

Guide de formation avec ateliers pratiques

Linux

Administration

Jean-François Bouchaudy

Les Guides
de formation

Tsoft

Plus de 40 000
personnes formées
à Unix et Linux

Tome 2

Administration système avancée

2^e édition

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.

Cette seconde édition met l'accent sur plusieurs techniques incluses récemment dans les distributions Linux : Ext4, SAN/iSCSI, Upstart, PXE, Grub 2, Bacula, Cgroup, Munin, Udev, LVM, création 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.

L'accent est mis sur le mode commande, mieux adapté aux besoins des administrateurs que les outils graphiques, et qui offre l'avantage d'une certaine indépendance vis-à-vis des distributions. Les travaux dirigés sont basés sur la distribution Red Hat Enterprise Linux 6 (elle-même dérivée de Fedora 12/13), mais les spécificités des autres distributions (Debian 5, Ubuntu 10, SUSE 11) sont également traitées dans l'ouvrage.

Au sommaire

Administration Linux professionnelle : automatiser, journaliser, documenter, communiquer • L'installation : organisation des données, installation automatique, Kickstart (Red Hat), serveur de dépôt, dépôt Debian, dépôt Yum, packages RPM/Debian • Le noyau : compilation, modules, paramétrage à chaud... • La gestion des périphériques : ajouter un périphérique, SysFS, Udev, lm_sensors, SMART, HAL • Les systèmes de fichiers : FS à journalisation, XFS, ReiserFS, Ext3, Ext4, NFS, CIFS, 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 FC/iSCSI, DAS (Proliant), RAID logiciel... • Gestion de disque avec le LVM : volumes logiques et groupes de volumes, snapshots, LVM et RAID... • La sauvegarde : sauvegarde incrémentale, commandes dump, restore, dd, sauvegarde réseau, Bare-Metal, Mondo, cartouches, Bacula • Le démarrage : grandes étapes, init, rc.sysinit, LILO, GRUB, Syslinux, PxeLinux, 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, cgroups • Le dépannage : méthodologie, exemple d'application • Compléments : l'environnement, la localisation, l'heure, le clavier, chkconfig, logrotate, travaux périodiques, anacron, terminaux et liaison série.

Linux

Administration

J.-F. BOUCHAUDY. – **Linux Administration. Tome 1 : les bases de l'administration système.**
N° 12624, 2^e édition, 2009, 270 pages.

J.-F. BOUCHAUDY. – **Linux Administration. Tome 3 : sécuriser un serveur Linux.**
N° 12245, 2^e édition, 2010, 468 pages.

J.-F. BOUCHAUDY. – **Linux Administration. Tome 4 : les services applicatifs Internet (Web, email, FTP).**
N° 12248, 2009, 400 pages.

R. BIZOÏ. – **Oracle 11g Administration.**
N° 12898, 2011, 880 pages.

R. BIZOÏ. – **Oracle 11g Sauvegarde et restauration.**
N° 12899, 2011, 420 pages.

F. JOUCLA, J.-F. ROUQUIÉ. – **Lotus Domino 8.5 Administration – Tome 2. Gestion et optimisation.**
N° 12791, 2010, 380 pages.

F. JOUCLA, J.-F. ROUQUIÉ. – **Lotus Domino 8.5 Administration – Tome 1. Installation et configuration.**
N° 12790, 2010, 420 pages.

Autres ouvrages

R. STALLMAN, S. WILLIAMS, C. MASUTTI. – **Richard Stallman et la révolution du logiciel libre.**
N° 12609, 2010, 324 pages (Collection Accès libre).

P. HANSTEEN, adapté par M. DERCHE. – **Le livre de Packet Filter (PF).**
N° 12516, 2009, 194 pages (Cahiers de l'Admin).

J. GABÈS. – **Nagios 3 pour la supervision et la métrologie. Déploiement, configuration et optimisation.**
N° 12473, 2009, 482 pages (Cahiers de l'Admin).

L. DRICOT. – **Ubuntu efficace. Ubuntu 9.04 « Jaunty Jackalope »**
N° 12362, 3^e édition, 2009, 344 pages avec CD-Rom (Collection Accès libre).

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, 3^e édition, 2007, 266 pages (Cahiers de l'Admin).

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

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

Linux

Administration

Tome 2

Administration système avancée

Jean-François Bouchaudy



2^e édition

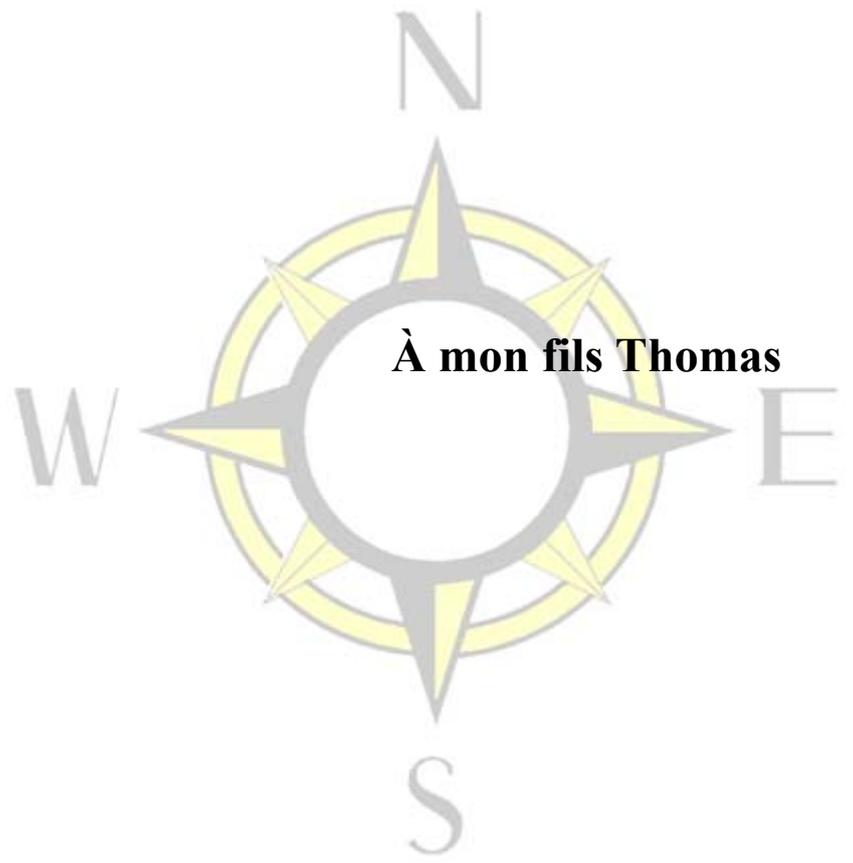
ÉDITIONS EYROLLES
61, bd Saint-Germain
75240 Paris Cedex 05
www.editions-eyrolles.com



Le code de la propriété intellectuelle du 1^{er} 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, 2011, ISBN : 978-2-212-12882-6

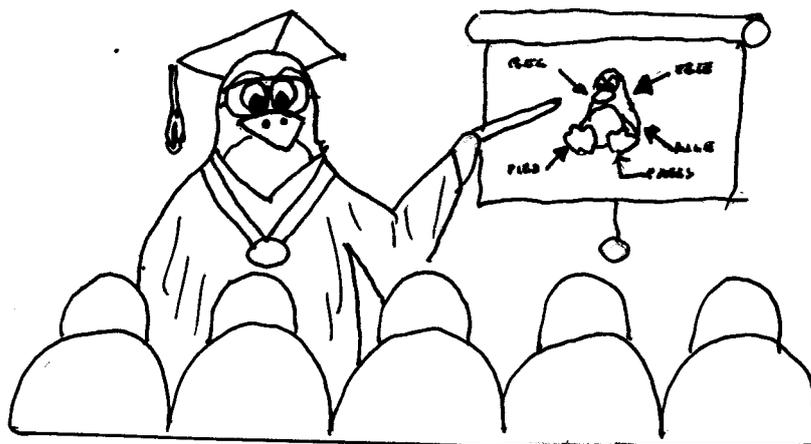


À mon fils Thomas

Remerciements

Je remercie Philippe Moreau, le directeur de cette collection pour son soutien moral. Sans lui, peut-être ce deuxième tome n'aurait-il jamais vu le jour.

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, Im_sensors, SMART, ACLs, xattr, FUSE, Ext4, LVM, cgroups, 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. À 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 accepté 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 la distribution RedHat, la plus utilisée dans les entreprises. 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, Ubuntu 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'autoformation. L'élève doit disposer d'un ordinateur qui sera dédié à Linux (on le reformate complètement). Les modules installation et sauvegarde nécessitent 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 soutenue, 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 avec des compléments sur Internet

Le site www.editions-eyrolles.com propose sur la page de présentation de l'ouvrage des liens dédiés à des compléments, par exemple le programme Breaksys qui sert à l'atelier de dépannage.

Pour accéder à cette page, rendez-vous sur le site www.editions-eyrolles.com, dans la zone <Recherche> saisissez **G12882** et validez par <Entrée>.

Table des matières

PROGRESSION PÉDAGOGIQUE	1
QUELQUES CONVENTIONS DE NOTATION	2
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 paquetage RPM.....	2-16
Créer un paquetage 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	3-9
Le paramétrage à chaud du noyau	3-11
IPC System V	3-12
ATELIERS	3-14
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
HAL.....	4-20
ATELIERS	4-22
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 Ext4.....	5-13
Le FS NFS.....	5-15
Le FS CIFS.....	5-17
Le FS ISO9960.....	5-19
FS spéciaux	5-21
Le montage à la volée.....	5-23
ATELIERS	5-24
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
Les stratégies de stockage d'information	7-5
SAN FC.....	7-6
SAN iSCSI	7-11
Les solutions de type DAS.....	7-13
Proliant (exemple de solution DAS)	7-14
Le RAID logiciel.....	7-15
Le RAID et le LVM	7-17
ATELIERS	7-19
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

Le LVM et le RAID	8-8
LVM – Compléments	8-10
ATELIERS	8-13
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
Bacula	9-14
ATELIERS	9-17
10 LE DÉMARRAGE	10-1
Les grandes étapes du démarrage	10-2
Dépannage du démarrage	10-5
Init – SystemV	10-7
Init – Upstart	10-10
Le script rc.sysinit	10-12
Démarrer avec Lilo	10-13
Démarrer avec Grub 1	10-15
Démarrer avec Grub 2	10-18
Démarrer avec Syslinux	10-20
PxeLinux	10-21
Initrd	10-23
Créer un live-CD, une live-key	10-25
ATELIERS	10-28
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
Test de performance – Benchmarks	11-14
La mémoire	11-17
Les cgroups	11-20
Munin	11-22

ATELIERS	11-24
12 LE DÉPANNAGE	12-1
L'approche générale du dépannage	12-2
Étape n° 1 : Faire une description claire du problème	12-3
Étape n°2 : Imaginer les causes possibles	12-4
Étape n°3 : Rassembler des informations	12-5
Étape 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
Les processus	12-18
Les journaux de bord	12-21
Rsyslog	12-24
ATELIERS	12-26
13 COMPLÉMENTS	13-1
L'environnement	13-2
La localisation	13-4
L'heure	13-7
Le clavier	13-9
Chkconfig	13-11
Logrotate	13-12
Les travaux périodiques	13-14
Anacron	13-15
Les terminaux et la liaison série	13-17
ATELIERS	13-19
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 G12882 dans la zone <Recherche> et validez par <Entrée>.	
INDEX.....	I-1

Progression pédagogique

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

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 bootables.

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 quantités 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 cronjobs. 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 :

```
ArrÃ¢t 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'accent et donc pas de problème ! (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

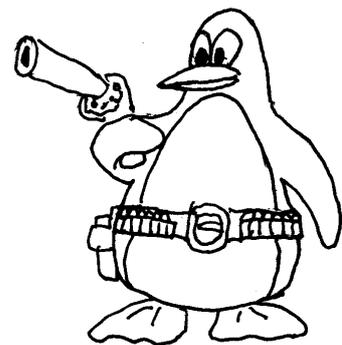
Administration Linux professionnelle

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 utilisateur 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.
- Évolutif.
- 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 ces 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 panne, vous pouvez faire des erreurs d'exploitation. À 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 *mailing 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/pdf/rhel-isa-fr.pdf>

Les mailing lists respectivement des systèmes RedHat, SuSE, Debian et Ubuntu.

<http://www.redhat.com/mailman/listinfo>

<http://lists.suse.com/archive/>

<http://www.debian.org/MailingLists/subscribe>

<https://lists.ubuntu.com/>

The USENIX Special Interest group for Sysadmins

<http://www.sage.org>

Annuaire 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 fiables les procédures. Toutes les tâches répétitives doivent être automatisées.

Le savoir concret

Les langages de script :

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

Les outils complémentaires d'automatisation.

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

Pour en savoir plus

Les pages de manuel

`awk(1)`, `perl(1)`, `python(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 (2002).

Introduction à Perl de Randal Schwartz, Tom Phoenix, Brian-D Foy, et Hervé Soulard chez O'Reilly (2006).

Apprendre à programmer avec Python, par Gérard Swinnen, chez O'Reilly (2005).

Programmer avec les outils GNU, par M. Loukides & Andy Oram, chez O'Reilly. Ce livre traite notamment des commandes make, RCS... (1997)

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

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

Les commandes RCS

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

Focus : Le cycle de vie avec RCS

- 1) On crée un répertoire RCS.
- 2) On archive une version avec `ci -l`.
- 3) On utilise le fichier ou on le modifie.
- 4) On boucle sur les opérations 2 et 3.

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. À 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 ou 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/Linux 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)`, `resintro(1)`, `ci(1)`, `co(1)`, `rsc(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`... (1997)

Documenter son système

La théorie

Il faut conserver la documentation écrite. Il faut savoir accéder à la documentation électronique. Il faut au 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

<code>troff</code>	Génère un fichier graphique (PS...) à partir d'une source <code>troff</code> .
<code>nroff</code>	Génère un fichier texte à partir d'un source <code>nroff</code> .
<code>groff</code>	La commande GNU qui émule les commandes <code>nroff/troff</code> .
<code>zcat</code>	Affiche le contenu d'un fichier texte compressé.

La commande groff (Gnu-roff)

<code>groff -Tascii -man fichier</code>	Génère un fichier texte.
<code>groff -Tps -man fichier</code>	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), zcat(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.
pidgin	Outil de messagerie instantanée (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

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

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 a 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 1^{re} colonne). 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.

Exemple :

```
%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.....	10 mn
Tâche 2 : Exemple d'utilisation des commandes expect et m4	5 mn
Tâche 3 : Journaliser avec script, logger, communiquer avec motd et wall	10 mn
Tâche 4 : Tracer les modifications d'un fichier de configuration avec RCS.....	10 mn
Tâche 5 : Écrire une page de manuel	15 mn
Tâche 6 : Utiliser la commande sudo	10 mn
Tâche 7 : Les commandes diff et patch	5 mn

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.

```
# vi /usr/local/sbin/disk_full
#!/bin/sh
# disk_full - teste si les disques sont plein
export LANG=C
: ${SEUIL:=80}
message=/tmp/bidon$$

df -P | 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
# chmod +x /usr/local/sbin/disk_full
```

2. Tester la commande.

Remarque : adapter le seuil de telle sorte que le script déclenche une alerte.

```
# SEUIL=20 sh disk_full
# mail
Heirloom Mail version 12.4 7/29/08. Type ? for help.
"/var/spool/mail/root": 1 message 1 new
>N 1 root          Fri Jul  9 11:01  18/606  "LES DISQUES SONT PLEI"
& 1
Message 1:
From root@linux01.pinguins  Fri Jul  9 11:01:50 2010
Return-Path: <root@linux01.pinguins>
X-Original-To: root
```

```
Delivered-To: root@linux01.pinguins
Date: Fri, 09 Jul 2010 11:01:49 +0200
To: root@linux01.pinguins
Subject: LES DISQUES SONT PLEINS
User-Agent: Heirloom mailx 12.4 7/29/08
Content-Type: text/plain; charset=us-ascii
From: root@linux01.pinguins (root)
Status: R
```

```
/ 26%
```

```
& q
```

```
Held 1 message in /var/spool/mail/root
```

3. Créer un crontab pour automatiser la surveillance.

```
# crontab -e
0 12 * * * /usr/local/sbin/disk_full
```

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. Installer `expect` au préalable.

```
# yum -y -q install expect
# 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
# chmod +x connexion.expect
# ./connexion.expect
# cat /tmp/date.log
ven. juil. 9 11:16:10 CEST 2010
```

Remarque : dans l'exemple, on suppose que l'on se soit déjà connecté au serveur par SSH.

2. Créer un fichier utilisant le préprocesseur `M4`.

```
# 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
```

```
# date > date.txt
# m4 source.m4
ven. juil.  9 11:19:28 CEST 2010
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.**

```
# script -a
Script started, file is typescript
# date
Sun Dec 31 23:33:33 CET 2006
# id
uid=0(root) gid=0(root)
groups=0(root),1(bin),2(daemon),3(sys),4(adm),6(disk),10(wheel)
# 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
# uptime
 23:33:51 up 2 days, 10:55,  1 user,  load average: 0.00, 0.00, 0.17
# exit
exit
Script done, file is typescript
# cat typescript
Script started on Sun 31 Dec 2006 11:33:28 PM CET
# date
Sun Dec 31 23:33:33 CET 2006
# id
uid=0(root) gid=0(root)
groups=0(root),1(bin),2(daemon),3(sys),4(adm),6(disk),10(wheel)
# 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
# uptime
 23:33:51 up 2 days, 10:55,  1 user,  load average: 0.00, 0.00, 0.17
# exit
exit

Script done on Sun 31 Dec 2006 11:33:56 PM CET
```

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.

```
# echo "local1.*          /var/log/journal_de_bord.log" >> /etc/rsyslog.conf
# service rsyslog restart
# logger -p local1.info "aujourd'hui j'installe Oracle"
# logger -p local1.warn "disques pleins, je fais du menage"
# tail /var/log/journal_de_bord.log
Jul  9 11:30:34 instru root: aujourd'hui j'installe Oracle
```

```
Jul  9 11:30:55 instru root: disques pleins, je fais du menage
#
```

3. Prévenir les utilisateurs en utilisant le fichier motd.

```
# vi /etc/motd
=====

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

=====

# 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

=====

$ exit
Connection to localhost closed.
# > /etc/motd
```

4. Prévenir immédiatement les utilisateurs avec wall.

```
# 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
```

Tâche 4 :

Tracer les modifications d'un fichier de configuration avec RCS

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

Remarque : on suppose que l'on utilise le serveur CUPS.

```
# yum -q -y install cups
# service cups start
# yum -y -q install rcs
# rpm -q rcs
rcs-5.7-36.el6.i686
# cd /etc/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.

```
# 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
# ls RCS
cupsd.conf,v
# 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.default

```

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.

```

# 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:*
# vi cupsd.conf
# $Id$
...
Listen localhost:631
Listen 192.168.0.1:631
...

```

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

```

# 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
# /etc/init.d/cups restart
Stopping cups:                [ OK ]
Starting cups:                [ OK ]
# 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:*

```

5. Visualiser les commentaires RCS.

```

# ident cupsd.conf
cupsd.conf:
    $Id: cupsd.conf,v 1.2 2006/12/31 21:38:22 root Exp $

```

6. Visualiser l'historique des modifications.

```

# 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
=====
# cd
```

Tâche 5 : **Écrire une page de manuel**

1. Écrire la commande news.

```
# 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

# chmod 555 /usr/local/bin/news
```

2. Écrire une page de manuel pour cette commande.

```
# mkdir -p /usr/local/man/man1
# 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.

```

# 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.

```

4. On la compresse.

```

# gzip /usr/local/man/man1/news.1
# 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.

```

# man news
No manual entry for news
# MANPATH=/usr/share/man:/usr/local/man
# export MANPATH
# 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.

```
# makewhatis /usr/local/man
# man -k news
news                (1)  - print news items
# more /var/cache/man/whatis
...
```

Remarque : le fichier `whatis` contient l'index des commandes. La commande `makewhatis` reconstruit ce fichier. Sans argument, la commande explore les différents répertoires contenant le manuel. Pour accélérer son exécution, on peut se limiter à une seule arborescence.

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

```
# 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`.

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

```
/var/news/msg2
:~::~:
...
```

Tâche 6 :

Utiliser la commande sudo

1. Créer un compte.

```
# useradd -m pierre
# echo pierre | passwd --stdin pierre
Changing password for user pierre.
passwd: all authentication tokens updated successfully.
```

2. Modifier la configuration : Pierre peut créer des comptes utilisateur.

```
# cp /etc/sudoers /etc/sudoers.000
# visudo
...
pierre linux01.pinguins = (root) /usr/sbin/useradd
```

3. Se connecter sous le compte pierre et créer un utilisateur.

```
# ssh -l pierre localhost
pierre@localhost's password: pierre
Last login: Sat Dec 30 10:33:02 2006 from raichu.pokemon
$ 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.
    #3) With great power comes great responsibility

[sudo] Password: pierre
$ id titeuf
uid=502(titeuf) gid=502(titeuf) groupes=502(titeuf)
$ exit
```

4. Créer une configuration plus réaliste.

```
# cp /etc/sudoers /etc/sudoers.$(date +%m%d%H%M')
# 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
# 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.

```
# groupadd admins
# useradd -m -g admins cathy
```

```
# echo cathy | passwd --stdin cathy
Changing password for user cathy.
passwd: all authentication tokens updated successfully.
```

6. Tester.

```
# su - cathy
$ 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
$ PATH=$PATH:/usr/sbin
$ sudo useradd -m vomito
$ id vomito
uid=506(vomito) gid=506(vomito) groups=506(vomito)
$ exit
```

7. Donner tous les droits aux membres du groupe admins (stratégie d'Ubuntu).

```
# visudo
#%admins ALL = (root) NOPASSWD: CMDS_USER
%admins ALL=(ALL) ALL
# su - cathy
$ sudo tail -1 /etc/shadow
[sudo] password for cathy: cathy
vomito:!!:14799:0:99999:7:::
$ exit
```

Tâche 7 :

Les commandes diff et patch

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

```
# cd /etc/cups
# diff -c cupsd.conf cupsd.conf.default > patch.txt
# head patch.txt
*** cupsd.conf 2007-01-30 15:30:22.000000000 +0100
--- cupsd.conf.default 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 $"
...
# cp cupsd.conf.default cupsd.conf.000
# cp cupsd.conf.default cupsd.conf.001
# yum -q -y install patch
# patch -p0 cupsd.conf.001 < patch.txt
patching file cupsd.conf.001
Reversed (or previously applied) patch detected! Assume -R? [n] y
# diff cupsd.conf cupsd.conf.001
# cd
```

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

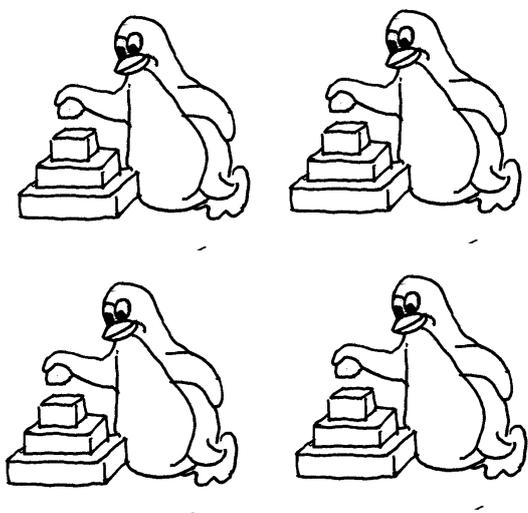
L'installation

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 paquetages.

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 paquetage RPM
- Créer un paquetage 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 principale, 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.
- Les baies de disques et la virtualisation des serveurs.

Exemples de stratégies

Remarque : on prend Apache comme exemple d'application.

Système monodisque :

- 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 les simplifier, voire les 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.

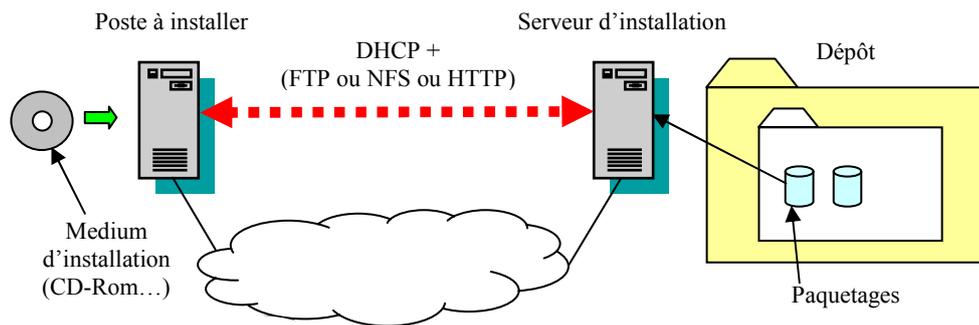


Fig. Installation réseau

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 CD-Rom 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 minisystème 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 forcément 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 clusters de calculs.

Ubuntu

L'installation finale repose sur le logiciel `debian-installer` mais il est possible d'utiliser KicStart comme frontal.

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 préconfiguration (`preseed.cfg`)

La manière la plus simple pour créer le fichier de préconfiguration 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
```

Éventuellement on édite ensuite le fichier.

Démarrer l'installation

Lors du démarrage à partir du medium d'installation (CD-Rom 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 paquetages et le chargeur.
- La configuration : des scripts de post-installation configurent 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 préconfiguration 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 CD-Rom d'installation dans un répertoire d'un serveur HTTP ou NFS.

Démarrer l'installation

Lors du démarrage à partir du medium d'installation (CD-Rom 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/lenny/example-preseed.txt>

Le manuel d'installation

<http://www.debian.org/releases/stable/i386/index.html>

Le logiciel FAI – Home Page

<http://www.informatik.uni-koeln.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/

Internet – Installation automatique d'une distribution Ubuntu

Automatic installation

<https://help.ubuntu.com/10.04/installation-guide/i386/automatic-install.html>

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

```
system-config-kickstart
```

Cette commande graphique génère de toutes pièces un fichier kickstart.

```
system-config-kickstart --generate fichier
```

Cette commande en mode texte génère un fichier kickstart qui reprend la liste des paquetages 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 paquetages à installer. Ce sont soit des paquetages individuels, soit des groupes de paquetages. 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 CD-Rom

```
install  
cdrom
```

Installation réseau

a) En NFS

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

b) En FTP ou HTTP (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 CD-Rom d'installation.

Focus : Démarrer l'installation

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

a) Si on utilise un CD-Rom (ou un DVD-Rom) 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 paquetage `/usr/share/doc/system-config-kickstart-*/`.

Internet

Redhat – Kickstart Installation

http://www.redhat.com/docs/en-US/Red_Hat_Enterprise_Linux/5.5/html/Installation_Guide/ch-kickstart2.html
http://www.redhat.com/docs/en-US/Red_Hat_Enterprise_Linux/6-Beta/html/Installation_Guide/ch-kickstart2.html

Chapter 31. PXE Network Installations

http://www.centos.org/docs/5/html/Installation_Guide-en-US/ch-pxe.html

KickStart Configurations & CFEngine

<http://sial.org/howto/kickstart/>
<http://www.cfengine.org/manuals/cf3-solutions.html>

Livre

Managing RPM-Based Systems with KickStart and Yum, par Ethan McCallum, chez O'Reilly (2007).

Créer un serveur de dépôt

La théorie

Pourquoi utiliser un serveur de dépôt

L'installation et la mise à jour des paquetages sont simplifiées si l'administrateur utilise des logiciels évolués comme Yum ou Apt-get qui gèrent les dépendances de paquetages et qui automatiquement téléchargent ces paquetages 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 sûr 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 paquetages 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 paquetages de base de la distribution. Ce sont les mêmes utilisés pour l'installation du système. On peut les obtenir en les téléchargeant à partir d'Internet ou en les recopiant à partir des CD-Rom 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 métadonnées qui mémorisent entre autre les relations de dépendances entre paquetages. Chaque outil utilise des métadonnées de nature différente. Éventuellement on peut créer plusieurs métadonnées. Ainsi le même dépôt pourrait être accessible aussi bien par Yum que par Apt-get.

Particularités des distributions

RedHat, Fedora, CentOS

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

Red Hat Network (RHN) Satellite est un système qui permet de gérer de manière centralisée le déploiement d'un ensemble de serveurs et stations RedHat.

Debian, Ubuntu

La distribution Debian et ses dérivés utilisent en standard l'outil **aptitude** pour l'installation et la mise à jour des packages. Le logiciel **debmirror** permet de créer un miroir au dépôt.

SUSE

Le cas de SUSE est assez complexe. Les versions récentes utilisent l'outil **rug** pour l'installation et la mise à jour des paquetages. Cet outil accepte de dialoguer avec plusieurs types de dépôts, 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 dépôt : 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 miroir 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 créer des miroirs aux 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.lftp
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
```