR(1)                                FSF                                VDIR(1)
NAME
       ls - zeigt Verzeichnisinhalt an
ÜBERSICHT
       ls [OPTION]... [DATEI]...
BESCHREIBUNG
       Auflistung  von  Informationen  der  DATEIen  (Standardvorgabe ist das
       momentane Verzeichnis). Alphabetisches Sortieren der  Einträge,  falls
       weder -cftuSUX noch --sort angegeben.
       -a, --all
[root@linux1 ~]#
```

Tâche 3 : L'heure

1. L'horloge interne et l'heure système.

a) On affiche l'heure de l'horloge interne et l'heure système.

```
[root@linux1 ~]# hwclock --show ; date
Mon Feb 26 21:53:44 2007  -0.955696 seconds
Mon Feb 26 21:49:31 CET 2007
```

b) On synchronise l'heure de l'horloge interne par rapport à l'heure système.

```
[root@linux1 ~]# hwclock --systohc
[root@linux1 ~]# hwclock --show ; date
Mon Feb 26 21:50:19 2007  -0.997352 seconds
Mon Feb 26 21:50:18 CET 2007
```

c) On fixe l'heure système : on la fait avancer d'environ 5 mn. Dans l'exemple : 26 Fev à 21h55.

```
[root@linux1 ~]# date 02262155
Mon Feb 26 21:55:00 CET 2007
```

d) On fixe l'heure système par rapport à l'heure de l'horloge interne.

```
[root@linux1 ~]# hwclock --hctosys
[root@linux1 ~]# hwclock --show ; date
Mon Feb 26 21:54:01 2007  -0.124711 seconds
Mon Feb 26 21:54:01 CET 2007
[root@linux1 ~]#
```

2. NTP.

a) On configure le serveur.

```
[root@instructeur ~]# rpm -q ntp
ntp-4.2.0.a.20040617-4.EL4.1
[root@linux1 ~]# /etc/init.d/ntpd stop
ArrÃªt de ntpd :                                            [ÃCHOUÃ]
[root@linux1 ~]# mv /etc/ntp.conf /etc/ntp.conf.000
```

Remarque : La configuration par défaut contient les adresses de deux serveurs de temps d'Internet : 0.pool.ntp.org et 1.pool.ntp.org.

```
[root@instructeur ~]# vi /etc/ntp.conf
server 127.127.1.1

driftfile /var/lib/ntp/drift
logfile /var/log/ntp.log
```

```
[root@instructeur ~]# touch /var/log/ntp.log
[root@instructeur ~]# chown ntp /var/log/ntp.log
[root@instructeur ~]# chkconfig ntpd on
[root@instructeur ~]# /etc/init.d/ntpd start
DÃ©marrage de ntpd :                                        [  OK  ]
[root@instructeur ~]# ps -e |grep ntp
12610 ?        00:00:00 ntpd
[root@instructeur ~]# netstat -an |grep ':123'
udp        0      0 192.168.218.99:123          0.0.0.0:*
udp        0      0 192.168.0.1:123             0.0.0.0:*
udp        0      0 127.0.0.1:123               0.0.0.0:*
udp        0      0 0.0.0.0:123                 0.0.0.0:*
udp        0      0 :::123                      :::*
[root@instructeur ~]# tail /var/log/messages
Mar  1 19:06:00 instructeur ntpd[12610]: ntpd 4.2.0a@1.1190-r Sun Aug 13
01:49:12 CDT 2006 (1)
Mar  1 19:06:00 instructeur ntpd: DÃ©marrage de ntpd succeeded
Mar  1 19:06:00 instructeur ntpd[12610]: precision = 2.000 usec
Mar  1 19:06:00 instructeur ntpd[12610]: Listening on interface wildcard,
0.0.0.0#123
Mar  1 19:06:00 instructeur ntpd[12610]: Listening on interface wildcard, ::#123
Mar  1 19:06:00 instructeur ntpd[12610]: Listening on interface lo,
127.0.0.1#123
Mar  1 19:06:00 instructeur ntpd[12610]: Listening on interface eth0,
192.168.0.1#123
Mar  1 19:06:00 instructeur ntpd[12610]: Listening on interface eth0:1,
192.168.218.99#123
Mar  1 19:06:00 instructeur ntpd[12610]: kernel time sync status 0040
Mar  1 19:06:00 instructeur ntpd[12610]: frequency initialized 0.000 PPM from
/var/lib/ntp/drift
[root@instructeur ~]# tail /var/log/ntp.log
 1 Mar 19:18:15 ntpd[2268]: synchronized to LOCAL(1), stratum 5
 1 Mar 19:18:15 ntpd[2268]: kernel time sync disabled 0041
 1 Mar 19:19:21 ntpd[2268]: kernel time sync enabled 0001
```

b) On configure le client.

```
[root@linux1 ~]# ssh 192.168.0.200 date ; date # heure distante et heure locale
root@192.168.0.200's password: seeret
Thu Mar  1 19:19:36 CET 2007
Thu Mar  1 19:20:48 CET 2007
[root@linux1 ~]# /etc/init.d/ntpd stop
ArrÃªt de ntpd :                                           [ÃCHOUÃ]
[root@linux1 ~]# mv /etc/ntp.conf /etc/ntp.conf.000
[root@linux1 ~]# vi /etc/ntp.conf
server 192.168.0.200

driftfile /var/lib/ntp/drift
logfile /var/log/ntp.log

[root@linux1 ~]# touch /var/log/ntp.log
[root@linux1 ~]# chown ntp /var/log/ntp.log

[root@linux1 ~]#
```

c) On essaye de combler l'écart de temps entre le client et le serveur.

```
[root@linux1 ~]# ntpd -q -g -d
ntpd 4.2.0a@1.1190-r Sun Aug 13 01:49:12 CDT 2006 (1)
addto_syslog: ntpd 4.2.0a@1.1190-r Sun Aug 13 01:49:12 CDT 2006 (1)
addto_syslog: precision = 2.000 usec
create_sockets(123)
bind() fd 4, family 2, port 123, addr 0.0.0.0, flags=8
addto_syslog: Listening on interface wildcard, 0.0.0.0#123
bind() fd 5, family 10, port 123, addr ::, flags=0
addto_syslog: Listening on interface wildcard, ::#123
bind() fd 6, family 2, port 123, addr 127.0.0.1, flags=0
addto_syslog: Listening on interface lo, 127.0.0.1#123
bind() fd 7, family 2, port 123, addr 192.168.0.1, flags=8
addto_syslog: Listening on interface eth0, 192.168.0.1#123
bind() fd 8, family 2, port 123, addr 192.168.218.99, flags=8
addto_syslog: Listening on interface eth0:1, 192.168.218.99#123
init_io: maxactivefd 8
local_clock: time 0 clock 0.000000 offset 0.000000 freq 0.000 state 0
key_expire: at 0
peer_clear: at 0 assoc ID 14004 refid INIT
newpeer: 192.168.0.1->192.168.0.200 mode 3 vers 4 poll 6 10 flags 0x201 0x1 ttl
0 key 00000000
addto_syslog: frequency initialized 0.000 PPM from /var/lib/ntp/drift
local_clock: time 0 clock 0.000000 offset 0.000000 freq 0.000 state 1
report_event: system event 'event_restart' (0x01) status 'sync_alarm,
sync_unspec, 1 event, event_unspec' (0xc010)
transmit: at 1 192.168.0.1->192.168.0.200 mode 3
auth_agekeys: at 1 keys 1 expired 0
timer: refresh ts 0
receive: at 1 192.168.0.1<-192.168.0.200 mode 4 code 1
peer 192.168.0.200 event 'event_reach' (0x84) status 'unreach, conf, 1 event,
event_reach' (0x8014)
clock_filter: popcorn 68.148361 0.000015
transmit: at 3 192.168.0.1->192.168.0.200 mode 3
receive: at 3 192.168.0.1<-192.168.0.200 mode 4 code 1
clock_filter: n 2 off -68.149129 del 0.000321 dsp 3.937512 jit 0.000767, age 2
transmit: at 5 192.168.0.1->192.168.0.200 mode 3
receive: at 5 192.168.0.1<-192.168.0.200 mode 4 code 1
clock_filter: n 3 off -68.149322 del 0.000320 dsp 1.937520 jit 0.000693, age 2
transmit: at 7 192.168.0.1->192.168.0.200 mode 3
receive: at 7 192.168.0.1<-192.168.0.200 mode 4 code 1
clock_filter: n 4 off -68.149522 del 0.000297 dsp 0.937526 jit 0.000718, age 2
report_event: system event 'event_peer/strat_chg' (0x04) status 'sync_alarm,
sync_ntp, 2 events, event_restart' (0xc621)
addto_syslog: synchronized to 192.168.0.200, stratum 6

clock_update: at 7 assoc 1
local_clock: assocID 14004 offset -68.149522500 freq 0.000 state 1
step_systime: step -68.149522 residual 0.000000
In ntp_set_tod
ntp_set_tod: settimeofday: 0: Interrupted system call
ntp_set_tod: Final result: settimeofday: 0: Interrupted system call
addto_syslog: time reset -68.149522 s
```

```
ntpd: time set -68.149522s
[root@linux1 ~]# ssh 192.168.0.200 date ; date
root@192.168.0.200's password: secret
Thu Mar  1 19:26:45 CET 2007
Thu Mar  1 19:26:45 CET 2007
[root@linux1 ~]#
```

d) On active le service chez le client.

```
[root@linux1 ~]# chkconfig ntpd on
[root@linux1 ~]# /etc/init.d/ntpd start
DÃ©marrage de ntpd :                                        [  OK  ]
[root@linux1 ~]# tail /var/log/ntp.log
 1 Mar 19:27:07 ntpd[12642]: synchronized to 192.168.0.200, stratum 6
 1 Mar 19:25:59 ntpd[12642]: time reset -68.149522 s
[root@linux1 ~]#
```

3. La TZ (Time Zone).

a) Quelle est la configuration par défaut ?

```
[root@linux1 ~]# echo $TZ

[root@linux1 ~]# date
Thu Mar  1 19:36:57 CET 2007
[root@linux1 ~]# TZ=GMT date
Thu Mar  1 18:37:09 GMT 2007
[root@linux1 ~]# ls -l /etc/localtime
-rw-r--r--  1 root root 1082 Jan 20 10:36 /etc/localtime
[root@linux1 ~]# file /etc/localtime
/etc/localtime: timezone data
[root@linux1 ~]# cmp /etc/localtime /usr/share/zoneinfo/Europe/Paris
```

Remarque : En général, /etc/localtime est un lien symbolique sur le fichier contenant les données de décalage se trouvant dans /usr/share/zoneinfo.

b) On joue avec TZ.

```
[root@linux1 ~]# TZ=GMT date     # l'heure Universelle
Thu Mar  1 18:43:00 GMT 2007
[root@linux1 ~]# export TZ=AST4 # Quebec (-4h)
[root@linux1 ~]# date
Thu Mar  1 14:43:08 AST 2007
[root@linux1 ~]# export TZ=JST-9 # Tokyo (+9h)
[root@linux1 ~]# date
Fri Mar  2 03:44:05 JST 2007
[root@linux1 ~]# export TZ=America/Martinique
[root@linux1 ~]# date
Thu Mar  1 14:44:42 AST 2007
[root@linux1 ~]# export TZ=Australia/Sydney
[root@linux1 ~]# date
Fri Mar  2 05:46:10 EST 2007
[root@linux1 ~]#
```

Tâche 4 : Le clavier

1. Afficher les caractéristiques du terminal.

```
[root@linux1 ~]# stty
speed 38400 baud; line = 0;
-brkint -imaxbel
```

```
[root@linux1 ~]# stty -a
speed 38400 baud; rows 24; columns 80; line = 0;
intr = ^C; quit = ^\; erase = ^?; kill = ^U; eof = ^D; eol = <undef>;
eol2 = <undef>; start = ^Q; stop = ^S; susp = ^Z; rprnt = ^R; werase = ^W;
lnext = ^V; flush = ^O; min = 1; time = 0;
-parenb -parodd cs8 -hupcl -cstopb cread -clocal -crtscts
-ignbrk -brkint -ignpar -parmrk -inpck -istrip -inlcr -igncr icrnl ixon -ixoff
-iuclc -ixany -imaxbel
opost -olcuc -ocrnl onlcr -onocr -onlret -ofill -ofdel nl0 cr0 tab0 bs0 vt0 ff0
isig icanon iexten echo echoe echok -echonl -noflsh -xcase -tostop -echoprt
echoctl echoke
[root@linux1 ~]#
```

2. Configurer le retour arrière.

```
[root@linux1 ~]# stty  erase  '^H'
[root@linux1 ~]# stty  erase   ^?
```

Remarque : Petite astuce, il suffit de taper sur la touche retour arrière et l'on voit à l'écran le code de contrôle associé.

3. Visualiser la configuration du clavier prise au démarrage.

```
[root@linux1 ~]# cat /etc/sysconfig/keyboard
KEYBOARDTYPE="pc"
KEYTABLE="fr"
```

4. Visualiser la keymap courante.

```
[root@linux1 ~]# dumpkeys | head
keymaps 0-6,8-9,12
keycode   1 = Escape
        shift   control keycode   1 = VoidSymbol
keycode   2 = +ampersand          +one              dead_acute
periodcentered
        alt     keycode   2 = Meta_one
        shift   alt     keycode   2 = Meta_ampersand
keycode   3 = eacute            +two              asciitilde        Eacute
nul               nul
        alt     keycode   3 = Meta_two
        shift   alt     keycode   3 = Meta_asciitilde
keycode   4 = +quotedbl           +three            numbersign        cedilla
Escape
[root@linux1 ~]#
```

5. Essayer de changer de keymap (on travaille sur la console maîtresse).

On change de clavier (US, DE et FR). A chaque essai on active la commande echo suivi des premières touches (AZERTY).

```
[root@linux1 ~]# loadkeys us
Loading /lib/kbd/keymaps/i386/qwerty/us.map.gz
[root@linux1 ~]# echo qwerty
qwerty
[root@linux1 ~]# loadkeys de
Loading /lib/kbd/keymaps/i386/qwertz/de.map.gz
[root@linux1 ~]# echo qwertz
qwertz
[root@linux1 ~]# loadkeys fr
Loading /lib/kbd/keymaps/i386/azerty/fr.map.gz
assuming iso-8859-1 cedilla
```

```
assuming iso-8859-1 acute
assuming iso-8859-1 diaeresis
assuming iso-8859-1 brokenbar
assuming iso-8859-1 threequarters
assuming iso-8859-1 currency
assuming iso-8859-1 onehalf
assuming iso-8859-1 onequarter
[root@linux1 ~]# echo azerty
azerty
[root@linux1 ~]#
```

Tâche 5 : Chkconfig

1. Récupérer et installer Apache à partir des sources.

```
[root@linux1 ~]# wget 'http://apache.multidist.com/httpd/apache_1.3.37.tar.gz'
[root@linux1 ~]# tar xvzf apache_1.3.37.tar.gz
[root@linux1 ~]# cd apache_1.3.37
[root@linux1 apache_1.3.37]# ./configure
[root@linux1 apache_1.3.37]# make
[root@linux1 apache_1.3.37]# make install
```

2. Recopier le RC d'Apache dans init.d et l'adapter à chkconfig.

```
[root@linux1 apache_1.3.37]# cd /usr/local/apache/bin
[root@linux1 bin]# cp apachectl /etc/init.d
[root@linux1 bin]# cd /etc/init.d
[root@linux1 init.d]# vi apachectl
#!/bin/sh
# chkconfig: 2345 99 00
# description: The Web Serveur
#...
[root@linux1 init.d]# cd
```

3. Demander à chkconfig sa prise en compte.

```
[root@linux1 ~]# chkconfig --add apachectl
```

4. Désactiver le service au prochain démarrage.

```
[root@linux1 ~]# chkconfig --list apachectl
apachectl       0:off   1:off   2:on    3:on    4:on    5:on    6:off
[root@linux1 ~]# chkconfig apachectl off
[root@linux1 ~]# chkconfig --list apachectl
apachectl       0:off   1:off   2:off   3:off   4:off   5:off   6:off
```

5. Activer le service au prochain démarrage. Désactiver la version RedHat.

```
[root@linux1 ~]# chkconfig httpd off
[root@linux1 ~]# chkconfig apachectl on
[root@linux1 ~]# /etc/init.d/apachectl start
/etc/init.d/apachectl start: httpd started
[root@linux1 ~]#
```

Tâche 6 : Logrotate

1. Visualiser la configuration courante.

```
[root@linux1 ~]# grep daily /etc/crontab
02 4 * * * root run-parts /etc/cron.daily
[root@linux1 ~]# ls /etc/cron.daily
00-logwatch         00webalizer  certwatch  prelink  slocate.cron  yum.cron
```

```
00-makewhatis.cron  0anacron     logrotate  rpm      tmpwatch
[root@linux1 ~]# grep logrotate /etc/cron.daily/logrotate
/usr/sbin/logrotate /etc/logrotate.conf
    /usr/bin/logger -t logrotate "ALERT exited abnormally with [$EXITVALUE]"
[root@linux1 ~]# more /etc/logrotate.conf
# see "man logrotate" for details
# rotate log files weekly
weekly

# keep 4 weeks worth of backlogs
rotate 4

# create new (empty) log files after rotating old ones
create

# uncomment this if you want your log files compressed
#compress

# RPM packages drop log rotation information into this directory
include /etc/logrotate.d

# no packages own wtmp -- we'll rotate them here
/var/log/wtmp {
    monthly
    create 0664 root utmp
    rotate 1
}

# system-specific logs may be also be configured here.
[root@linux1 ~]# ls /etc/logrotate.d
acpid  httpd   ppp     rpm    squid   tux      vsftpd.log
cups   mgetty  psacct  samba  syslog  up2date  yum
[root@linux1 ~]#
```

2. Jouer avec logrotate.

a) On crée un fichier journal.

```
[root@linux1 ~]# find /usr | head -300 > fichier.log
[root@linux1 ~]# ls -lh fichier.log
-rw-r--r--  1 root root 18K Mar  3 13:52 fichier.log
[root@linux1 ~]#
```

b) On crée un fichier de configuration logrotate pour ce fichier.

```
[root@linux1 ~]# vi fichier.conf
/root/fichier.log {
        compress
        size=10k
        create 0600 root root
        rotate 2
}
```

c) On réalise la rotation des journaux.

```
[root@linux1 ~]# logrotate /root/fichier.conf
[root@linux1 ~]# ls -lh fichier.log*
-rw-------  1 root root    0 Mar  3 13:54 fichier.log
-rw-r--r--  1 root root 1.4K Mar  3 13:54 fichier.log.1.gz
[root@linux1 ~]#
```

d) On recommence, mais le journal n'a pas atteint 10k. La rotation n'a pas lieu.

```
[root@linux1 ~]# find /etc | head -50 >> fichier.log
[root@linux1 ~]# ls -lh fichier.log
-rw-------  1 root root 1.1K Mar  3 13:57 fichier.log
[root@linux1 ~]# logrotate /root/fichier.conf
[root@linux1 ~]# ls -lh fichier.log*
-rw-------  1 root root 1.1K Mar  3 13:57 fichier.log
-rw-r--r--  1 root root 1.4K Mar  3 13:54 fichier.log.1.gz
[root@linux1 ~]#
```

e) On ajoute des données au journal et on active la rotation.

```
[root@linux1 ~]# find /etc | head -500 >> fichier.log
[root@linux1 ~]# ls -lh fichier.log*
-rw-------  1 root root  19K Mar  3 13:59 fichier.log
-rw-r--r--  1 root root 1.4K Mar  3 13:54 fichier.log.1.gz
[root@linux1 ~]# logrotate /root/fichier.conf
[root@linux1 ~]# ls -lh fichier.log*
-rw-------  1 root root    0 Mar  3 13:59 fichier.log
-rw-------  1 root root 3.1K Mar  3 13:59 fichier.log.1.gz
-rw-r--r--  1 root root 1.4K Mar  3 13:54 fichier.log.2.gz
```

f) On recommence, mais là, on perd l'archive la plus ancienne.

```
[root@linux1 ~]# find /var | head -500 >> fichier.log
[root@linux1 ~]# ls -lh fichier.log
-rw-------  1 root root  15K Mar  3 14:00 fichier.log
[root@linux1 ~]# logrotate /root/fichier.conf
[root@linux1 ~]# ls -lh fichier.log*
-rw-------  1 root root    0 Mar  3 14:01 fichier.log
-rw-------  1 root root 2.4K Mar  3 14:01 fichier.log.1.gz
-rw-------  1 root root 3.1K Mar  3 13:59 fichier.log.2.gz
[root@linux1 ~]#
```

Tâche 7 : At et batch

1. Créer un script.

```
[root@linux1 ~]# cat un_script.sh
#!/bin/sh
exec > /tmp/result
echo "==== voici la date ===="
date
echo "------- FIN -----------"

[root@linux1 ~]# chmod u+x un_script.sh
```

2. L'exécuter avec at. L'exécution est demandée dans quelques minutes.

```
[root@linux1 ~]# date
Sat Mar  3 14:10:24 CET 2007
[root@linux1 ~]# at now + 2 minutes < un_script.sh
job 2 at 2007-03-03 14:12
[root@linux1 ~]# at 1415 < un_script.sh
job 3 at 2007-03-03 14:15
[root@linux1 ~]# at -l
2       2007-03-03 14:12 a root
3       2007-03-03 14:15 a root
```

```
[root@linux1 ~]# date
Sat Mar  3 14:12:45 CET 2007
[root@linux1 ~]# tail /tmp/result
==== voici la date ====
Sat Mar  3 14:12:00 CET 2007
------- FIN -----------
```

3. Supprimer un travail en attente.

```
[root@linux1 ~]# date
Sat Mar  3 14:13:32 CET 2007
[root@linux1 ~]# at -l
3       2007-03-03 14:15 a root
[root@linux1 ~]# at -r 3
[root@linux1 ~]# at -l
[root@linux1 ~]#
```

4. Activer un travail lorsque le système est à faible charge.

```
[root@linux1 ~]# batch < un_script.sh
job 7 at 2007-03-03 14:31
[root@linux1 ~]# at -l
[root@linux1 ~]# cat /tmp/result
==== voici la date ====
Sat Mar  3 14:31:20 CET 2007
------- FIN -----------
[root@linux1 ~]#
```

Tâche 8 : Anacron

1. Visualiser la configuration courante.

```
[root@linux1 ~]# grep daily /etc/crontab
02 4 * * * root run-parts /etc/cron.daily
[root@linux1 ~]# ls /etc/cron.daily/
00-logwatch         00webalizer  certwatch  prelink  slocate.cron  yum.cron
00-makewhatis.cron  0anacron     logrotate  rpm      tmpwatch
[root@linux1 ~]# grep -v -e '^#' -e '^$' /etc/cron.daily/0anacron
anacron -u cron.daily
[root@linux1 ~]# grep 'anacron.*-s' /etc/init.d/anacron
    daemon +19 anacron -s
[root@linux1 ~]# chkconfig --list anacron
anacron         0:off   1:off   2:on    3:on    4:on    5:on    6:off
[root@linux1 ~]# cat /etc/anacrontab
# /etc/anacrontab: configuration file for anacron

# See anacron(8) and anacrontab(5) for details.

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

1       65      cron.daily              run-parts /etc/cron.daily
7       70      cron.weekly             run-parts /etc/cron.weekly
30      75      cron.monthly            run-parts /etc/cron.monthly
[root@linux1 ~]#
```

2. Jouer avec anacron.

a) On crée un script (cf. item 1 tâche 7).

b) On crée un fichier de configuration d'anacron.

```
[root@linux1 ~]# cat mon_anacron
1   3   mon_job    /root/un_script.sh
```

c) On active anacron.

```
[root@linux1 ~]# ls /var/spool/anacron/
cron.daily  cron.monthly  cron.weekly
[root@linux1 ~]# date
Sat Mar  3 15:01:56 CET 2007
[root@linux1 ~]# anacron -d -t /root/mon_anacron mon_job
Anacron 2.3 started on 2007-03-03
Will run job `mon_job' in 3 min.
Job `mon_job' started
Job `mon_job' terminated
Normal exit (1 jobs run)
[root@linux1 ~]# cat /tmp/result
==== voici la date ====
Sat Mar  3 15:05:04 CET 2007
------- FIN -----------
[root@linux1 ~]# ls /var/spool/anacron/
cron.daily  cron.monthly  cron.weekly  mon_job
[root@linux1 ~]# tail /var/log/cron
Mar  3 15:02:04 linux1 anacron[11699]: Anacron 2.3 started on 2007-03-03
Mar  3 15:02:04 linux1 anacron[11699]: Will run job `mon_job' in 3 min.
Mar  3 15:05:04 linux1 anacron[11699]: Job `mon_job' started
Mar  3 15:05:04 linux1 anacron[11699]: Job `mon_job' terminated
Mar  3 15:05:04 linux1 anacron[11699]: Normal exit (1 jobs run)
```

Remarques : les options d'anacron :
L'option –d demande une exécution en avant-plan (pratique pour le dépannage).
L'option –t permet de préciser l'

d) On active de nouveau anacron : rien ne se passe, il a déjà accompli le travail.

```
[root@linux1 ~]# anacron -d -t /root/mon_anacron mon_job
Anacron 2.3 started on 2007-03-03
Normal exit (0 jobs run)
```

e) On relance anacron après avoir détruit le fichier repère.

```
[root@linux1 ~]# rm /var/spool/anacron/mon_job
rm: remove regular file `/var/spool/anacron/mon_job'? y
[root@linux1 ~]# anacron  -t /root/mon_anacron mon_job
[root@linux1 ~]# ps -e |grep anacron
11710 ?        00:00:00 anacron
[root@linux1 ~]# date
Sat Mar  3 15:10:22 CET 2007
[root@linux1 ~]# date
Sat Mar  3 15:15:48 CET 2007
[root@linux1 ~]# tail /var/log/cron
Mar  3 15:10:05 linux1 anacron[11710]: Anacron 2.3 started on 2007-03-03
Mar  3 15:10:05 linux1 anacron[11710]: Will run job `mon_job' in 3 min.
Mar  3 15:13:06 linux1 anacron[11710]: Job `mon_job' started
Mar  3 15:13:06 linux1 anacron[11710]: Job `mon_job' terminated
Mar  3 15:13:06 linux1 anacron[11710]: Normal exit (1 jobs run)
```

A

Annexes

Annexe A : L'éditeur vi

Annexe B : Quelques numéros ports udp/tcp

Annexe A : L'éditeur vi

Les commandes du mode vi

Insertion/Remplacement de texte

i, a	Insertion avant, après le curseur.
o, O	Insertion avant, après la ligne courante.
R	Bascule dans le mode remplacement (jusqu'à la fin de la ligne courante).

Remarque : On termine ces commandes en appuyant sur la touche échappement (ESC).

Déplacement

←↓↑→

ou h,j,k,l	Déplacement dans les quatre directions.
0, $	On se déplace en début ou fin de ligne.
Ctrl-F, Ctrl-B	On se déplace à l'écran suivant, précédent.
G, 7G	On se déplace en fin de fichier, à la 7ième ligne du fichier.

Remplacement et destruction

J	Jointure de la ligne courante et de la ligne suivante.
x, dd, 4dd	Suppression du caractère courant, de la ligne courante, de 4 lignes.

Rechercher des chaînes

/chaîne, ?chaîne	Effectuer une recherche avant, arrière d'une chaîne.
n, N	Continuer la recherche en avant, en arrière.

Autres

u	Annule la dernière action (undelete).
yy, 4yy	Copier la ligne courante, 4 lignes dans le tampon.
p, P	Coller le tampon après, avant la ligne courante

Les commandes du mode ex (on doit les valider)

Les commandes de sortie

:q!	Abandon.
:wq	Sauvegarde et sortie de l'éditeur.

Autres commandes

:1,$s/ch1/ch2/g	Substitue toutes les occurrences de la chaîne ch1 par la chaîne ch2.
:!ls	Exécute une commande, dans l'exemple `ls`.
:w	Sauvegarde le fichier, on reste sous l'éditeur.
:r fic	Insère le fichier fic après la ligne courante.
:0r fic	Insère le fichier fic avant la première ligne.

Annexe B : Quelques numéros de ports udp/tcp

20	FTP-Data	143	IMAP	1646	Radius
21	FTP-Transfer	144	News	1701	L2TP
22	SSH	161	SNMP	1719	H 323 (VoIP)
23	telnet	177	XDMCP	1720	H.323/Q.931 (VoIP)
25	SMTP	179	BGP	1723	PPTP
42	Nameserver (WINS)	389	LDAP	1812	Radius
43	Whois	443	SSL (https)	1813	Radius
53	Domain (DNS)	445	SMB over TPC/IP	2049	NFS
66	sql*net (Oracle)	464	Kerberos	2869	UPNPHost
67	Bootp (DHCP)	512	Exec (BSD)	3128	Squid
68	Bootp (DHCP)	513	Login (rlogin)	3130	ICP (Cache)
69	TFTP	514	Syslog, RSH	3306	MySQL
70	Gopher	515	Printer (LPD)	3389	RDP (Terminal Server)
79	Finger	517	Talk	4444	Kerberos
80	HTTP (Web)	593	RPC sur https	5060	SIP (VoIP)
88	Kerberos (v5)	631	IPP	5632	PCAnywhere
98	Linuxconf	636	LAPD-SSL	5900	VNC
110	POP-3	666	Doom	6000	X-Windows
111	Portmap (RPC-ONC)	734	Kerberos	6668	IRC
113	Idents/auth	749	Kerberos	6699	Napster
119	NNTP (News)	873	Rsync	7100	XFS (X Font Server)
123	NTP	901	Swat	9100	JetDiret
135	Portmap (RPC-DCE)	993	S-IMAP	26000	Quake
137	Netbios	995	S-POP	27960	QuakeIII
138	Netbios	1434	SQL server	10000	Webmin
139	Netbios	1435	SQL monitor		
		1494	ICA (Citrix)		
		1521	Oracle SQL		
		1645	Radius		

Pour en savoir plus

Fichier :

/etc/services

Internet

IETF assigned port numbers
http://www.iana.org/assignments/port-numbers

List of TCP and UDP port numbers
http://en.wikipedia.org/wiki/List_of_TCP_and_UDP_port_numbers

Vue d'ensemble des services et exigences de ports réseau pour le système Windows Server
http://support.microsoft.com/kb/832017

Trojan list sorted on trojan port
http://www.simovits.com/nyheter9902.html

Index

/

/boot/grub/grub.conf 3-3
/etc/anacrontab 13-14
/etc/apt/sources.list 2-12
/etc/bashrc 13-3
/etc/dumpdates 9-6
/etc/exports 5-13
/etc/fstab 6-2, 6-10
/etc/inittab 10-6
/etc/lilo.conf 10-10
/etc/localtime 13-6
/etc/logrotate.conf 13-11
/etc/lvm/lvm.conf 8-10
/etc/mdadm.conf 7-12
/etc/modprobe.conf 3-7, 4-11
/etc/modules.conf 3-7, 4-11
/etc/ntp.drift 13-6
/etc/ntpd.conf 13-6
/etc/profile 13-3
/etc/profile.d/* 13-3
/etc/raidtab 7-11
/etc/rc.d/rc.sysinit 10-9
/etc/security/pam_env.conf 13-2
/etc/sensors.conf 4-18
/etc/skel/ 13-3
/etc/smartd.conf 4-19
/etc/sudoers 1-10
/etc/sysctl.conf 3-10
/etc/yum.conf 2-14
/proc 10-9
/root/anaconda-ks.cfg 2-6
/sys 10-9

A

ac 11-10
accounting 11-10
accton 11-10
ACL 6-4, 6-7
ACPI 4-5, 10-9
ACU 7-10
AGP 4-4
anacron 13-13, 13-14
APM 4-5
apt-ftparchive 2-12
apt-mirror 2-12
aquota.group 6-2
aquota.user 6-2
ar 2-19
arch 3-3
at 13-13
atd 13-13
attributs étendus 6-7
attributs Ext2 6-6
autorun 5-17
AutoYaST 2-4

B

Bacula 9-3, 9-11
baies de stockage 7-5
bash 13-2
bash_profile 13-3
batch 13-13
BIOS 4-5, 10-2
blockdev 4-10
bottle-neck 11-2
buffer cache 5-4
bus 4-2
bus local 4-4
bus mastering 4-4
bzImage 3-5

C

carte mère 4-2
cartouche magnétique 9-13
cciss 7-10
cdrecord 5-15
Cfengine 1-6
chargeur 10-2
chattr 6-6, 11-13
checklist 12-8
chipsets 4-3
chkconfig 13-10

chunk 7-2, 8-9
chunk-size 7-2, 7-12
ci 1-5
clavier 10-9, 13-8
clonezilla 9-3, 9-11
clvmd 8-9
cmdline 3-8
co 1-5
comptabilité 11-10
contrôleurs 4-2
CPU 4-2, 11-11
cpuinfo 3-8
createrepo 2-14
cron 13-13
crontab 13-13
cycle de rétention 9-4
cycle de sauvegarde 9-4

D

DAS 7-5, 7-9
date 13-6
DCE 13-15
dd 9-8, 9-11
Debian 2-19
  création d'un paquet 2-19
  dépôt 2-11
debugfs 5-11
debugreiserfs 5-8
défaut de page 6-9
dépannage 12-2
depmod 3-7
dépôt 2-8
dépôt Yum 2-14
device driver 4-8
diff 1-5
Dirty pages 6-10
dirty_background_ratio 11-12
dirty_expire_centisecs 11-12
dirty_writeback_centisecs 11-12
discover 4-11
DISPLAY 13-2
disque dédié 6-8
DMA 4-4
dmesg 3-3, 4-6, 4-9, 10-3
dmidecode 4-6
Docbook 1-8
dpkg 2-20
dpkg-deb 2-20
dpkg-scanpackages 2-12
dpkg-scansources 2-12
DTE 13-15
dump 6-5, 6-7, 9-5, 9-6
dumpe2fs 5-11
dumpkeys 13-8
duplexing 7-5

E

e2fsck 5-11
e2label 5-11
edquota 6-3
EISA 4-4
eject 5-16
env 13-2
environnement 13-2
expect 1-4
export 13-2
exportfs 5-13
Ext2 6-2, 6-4, 6-6, 6-7
ext2online 5-11, 8-6
Ext3 5-9, 6-2, 6-7

F

fakeroot 2-20
fcron 13-13
Fiber Channel 7-5
fiches incidents 12-3
file-max 3-10, 11-12
free 6-10, 11-4
FS 5-4

G

getfacl 6-4
getfattr 6-7
GMT 13-6
Google 12-11
groff 1-7
Groupe de volume 8-2
growisofs 5-16
GRUB 10-12
grub-install 10-13

H

HBA 7-7
hdparm 4-6, 4-10, 11-12
HOME 13-2
horloge 10-9, 13-6
hotplug 4-15
HotPlug 4-13
hwclock 13-6
hwinfo 4-11

I

i18n 13-5
icmp_echo_ignore_all 3-10
icmp_echo_ignore_broadcasts 3-10
ident 1-5
init 10-6, 13-2
initrd 10-2, 10-16
inode 5-3
inode-max 3-10, 11-12
inodes 6-2

insmod 3-7
iostat 11-4
ip_forward 3-10
IPC 3-2, 3-11
ipcrm 3-11
ipcs 3-11
IRQ 4-3
ISA 4-3
iSCSI 7-5

J

journal 5-12
journalisation 5-6, 5-11

K

KDE 5-17
Kerberos 9-10
Kernel 10-2
keycode 13-8
keymap 13-8
Kickstart 2-6
KickStart 9-11
KML 4-8
ksconfig 2-6
kudzu 4-11

L

l'APIC 4-3
LANGUAGE 13-5
lastcomm 11-10
*LC_** *13-4*
LC_ALL 13-4
LD_LIBRARY_PATH 13-2
LE *Voir* Logical Extend
lftp 2-9
liaison série 13-15
lilo 10-10
LILO 10-10
LINGUAS 13-5
lintian 2-20
linuxrc 10-16
live-cd 10-17
lm_sensors 4-18
loader 10-2
loadkeys 13-8
locale 13-5
localisation 13-4
logger 1-5
Logical Extend 8-4
Logical Volume *Voir* volume logique
login 11-10, 13-2
LOGNAME 13-2
logrotate 13-11
loop 5-15
LPDEST 13-2
lsattr 6-6
lsdev 4-6
lsmod 3-3, 3-7, 4-9
lsof 12-8
lspci 4-6
lsraid 7-11
lsscsi 4-6
lsusb 4-6
LV *Voir* volume logique
lvchange 8-10
lvcreate 8-4, 8-10
lvdisplay 8-4
lvextend 8-4
LVM 8-2, 10-9
  et RAID 7-13
  LVM1 et LVM2 8-8
lvmdiskscan 8-4
lvreduce 8-4
lvremove 8-4
lvs 8-4
lvscan 8-4

M

m4 1-4
Mailing List 1-3
make 1-4, 3-5
MAKEDEV 4-9, 4-10
MD 7-11
mdadm 7-11
meminfo 11-4
mémoire 6-9
mémoire partagée 3-11
metadata 2-8, 2-14, 8-8
mgetty 13-15
minicom 13-15
mirroring 7-2
mke2fs 5-11
mkfs.xfs 5-7
mkinitrd 3-5, 8-9, 10-16
mkisofs 5-15
mknod 4-10
mkraid 7-11
mkreisrfs 5-8
mkswap 6-10
MMU 6-9
modinfo 3-7, 4-9
modprobe 3-7
Mondo 9-11, 9-12
mondoarchive 9-12
mondorestore 9-12
motherboard 4-2
mount 6-4, 6-6, 6-7
mountd 5-13
MOUNTD 5-13
mpstat 11-4
mt 9-13
mtx 9-13
Multi-path IO 7-6

N

NAS 7-5
netstat 11-4
NFS 5-13, 6-2, 6-4
nfsd 5-13
nice 11-13
niveau de sauvegarde 9-4
noyau 3-2
nroff 1-7
NTP 13-7
ntpd 13-6
ntpdate 13-6
NULL-Modem 13-15

O

OHCI 4-5
opcontrol 11-4
ORCA 7-10

P

page 6-9
page de manuel 1-7
page-cluster 11-12
Page-in 6-9
Page-out 6-10
PAM 11-13
pam_env.so 13-2
partimage 9-3
patch 1-5, 3-5
PATH 13-2
PCI 4-4
PCI Express 4-5
PCI-X 4-5
pdflush 11-12
PE 11-11, *Voir* Physical Extend
Physical Extend 8-3
Physical Volume *Voir* volume physique
PIC 4-3
pilote de périphérique 4-8
PnP 4-4
pnpdump 4-6
portmap 5-13
PORTMAP 5-13
POSIX 13-4
PowerEdge 4-11, 7-9
PRINTER 13-2
procinfo 11-4
profile 13-3
Proliant 4-11, 7-9, 7-10
ps 11-4
psacct 11-10
PV 8-8, *Voir* volume physique
pvcreate 8-4
pvdisplay 8-4
pvmove 8-9
PVRA 8-8
pvremove 8-9
pvresize 8-9
pvs 8-4
pvscan 8-4

Q

quota 6-2, 6-3
Quota 10-9
quotacheck 6-3
quotaoff 6-3
quotaon 6-3

R

RAID 7-2, 10-9
  logiciel 7-4
  matériel 7-4
  niveau de 7-2
  RAID 0 7-3
  RAID 01 ou 0+1 7-4
  RAID 1 7-3
  RAID 10 ou 1+0 7-3
  RAID 4 7-3
  RAID 5 *Voir*
raidhotadd 7-11
raidhotremove 7-11
raidsetfaulty 7-11
raidstart 7-11
raidstop 7-11
RAM 4-2, 6-9
ram0 5-15
Ramdisk 5-15
raw 6-8
raw device 6-8
RC 10-2
rc.sysinit 10-9
rcs 1-5
RCS 1-5
ReiserFS 5-8, 6-2, 6-4, 6-6, 6-7, 9-5
reiserfsck 5-8
reiserfstune 5-8
repquota 6-3
resize_reiserfs 5-8, 8-6
resize2fs 5-11, 8-6
restore 9-6
rlog 1-5
rmmod 3-7
rmt 9-9
RMT 9-7
RPC 5-13
rpcinfo 5-13
RPM 2-16
  création 2-16
rpm2cpio 2-18
rpmbuild 2-18
rpmdeps 2-18
rpmsign 2-18
rsh 9-9

RSH 9-7, 9-9, 9-10
rsync 2-9
RSYNC 9-10
RSYNC_RSH 9-9
rug 2-9
runlevel 10-6

S

sa 11-10
sa1 11-8
sa2 11-8
sadc 11-8
SAN 7-5, 7-7
sar 11-4, 11-8
Sauvegarde 9-3
  Bare-Metal 9-11
  différentielle 9-4
  incrémentale 9-4
scancode 13-8
Scheduler 3-2
script 1-5
SCSI 4-5
scsiinfo 4-6
SELinux 6-7, 10-9
sémaphores 3-11
sensors 4-18
sensors-detect 4-18
serveur de dépôt 2-8
setfacl 6-4
setfattr 6-7
setquota 6-3
SHELL 13-2
shmmax 3-10
showmount 5-13
slabinfo 11-6
SMART 4-19
Smart-Array 7-10
smartctl 4-19
smartd 4-19
Snapshot 8-7
Spare 7-4
ssh 9-9
SSH 9-10
Stage1 10-12
Stage1_5 10-12
Stage2 10-12
strace 3-3, 12-7
strip 7-2
stripping 7-2
stty 13-8, 13-15
sudo 1-10
Superbloc 5-3
swap 6-10, 10-9
swapon 6-10
sysctl 3-10
sysfs 4-13
Syslinux 10-15
systool 4-13

T

TAPE 9-7, 9-9
telinit 10-6
TERM 13-16
TERMCAP 13-15
TERMINFO 13-15
time 11-4
tmpfs 5-15
top 11-4
Tour de Hanoi 9-4
troff 1-7
Troubleshooting 11-2, 12-2
tset 13-15
tune2fs 5-11
TZ 13-7

U

udev 4-15
Udev 4-15, 5-17, 10-9
udevd 4-15
udevinfo 4-15
udevsend 4-15
udevstart 4-15
udevtest 4-15
UHCI 4-5
ulimit 11-13
uname 3-3
Unicode 13-4
unset 13-2
uptime 11-4
USB 4-5
USB bootable 10-18
usbmodules 4-10
UTC 13-6
UTF 13-4
UUID 8-8

V

VFS 3-2, 5-2, 6-4
VG *Voir* Groupe de volumes
vgcfgbackup 8-10
vgcfgrestore 8-10
vgchange 8-4, 8-5, 8-10
vgck 8-10
vgconvert 8-10
vgcreate 8-4
vgdisplay 8-4
vgexport 8-9
vgextend 8-4
vgimport 8-9
vgmerge 8-10
vgmknodes 8-10
VGRA 8-8
vgreduce 8-4
vgremove 8-4
vgrename 8-10

vgs 8-4
vgscan 8-4, 8-5
vgsplit 8-10
vmlinuz 3-3
vmstat 6-10, 11-4, 11-6
Volume Group *Voir* Groupe de volumes
volume logique 8-2
volume physique 8-2

W

w 11-4
who 10-6

X

x86info 4-6
xattr 6-7
Xfs 6-4
XFS 5-7, 6-2, 6-7
xfs_check 5-7
xfs_growfs 5-7, 8-6
xfs_info 5-7
xfs_repair 5-7
xmodmap 13-8
xorg.conf 13-8
xSeries 4-11, 7-9

Guide de formation avec ateliers pratiques

Linux Administration

Les Guides de formation Tsoft
Rédigés par des professionnels de la formation, *les Guides de formation Tsoft* ont été adoptés par de nombreuses entreprises comme supports de cours ou manuels d'autoformation. Chaque manuel est découpé en modules thématiques présentés sous forme de fiches descriptives illustrées de nombreux exemples de commandes et de scripts. Chaque module se termine par une série de travaux dirigés minutés pour mettre immédiatement en pratique les notions acquises.

Après des études de doctorat en biophysique, **Jean-François Bouchaudy** s'est orienté vers l'informatique en se spécialisant dans le développement d'applications en C et d'applications TCP/IP sous Unix. Il anime des formations sur les thèmes suivants : administration Unix et Linux, programmation système Unix, langages C, C++ et Perl, TCP/IP, Samba, sécurité réseau, etc.

Acquérir des compétences professionnelles en administration Linux

Après un premier tome dédié aux *Bases de l'administration système*, ce deuxième tome de la série Linux Administration s'adresse aux administrateurs souhaitant approfondir tous les aspects de la gestion d'un serveur Linux en entreprise : compilation du noyau, gestion des périphériques, gestion des espaces disque, sauvegarde, optimisation des performances et dépannage.

Il met l'accent sur plusieurs techniques incluses récemment dans les distributions Linux : Kickstart, Yum, SysFS, Udev, lm_sensors, SMART, ACL, xattr, LVM, création de live-cd ou de clé USB bootable, etc.

Très pragmatique dans son approche, l'auteur va à l'essentiel avec des fiches de cours synthétiques, illustrées de nombreux exemples de commandes et de scripts. Chaque module se termine par une série de travaux dirigés minutés pour se mettre immédiatement en situation et maîtriser les tâches que doit assurer au quotidien un administrateur Linux.

Au sommaire

Administration Linux professionnelle : automatiser, journaliser, documenter, communiquer • L'installation : organisation des données, installation automatique, Kickstart (Redhat), serveur de dépôt, dépôt Debian, dépôt Yum, création de packages RPM/Debian • Le noyau : compilation, modules, paramétrage à chaud… • La gestion des périphériques : ajouter un périphérique, SysFS, Udev, lm_sensors, SMART • Les systèmes de fichiers : FS à journalisation, XFS, ReiserFS, Ext3, NFS, FS spéciaux, montage à la volée • Compléments sur les disques et les FS : quotas, ACL, attributs Ext2, xattr, disques dédiés, SWAP • Systèmes RAID : RAID matériel, SAN, DAS, RAID logiciel… • Gestion de disque avec le LVM : volumes logiques et groupes de volumes, snapshots… • La sauvegarde : sauvegarde incrémentale, commandes dump/restore, dd, sauvegarde réseau, Bare-Metal, Mondo, cartouches • Le démarrage : grandes étapes, init, rc.sysinit, LILO, GRUB, Syslinux, initrd, créer un live-cd, une live-key • La gestion des performances : surveiller le système, commandes vmstat, sar, augmenter les ressources, diminuer l'usage des ressources • Le dépannage : méthodologie, exemple d'application • Compléments : l'environnement, la localisation, l'heure, le clavier, chkconfig, logrotate, travaux périodiques, anacron, terminaux et liaison série.

Code éditeur : G12154
ISBN-13 : 978-2-212-12154-4



25 €

Tsoft EDITEUR

www.editions-eyrolles.com
Groupe Eyrolles | Diffusion Geodif | Distribution Sodis

www.barbary-courte.com