Collection Savoirs francophones
Série Technologies de l'information

Préparation à l'examen 201
pour la certification de l'Institut professionnel de Linux, niveau avancé (LPIC-2)

Zied Bouziri, Hedi Magroun

Pour citer cet ouvrage

Zied Bouziri, Hedi Magroun (2012). *Préparation à l'examen 201 pour la certification de l'Institut professionnel de Linux, niveau avancé (LPIC-2)*. Agence universitaire de la Francophonie, Paris. Disponible sur le Web : www.lpi-francophonie.org/spip.php?article266

Les auteurs remercient Véronique Pierre pour son appui à la relecture et à la mise en forme de l'ouvrage.

Agence universitaire de la Francophonie (AUF)
Direction de l'innovation pédagogique et de l'économie de la connaissance
4 place de la Sorbonne
75005 PARIS
France
www.auf.org

Accès et utilisation

Cet ouvrage est diffusé exclusivement au format numérique, gratuitement. Il est téléchargeable au format PDF sur le site **LPI Francophonie**, www.lpi-francophonie.org.

Le contrat Creative Commons BY-NC-SA sous lequel il est mis à disposition vous donne un certain nombre de droits, mais vous impose également de respecter un certain nombre de conditions :

Les droits

Vous êtes libre de reproduire, distribuer et communiquer cet ouvrage, tel quel ou après modification. L'ouvrage peut vous être fourni dans un format numérique modifiable sur simple demande, à envoyer à innovation@lpi-francophonie.org.

Les conditions à respecter

BY = Paternité (*by*) : les noms des auteurs et éditeurs de l'ouvrage devront toujours être mentionnés, en utilisant le modèle donné (*cf.* page précédente), ceci même si vous apportez des modifications et, dans ce cas, d'une manière qui ne risque pas de suggérer qu'ils soutiennent ou approuvent les modifications apportées ;

NC = Pas d'utilisation commerciale (*Non Commercial*) **:** toute diffusion payante, même après modification, est interdite ;

SA = Partage des conditions initiales à l'identique (*Share Alike*) : si vous modifiez, transformez ou adaptez cet ouvrage, vous n'avez le droit de distribuer la création qui en résulte qu'en donnant les mêmes droits, et sous les mêmes conditions.

À chaque réutilisation ou distribution de cet ouvrage, ou de toute œuvre qui en serait dérivée, vous devez faire apparaître clairement au public les conditions contractuelles de sa mise à disposition. La meilleure manière de les indiquer est un lien vers cette page web :

http://creativecommons.org/licenses/by-nc-sa/2.0/fr/

Chacune de ces conditions peut être levée si vous obtenez l'autorisation du titulaire des droits sur cette œuvre.

Rien dans ce contrat ne diminue ni ne restreint le droit moral de l'auteur.

Table des matières

Introduction

La certification de l'**Institut professionnel de Linux** – *Linux professional Institute (LPI)* – permet de valider les connaissances et l'expérience des administrateurs systèmes et réseaux qui travaillent avec le système d'exploitation GNU/Linux.

Cet ouvrage permet de préparer l'**examen 201**, qui constitue le premier examen à passer pour obtenir la certification de niveau 2 – *Advanced Level Linux Certification* – abrégée en « **LPIC-2 »**.

Il est publié par l'Agence universitaire de la Francophonie (AUF) dans le cadre du **LPI Francophonie**. Il a reçu le label « Support de formation agréé Institut professionnel de Linux » (LATM, *LPI Approved Training Material*).

Il prend en compte les objectifs détaillés de l'examen 201 mis à jour en août 2012 :

- version originale sur le site du LPI *Exam 201 : Detailed Objectives* : www.lpi.org/linux-certifications/programs/lpic-2/exam-201/, mise à jour août 2012 : www.lpi.org/content/exam-201-objective-changes-august-1-2012 ;
- traduction en français sur le site du LPI Francophonie : www.lpi-francophonie.org/spip.php?rubrique20

Chaque chapitre traite d'un sujet du programme de certification.

Pourquoi une certification Linux ?

Les objectifs de la certification créée par le LPI sont multiples. En voici quelques uns :

- pouvoir répondre aux détracteurs des logiciels libres en démontrant que la communauté du logiciel libre est capable de s'organiser ;
- donner aux employeurs un outil permettant de juger les connaissances et l'expérience d'une personne ;
- fournir aux centres de formations une structure commune pour l'enseignement de l'administration système/réseau basée sur l'utilisation de GNU/Linux.

Par la création d'une certification, l'idée est également de participer à la promotion de l'utilisation des logiciels libres et à son développement, en particulier du système d'exploitation GNU/Linux dans le domaine de l'administration « système/réseau ».

Une certification indépendante fonctionnant sur le modèle du logiciel libre.

La certification LPI valide les connaissances et l'expérience acquises par les administrateurs utilisant les logiciels libres associés au système GNU/Linux.

Elle est indépendante des différentes distributions GNU/Linux, même si de nombreux acteurs du logiciel libre sont partenaires de l'initiative.

La communauté du logiciel libre est associée au programme de la certification. Son évolution, sa réactivité et son indépendance sont ainsi garanties.

Le LPI, un organisme neutre fondé par la communauté du logiciel libre.

Le LPI est une association à but non lucratif basée au Canada. Il est soutenu par une large communauté de clients d'entreprises, de gouvernements, de centres d'examen, d'éditeurs de livres, de fournisseurs de supports pédagogiques et d'établissements éducatifs et de formation dans le monde.

Le LPI ne prépare pas à la certification, il n'a pas vocation à être un centre de formation ni à vendre des supports de formation. Il délivre toutefois des agréments de qualité pour les centres de formation et pour les contenus pédagogiques qui préparent à ses certifications. Son action reste prioritairement concentrée sur la création et la gestion des certifications. Les certifications représentent son seul « capital ».

Le LPI présente les premières certifications dans les technologies de l'information ayant obtenu une accréditation professionnelle. Il favorise ainsi l'adoption et le développement de normes ouvertes en association avec les acteurs spécialisés du domaine. Il participe au développement d'outils se basant sur des logiciels libres pour faire progresser les procédures de développement des examens.

Le LPI Francophonie.

L'Agence universitaire de la Francophonie (AUF) et le LPI ont créé le LPI Francophonie en 2003.

Ce partenariat a permis d'organiser des sessions de préparation à la certification LPI via les Centres Linux et logiciels libres pour le développement (C3LD). Un des objectifs est de promouvoir l'usage des logiciels libres et la certification des compétences humaines.

Chapitre 1. Le noyau Linux

Objectifs

Composants du noyau

– identifier les versions d'un noyau stable ou en développement ;

– utiliser les composants du noyau qui sont nécessaires aux matériels spécifiques, pilotes, ressources et besoins du système ;

– identifier les différents types d'images du noyau.

Compilation d'un noyau

– personnaliser la configuration du noyau ;

– compiler un noyau Linux 2.6 en incluant ou désactivant des composants spécifiques du noyau.

Mise à jour d'un noyau

– appliquer les mises à jour du noyau Linux pour accueillir de nouveaux périphériques ;

– désinstaller correctement des mises à jour.

Personnalisation, construction et installation d'un noyau et des modules noyau

– personnaliser et construire un noyau 2.6 pour des besoins spécifiques du système, par la mise à jour, la compilation ou la modification des fichiers de configuration ;

– construire et configurer les modules du noyau ;

– créer une image d'initialisation système (`initrd`) et installer un nouveau noyau.

Gestion/interrogation du noyau et des modules noyau en exécution

– gérer et interroger les modules du noyau 2.6.x ;

– charger et décharger manuellement les modules du noyau.

Points importants

– Fichiers de configuration de GRUB.

– Les cibles de la commande make pour le noyau 2.6.x.

– Les cibles de la commande make pour le noyau 2.6.x.

– Le fichier Makefile.

– Personnalisation de la configuration du noyau courant.

– Construction d'un nouveau noyau et des modules noyau appropriés.

– Installation d'un nouveau noyau et de tout module nécessaire.

– Localisation du chargeur de démarrage du nouveau noyau et des

	fichiers associés. – /usr/src/linux/ – Fichiers de configuration des modules. – Utilisation des commandes de récupération des informations sur le noyau et sur ses modules en cours d'exécution. – Chargement et déchargement manuels des modules du noyau. – Détermination des paramètres acceptés par le module.
Mots clés	/usr/src/linux, zImage, bzImage, mkinitrd, mkinitramfs, make, patch, depmod, insmod, lsmod, rmmod, modinf, modprobe, uname Les cibles de make : config, xconfig, menuconfig, oldconfig, mrproper, zImage, bzImage, modules, modules_install

Une des caractéristiques les plus intéressantes du noyau Linux est qu'il n'est pas un produit commercial, il est issu d'un projet collaboratif développé sur Internet. Bien que Linus Torvalds, qui a développé la première version du noyau Linux, reste le premier responsable de sa maintenance, il peut s'appuyer sur une très large communauté de contributeurs.

Le noyau Linux est diffusé sous licence GNU General Public License (GPL) version 2.0. On peut ainsi télécharger le code source, et également le modifier.

Le noyau Linux est développé essentiellement en langage C, ce qui permet d'offrir une couche d'abstraction logicielle qui cache les particularités matérielles de bas niveau.

1. Les versions du noyau Linux

Le premier noyau Linux a été créé en 1991 par Linus Torvalds. Il a évolué au cours de temps et évolue encore.

Jusqu'à la version 2.6, les différentes versions du noyau Linux sont désignées par un numéro composé de trois séquences de chiffres délimitées par un point :

- la première séquence est le numéro de la version **majeure**. Ce numéro a été modifié uniquement pour des changements majeurs dans le code et le concept du noyau, en 1994 (version 1.0) et en 1996 (version 2.0) ;
- la deuxième séquence est le numéro de la version **mineure**. Un nombre pair indique qu'il s'agit d'un noyau stable, un nombre impair qu'il s'agit d'un noyau en développement. Ainsi, les versions 2.2 et 2.4 sont des versions stables, les versions 2.3 et 2.5 des versions en développement. Les versions stables sont destinées à être déployées dans des environnements de production. Elles sont livrées pour fournir des corrections ou des nouveaux pilotes de périphériques. Les versions en développement sont destinées aux tests de nouvelles fonctionnalités ;
- la troisième séquence est le numéro de **révision**. Ce numéro est incrémenté chaque fois qu'une nouvelle version du noyau est diffusée, que ce soit pour des correctifs de sécurité, des corrections de bogues, l'ajout de nouvelles fonctionnalités ou de

nouveaux pilotes.

En 2004, après la publication de la version 2.6.0 du noyau Linux, les développeurs du noyau décident d'abandonner ce modèle de numérotation stable/développement. Ils estiment en effet que le noyau 2.6 est suffisamment mature et stable et que les nouvelles fonctionnalités, qui risqueraient de le déstabiliser, sont inutiles. Les numéros mineurs pairs ou impairs n'ont donc plus de signification particulière. La série des noyaux 2.6 est prolongée et l'introduction de la série des noyaux 2.7 en développement abandonnée. Le cycle de développement de chaque révision 2.6 est plus rapide, chaque version comportant une série de mini-développements.

Depuis lors, les numéros de versions sont composés de la façon suivante :

- les deux premiers numéros, « 2.6 », sont restés inchangés depuis 2003 ;
- le troisième numéro est la version courante du noyau ;
- les développeurs du noyau ont ensuite introduit une quatrième séquence, le numéro de la version stable. Sa première utilisation date de la version du noyau 2.6.8, quand une grave erreur, qui exigeait une correction immédiate, a été identifiée dans l'implémentation du protocole NFS. Comme il n'y avait pas assez de changements pour justifier la publication d'une nouvelle version du noyau (qui aurait été la 2.6.9), la version 2.6.8.1 a été publiée, avec pour seul changement le correctif de cette erreur.

À partir de la version 2.6.11, cette quatrième séquence a été officiellement adoptée dans la nomenclature des versions du noyau Linux pour indiquer que des corrections d'erreurs et des correctifs de sécurité ont été apportés à la version de base du noyau.

Le 29 mai 2011, Linus Torvalds annonça la version 3.0 du noyau, en l'honneur du 20^{e} anniversaire de Linux. Cette version 3.0 n'est qu'une simple évolution de la 2.6.39.

2. Modification et configuration du noyau

Le noyau Linux est diffusé avec une configuration générique conçue pour supporter n'importe quelle application sur n'importe quel matériel.

Cette configuration générique comprend de nombreux pilotes de périphériques, mais aussi des paramètres pour le noyau. On peut modifier ces paramètres afin d'adapter le noyau à des besoins spécifiques, augmenter les performances, renforcer la sécurité, ou encore la fiabilité du système.

Dans certains cas, la modification du noyau est nécessaire afin d'ajouter de nouveaux pilotes de périphériques. Le code source du pilote ajouté doit être intégré dans les structures de données du noyau. Ceci peut exiger la re-compilation du noyau.

Il existe quatre méthodes pour intervenir sur la configuration d'un noyau Linux :

- modification dynamique des paramètres de configuration du noyau ;
- construction d'un noyau à partir de zéro (compilation du code source, avec éventuellement des modifications et des ajouts) ;

- chargement de modules dans un noyau existant, à la volée ;
- passage de paramètres en utilisant le chargeur de démarrage : LILO ou GRUB.

Ces méthodes sont applicables dans des situations différentes. La modification dynamique des paramètres est la plus facile et la plus courante, tandis que la construction d'un noyau à partir des fichiers sources est la plus difficile et la moins souvent nécessaire.

3. La modification dynamique des paramètres du noyau

Le noyau peut être ajusté dynamiquement à travers des paramètres du système. Ces paramètres sont accessibles et modifiables à partir des fichiers du répertoire `/proc/sys`.

EXEMPLE

Pour modifier le nombre maximal de fichiers que le système peut ouvrir simultanément, on peut modifier le fichier `/proc/sys/fs/file-max` de la façon suivante :

```
# echo 32768 > /proc/sys/fs/file-max
```

Quelques autres paramètres ajustables du noyau sont décrits dans le *tableau 1*.

Tableau 1. Quelques paramètres ajustables du noyau Linux

Fichier	Description
`/proc/sys/fs/file-max`	Indique le nombre maximal de fichiers que le noyau peut manipuler simultanément
`/proc/sys/kernel/ctrl-alt-del`	Contrôle la gestion de la séquence Ctrl-Alt-Supp du clavier. S'il contient la valeur zéro, Ctrl-Alt-Supp est capturé et envoyé au programme `init` pour relancer le système correctement.
`/proc/sys/net/ipv4/icmp_echo_ignore_all`	Bloque les réponses au ping.
`/proc/sys/net/ipv4/icmp_echo_ignore_broadcasts`	Ignore les messages de diffusion ICMP (Internet Control Message Protocol)
`/proc/sys/net/ipv4/ip_forward`	Active ou désactive le relayage (forwarding) entre les cartes réseaux. Activer le relayage est nécessaire pour faire fonctionner le système comme un routeur.
`/proc/sys/kernel/hostname`	Permet de changer le nom de la machine.

Puisque le système de fichiers `/proc` est virtuel et n'a d'existence qu'au niveau de la mémoire centrale, la modification des fichiers qu'il contient est temporaire, et sera perdue lors du redémarrage du système.

Pour mémoriser les paramètres à appliquer au démarrage du système, on utilise le fichier `/etc/sysctl.conf`.

Si le fichier `/etc/sysctl.conf` contient la ligne :

```
net.ipv4.ip_forward = 0
```

alors au démarrage du système le paramètre `ip_forward` aura pour valeur 0, donc le relayage ne sera pas activé.

En cours d'exécution, les paramètres du noyau peuvent aussi être modifiés par la commande `sysctl` :

```
# sysctl –w net.ipv4.ip_forward=1
```

Cette commande donne au paramètre `ip_forward` la valeur 1, ce qui a pour effet d'activer le relayage.

4. Construction d'un noyau Linux

On peut être amené à compiler et installer un nouveau noyau Linux pour intégrer des correctifs (*patchs*), des pilotes de périphériques ou de nouvelles fonctionnalités. Le code source et les correctifs des noyaux Linux sont disponibles sur le site www.kernel.org.

On doit peser les besoins et les risques lors de la planification des améliorations et des correctifs du noyau. Bien sûr, la nouvelle version peut apporter des fonctionnalités attrayantes, mais est-elle aussi stable que la version courante ? Une bonne règle est de mettre à jour le noyau ou d'appliquer les correctifs lorsque des gains en terme de productivité (souvent mesurés en terme de fiabilité et de performance) dépassent l'effort et la perte de temps nécessaires pour effectuer ces mises à jour. Si on rencontre des difficultés à quantifier ce gain, c'est signe que le correctif peut attendre un autre jour.

EXEMPLE

Dans l'exemple ci-dessous, le code source du noyau 2.6.35 a été téléchargé, à partir du lien www.kernel.org/pub/linux/kernel/v2.6, sous forme d'une archive nommée `tar.bz2`. Le répertoire `linux-2.6.35.5/` est créé à l'issue de la décompression et de l'extraction de cet archive par la commande `tar xjfv linux-2.6.35.5.tar.bz2`. À l'intérieur de ce répertoire se trouve l'arborescence du code source du noyau Linux 2.6.35.

```
$ tar xjfv linux-2.6.35.5.tar.bz2
$ cd linux-2.6.35.5/
$ ls
arch   COPYING  crypto         drivers   fs       init  Kbuild  lib
Makefile  net     REPORTING-BUGS  scripts   sound  usr
block  CREDITS  Documentation  firmware  include  ipc   kernel  MAINTAINERS
mm        README  samples         security  tools  virt
```

4.1. Configuration du noyau

Les informations sur la configuration du noyau sont stockées dans le fichier `.config` situé à la racine du répertoire source du noyau.

Il est déconseillé de modifier ce fichier manuellement. Linux offre plusieurs outils permettant de configurer le noyau et d'écrire le résultat dans le fichier `.config`.

4.1.1. Make config

L'outil le plus élémentaire est `make config`. Il s'exécute en mode console. L'utilisateur doit ensuite spécifier toutes les options de configuration. `make config` demande pour chaque fonction si elle doit ou non être activée, en proposant quatre choix possibles sous la forme [Y / m / n /?] :

- `Y` (« yes ») pour intégrer la fonction directement dans le noyau. C'est le choix par défaut, il peut être sélectionné en appuyant simplement sur la touche **[Entrée]** ;
- `m` (« module ») pour construire un module qui va être chargé de façon dynamique ;
- `n` (« no ») pour ne pas activer la fonction ;
- `?` pour afficher un message décrivant la fonction.

4.1.2. Make oldconfig

Le noyau contient près de deux mille options de configuration, et répondre à toutes les questions prend beaucoup de temps. Heureusement, il y a un moyen plus rapide de configurer le noyau : utiliser l'outil `make oldconfig` qui construit une configuration basée sur une configuration pré-construite.

EXEMPLE

On va construire une configuration pour le nouveau noyau 2.6.35.5 à partir de la configuration pré-construite pour le noyau courant (version 2.6.32.24) :

```
$ make oldconfig
scripts/kconfig/conf -o arch/x86/Kconfig
#
# using defaults found in /boot/config-2.6.32-24-generic
#
/boot/config-2.6.32-24-generic:556:warning: symbol value 'm' invalid for
PCCARD_NONSTATIC
/boot/config-2.6.32-24-generic:3031:warning: symbol value 'm' invalid for
MFD_WM831X
/boot/config-2.6.32-24-generic:3032:warning: symbol value 'm' invalid for
MFD_WM8350
/boot/config-2.6.32-24-generic:3033:warning: symbol value 'm' invalid for
MFD_WM8350_I2C
/boot/config-2.6.32-24-generic:3038:warning: symbol value 'm' invalid for
```

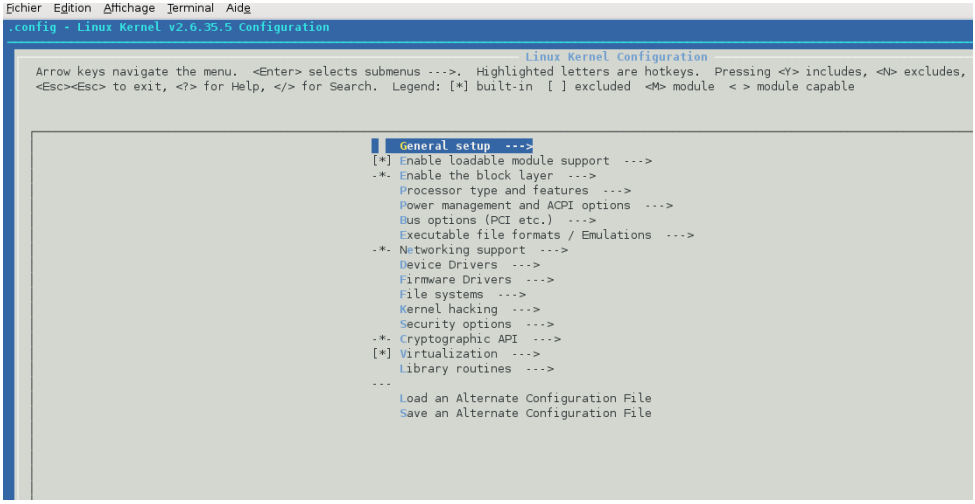
```
AB3100_CORE
/boot/config-2.6.32-24-generic:3505:warning: symbol value 'm' invalid for
FB_VESA
/boot/config-2.6.32-24-generic:4174:warning: symbol value 'm' invalid for
MMC_RICOH_MMC
*
* Restart config...
*
*
* General setup
*
Prompt for development and/or incomplete code/drivers (EXPERIMENTAL)
[Y/n/?] y
Cross-compiler tool prefix (CROSS_COMPILE) [] (NEW)
```

4.1.3. Make menuconfig

Il existe trois autres outils interactifs pour la configuration du noyau.

Le premier outil est `make menuconfig`, il permet une configuration en mode console, la navigation entre les options de configuration est faite à l'aide des touches fléchées du clavier (*figure 1*).

Figure 1. Outil `make menuconfig`



La console de l'outil `make menuconfig` est subdivisée en plusieurs sections. Chaque section contient des options qui correspondent à un thème spécifique. Par exemple la section

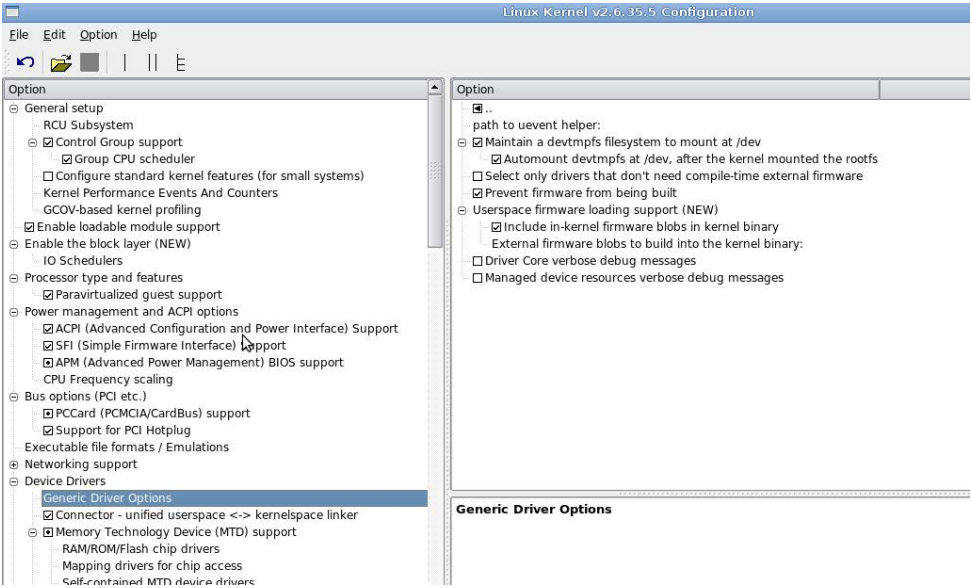
Device Drivers contient des options concernant les pilotes des périphériques.

4.1.4. Make gconfig, make xconfig

Les deux autres outils, `make gconfig` et `make xconfig`, utilisent des interfaces graphiques permettant de modifier la configuration du noyau. Ces deux méthodes sont presque identiques, la seule différence est la bibliothèque d'outils graphiques avec laquelle elles sont écrites. L'outil `make gconfig` est écrit avec GTK+ et l'outil `make xconfig` est écrit avec QT.

Pour ces deux outils on utilise la souris pour naviguer dans les menus et sélectionner des options. Par exemple, la *figure* 2 illustre la fenêtre « xconfig Generic Driver Options » de la section Device Drivers.

Figure 2. Fenêtre « *xconfig Generic Driver Options* »



Notons qu'avec l'outil `gconfig`, une case cochée signifie que l'option sera intégrée au noyau, tandis qu'une ligne dans la case signifie que l'option sera construite comme un module. Avec l'outil `xconfig`, une option intégrée comme un module est représentée par un point dans une case.

4.2. Compilation du noyau

Une fois que la configuration du noyau est créée, on peut compiler le noyau en utilisant la commande `make` :

```
$ make
  CHK     include/linux/version.h
  UPD     include/linux/version.h
  CHK     include/generated/utsrelease.h
  UPD     include/generated/utsrelease.h
  CC      kernel/bounds.s
  GEN     include/generated/bounds.h
  CC      arch/x86/kernel/asm-offsets.s
  GEN     include/generated/asm-offsets.h
  CALL    scripts/checksyscalls.sh
  HOSTCC  scripts/genksyms/genksyms.o
  SHIPPED scripts/genksyms/lex.c
  SHIPPED scripts/genksyms/parse.h
  SHIPPED scripts/genksyms/keywords.c
  HOSTCC  scripts/genksyms/lex.o
  SHIPPED scripts/genksyms/parse.c
  HOSTCC  scripts/genksyms/parse.o
  HOSTLD  scripts/genksyms/genksyms
  CC      scripts/mod/empty.o
  HOSTCC  scripts/mod/mk_elfconfig
  MKELF   scripts/mod/elfconfig.h
  HOSTCC  scripts/mod/file2alias.o
  HOSTCC  scripts/mod/modpost.o
  HOSTCC  scripts/mod/sumversion.o
  HOSTLD  scripts/mod/modpost
  HOSTCC  scripts/selinux/genheaders/genheaders
  HOSTCC  scripts/selinux/mdp/mdp
```

Cette commande a pour effet de compiler le noyau avec la configuration définie dans l'étape précédente, ainsi que tous les modules nécessaires à cette configuration.

Lors de la compilation du noyau, chaque fichier source compilé est affiché individuellement, avec des messages d'avertissement ou d'erreur éventuels.

Si la compilation du noyau se termine sans erreur, le résultat est un fichier binaire – le fichier image du noyau – qui doit être installé avant qu'on puisse l'utiliser au démarrage de la machine.

4.3. Installation et démarrage du nouveau noyau

Maintenant que le noyau est sous forme d'un fichier binaire, ainsi que les modules que le noyau va utiliser de façon dynamique, il est temps d'installer le nouveau noyau et d'essayer de le démarrer.

Dans cette étape, à la différence des étapes précédentes, toutes les commandes doivent être exécutées avec les droits root. L'installation du nouveau noyau peut être réalisée soit

en utilisant des scripts offerts par la distribution installée, soit de façon manuelle.

4.3.1. Utilisation des scripts d'installation d'une distribution

La plupart des distributions Linux sont livrées avec un script appelé `installkernel` qui peut être utilisé lors de l'installation d'un nouveau noyau compilé. Ce script permet de copier le nouveau noyau dans le répertoire approprié et de modifier le chargeur de démarrage afin que ce nouveau noyau puisse être sélectionné lors du démarrage du système.

Si on a choisi d'utiliser des modules externes dynamiques, on doit d'abord les installer avec la commande :

```
# make modules_install
```

Ceci permet de placer tous les modules compilés dans le répertoire approprié dans l'arborescence standard Linux, où le noyau pourra les retrouver en cas de besoin. Dans notre exemple les modules dynamiques sont placés dans le répertoire `/lib/modules/2.6.35.5`.

Une fois les modules installés correctement, l'image du nouveau noyau doit être aussi installée par la commande :

```
# make install
```

Cette commande déclenche le processus suivant :

1. Vérification de la bonne construction du nouveau noyau.
2. Installation de la partie statique du noyau dans le répertoire `/boot`
3. Création de toutes les images RAM Disk nécessaires[1], en utilisant les modules installés lors de l'exécution de la commande `make modules_install`.
4. Ajout d'une entrée dans le fichier de configuration du chargeur de démarrage pour le nouveau noyau.

Une fois ce processus terminé, le noyau est installé avec succès, on peut donc redémarrer et tester la nouvelle image du noyau. Notons ici que cette installation n'écrase pas l'ancienne image du noyau. Si un problème survient on peut y revenir au moment du démarrage.

4.3.2. Installation manuelle

Si la distribution ne dispose pas du script `installkernel`, ou si l'on souhaite tout simplement faire l'installation manuellement, on doit procéder de la manière suivante :

Installer les modules :

[1] Image RAM Disk (ou « image intrd ») : voir explication dans la section *4.3.2 Installation manuelle*.

```
# make modules_install
```

Le noyau doit être copié dans le répertoire `/boot`. Pour un noyau basé sur l'architecture 386, on exécute la commande :

```
# cp arch/i386/boot/bzImage /boot/vmlinuz-2.6.35.5
# cp System.map /boot/System.map-2.6.35.5
```

On doit ensuite modifier le fichier de configuration du chargeur de démarrage afin qu'il puisse reconnaître le nouveau noyau.

Si le processus de démarrage ne fonctionne pas correctement, c'est généralement parce qu'une image **initrd** est nécessaire.

initrd est une abréviation de Initial RAM Disk (disque virtuel initial). Cette image initrd est utilisée par le noyau pour charger les pilotes avant le démarrage du système. Le but est de permettre aux utilisateurs de construire des noyaux modulaires. Ceci permet par exemple de démarrer à partir de n'importe quel disque SCSI avec des noyaux ne contenant pas tous les pilotes des contrôleurs SCSI. Dans ce cas, l'image initrd contient les pilotes SCSI nécessaires et tous les autres pilotes nécessaires pour le démarrage du système.

Une image initrd est généralement construite par la commande `mkinitrd`. Pour la distribution Red Hat et ses dérivées on exécute la commande :

```
mkinitrd -v /boot/initrd-2.6.35.5.img 2.6.35.5
```

Pour la distribution Debian et ses dérivées on exécute la commande :

```
mkinitrd -o /boot/initrd-2.6.35.5.img /lib/modules/2.6.35.5
```

5. Mise à jour d'un noyau

On peut être confronté à la question de mise à niveau d'un noyau personnalisé suite à la correction de quelques bugs dans la dernière version du noyau, un problème de sécurité corrigé, ou autre. On ne veut pas perdre le temps déjà déployé pour la personnalisation et la compilation du noyau.

Il est facile de mettre à jour un noyau d'une version ancienne tout en conservant toutes les options de la configuration précédente.

Tout d'abord, dans le répertoire source du noyau on sauvegarde le fichier `.config`. On a consacré du temps des efforts à sa création, il est important de pouvoir disposer d'une sauvegarde en cas de problème.

```
$ cd linux-2.6.35.5
$ cp .config ../good_config
```

Cinq étapes sont nécessaires pour mettre à niveau un noyau déjà compilé et installé :

1. Obtenir le nouveau code source.
2. Appliquer les modifications sur l'ancienne arborescence du code source du noyau pour le mettre à niveau.
3. Reconfigurer le nouveau noyau en se basant sur la configuration précédente.
4. Compiler le nouveau noyau.
5. Installer le nouveau noyau.

Les trois dernières étapes sont identiques à celles décrites précédemment, seules les deux premières sont donc présentées ici.

La version 2.6.35.5 du noyau de Linux est déjà compilée et installée. Il s'agit maintenant de mettre à jour ce noyau vers la version 2.6.35.7.

Un fichier de patch, ou « patch », est appliqué sur l'arborescence du code source existant, créant ainsi une nouvelle arborescence. Le *patch* contient les changements à apporter aux anciens fichiers, et de nouveaux fichiers.

Voici comment les *patchs* peuvent être appliqués :

- les *patchs* du noyau stable peuvent être appliqués sur une version de base du noyau, par exemple le *patch* 2.6.35.6 peut être appliqué seulement sur la version 2.6.35 du noyau. Ainsi, il ne peut pas être appliqué sur la précédente version du noyau (version 2.6.35.5) ;
- un *patch* d'une version de base ne peut être appliqué que sur un noyau de la version de base précédente. Cela signifie que le patch 2.6.36 ne s'applique qu'à la version 2.6.35 du noyau. Il ne s'appliquera pas à la dernière version stable du noyau 2.6.35.y ni à toute autre version ;
- les patchs incrémentaux permettent la mise à niveau d'une version donnée vers la version qui la suit immédiatement. Ainsi les développeurs ne sont pas obligés de télécharger un noyau, puis de le mettre à jour, juste pour passer d'une version stable à la suivante (on rappelle que les patchs d'une version stable ne s'appliquent que sur une version de base du noyau, et non sur la précédente version stable).

EXEMPLE

On veut passer de la version 2.6.35.5 à la version 3.6.35.7. On a donc besoin de télécharger deux patchs : un patch pour passer de la version 2.6.35.5 à la version 2.6.35.6, puis un autre patch pour passer de la version 2.6.35.6 à la version 2.6.35.7.

On les trouve à la page www.kernel.org/pub/linux/kernel/v2.6/incr/.

Comme ils sont compressés, la première chose à faire est de les décompresser avec la commande `bzip2` :

```
$ bzip2 -dv patch-2.6.35.5-6.bz2
```

```
patch-2.6.17.9-10.bz2: done
$ bzip2 -dv patch-2.6.35.6-7.bz2
```

Maintenant, il faut appliquer les fichiers de patch dans le répertoire du noyau. On se place dans le répertoire `linux-2.6.35.5` et on lance la commande `patch` :

```
$ cd linux-2.6.35.5
$ patch -p1 < ../patch-2.6.35.5-6
patching file Makefile
patching file arch/alpha/kernel/err_marvel.c
patching file arch/alpha/kernel/proto.h
patching file arch/alpha/kernel/sys_cabriolet.c
patching file arch/alpha/kernel/sys_takara.c
patching file arch/arm/mach-at91/at91sam9g45_devices.c
patching file arch/ia64/kernel/fsys.S
patching file arch/sparc/include/asm/oplib_64.h
patching file arch/sparc/prom/cif.S
patching file arch/sparc/prom/console_64.c
patching file arch/sparc/prom/devops_64.c
patching file arch/sparc/prom/misc_64.c
patching file arch/sparc/prom/p1275.c
patching file arch/sparc/prom/tree_64.c
patching file arch/x86/kvm/emulate.c
patching file arch/x86/kvm/mmu.c
patching file arch/x86/kvm/paging_tmpl.h
```

On peut examiner le fichier `Makefile` pour voir le changement de version du noyau :

```
$ head Makefile
VERSION = 2
PATCHLEVEL = 6
SUBLEVEL = 35
EXTRAVERSION = .6
NAME = Sheep on Meth
# *DOCUMENTATION*
# To see a list of typical targets execute "make help"
# More info can be located in ./README
# Comments in this file are targeted only to the developer, do not
```

On utilise à nouveau la commande `patch` pour passer de la version 2.6.35.6 à la version 2.6.35.7 :

```
$ patch -p1 < ../patch-2.6.35.6-7
patching file Makefile
patching file drivers/xen/events.c
```

```
$ head Makefile
VERSION = 2
PATCHLEVEL = 6
SUBLEVEL = 35
EXTRAVERSION = .7
NAME = Yokohama
```

6. Modules du noyau

Le noyau Linux supporte un chargement dynamique des modules : il est possible de les charger ou de les supprimer quand le noyau est en cours d'exécution.

Les modules du noyau sont stockés dans le répertoire `/lib/modules/version_noyau`, où `version_noyau` est la version du noyau Linux en cours (retournée par la commande `uname -r`). On peut lister les modules présents avec la commande `lsmod` :

```
$ lsmod
Module                  Size  Used by
nls_utf8                1069  0
isofs                  29250  0
nls_iso8859_1           3249  2
nls_cp437               4919  2
vfat                    8933  2
fat                    47767  1 vfat
usb_storage            39425  2
binfmt_misc             6587  1
ppdev                   5259  0
vboxnetadp              6326  0
vboxnetflt             15280  0
vboxdrv               190594  2 vboxnetadp,vboxnetflt
joydev                  8708  0
snd_hda_codec_atihdmi     2367  1
snd_hda_codec_conexant    22641  1
snd_hda_intel          21941  4
```

La commande `insmod` permet de charger manuellement un module du noyau. Dans l'exemple suivant on ajoute le module *msdos* :

```
# insmod /lib/modules/2.6.35-22-generic/kernel/fs/fat/msdos.ko
insmod: error inserting '/lib/modules/2.6.35-22-
generic/kernel/fs/fat/msdos.ko': -1 Unknown symbol in module
```

La commande a retourné une erreur indiquant qu'un symbole n'est pas résolu. En effet lors du chargement de ce module des variables ou des fonctions n'ont pas été retrouvées. Ce

module dépend nécessairement d'autres modules et on peut voir cette dépendance par la commande `modinfo` :

```
# modinfo /lib/modules/2.6.35-22-generic/kernel/fs/fat/msdos.ko
filename:       /lib/modules/2.6.35-22-generic/kernel/fs/fat/msdos.ko
description:    MS-DOS filesystem support
author:         Werner Almesberger
license:        GPL
srcversion:     44046DD818C31AAF3D90191
depends:        fat
vermagic:       2.6.35-22-generic SMP mod_unload modversions 686
```

Ainsi le module *msdos* dépend du module fat qu'il faut donc charger en premier :

```
# insmod /lib/modules/2.6.35-22-generic/kernel/fs/fat/fat.ko
```

Et maintenant le module *msdos* peut être chargé :

```
# insmod /lib/modules/2.6.35-22-generic/kernel/fs/fat/msdos.ko
# lsmod | grep fat
fat                    48240  1 msdos
```

La commande `rmmod` est utilisée pour supprimer des modules du noyau en cours d'exécution.

```
# rmmod fat
ERROR: Module fat is in use by msdos
```

Le message d'erreur indique que la suppression du module *fat* échoue parce qu'il est en cours d'utilisation par le module *msdos*. Il faut donc commencer par supprimer le module *msdos* :

```
# rmmod msdos
# rmmod fat
```

La commande `modprobe` peut déterminer les dépendances entre les modules et installer automatiquement les modules nécessaires. Pour ce faire, `modprobe` examine le fichier `/lib/modules/version_noyau/modules.dep`.

```
# modprobe msdos
# lsmod | grep fat
fat                    48240  1 msdos
```

Les lignes dans le fichier `modules.dep` sont sous la forme :

`nom_module.ko : dependance1 dependance2 ….`

Par exemple :

```
$ grep msdos /lib/modules/2.6.35.5/modules.dep
kernel/fs/fat/msdos.ko: kernel/fs/fat/fat.ko
```

Le fichier `modules.dep` doit être cohérent afin d'assurer le bon fonctionnement de la commande `modprobe`. Il est généralement régénéré à chaque démarrage du système par la commande `depmod -a` placée dans les scripts de démarrage.

Le fichier `/etc/modprobe.conf` (ou les fichiers sous le répertoire `/etc/modprobe.d`) est le fichier de configuration de la commande `modprobe`. Il sert à changer le comportement de `modprobe` lors du chargement d'un ou de plusieurs modules. Il est possible d'y définir des alias pour les noms de modules, les options des modules ou encore de lancer des commandes au lieu de charger ou décharger un module.

7. Passage de paramètres au noyau à partir du chargeur de démarrage

On peut avoir besoin de passer quelques options au noyau Linux lors du démarrage du système, telles que la partition racine qu'il doit utiliser ou les cartes Ethernet multiples qu'il doit détecter. Le chargeur de démarrage (LILO ou GRUB) est responsable du transfert de ces options au noyau.

Pour que ces options soient prises en compte à chaque démarrage du système, on doit les ajouter dans les fichiers de configurations `/etc/lilo.conf` ou `/boot/grub/grub.conf`, selon le chargeur de démarrage utilisé.

On peut aussi utiliser l'invite de commandes du chargeur de démarrage.

EXEMPLE

Pour indiquer à LILO de charger le noyau spécifié par l'étiquette « linux », d'utiliser la partition racine `/dev/sda1` et de détecter deux cartes Ethernet, on utilise la commande LILO suivante:

```
LILO: linux root=/dev/sda1 ether=0,0,eth0 ether=0,0,eth1
```

Pour indiquer la même chose à GRUB on utilise la commande :

```
grub> kernel /vmlinuz root=/dev/sda1 ether=0,0,eth0 ether=0,0,eth1
grub> boot
```

Exercices

1. **Laquelle des commandes suivantes, lorsqu'elle est exécutée dans le répertoire `/usr/src/linux` après la configuration du noyau, permet de compiler le noyau Linux et ses principaux modules ?**

   - ❑ A. `make bzImage`
   - ❑ B. `make modules`
   - ❑ C. `make xconfig`
   - ❑ D. `make`

2. **Laquelle des commandes suivantes pourriez vous taper dans `/usr/src/linux` après avoir copié le fichier de configuration à partir d'un vieux noyau pour appliquer les options de l'ancien noyau dans le nouveau noyau ?**

   - ❑ A. `make config`
   - ❑ B. `make allmodconfig`
   - ❑ C. `make oldconfig`
   - ❑ D. `make mrproper`

3. **Vous avez configuré et compilé un nouveau noyau, sa version 2.6.35.4. Vous avez maintenant tapé `make modules_install`. Où pouvez-vous trouver les fichiers de modules ?**

   - ❑ A. `/lib/modules/modules-2.6.35.4`
   - ❑ B. `/usr/src/linux/2.6.35.4`
   - ❑ C. `/lib/modules/2.6.35.4`
   - ❑ D. `/usr/lib/2.6.35.4`

Chapitre 2. Démarrage du système

Objectifs	**Personnalisation du processus de démarrage système** – interroger et modifier le comportement des services systèmes dans différents niveaux d'exécution ; – configurer la procédure de démarrage des services. **Récupération du système** – manipuler la séquence de démarrage et le mode récupération ; – utiliser les outils init et les options du noyau relatives à init.
Points importants	– Spécification du standard LSB (*Linux Standard Base*). – Shell Grub. – Commandes de gestion de service.
Mots clés	`/etc/inittab`, `/etc/init.d/`, `/etc/rc.d/`, `chkconfig`, `update-rc.d`, `mount`, `fsck`, `init`, `telinit`

Le démarrage du système implique quatre programmes lancés successivement : le BIOS (*Basic Input Output System*), le chargeur de démarrage, le noyau et le processus `init`.

Le BIOS est le premier programme exécuté, il réside au niveau du ROM. Il charge les 512 premiers octets, ces 512 octets constituent le secteur d'amorçage ou le (Master Boot Record).

Le MBR contient plusieurs informations sur les partitions du disque. Il inclut aussi le chargeur de démarrage (ou une partie du chargeur de démarrage).

Le chargeur de démarrage Grub (Grand Unified Bootloader) est utilisé sur la plupart des distributions Linux. Il est décomposé en deux parties. La première partie réside sur le MBR, elle charge la deuxième partie qui se trouve dans une partition du disque. Une fois que la deuxième partie du Grub est chargée, une interface est affichée permettant à l'utilisateur de choisir quel système d'exploitation démarrer.

Le chargeur de démarrage exécute le noyau. Ce dernier continue le démarrage de la machine, il détecte et initialise les périphériques, monte la partition racine et démarre le processus `init`.

1. Processus init et niveau d'exécution

Le processus `init` est le premier processus utilisateur créé par le noyau lors du démarrage du système. Il utilise les niveaux d'exécution pour définir l'état du système à un instant donné. Chaque niveau d'exécution contient une liste de services à arrêter ou à démarrer.

Lors du démarrage du système, le processus `init` détermine, à partir du fichier de configuration `/etc/inittab`, le niveau d'exécution par défaut, et démarre les applications et les services requis dans ce niveau.

Lors de l'arrêt du système, `init` bascule vers le niveau d'exécution numéro 0. Ce niveau d'exécution est configuré de telle sorte que toutes les applications et tous les services seront arrêtés.

Il existe sept niveaux d'exécution possibles, allant de 0 à 6. Chaque distribution les définit à sa manière, mais certains niveaux d'exécution sont les mêmes pour toutes les distributions. C'est le cas des niveaux d'exécution 0, 1 et 6. Le niveau d'exécution 0 est utilisé pour l'arrêt du système, le niveau d'exécution 1 est utilisé pour le démarrage du système en mode mono-utilisateur – ou mode dépannage – et le niveau d'exécution 6 est utilisé lors du redémarrage du système.

Sur la distribution Debian et ses dérivés les niveaux d'exécution de 2 à 5 sont tous en mode multi-utilisateurs. Et tous les services requis sont généralement configurés pour être démarrés pour ces niveaux d'exécution.

La distribution Red Hat et ses dérivés utilisent les niveaux d'exécution de la manière suivante :

- 1 : mode mono-utilisateur, utilisé pour la maintenance et la récupération du système (recovery mode). Dans ce niveau d'exécution, l'administrateur peut changer la configuration et effectuer les tâches de maintenance critique, comme le redimensionnement des partitions ou la vérification du système de fichiers racine. Typiquement, le niveau d'exécution s ou S produit un shell root sans montage des systèmes de fichiers, tandis que le niveau d'exécution 1 essaie de monter des systèmes de fichiers et lance quelques programmes systèmes.
- 2 : mode console, multi-utilisateurs, sans le support réseau ;
- 3 : mode multi-utilisateurs avec le support réseau ;
- 5 : mode multi-utilisateurs avec interface graphique.

Les commandes `init` et `telinit` permettent de changer de niveau d'exécution.

EXEMPLE

Pour basculer du niveau d'exécution courant vers le niveau d'exécution 1 :

```
# telinit 1
```

Le fichier de configuration de `init` est `/etc/inittab`. Ce fichier spécifie le niveau d'exécution par défaut du système. Il détaille également les autres niveaux d'exécution, et

où trouver la liste des services à démarrer ou arrêter à chaque niveau d'exécution.

Le format d'une ligne du fichier `/etc/inittab` est :

```
Id:[niveaux]:action:commande
```

- `Id` : identifie une entrée dans `inittab`. Pour les programmes de connexion comme `getty` ou `mingetty`, le champ *id* doit être le numéro du tty correspondant à la console, par exemple *1* pour *tty1* ;
- `niveaux` : la liste des niveaux d'exécution pour lesquels l'action doit être réalisée ;
- `action` : décrit l'action à réaliser ;
- `commande` : spécifie la commande à exécuter.

Voici les principales actions valides dans le champ *action* :

- `respawn` : le programme est redémarré à chaque fois qu'il se termine (exemple : `getty`) ;
- `wait` : le programme n'est démarré qu'une seule fois dans son niveau d'exécution et `init` attend la fin de l'exécution avant de continuer ;
- `once` : le programme est exécuté lorsque le niveau d'exécution spécifié est appelé ;
- `boot` : le programme est exécuté pendant le démarrage du système. Le champ *niveaux* est ignoré ;
- `bootwait` : le programme est exécuté pendant le démarrage du système, et `init` attend qu'il se termine. Le champ niveaux est ignoré ;
- `off` : ne fait rien ;
- `initdefault` : indique le niveau d'exécution par défaut ;
- `sysinit` : le programme est exécuté pendant le démarrage du système. Il est exécuté avant les entrées *boot* ou *bootwait*. Le champ niveaux est ignoré ;
- `powerwait` : le programme est exécuté quand la machine est sur le point de s'éteindre. Ce cas se produit quand la machine est branchée sur un onduleur et que la batterie est presque déchargée ;
- `powerfail` : équivalent à powerwait, mis à part qu'`init` n'attend pas la fin du programme avant de continuer ;
- `powerokwait` : le programme est exécuté dès qu'`init` est informé que l'alimentation de la machine est rétablie ;
- `powerfailnow` : le programme est exécuté quand `init` est informé que la tension faiblit ;
- `ctrlaltdel` : le programme est exécuté lorsque `init` reçoit le signal `SIGINT`. Cela signifie que quelqu'un sur la console système a tapé la combinaison de touches [Ctrl]-[Alt]-[Supp].

EXEMPLE

Voici un exemple de fichier `/etc/inittab` :

```
# Le niveau d'exécution par défaut (le 2)
id:2:initdefault:

# Le premier script à démarrer
si::sysinit:/etc/init.d/rcS


# /etc/init.d exécute les scripts S et K scripts lors
# d'un changement de mode de démarrage.
#
# Runlevel 0 est l'arrêt.
# Runlevel 1 est le mode simple utilisateur.
# Runlevels 2-5 sont des modes multi-utilisateur.
# Runlevel 6 est le mode de redémarrage.

l0:0:wait:/etc/init.d/rc 0
l1:1:wait:/etc/init.d/rc 1
l2:2:wait:/etc/init.d/rc 2
l3:3:wait:/etc/init.d/rc 3
l4:4:wait:/etc/init.d/rc 4
l5:5:wait:/etc/init.d/rc 5
l6:6:wait:/etc/init.d/rc 6

#  Intercepter les touches CTRL-ALT-DELETE pour un redémarrage
ca::ctrlaltdel:/sbin/shutdown -t5 -rf now

# Création des différentes consoles (CTRL ALT F[1-6]
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
```

Le processus `init` utilise une série de scripts stockés sous les répertoires `/etc/rcN.d` où N est le niveau d'exécution. Par exemple le répertoire `/etc/rc3.d` contient les services du niveau d'exécution 3.

2. Gestion des services

Un serveur est conçu pour être constamment en activité et écouter les requêtes des clients, c'est le cas des serveurs de courriers électroniques, serveurs web ou serveurs de fichiers. Un serveur est aussi appelé *démon* ou *service*.

Les niveaux d'exécution sont utilisés par le système lors du démarrage pour savoir quels services démarrer et quels services arrêter.

Par exemple les fichiers dans le répertoire `/etc/rc3.d` sont des liens symboliques vers des scripts stockés dans le répertoire `/etc/init.d`. Ces scripts contiennent les procédures pour démarrer, arrêter ou déterminer l'état de chaque service :

```
lrwxrwxrwx. 1 root root 15 janv. 10 10:33 K15httpd -> ../init.d/httpd
lrwxrwxrwx. 1 root root 13 janv. 10 10:34 K20nfs -> ../init.d/nfs
lrwxrwxrwx. 1 root root 23 janv. 25 06:27 K32clamd.amavisd ->
../init.d/clamd.amavisd
lrwxrwxrwx. 1 root root 15 janv. 10 10:44 K50snmpd -> ../init.d/snmpd
lrwxrwxrwx. 1 root root 19 janv. 10 10:44 K50snmptrapd ->
../init.d/snmptrapd
lrwxrwxrwx. 1 root root 14 janv. 10 11:52 K74ntpd -> ../init.d/ntpd
lrwxrwxrwx. 1 root root 17 janv. 10 11:52 K75ntpdate -> ../init.d/ntpdate
lrwxrwxrwx. 1 root root 16 janv. 10 10:42 K84btseed -> ../init.d/btseed
lrwxrwxrwx. 1 root root 17 janv. 10 10:31 K90network -> ../init.d/network
lrwxrwxrwx. 1 root root 19 janv. 10 11:52 K95firstboot ->
../init.d/firstboot
lrwxrwxrwx. 1 root root 18 janv. 10 10:29 S06cpuspeed -> ../init.d/cpuspeed
lrwxrwxrwx. 1 root root 19 janv. 10 10:28 S08ip6tables ->
../init.d/ip6tables
lrwxrwxrwx. 1 root root 18 janv. 10 10:28 S08iptables -> ../init.d/iptables
lrwxrwxrwx. 1 root root 16 janv. 10 10:34 S11auditd -> ../init.d/auditd
lrwxrwxrwx. 1 root root 21 janv. 10 10:28 S11portreserve ->
../init.d/portreserve
lrwxrwxrwx. 1 root root 17 janv. 10 10:31 S12rsyslog -> ../init.d/rsyslog
lrwxrwxrwx. 1 root root 14 janv. 10 10:33 S25cups -> ../init.d/cups
lrwxrwxrwx. 1 root root 24 janv. 10 10:44 S27NetworkManager ->
../init.d/NetworkManager
lrwxrwxrwx. 1 root root 14 janv. 10 10:34 S55sshd -> ../init.d/sshd
lrwxrwxrwx. 1 root root 17 janv. 11 15:35 S65dovecot -> ../init.d/dovecot
lrwxrwxrwx. 1 root root 22 janv. 25 14:59 S78spamassassin ->
../init.d/spamassassin
lrwxrwxrwx. 1 root root 17 janv. 25 09:20 S79amavisd -> ../init.d/amavisd
lrwxrwxrwx. 1 root root 17 janv. 11 10:15 S80postfix -> ../init.d/postfix
lrwxrwxrwx. 1 root root 15 janv. 10 10:32 S90crond -> ../init.d/crond
lrwxrwxrwx. 1 root root 13 janv. 10 10:40 S95atd -> ../init.d/atd
```

```
lrwxrwxrwx. 1 root root 11 janv. 10 10:31 S99local -> ../rc.local
```

Tous ces liens symboliques sont préfixés par `Sn` pour les scripts à démarrer (« start ») et `Kn` (« kill ») pour les scripts à arrêter, où `n` indique l'ordre de traitement.

`init` parcourt le répertoire `/etc/init.d`, il commence par arrêter les processus préfixés par un `K`, ensuite il démarre ceux préfixés par un `S`. Il démarre ou arrête ces services en ordre, du nombre `n` le plus petit vers le nombre `n` le plus grand.

Dans notre exemple, on peut voir que le service `postfix` démarre après le service `iptables`. En effet d'après les noms des fichiers liens symboliques, l'ordre du service iptables, « 08 », est inférieur à celui du service postfix (« 80 »). Ce système de numérotation est utilisé afin d'assurer que les services démarrent dans un ordre correct. Le service `iptables` doit démarrer avant `postfix` parce que le pare-feu iptables protège le système contre les intrusions.

Les services peuvent être lancés à partir des scripts du répertoire `/etc/init.d`. Ces scripts peuvent être appelés avec différents arguments :

- `start` : pour démarrer le service ;
- `stop` : pour arrêter le service ;
- `restart` : pour arrêter puis redémarrer le service ;
- `reload` : pour envoyer un signal SIGHUP au processus du service en cours d'exécution. Ce signal le force à relire ses fichiers de configuration ;
- `status` : indique si le service est en cours d'exécution.

EXEMPLE

La séquence de commandes ci-dessous démarre puis arrête le service `postfix`.

```
# /etc/init.d/postfix start
Démarrage de postfix :
# /etc/init.d/postfix stop
Arrêt de postfix :
```

3. Gestion des services sous Red Hat

Pour comprendre comment Red Hat gère les services, examinons par exemple l'en-tête du script `postfix` dans le répertoire `/etc/init.d`.

```
$ head /etc/init.d/postfix
#!/bin/bash
#
# postfix      Postfix Mail Transfer Agent
#
```

```
# chkconfig: 2345 80 30
# description: Postfix is a Mail Transport Agent, which is the program \
#              that moves mail from one machine to another.
# processname: master
# pidfile: /var/spool/postfix/pid/master.pid
# config: /etc/postfix/main.cf
```

On peut voir à la quatrième ligne :

```
chkconfig : 2345 80 30
```

Cette information est utilisée par le programme `chkconfig` pour créer des liens symboliques vers le script `/etc/init.d/postfix` dans les répertoires `/etc/rc2.d`, `/etc/rc3.d`, `/etc/rc4.d` et `/etc/rc5.d`. Ces liens symboliques ont les préfixes S80 et K30. Ainsi le service `postfix` démarre dans les niveaux d'exécution 2, 3, 4 et 5 avec l'ordre 80, et s'arrête avec l'ordre 30.

Le standard LSB (Linux Standard Base) préconise d'écrire dans l'en-tête des scripts du répertoire `/etc/init.d` des mots clés suivis d'une liste d'arguments.

EXEMPLE

Voici un extrait de l'en-tête du script `/etc/init.d/postfix` :

```
### BEGIN INIT INFO
# Provides: postfix mail-transport-agent
# Required-Start : $local_fs $remote_fs $syslog $named $network $time
# Required-Stop : $local_fs $remote_fs $syslog $named $network
# Should-Start : postgresql mysql clamav-daemon postgrey spamassassin
# Should-Stop : postgresql mysql clamav-daemon postgrey spamassassin
# Default-Start : 2 3 4 5
# Default-Stop : 0 1 6
# Short- Description : start and stop the Postfix Mail Transport Agent
# Description: postfix is a Mail Transport agent
### END INIT INFO
```

Dans cet exemple le mot clé Required_Start indique les services requis pour le démarrage du service SMTP `postfix`. Les arguments $local_fs et $remote_fs indiquent que les systèmes de fichiers locaux et distants doivent être montés, les autres arguments – $syslog, $named, $network et $time – indiquent respectivement que le service de gestion des logs, le service DNS, le service réseau et le serveur de temps doivent être démarrés.

Les mots-clés les plus utilisés sont présentés dans le *tableau 2*.

Tableau 2. Quelques mots clés LSB

Mots clés	Description
Provides	Donne une brève indication sur ce que le service fournit. Cette information est utilisée par d'autres services.
Required-Start	Liste les services qui doivent être disponibles pour que le service puisse démarrer.
Required-Stop	Indique que le service doit être arrêté avant que les services listés ici soient arrêtés.
Should-Start	Définit les services qui peuvent être démarrés, mais non obligatoirement, avant que ce service démarre.
Should-Stop	Indique que le service peut être arrêté, mais non obligatoirement, avant les services listés ici.
Default-Start	Par défaut, le service doit démarrer dans les niveaux d'exécution listés.
Default-Stop	Par défaut, le service ne doit pas démarrer dans les niveaux d'exécution listés.
Description	Donne la description de ce service.

Avec l'option `--list` de la commande `chkconfig` on peut lister tous les services, activés ou désactivés, disponibles sur le système :

```
# chkconfig --list
NetworkManager  0:arrêt 1:arrêt 2:marche 3:marche 4:marche 5:marche 6:arrêt
acpid           0:arrêt 1:arrêt 2:marche 3:marche 4:marche 5:marche 6:arrêt
amavisd         0:arrêt 1:arrêt 2:marche 3:marche 4:marche 5:marche 6:arrêt
atd             0:arrêt 1:arrêt 2:arrêt 3:marche 4:marche 5:marche 6:arrêt
clamd.amavisd   0:arrêt 1:arrêt 2:arrêt 3:arrêt 4:arrêt 5:arrêt 6:arrêt
crond           0:arrêt 1:arrêt 2:marche 3:marche 4:marche 5:marche 6:arrêt
cups            0:arrêt 1:arrêt 2:marche 3:marche 4:marche 5:marche 6:arrêt
dovecot         0:arrêt 1:arrêt 2:marche 3:marche 4:marche 5:marche 6:arrêt
httpd           0:arrêt 1:arrêt 2:arrêt 3:arrêt 4:arrêt 5:arrêt 6:arrêt
ip6tables       0:arrêt 1:arrêt 2:marche 3:marche 4:marche 5:marche 6:arrêt
iptables        0:arrêt 1:arrêt 2:marche 3:marche 4:marche 5:marche 6:arrêt
network         0:arrêt 1:arrêt 2:arrêt 3:arrêt 4:arrêt 5:arrêt 6:arrêt
nfs             0:arrêt 1:arrêt 2:arrêt 3:arrêt 4:arrêt 5:arrêt 6:arrêt
ntpd            0:arrêt 1:arrêt 2:arrêt 3:arrêt 4:arrêt 5:arrêt 6:arrêt
ntpdate         0:arrêt 1:arrêt 2:arrêt 3:arrêt 4:arrêt 5:arrêt 6:arrêt
postfix         0:arrêt 1:arrêt 2:marche 3:marche 4:marche 5:marche 6:arrêt
```

Afin d'activer le service `postfix` dans les niveaux d'exécution définis dans l'en-tête du script `/etc/init.d/postfix`, on exécute la commande :

```
# chkconfig postfix on
```

On peut aussi définir manuellement les niveaux d'exécution dans lesquels le service `postfix` va démarrer.

EXEMPLE

Pour activer `postfix` dans les niveaux d'exécution 3 et 5, on utilise la commande :

```
# chkconfig --level 35 postfix on
```

Ainsi, lors du démarrage du système, le service `postfix` démarre automatiquement dans les niveaux d'exécution 3 ou 5, mais pas dans les niveaux d'exécution 2 et 4.

Si on souhaite désactiver le service `postfix` de sorte qu'il ne démarre pas automatiquement lors du démarrage du système, on utilise la commande :

```
# chkconfig postfix off
```

4. Gestion des services sous Debian et ses dérivés

Sous Débian et ses dérivés on utilise la commande `update-rc.d` pour maintenir les liens symboliques vers les scripts de démarrage dans les répertoires `/etc/rcN.d`

SYNTAXE

`update-rc.d nom_service { start | stop } numéro_sequence numéros_niveaux_execution .`

La commande `update-rc.d` a comme arguments :

- un numéro de séquence, utilisé par `init` pour décider de l'ordre d'exécution des scripts ;
- la liste des niveaux d'exécution applicables, impérativement suivie par un point.

EXEMPLE

Pour configurer le service d'impression `cups` de telle sorte qu'il démarre dans les niveaux d'exécution 2, 3 4 et 5 avec un ordre de 80 et s'arrête dans les niveaux d'exécution S, 1 et 6 avec un ordre de 20 :

```
# update-rc.d cups start 80 2 3 4 5 . stop 20 S 1 6 .
Adding system startup for /etc/init.d/cups ...
/etc/rc1.d/K20cups -> ../init.d/cups
/etc/rc6.d/K20cups -> ../init.d/cups
/etc/rcS.d/K20cups -> ../init.d/cups
/etc/rc2.d/S80cups -> ../init.d/cups
/etc/rc3.d/S80cups -> ../init.d/cups
/etc/rc4.d/S80cups -> ../init.d/cups
/etc/rc5.d/S80cups -> ../init.d/cups
```

Quand on appelle `update-rc.d` avec l'option `remove` les liens dans les répertoires `/etc/rcN.d` qui pointent vers le script correspondant du répertoire `/etc/init.d` seront supprimés. Ce script doit auparavant avoir été lui-même supprimé. S'il est toujours présent

dans `/etc/init.d`, `update-rc.d` affiche un message d'erreur indiquant qu'il faut utiliser l'option -f pour forcer la suppression du script, comme l'illustre l'exemple suivant.

EXEMPLE

```
# update-rc.d atd remove
update-rc.d: /etc/init.d/atd exists during rc.d purge (use -f to force)
# update-rc.d -f atd remove
Removing any system startup links for /etc/init.d/atd ...
  /etc/rc1.d/K11atd
  /etc/rc2.d/S89atd
  /etc/rc3.d/S89atd
  /etc/rc4.d/S89atd
  /etc/rc5.d/S89atd
```

5. Upstart

Plusieurs distributions Linux récentes, y compris les dernières versions d'Ubuntu et de Fedora, ont remplacé le système classique init d'UNIX système V, SysV init, basé sur les niveaux d'exécution, par un nouveau système Upstart. Upstart est un système de gestion de services fonctionnant avec les événements. Il supervise les services pendant que le système fonctionne. Ainsi il démarre ou arrête des services en réponse aux événements tels que l'ajout d'un périphérique.

Pour un besoin de compatibilité, il émule également les niveaux d'exécution traditionnels de `init`.

Un système qui utilise Upstart remplace à la fois le fichier `/etc/inittab` et les répertoires spécifiques des niveaux d'exécution par les scripts placés dans le répertoire `/etc/init` (ce répertoire a été appelé `/etc/event.d` sur des versions antérieures d'Upstart).

6. Récupération du système

Des problèmes de MBR ou de système de fichiers de la partition racine empêchent Linux de démarrer. L'administrateur doit donc utiliser un CD de dépannage permettant de démarrer le système à partir d'un noyau Linux sur ce CD, ensuite procéder à la réparation des problèmes de démarrage du système.

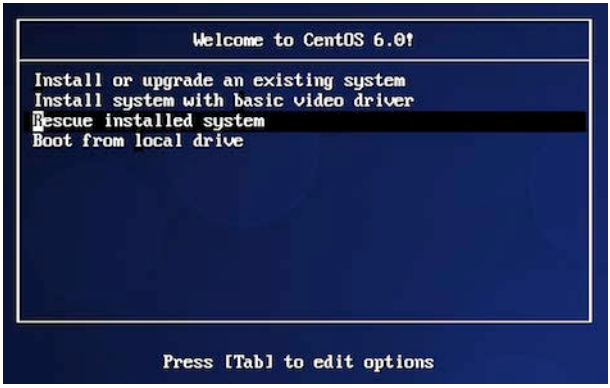
6.1 Récupération du chargeur de démarrage Grub

Si on n'obtient pas l'écran de Grub au démarrage du système, il se peut que le MBR (Master Boot Record) soit endommagé, il faut donc réinstaller Grub sur le MBR du disque d'amorçage.

PROCEDURE

– Démarrage du système à partir du premier cédérom d'installation de la distribution. La distribution CentOs 6.0 propose le menu suivant permettant un démarrage en mode dépannage (*figure 3*). Ce mode est aussi disponible dans les autres distributions Linux.

Figure 3. Menu de démarrage en mode démarrage de la distribution CentOS

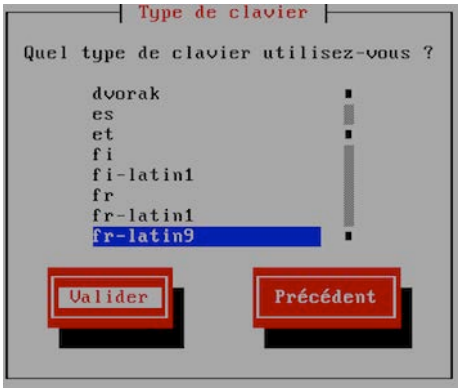


– Configuration de la langue (*figure 4*).

Figure 4. Configuration de la langue

– Configuration du type de clavier (*figure 5*).

Figure 5. Configuration du type de clavier



– Montage du système : le mode dépannage cherche l'installation Linux et la monte sous le répertoire `/mnt/sysimage` (*figure 6)* :

Figure 6. Montage du système

– Changement de répertoire racine (`chroot`) : le prompt « # » apparaît, on définit le répertoire `/mnt/sysimage` comme la nouvelle racine « / » du système :

```
chroot /mnt/sysimage
```

– Installation du Grub sur le MBR avec les paramètres définis dans le fichier de configuration `grub.conf` :

```
# /sbin/grub-install /dev/sda
Installation finished. No error reported.
This is the contents of the device map /boot/grub/device.map.
Check if this is correct. If any of the lines are incorrect,
fix it and re-run the script 'grub-install.'
# this device map was generated by anaconda
```

```
(hd0) /dev/sda
```

Lors du redémarrage du système on devra voir l'écran du Grub permettant de sélectionner le système d'exploitation à démarrer.

Dans le cas où la partition de démarrage est endommagée, la commande `grub-install` ne fonctionne pas car elle ne peut pas localiser le fichier de configuration `grub.conf`. On doit donc utiliser le shell Grub pour localiser la partition contenant l'installation précédente du Grub puis le réinstaller sur le MBR :

– Exécution de la commande `grub`, le prompt change indiquant que le shell Grub est lancé :

```
# grub

    GNU GRUB version 0.97 (640K lower / 3072K upper memory)

  [ Minimal BASH-like line editing is supported. For the first word, TAB
    lists possible command completions. Anywhere else TAB lists the
possible
    completions of a device/filename.]
grub >
```

– Recherche de la partition contenant l'installation du Grub :

```
grub> find /grub/stage1
find grub/stage1
(hd0,0)
```

Le résultat de la commande précédente indique que l'installation du Grub est sur la première partition du premier disque dur.

– Définition de la partition contenant le répertoire `/grub` :

```
grub> root (hd0,0)
Filesystem type is ext2fs, partition type 0x83
```

– Réinstallation du Grub sur le MBR :

```
grub> setup (hd0)
setup (hd0)
 Checking if "/boot/grub/stage1" exists... no
 Checking if "/grub/stage2" exists... yes
 Checking if "/grub/e2fs_stage1_5" exists... yes
 Running "embed /grub/e2fs_stage1_5 (hd0)"...  26 sectors are embedded.
succeeded
 Running "install /grub/stage1 (hd0) (hd0)1+26 p (hd0,0)/grub/stage2
/grub/grub.conf"... succeeded
```

```
Done.
```

6.2 Récupération de la partition racine

Si le système de fichiers de la partition racine n'est pas démonté correctement, la commande `fsck` est lancée automatiquement lors de redémarrage du système afin de vérifier et corriger les systèmes de fichiers.

On peut aussi vérifier le système de fichiers manuellement avec le mode de dépannage lancé à partir du premier cédérom d'installation de la distribution.

Il est recommandé d'exécuter la commande `fsck` sur un système de fichiers démonté. Comme on ne peut pas démonter le système de fichiers racine d'un système en cours d'exécution, la solution est de remonter la partition racine en lecture seule :

```
# mount -o remount,ro /
```

Ensuite vérifier et corriger le système de fichiers de la partition racine :

```
# fsck /
```

Exercices

1. **Dans quels répertoires pourriez-vous trouver des liens symboliques vers les scripts de démarrage des services d'UNIX système V ?**
   - ❑ A. `/etc/rc?.d`
   - ❑ B. `/etc/inittab`
   - ❑ C. `/etc/init.d/rc?.d`
   - ❑ D. `/etc/init`

2. **Sur une machine Red Hat, quelle commande allez-vous utiliser pour savoir dans quels niveaux d'exécution le serveur postfix est activé ?**
   - ❑ A. `ps -ef | grep postfix`
   - ❑ B. `chkconfig --list postfix`
   - ❑ C. `info postfix`
   - ❑ D. `update-rc.d postfix`

3. **Lesquelles des propositions suivantes pourraient être utilisées comme un niveau d'exécution par défaut ?**
   - ❑ A. 0

- ❑ B. 2
- ❑ C. 3
- ❑ D. 5

4. Sur une machine Debian, laquelle des commandes suivantes est utilisée pour configurer le service cron de telle sorte qu'il s'arrête dans le niveau d'exécution 2 avec un ordre de 45 ?

- ❑ A. `chkconfig cron off 45 2 .`
- ❑ B. `update-rc.d cron stop 45 2 .`
- ❑ C. `disable cron 45 2`
- ❑ D. `update-rc.d cron off 45 2`

Chapitre 3. Les systèmes de fichiers Linux

Objectifs	**Intervention sur le système de fichiers Linux** – configurer et monter plusieurs types de systèmes de fichiers Linux ; – activer et désactiver le swap. **Maintenance d'un système de fichiers Linux** – corriger les erreurs d'un système de fichiers ; – manipuler un système de fichiers. **Création et configuration des options du système de fichiers** – configurer le montage automatique d'un système de fichiers avec le service autofs ; – créer un système de fichiers pour des périphériques tels que les cédéroms. – connaître les caractéristiques de base des systèmes de fichiers cryptés **Gestion de périphérique udev** – comprendre la détection et la gestion de périphériques avec udev ; – maintenir les règles udev.
Points importants	– Le fichier de configuration fstab. – Outils pour manipuler les partions et les fichiers SWAP. – Utilisation des UUIDs. – Outils pour manipuler les systèmes de fichiers ext2, ext3, ext4, reiserfs et xfs. – Fichiers de configuration autofs. – Outils de manipulation de UDF et ISO9660. – Règles udev. – interface du noyau.
Mots-clés	`/etc/fstab`, `/proc/mounts`, `sync`, `swapoff`, `/etc/mtab`, `mount et umount`, `swapon`, `fsck (fsck.*)`, `mkfs (mkfs.*)`, `debugfs`, `debugreiserfs`, `mkswap`, `xfs_check`, `badblocks`, `dumpe2fs`, `xfsdump`, `xfsrestore`, `tune2fs`, `reiserfstune`, `xfs_info`, `xfs_repair`, `/etc/auto.master`, `mkisofs`, `mke2fs`, `/etc/auto.[dir]`, `dd`, `udevmonitor`, `/etc/udev`, `xfs_info`, `xfs_check and xfs_repair`

1. Les types de systèmes de fichiers

minix, le système de fichier de Minix, est le premier système de fichiers utilisé sous Linux. Le système de fichiers étendu **ext**, une extension de **minix**, a été développé en avril 1992. C'est le premier système de fichiers utilisant l'API VFS. Il a été inclus dans la version 0.96c du noyau Linux.

Dans cette section, certains des systèmes de fichiers les plus couramment utilisés sous Linux seront examinés. Ils sont énumérés dans le *tableau 3* avec leurs principales caractéristiques.

Tableau 3. Principaux systèmes de fichiers supportés par Linux

Système de fichiers	Caractéristiques
ext2	Extension du système de fichiers **ext**. Stable, usage général, peut être rétréci ou agrandi. Ce système de fichiers n'a pas la fonctionnalité de journalisation.
ext3	Une amélioration de **ext2**. La fonctionnalité de journalisation est ajoutée pour permettre la récupération rapide du système de fichiers en cas de crash.
ext4	Une amélioration de **ext3** pour supporter de grandes tailles de fichiers et de systèmes de fichiers.
XFS	Stable, usage général, une récupération rapide, peut être étendu en ligne
JFS	Stable, usage général, une récupération rapide

1.1 ext2

Le système de fichier **ext2** a été, jusqu'à la fin des années 1990, le système de fichiers de Linux. Il a la réputation d'un système de fichiers fiable et stable. Il a été éclipsé par d'autres systèmes de fichiers à journal, mais il a toujours son utilité. En particulier, **ext2** peut être un bon choix pour une petite partition ou pour les disques amovibles de petites tailles. Sur ces petites partitions, la taille du journal utilisé par ces systèmes de fichiers à journal peut servir à stocker les données.

1.2 ext3

Le système de fichiers **ext3** est une extension de **ext2**. Il ajoute une fonctionnalité de journalisation qui augmente sa fiabilité, sans changer la structure fondamentale héritée de ext2. On peut toujours monter un système de fichiers **ext3** comme étant un système de fichier ext2 en désactivant la fonctionnalité de journalisation.

Il est possible de convertir un système de fichiers **ext2** en un système de fichiers **ext3**

avec l'option `-j` de la commande `tune2fs` .

EXEMPLE

Pour convertir la partition **ext2** `/dev/sda2` en **ext3** :

```
#tune2fs -j /dev/sda2
```

Le système de fichiers **ext3** réserve une zone du disque pour le fichier journal[2]. Quand une opération sur le système de fichiers se produit, les modifications nécessaires sont d'abord écrites dans le fichier journal. Ensuite le système de fichiers est modifié. Si une coupure de courant ou une panne système survient pendant la mise à jour, on peut se référer au fichier journal pour reconstruire un système de fichiers cohérent.

La technique de journalisation permet de réduire considérablement le temps nécessaire au système de fichiers pour effectuer des contrôles de cohérence. Sauf pour des pannes matérielles, l'état d'un système de fichiers **ext3** peut être presque instantanément évalué et restauré.

1.3 ext4

Le système de fichier **ext4** est la nouvelle génération de la famille **ext**. **ext4** ajoute la possibilité de travailler avec des disques très volumineux (plus de 32 téraoctets) ou de très gros fichiers (ceux de plus de 2 To), ainsi que des extensions destinées à améliorer les performances et à la défragmentation en ligne.

1.4 XFS

Silicon Graphics (SGI) a crée le système de fichier **XFS** pour son système d'exploitation IRIX, par la suite elle a donné le code source de **XFS** à Linux. **XFS** a acquis une réputation de robustesse, de rapidité et de flexibilité sur IRIX. La taille d'un système de fichier **XFS** peut être agrandie en ligne, à condition qu'il y ait de l'espace non alloué disponible sur le disque contenant le système de fichiers.

1.5 JFS

IBM a développé le système de fichiers journalisé JFS (*Journaled File System*) pour son système d'exploitation AIX. JFS est considéré comme un bon système de fichiers journalisé qui fonctionne bien avec des fichiers de tailles variables. Il est également considéré comme un système de fichiers léger qui n'utilise pas de grandes quantités de CPU lors d'une activité importante du disque.

[2] Sur les fichiers journaux, voir manuel LPI 102, chapitre 6 « Administration du système GNU/Linux »

2. Création de systèmes de fichiers

2.1 Commande mkfs

La commande `mkfs` permet de construire un système de fichiers sur une partition du disque. Elle fait appel à d'autres programmes en fonction du type de système de fichiers sélectionné. Linux prend en charge différents types de systèmes de fichiers, y compris plusieurs systèmes de fichiers journalisés et les systèmes de fichiers Windows.

La création d'un système de fichiers peut être faite avec la commande qui a comme préfixe `mkfs` et comme suffixe le nom du type de système de fichiers.

Par exemple :

- pour créer un système de fichiers ext3 on utilise la commande `mkfs.ext3` ;
- pour les systèmes de fichiers ReiserFS, la commande est `mkfs.reiserfs` ;
- pour les systèmes de fichiers 6 bits de Windows (95/98), la commande est `mkfs.vfat`.

La commande `mke2fs` permet aussi de créer un système de fichiers ext2, et en utilisant l'option `-j` on peut créer un système de fichiers ext3.

La syntaxe de la commande `mkfs` est :

`mkfs` `[-t type-sys-fichiers][options-sys-fichiers] partition [nombre-blocs]`

Si le type du système de fichiers (`type-sys-fichiers`) n'est pas indiqué, le système de fichiers utilisé par défaut est **ext2**. On peut ajouter des options du système de fichiers à créer (`options-sys-fichiers`). Enfin on peut spécifier le nombre de blocs à utiliser pour le système de fichiers (`nombre-blocs`).

Le *tableau 4* résume les principales options ext3 de la commande `mkfs`.

Tableau 4. Principales options ext3 de la commande mkfs

Option	Description
`-V`	Affiche le numéro de la version de la commande `mkfs`.
`-v`	Sortie bavarde
`-c`	Vérifie le périphérique pour éliminer les secteurs défectueux avant de formater le système de fichiers.
`-l nom-fichier`	Lit la liste des secteurs défectueux depuis le fichier `nom-fichier`.
`-L étiquette-du-volume`	Définit le nom de volume pour le système de fichiers
`-b taille-de-bloc`	Définit la taille de bloc

`-N nombre-d-inodes`	Définit le nombre d'inodes
`-m pourcentage-de-blocs-réservés`	Définit le pourcentage de blocs réservés pour le super utilisateur

EXEMPLE

L'exemple suivant illustre la création d'un système de fichiers ext3 sur la partition `/dev/sdb1` :

```
# mkfs.ext3 /dev/sdb1
mke2fs 1.40.8 (
13-Mar-2008)
Filesystem label=
OS type: Linux
Block size=4096 (log=2)
Fragment size=4096 (log=2)
122880 inodes, 489974 blocks
24498 blocks (5.00%) reserved for the super user
First data block=0
Maximum filesystem blocks=503316480
15 block groups
32768 blocks per group, 32768 fragments per group
8192 inodes per group
Superblock backups stored on blocks:
32768, 98304, 163840, 229376, 294912
Writing inode tables: done
Creating journal (8192 blocks): done
Writing superblocks and filesystem accounting information: done
This filesystem will be automatically checked every 25 mounts or
180 days, whichever comes first. Use tune2fs -c or -i to override.
```

On peut voir dans la sortie de la commande précédente que l'étiquette du système de fichiers n'est pas définie (« `Filesystem label=` »). Elle peut être spécifiée avec l'option `-L` de la commande `mkfs.ext3`.

L'étiquette – ou nom du système de fichier – peut être utilisée dans le montage d'une partition. Ainsi, si la partition `/dev/sdb1` a l'étiquette « DISK1 », elle sera montée par la commande suivante :

```
# mount LABEL=DSIK1 /media
```

On peut aussi voir une série de statistiques sur la taille du système de fichiers et comment l'espace de stockage a été alloué.

EXEMPLE

```
Block size=4096 (log=2)
….
122880 inodes, 489974 blocks
```

Le système de fichiers contient 489 974 blocs et 12 2880 inodes. La taille d'un bloc est 4 096 octets. Chaque fichier correspond à un inode unique. L'inode est une structure de données contenant des informations sur le fichier associé. Les informations stockées dans un inode sont : l'utilisateur propriétaire du fichier, le groupe propriétaire du fichier, le type de fichier, les droits d'accès, la date du dernier accès au fichier, la date de dernière modification du fichier, la date de dernière modification de l'inode, la taille du fichier et les adresses des blocs disques contenant le fichier.

Et on peut noter que des blocs sont réservés pour le super-utilisateur (*root*) :

```
24498 blocks (5.00%) reserved for the super user
```

Les blocs réservés pour le super-utilisateur sont définis de sorte qu'un utilisateur autre que *root* ne puisse pas remplir tout un système de fichiers. Si c'était le cas, le super-utilisateur ne pourrait plus se connecter, et les services s'exécutant en tant que super-utilisateur seraient incapables d'écrire des données sur le disque.

La taille par défaut des blocs réservés pour le super-utilisateur est de 5 % de la taille du système de fichier. Ce pourcentage convient si le système de fichiers racine a une taille de quelques gigaoctets. En revanche, si le système de fichiers à une taille d'un téraoctet, la taille des blocs réservés pour le super-utilisateur est de 50 gigaoctets. Dans ce cas, pour éviter qu'un tel espace ne puisse être utilisé pour le stockage des données des utilisateurs, il est logique de modifier le pourcentage.

L'option `-m` de la commande `mkfs.ext3` permet de définir ce pourcentage de blocs réservés pour le super-utilisateur. Sa valeur est 0 lors de la création du système de fichiers, et on peut la modifier plus tard avec la commande `tune2fs`.

Enfin, la sortie de la commande `mkfs.ext3` indique qu'on peut modifier les conditions déclanchement de la vérification automatique du système de fichiers :

```
This filesystem will be automatically checked every 25 mounts or
180 days, whichever comes first. Use tune2fs -c or -i to override.
```

Par défaut, une nouvelle vérification est faite dès que l'une des deux conditions suivantes est réalisée : 25 montages ont été faits depuis la dernière vérification, ou 180 jours se sont écoulés.

2.2 Le superbloc

Les systèmes de fichiers **ext3** et **ext2** ont la même structure de base. Cette structure est composée par un superbloc suivi de plusieurs groupes de blocs :

Superbloc	Groupe de blocs 1	Groupe de blocs 2	...	Groupe de blocs n

Le superbloc est un enregistrement qui décrit les caractéristiques du système de fichiers. Il contient des informations sur la taille des blocs, la taille et l'emplacement des tables des inodes, la taille des groupes de blocs, la taille du système de fichiers, l'espace disque disponible et quelques autres paramètres importants du système de fichiers.

Avec la perte du superbloc, on n'a aucun moyen pour déterminer l'emplacement des blocs de données des fichiers, ce qui risque d'entraîner la perte des données. Pour cette raison des copies de sauvegarde du superbloc sont maintenues dans des lieux dispersés au début de chaque groupe de blocs.

Les données d'un fichier sont donc réparties entre les groupes de blocs de sorte que les blocs de données, qui doivent être accessibles ensemble, sont stockés à proximité les uns des autres sur le disque. Ce regroupement permet de réduire la nécessité de rechercher sur tout le disque lors de l'accès aux blocs d'un même fichier.

La commande `dumpe2fs` permet d'afficher les informations sur le superbloc et les groupes de blocs et de localiser les adresses des copies du superbloc sur une partition du disque.

EXEMPLE

Pour localiser les adresses de copies du superbloc sur la partition `/dev/sda1` on exécute la commande suivante :

```
# dumpe2fs /dev/sda1
Filesystem volume name:   <none>
Last mounted on:          <not available>
Filesystem UUID:          bb19cad6-586d-48bc-83c1-5bed78bdcd2e
Filesystem magic number:  0xEF53
Filesystem revision #:    1 (dynamic)
Filesystem features:      has_journal ext_attr resize_inode dir_index
filetype needs_recovery
Inode count:              305824
Block count:              1220932
Reserved block count:     61046
Free blocks:              718222
Free inodes:              280072
First block:              0
Block size:               4096
Fragment size:            4096
Reserved GDT blocks:      298
Blocks per group:         32768
Fragments per group:      32768
Inodes per group:         8048
Inode blocks per group:   503
```

```
Filesystem created:       Sun Sep  5 11:50:38 2010
Last mount time:          Fri Feb 11 09:27:06 2011
Last write time:          Sat Jan 29 09:38:02 2011
Mount count:              31
Maximum mount count:      37
First orphan inode:       173244
Default directory hash:   half_md4
Directory Hash Seed:      1fa74e4e-edbf-429b-9214-86447611ee3e
Journal backup:           inode blocks
Journal features:         journal_incompat_revoke
Taille du journal:        128M
Journal length:           32768
Journal sequence:         0x00031af3
Journal start:            1


....
Groupe 1 : (Blocs 32768-65535)
 superbloc Secours à 32768, Descripteurs de groupes à 32769-32769
  Blocs réservés GDT à 32770-33067
 Bitmap de blocs à 33068 (+300), Bitmap d'i-noeuds à 33069 (+301)
 Table d'i-noeuds à 33070-33572 (+302)
  0 blocs libres, 6818 i-noeuds libres, 243 répertoires
 Blocs libres :
 I-noeuds libres : 9279-16096
```

2.3 Configuration de l'espace swap

La commande `mkswap` permet de créer un espace swap sur un périphérique et de l'ajouter au système.

Le périphérique est habituellement une partition de disque, mais peut aussi être un fichier. On peut également spécifier la taille de l'espace swap, mais ce n'est pas recommandé. Quand la taille est omise, `mkswap` utilise simplement la totalité de la partition ou du fichier spécifié.

La commande `swapon` permet d'activer des périphériques et des fichiers à utiliser pour le swap. `swapon` vérifie également que l'espace swap a été correctement ajouté.

EXEMPLE

Dans cet exemple les commandes `mkswap` et `swapon` sont utilisées pour créer puis activer un espace swap sur la partition `/dev/sda2`. L'option `-s` de la commande `swapon` permet d'afficher l'utilisation du swap pour chaque périphérique.

```
# mkswap /dev/sda2
```

```
Setting up swapspace version 1, size = 2105667584 bytes
# swapon /dev/sda2
# swapon -s
Filename Type Size Used Priority
/dev/hda5 partition 133020 688 -1
/dev/sda2 partition 2056316 0 -2
```

3. Ajustement des paramètres des systèmes de fichiers ext[234]

La commande `tune2fs` permet de modifier les paramètres ajustables d'un système de fichiers **ext2**, **ext3** ou **ext4**. Les principaux paramètres ajustables sont le label, le journal et les blocs réservés pour le super-utilisateur.

Le *tableau* 5 illustre les principales options de la commande `tune2fs`.

Tableau 5. Principales options de la commande tune2fs

Option	Fonction
`-c` *N*	Ajuste le nombre maximal de montages entre deux vérifications du système de fichiers. Si N vaut 0 ou -1, le nombre de fois que le système de fichiers a été monté sera ignoré par la commande `e2fsck` et par le noyau.
`-l`	Affiche le contenu du superbloc du système de fichiers.
`-m` *N*	Définit le pourcentage de blocs réservés pour le super-utilisateur dans le système de fichiers.
`-r` *N*	Définit le nombre de blocs réservés pour le super-utilisateur dans le système de fichiers.
`-j`	Ajoute un journal ext3 au système de fichiers.
`-L` *label*	Définit le nom de volume du système de fichiers.

4. Vérification et réparation d'un système de fichiers

Généralement les systèmes de fichiers modernes sont fiables face aux pannes du système et du matériel. Des pannes du noyau et des coupures du courant électrique peuvent engendrer des petites incohérences dans le système de fichiers. Ces dommages peuvent être résolus avec la commande `fsck`.

La commande `fsck` analyse les systèmes de fichiers et corrige les erreurs détectées, telles que :

- blocs de données non utilisés et non enregistrés ;

- blocs de données marqués comme libres mais également utilisés dans un fichier ;
- inodes non référencés ;
- répertoires non reliés au système de fichiers ;
- informations incorrectes dans le superbloc.

Les disques sont généralement analysés au démarrage du système avec la commande `fsck -p`. Cette commande examine et corrige les systèmes de fichiers listés dans le fichier `/etc/fstab`. Linux garde une trace des systèmes de fichiers qui ont été correctement démontés. Concernant les systèmes de fichiers qui ne sont pas démontés correctement, Linux procède à leur analyse et à leur correction. Si une certaine forme de journalisation est activée, `fsck` indique que le système de fichiers est journalisé et applique le dernier état cohérent du système de fichiers.

EXEMPLE

Pour analyser le système de fichiers `/dev/sda5` :

```
# fsck -p /dev/sda5
```

La commande `fsck -p` lit le fichier `/etc/fstab` pour savoir quels sont les systèmes de fichiers à contrôler. Elle les traite par ordre numérique croissant, en se basant sur la valeur contenue dans la dernière colonne de chaque ligne. La partition racine est toujours vérifiée en premier.

Même si tous les démontages des systèmes de fichiers ont été faits correctement, on peut forcer la vérification après un certain nombre de montages. La valeur par défaut est 25 démontages. C'est une bonne précaution, toutefois, sur les ordinateurs de bureau où les montages de systèmes de fichiers sont fréquents, faire un `fsck` tous les 25 montages peut devenir fastidieux. On peut donc fixer à 50 le nombre maximum de montages entre deux vérifications.

EXEMPLE

```
# tune2fs -c 50 /dev/sda3
tune2fs 1.35 (28-Feb-2004)
Setting maximal mount count to 50
```

5. Système de fichiers XFS

XFS est un système de fichiers développé par Silicon Graphics Inc pour son système d'exploitation Unix propriétaire IRIX. Actuellement, il est disponible en code source libre sous Linux. XFS est un système de fichiers journalisé et peut être redimensionné à chaud.

Les principales commandes de gestion d'un système de fichiers XFS sont : `mkfs.xfs`, `xfs_info`, `xfs_check`, `xfs_repair`, `xfs_growfs`, `xfsdump` et `xfsrestore`.

Ces commandes appartiennent aux paquetages `xfsprogs` et `xfsdump`.

5.1. Création et montage d'un système de fichiers XFS

La commande `mkfs.xfs` permet de créer un système de fichiers XFS. Ses principales options sont :

- `-l size` : spécifie la taille du journal ;
- `-l logdev` : spécifie le nom du périphérique qui contient le journal ;
- `-i size` : spécifie la taille de l'inode ;
- `-i maxpct` : indique le pourcentage maximal alloué aux inodes ;
- `-b size` : spécifie la taille de block ;
- `-d size` : spécifie la taille de l'espace réservé aux données ;
- `-f` : force la création du système de fichiers.

EXEMPLE

- Création d'un système XFS sur la partition `/dev/sdb1`. L'option `-f` est nécessaire si cette partition contient déjà un système de fichiers.

```
# mkfs.xfs -f /dev/sdb1
meta-data=/dev/sdb1              isize=256    agcount=4, agsize=152114 blks
         =                       sectsz=512   attr=2, projid32bit=0
data     =                       bsize=4096   blocks=608454, imaxpct=25
         =                       sunit=0      swidth=0 blks
naming   =version 2              bsize=4096   ascii-ci=0
log      =internal log           bsize=4096   blocks=2560, version=2
         =                       sectsz=512   sunit=0 blks, lazy-count=1
realtime =none                   extsz=4096   blocks=0, rtextents=0
```

- Montage du système de fichiers créé :

```
# mount /dev/sdb1 /mnt
```

- Visualisation des caractéristiques du système de fichiers créé :

```
# xfs_info /mnt
meta-data=/dev/sdb1              isize=256    agcount=4, agsize=152114 blks
         =                       sectsz=512   attr=2
data     =                       bsize=4096   blocks=608454, imaxpct=25
         =                       sunit=0      swidth=0 blks
naming   =version 2              bsize=4096   ascii-ci=0
log      =internal               bsize=4096   blocks=2560, version=2
```

```
         =                       sectsz=512   sunit=0 blks, lazy-count=1
realtime =none                   extsz=4096   blocks=0, rtextents=0
```

5.2. Redimensionnement d'un système de fichiers XFS

La commande `xfs_growfs` permet de modifier la taille d'un système de fichiers XFS. Sans option, elle étend la taille du système de fichiers à la taille de la partition.

EXEMPLE

- Création d'un système de fichiers XFS sur une partie de l'espace disponible de la partition `/dev/sdb1` :

```
# mkfs.xfs -f -d size=1g /dev/sdb1
meta-data=/dev/sdb1              isize=256    agcount=4, agsize=65536 blks
         =                       sectsz=512   attr=2, projid32bit=0
data     =                       bsize=4096   blocks=262144, imaxpct=25
         =                       sunit=0      swidth=0 blks
naming   =version 2              bsize=4096   ascii-ci=0
log      =internal log           bsize=4096   blocks=2560, version=2
         =                       sectsz=512   sunit=0 blks, lazy-count=1
realtime =none                   extsz=4096   blocks=0, rtextents=0
```

- Montage du système de fichiers sur le répertoire `/mnt`:

```
# mount /dev/sdb1 /mnt
```

- Redimensionnement du système de fichiers. La capacité actuelle du système de fichiers est de 262 144 blocks, on utilise la commande `xfs_growfs` pour la doubler :

```
 xfs_growfs -D 524288 /mnt
meta-data=/dev/sdb1              isize=256    agcount=4, agsize=65536 blks
         =                       sectsz=512   attr=2
data     =                       bsize=4096   blocks=262144, imaxpct=25
         =                       sunit=0      swidth=0 blks
naming   =version 2              bsize=4096   ascii-ci=0
log      =internal               bsize=4096   blocks=2560, version=2
         =                       sectsz=512   sunit=0 blks, lazy-count=1
realtime =none                   extsz=4096   blocks=0, rtextents=0
data blocks changed from 262144 to 524288
```

- Vérification de la nouvelle taille du système de fichiers avec les commandes `xfs_info` et `df` :

```
# xfs_info /mnt
meta-data=/dev/sdb1              isize=256    agcount=8, agsize=65536 blks
```

```
         =                       sectsz=512   attr=2
data     =                       bsize=4096   blocks=524288, imaxpct=25
         =                       sunit=0      swidth=0 blks
naming   =version 2              bsize=4096   ascii-ci=0
log      =internal               bsize=4096   blocks=2560, version=2
         =                       sectsz=512   sunit=0 blks, lazy-count=1
realtime =none                   extsz=4096   blocks=0, rtextents=0
# df -h /mnt
Sys. de fichiers    Taille  Uti. Disp. Uti% Monté sur
/dev/sdb1             2,0G  4,3M  2,0G   1% /mnt
```

5.3 Sauvegarde et restauration d'un système de fichiers XFS

Le paquetage `xfsdump` contient les commandes `xfsdump`, `xfsrestore` et d'autres outils de gestion du système de fichiers XFS.

La commande `xfsdump` permet de sauvegarder un système de fichiers XFS, elle analyse le système de fichiers afin de déterminer ceux qui doivent être sauvegardés et les copie sur un disque, une bande ou un autre support de stockage.

La commande `xfsrestore` exécute l'opération inverse, elle restaure une sauvegarde entière d'un système de fichiers. Des sauvegardes incrémentales ultérieures peuvent être faites après la sauvegarde complète (voir chapitre 6). La commande `xfsrestore` peut aussi restaurer des fichiers et des répertoires à partir des sauvegardes complètes ou incrémentales.

EXEMPLE

- Création des répertoires et des fichiers sur un système de fichiers XFS

```
# mkdir rep1 rep2
# touch rep1/file1 rep1/file2 rep1/file3
```

- Démontage et montage du système de fichiers afin de s'assurer que toutes les données et les métadonnées ont été écrites sur le disque. La commande `sync` ne garantit pas cela avec le système de fichiers XFS. En effet, les modifications de métadonnées peuvent être dans le journal sur le disque, mais pas encore dans les inodes.

```
# umount /mnt
# mount /dev/sdb1 /mnt
```

- Sauvegarde du système de fichiers :

```
# xfsdump -f /root/sauv /mnt
xfsdump: using file dump (drive_simple) strategy
xfsdump: version 3.0.4 (dump format 3.0) - Running single-threaded
```

```
 ============================ dump label dialog
============================
please enter label for this dump session (timeout in 300 sec)
...
```

– - Suppression des fichiers et des répertoires sur `/mnt`

```
# rm -rf /mnt/*
# ls -la /mnt
total 4
drwxr-xr-x  2 root root    6 10 sept. 10:18 .
drwxr-xr-x 22 root root 4096  7 sept. 10:05 ..
```

– - Restauration des fichiers et des répertoires supprimés :

```
# xfsrestore -f /root/sauv .
xfsrestore: using file dump (drive_simple) strategy
xfsrestore: version 3.0.4 (dump format 3.0) - Running single-threaded
xfsrestore: searching media for dump
xfsrestore: examining media file 0
xfsrestore: dump description:
xfsrestore: hostname: debian
xfsrestore: mount point: /mnt
xfsrestore: volume: /dev/sdb1
…...
```

6. Systèmes de fichiers cryptés

Les systèmes de fichiers `ext[234]` ne permettent pas de garantir la confidentialité des données enregistrées sur un périphérique de stockage. En effet les droits d'accès permettent de se protéger contre un accès non autorisé aux fichiers. Par contre, il est possible d'accéder directement au système de fichiers en plaçant le périphérique de stockage sur une autre machine.

Il existe plusieurs techniques de création de systèmes de fichiers cryptés. La technique TCFS (*Transparent Cryptographic File System*) consiste à définir une couche au dessus des autres systèmes de fichiers (`ext2`, `ext3`, `ext4`, `xfs` ..) permettant de chiffrer des données lors de leur stockage. Ainsi, les informations contenues dans un fichier seront chiffrées par cette couche avant d'être traduites sous forme de blocs de données, et écrites sur le média.

eCryptFS est un système de fichiers cryptés de type TCFS. Initialement développé par IBM, il est intégré dans le noyau Linux depuis la version 2.6.19.

EXEMPLE

L'exemple suivant illustre l'utilisation de eCryptFS pour le cryptage des fichiers sur une partition de type XFS.

- Installation du paquetage `ecryptfs-utils` :

```
apt-get install ecryptfs-utils
```

- Montage de la partition `/dev/sdb1` sur le répertoire `/mnt` :

```
mount /dev/sdb1 /mnt
```

- Montage de la partition (de type eCryptFS) à chiffrer. Ceci définit une couche permettant de crypter les données avant d'être traduites sous forme de blocs de données, et écrites sur la partition `/dev/sdb1` :

```
# mount -t ecryptfs /mnt /mnt
```

- Définition de mot de passe utilisé pour le cryptage et le décryptage des données :

```
Passphrase:
```

- Sélection de l'algorithme de chiffrement :

```
Select cipher:
 1) aes: blocksize = 16; min keysize = 16; max keysize = 32 (not loaded)
 2) blowfish: blocksize = 16; min keysize = 16; max keysize = 56 (not
loaded)
 3) des3_ede: blocksize = 8; min keysize = 24; max keysize = 24 (not
loaded)
 4) twofish: blocksize = 16; min keysize = 16; max keysize = 32 (not
loaded)
 5) cast6: blocksize = 16; min keysize = 16; max keysize = 32 (not loaded)
 6) cast5: blocksize = 8; min keysize = 5; max keysize = 16 (not loaded)
Selection [aes]:
```

- Sélection de la taille de la clé de chiffrement :

```
Select key bytes:
 1) 16
 2) 32
 3) 24
Selection [16]:
```

- Activation du cryptage des noms des fichiers :

```
Enable filename encryption (y/n) [n]: y
```

```
Filename Encryption Key (FNEK) Signature [192ce7a9c1e94e74]:
Attempting to mount with the following options:
  ecryptfs_unlink_sigs
  ecryptfs_fnek_sig=192ce7a9c1e94e74
  ecryptfs_passthrough
  ecryptfs_key_bytes=16
  ecryptfs_cipher=aes
  ecryptfs_sig=192ce7a9c1e94e74
WARNING: Based on the contents of [/root/.ecryptfs/sig-cache.txt],
it looks like you have never mounted with this key
before. This could mean that you have typed your
passphrase wrong.
Would you like to proceed with the mount (yes/no)? : yes
Would you like to append sig [192ce7a9c1e94e74] to
[/root/.ecryptfs/sig-cache.txt]
in order to avoid this warning in the future (yes/no)? : yes
Successfully appended new sig to user sig cache file
Mounted eCryptfs
```

– Création d'un fichier crypté sous le répertoire `/mnt`. Tant que le montage eCryptFS est actif, on peut lire le contenu de ce fichier :

```
# echo "test ecryptfs">> /mnt/fichier1
# cat /mnt/fichier1
test ecryptfs
```

– - Démontage du système de fichiers eCryptFS monté sur le répertoire `/mnt` :

```
#  umount /mnt
```

– - Visualisation de contenu du fichier `/mnt/fichier1` :

```
# cat /mnt/fichier1
cat: /mnt/file1: Aucun fichier ou dossier de ce type
# cd /mnt
# ls
ECRYPTFS_FNEK_ENCRYPTED.FWYN9CSdkSZCR-SvSR-
ejPZ3C2geW1FVeA7mPFMk7S6Wbuvx8aelsCNP0---
```

On peut voir aussi que le nom de fichier `/mnt/fichier1` est crypté. On pourra lire à nouveau ce fichier une fois qu'on monte `/mnt` en utilisant eCryptFS avec la même clé définie précédemment.

7. Gestion des disques optiques

7.1 Systèmes de fichiers cédérom et DVD

Le système de fichiers ISO 9660 est conçu pour les disques optiques tels que les cédéroms et les DVD. Initialement le système de fichiers ISO 9660 a imposé plusieurs limites, par exemple, les noms de fichiers sont réduits à 8 caractères avec une extension de 3 caractères. Pour remédier à ces limitations et pouvoir utiliser les supports optiques sur de différents systèmes d'exploitations, des extensions ISO 9660 sont définies :

- l'extension Joliet est définie par Microsoft, elle permet d'utiliser des noms de fichiers longs allant jusqu'à 64 caractères, comprenant des espaces et des caractères accentués ;
- l'extension Rock Ridge est compatible avec le standard POSIX permettant d'utiliser des noms de fichiers composés de plus de 255 caractères, des droits d'accès Unix, des liens symboliques, etc. ;
- l'extension El Torido est utilisée sur les cédéroms et DVD destinés au démarrage du système.

Le système de fichiers UDF (*Universal Disk Format*) est conçu pour remplacer l'ISO 9660, il supporte de plus grands fichiers et de plus grands disques. Il est supporté par les systèmes d'exploitation Linux, Windows et Mac OS. Il est utilisé sur les DVD de données, DVD vidéo, DVD audio, DVD réinscriptibles, etc.

Le système de fichiers HFS (*Hierarchical File System*) est développé par Apple et utilisé sur les disques durs et les supports optiques.

7.2 Création de systèmes de fichiers

La commande `mkisofs` crée un fichier image contenant un système de fichiers ISO 9660.

EXEMPLE

Créer une image nommée `image.iso` contenant les fichiers du répertoire `~/work/lpi/auf/` :

```
$ mkisofs -o image.iso ~/work/lpi/auf/
```

La commande `mkisofs` dispose de plusieurs options, certaines sont utilisées pour ajouter les extensions ISO 9660 :

- `-R` : ajoute l'extension Rock Ridge, les fichiers conservent alors leurs droits d'accès et leurs propriétaires ;
- `-r` : ajoute l'extension Rock Ridge, change les propriétaires des fichiers à *root* et donne un accès en lecture seul à tous les utilisateurs ;
- `-J` : ajoute l'extension Joliet ;
- `-T` : Crée des tables de traduction des fichiers destinés aux systèmes ne supportant pas l'extension Rock Ridge ;

- `-V` : définit l'étiquette ou le nom du volume du cédérom ;
- `-v` : affiche en mode bavard (*verbose*) la progression et les messages de la création d'image.

EXEMPLE

- Créer une image nommée `image.iso` contenant les fichiers du répertoire `~/work/lpi/auf/`, avec les extensions Rock Ridge. Le nom de l'étiquette de l'image ISO créée est LABEL :

```
# mkisofs -r -V 'LABEL' -v -o image.iso ~/work/lpi/auf
```

- Vérifier l'image créée en faisant le montage du système de fichiers de l'image sur le répertoire `/mnt` :

```
# mount -o loop -t iso9660 -o ro image.iso /mnt/
```

La commande `dd` permet aussi de créer un fichier image à partir d'un cédérom.

EXEMPLE

- Chercher le périphérique associé au cédérom :

```
# mount
/dev/sda1 on / type ext3 (rw,errors=remount-ro,commit=0)
proc on /proc type proc (rw,noexec,nosuid,nodev)
none on /sys type sysfs (rw,noexec,nosuid,nodev)
/dev/sda7 on /home type ext3 (rw,commit=0)
/dev/sda5 on /usr type ext3 (rw,commit=0)
...
/dev/sr0 on /media/disk type iso9660
(ro,nosuid,nodev,uhelper=udisks,uid=1000,gid=1000,iocharset=utf8,mode=0400,
dmode=0500)
```

  On peut voir que le périphérique cédérom `/dev/sr0` est monté sous le répertoire `/media/disk`

- Copier le contenu d'un cédérom vers vers un fichier image :

```
# dd if=/dev/sr0 of=monimage.iso
```

7.3 Gravure d'une image iso

La commande cdrecord permet de graver un fichier image, contenant un système de fichier ISO 9660, sur un cédérom ou un DVD.

EXEMPLE

- Récupérer les paramètres du graveur

```
$ cdrecord -scanbus
scsibus1:
1,0,0 100) 'MATSHITA' 'DVD-RAM UJ880AS ' '1.50' Removable CD-ROM
1,1,0 101) *
1,2,0 102) *
1,3,0 103) *
1,4,0 104) *
1,5,0 105) *
1,6,0 106) *
1,7,0 107) *
```

Le graveur est un « MATSHITA' 'DVD-RAM UJ880AS » identifié par l'adresse 1,0,0.

- Graver l'image :

```
# cdrecord -v speed=16 dev=1,0,0 image.iso
```

8. Gestion des périphériques avec udev

8.1 Principe de fonctionnement

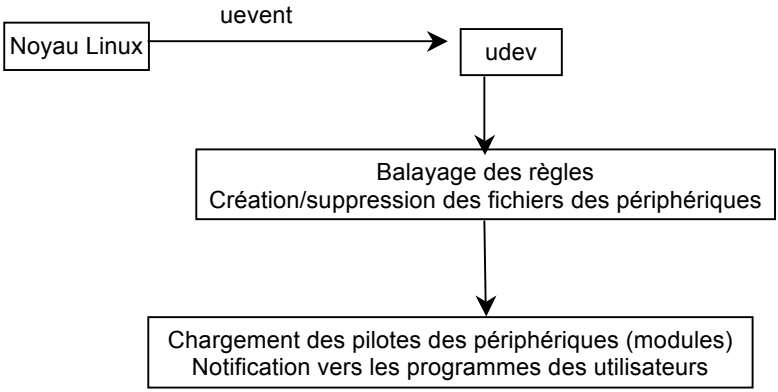
Au départ, les fichiers spéciaux des périphériques, appelés **nœuds** ou **nodes**, étaient créés manuellement dans le répertoire `/dev`. Les mainteneurs des distributions Linux étaient obligés de peupler le répertoire `/dev` avec tous les fichiers de périphériques possibles. Cette approche n'était pas idéale ni pratique.

Depuis la version 2.6.2 du noyau Linux, le service `udev` gère dynamiquement la création et la suppression des nœuds dans le répertoire `/dev` en fonction de l'ajout ou de la suppression des périphériques du système.

Le service `udev` peut aussi charger automatiquement les pilotes des périphériques, assurer que les périphériques gardent toujours les mêmes noms, indépendamment de leur ordre de branchement, et notifier aux autres programmes la présence d'un périphérique.

La *figure 7* illustre le principe de fonctionnement du service `udev` : il écoute les messages du noyau, appelés `uevent`, concernant les changements d'état des périphériques. Il peut effectuer une variété d'actions quand un périphérique est découvert ou déconnecté, pour cela il se base sur des règles définies dans les fichiers des répertoires `/etc/udev/rules.d` et `/lib/udev/rules.d/`.

Figure 7



EXEMPLE

On utilise la commande `udevadm monitor --environment` pour capturer le message du noyau (`uevent`), envoyé suite à la connexion d'un disque USB.

```
$ udevadm monitor --environment

monitor will print the received events for:
UDEV - the event which udev sends out after rule processing
KERNEL - the kernel uevent
KERNEL[1321431768.343421] add      /devices/pci0000:00/0000:00:1d.7/usb2/2-
2 (usb)
UDEV_LOG=3
ACTION=add
DEVPATH=/devices/pci0000:00/0000:00:1d.7/usb2/2-2
SUBSYSTEM=usb
DEVNAME=bus/usb/002/004
DEVTYPE=usb_device
PRODUCT=781/5406/10
TYPE=0/0/0
BUSNUM=002
DEVNUM=004
SEQNUM=1725
MAJOR=189
MINOR=131
```

Lorsque le noyau détecte le disque USB, il crée une entrée dans le système de fichiers **sysfs**. L'attribut `DEVPATH` représente l'emplacement de cette entrée dans le répertoire

/sys. L'attribut DEVNAME indique que ce périphérique est le numéro 4 sur le bus USB numéro 2.

L'attribut PRODUCT indique que le code vendeur est 781, et le code produit 5406. En se référant à la page www.linux-usb.org/usb.ids[3] on peut voir que ces codes correspondent au vendeur « SanDisk Corp » et à la clé USB « Cruzer Micro U3 ».

Le numéro de séquence est incrémenté pour chaque message uevent émis. Enfin, le numéro majeur (189) et le numéro mineur (131) de périphérique sont inclus dans le message uevent. Tous les périphériques gérés par un même pilote ont le même numéro majeur. Le numéro mineur est utilisé par le noyau afin de distinguer les périphériques gérés par un même pilote.

Udev examine les attributs du périphérique détecté pour sélectionner et appliquer les règles correspondantes.

8.2 Le système de fichiers sysfs

Les informations sur les bus et les périphériques sont exportées sous forme d'**objets**, de l'espace du noyau vers l'espace des utilisateurs, par le biais du système de fichiers **sysfs**.

Le système de fichiers **sysfs** est monté sur le répertoire /sys. Les objets sont représentés par des **sous-répertoires**, les attributs d'un objet sont représentés par des **fichiers** et la valeur d'un attribut correspond au contenu du fichier.

Par exemple le fichier /sys/block/sdc/size représente la taille du disque sdc :

```
$ cat  /sys/block/sdc/size
7978637
```

L'objet ici est un disque dur sdc représenté par le répertoire /sys/block/sdc/. L'attribut size du disque est représenté par le fichier /sys/block/sdc/size et a pour valeur la chaîne de caractères «7978637 » contenue dans le fichier /sys/block/sdc/size.

La commande udevadm permet aussi de recueillir les attributs d'un périphérique à partir de sa représentation sysfs. Ces informations permettent de construire les règles udev.

```
$ udevadm  info -a -p /sys/block/sdc

Udevadm info starts with the device specified by the devpath and then
walks up the chain of parent devices. It prints for every device
found, all possible attributes in the udev rules key format.
A rule to match, can be composed by the attributes of the device
and the attributes from one single parent device.
```

[3] On trouve dans cette page la liste de tous les périphériques USB connus. Chaque périphérique USB est identifié par un code constructeur et un code produit.

```
  looking at device '/devices/pci0000:00/0000:00:1d.7/usb2/2-2/2-
2:1.0/host9/target9:0:0/9:0:0:0/block/sdc':
    KERNEL=="sdc"
    SUBSYSTEM=="block"
    DRIVER==""
    ATTR{range}=="16"
    ATTR{ext_range}=="256"
    ATTR{removable}=="1"
    ATTR{ro}=="0"
    ATTR{size}=="7978637"
    ATTR{alignment_offset}=="0"
    ATTR{discard_alignment}=="0"
    ATTR{capability}=="51"
    ATTR{stat}=="     308     8935    10390      432        1        0
1        4        0      316      436"
    ATTR{inflight}=="       0        0"
```

8.3 Les règles udev

Chaque règle `udev` est composée de deux éléments :

- les **conditions** de déclenchement des actions ;
- les **actions** proprement dites.

Les conditions sont définies par des **clés de comparaison**, et les actions par des **clés d'assignation**.

EXEMPLE

Voici un exemple de règle `udev` extrait du fichier `/lib/udev/rules.d/97-bluetooth.rules`

```
ACTION=="add", SUBSYSTEM=="bluetooth", RUN+="/usr/sbin/bluetoothd -udev"
```

Ici, si l'action est un ajout et le périphérique est de type **Bluetooth**, le programme `/usr/sbin/bluetoothd` sera exécuté automatiquement.

Les opérateurs suivants peuvent être utilisés avec les clés de comparaison :

- == pour tester l'égalité entre la clé et une valeur ;
- != pour tester la différence entre la clé et une valeur.

Dans l'exemple précédent on a testé l'égalité entre la clé `ACTION` et la valeur « `add` » et entre la clé `SUBSYSTEM` et la valeur « `bluetooth` ».

Les opérateurs suivants sont utilisés avec les clés d'assignation :

- = pour assigner une nouvelle valeur à une clé. Si la clé contient déjà une ou plusieurs valeurs, cette nouvelle valeur les remplace ;
- += pour ajouter une valeur à la clé. Si la clé contient déjà une liste de valeurs, cette nouvelle valeur est ajoutée à cette liste ;
- := pour assigner une nouvelle valeur à une clé et interdire toute modification ultérieure par d'autres règles.

Les principales clés utilisées dans les règles `udev` sont :

- `ACTION`, contient le nom de l'évènement uevent ;
- `DEVPATH`, définit le chemin d'accès sysfs aux informations sur le périphérique ;
- `KERNEL`, indique le nom attribué par le noyau au périphérique ;
- `NAME`, définit le nom de fichier périphérique à créer sur le répertoire `/dev` ;
- `SYMLINK`, contient les noms des liens symboliques à créer ;
- `SUBSYSTEM`, indique le type de périphérique : block ou character ;
- `DRIVER`, définit le pilote de périphérique ;
- `LABEL`, contient une étiquette qui peut être utilisée avec la clé `GOTO` ;
- `GOTO`, permet de passer à l'étiquette définie avec la clé `LABEL` ;
- `RUN`, indique le nom d'un programme à exécuter.

Des caractères jokers peuvent être utilisés dans les valeurs des clés :

- `*` : remplace zéro, un ou plusieurs caractères ;
- `?` : remplace un et un seul caractère ;
- une liste de caractères entre crochets, par exemple `[13478]`, remplace un et un seul caractère, qui doit être l'un des caractères listés. Il est possible de spécifier des intervalles de caractères, par exemple le caractère remplacé par `[0-9]` peut être n'importe quel chiffre ;
- une liste de caractères entre crochets et précédée par un point d'exclamation, par exemple `[!13478]`, remplace un et un seul caractère, qui ne doit être aucun des caractères listés.

EXEMPLES

`KERNEL=="sd[!0-9]` correspond aux noms des périphériques `sda`, `sdb`, `sdc`, etc ...

`KERNEL=="tty[SR]` correspond soit à « ttyS » soit à « ttyR »

Enfin on peut utiliser dans les règles `udev` des variables prédéfinies. Les variables les plus utilisées sont :

- `%k` ou `$kernel`, nom assigné par le noyau à un périphérique ;

- `%n` ou `$number`, numéro de périphérique ;
- `$driver`, nom de pilote ;
- `%M`, numéro majeur ;
- `%m`, numéro mineur.

EXEMPLE

Voici trois règles `udev` :

```
ACTION=="add", KERNEL=="sd[a-d][0-9]", SYMLINK+="usbdisk%n", NAME="%k"
ACTION=="add", KERNEL=="sd[a-d][0-9]", RUN+="/bin/mkdir -p /media/usbdisk%n"
ACTION=="add", KERNEL=="sd[a-d][0-9]", RUN+="/bin/mount /dev/%k /media/usbdisk%n"
```

Lors de la détection de périphérique, `udev` crée un lien symbolique vers le périphérique réel appelé `/dev/usbdiskn`, où « n » est le numéro du périphérique. Par exemple, pour le premier périphérique détecté, un lien symbolique appelé `/dev/usbdisk0` pointe vers le périphérique réel. Ensuite, la directive `RUN` permet de créer un répertoire sous `/media` avec le même nom. La dernière action consiste à monter le nouveau périphérique sur un nouveau point de montage créé sur le répertoire `/media`.

9. Le montage automatique : le service autofs

Le montage automatique des systèmes de fichiers définis dans `/etc/fstab` crée un certain nombre de problèmes dans les grands réseaux. En effet, si les systèmes de fichiers sont montés via NFS (*Network File System*) à partir des serveurs sur le réseau, c'est le chaos quand un de ces serveurs tombe en panne. En plus chaque commande qui accède à des points de montage se bloque. Ainsi, la performance sera affectée lorsque le système maintient plusieurs montages de systèmes de fichiers en même temps.

Une solution alternative au fichier `/etc/fstab` est d'utiliser le service `autofs`. Ce service permet de monter des systèmes de fichiers de manière automatique et transparente au moment où on y accède et les démonte après une période d'inactivité, ce qui améliore la performance du système.

Le service `autofs` est géré par le script `/etc/init.d/autofs`. Ce dernier accepte les arguments `start`, `restart` et `stop` pour le démarrage, le redémarrage et l'arrêt du service `autofs`. L'argument `reload` permet de charger les modifications apportées à la configuration du service `autofs`. Et l'argument `status` indique s'il est démarré ou non.

Le fichier `/etc/auto.master` est le fichier de configuration principale pour `autofs`. Chaque ligne décrit un point de montage, un fichier de configuration du montage et des options de montage.

EXEMPLE

Voici un exemple de fichier de configuration principale d'`autofs` :

```
# cat /etc/auto.master
/mnt/nfs /etc/auto.nfs --timeout=10
```

Le répertoire `/mnt/nfs` est le point de montage. Le fichier de configuration `/etc/auto.nfs` contient les options de montage associées au point de montage `/mnt/nfs`.

Voici le contenu du fichier `/etc/auto.nfs` :

```
# cat /etc/auto.nfs
zied  -rw 172.31.209.135:/home/zied
hedi  -rw 172.31.209.135:/home/hedi
```

Deux sous-répertoires de montage `zied` et `hedi` seront créés sous `/mnt/nfs` et correspondent respectivement aux partages NFS `/home/zied` et `/home/hedi` sur la machine 172.31.209.135.

Les sous-répertoires `zied` et `hedi` sont créés uniquement lorsqu'on essaie d'y accéder.

Exercices

1. Quel point de montage doit-on associer avec les partitions swap ?

- ❑ A. /home
- ❑ B. /
- ❑ C. /proc
- ❑ D. aucun

2. Laquelle des options suivantes est utilisée avec la commande fsck, afin d'utiliser un type de système de fichiers ?

- ❑ A. t
- ❑ B. A
- ❑ C. f
- ❑ D. C

3. Laquelle des caractéristiques suivantes pouvez-vous ajuster avec tune2fs? ?

- ❑ A. le propriétaire du système du fichier
- ❑ B. la valeur UUID du système de fichiers
- ❑ C. la taille de système de fichier
- ❑ D. l'ajout d'un journal

Chapitre 4. RAID et LVM

Objectifs	**Configuration RAID** – configurer et implémenter un RAID logiciel ; – utiliser et configurer les niveaux 0, 1 et 5 du RAID logiciel. **Gestionnaire de volumes logiques** – créer et supprimer des volumes logiques, des groupes de volumes et des volumes physiques ; – prendre des instantanés (*snapshots*) et redimensionner des volumes logiques. **Ajustement des paramètres des périphériques de stockage** – configurer les options du noyau permettant de supporter différents pilotes de périphériques de stockage ; – utiliser les outils logiciels pour visualiser et modifier les paramètres du disque dur.
Points importants	– Fichiers de configuration et utilitaires de RAID logiciel. – Commandes de gestion des volumes LVM. – Redimensionnement, renommage, création et suppression des volumes logiques, groupes de volumes et volumes physiques. – Création et maintenance des instantanés (*snapshots*). – Activation des groupes de volumes. – Outils logiciels pour visualiser et modifier les paramètres du disque dur. – Commandes de configuration de DMA pour les périphériques IDE, ATAPI et SATA. – Outils de manipulation et d'analyse des ressources systèmes associées aux disques durs (*e.g.* interruptions).
Mots clés	`mdadm.conf`, `mdadm`, `/proc/mdstat`, `fdisk`, `hdparm`, `tune2fs`, `/sbin/pv*`, `/sbin/lv*`, `/sbin/vg*`, `mount`, `/dev/mapper/`

Un volume RAID (*Redudent Array of Independent Disk*) est constitué d'un ensemble de disques ou de partitions de disques. L'objectif est d'assurer une tolérance aux pannes, en dupliquant les données sur plusieurs disques. La performance est améliorée en permettant

la lecture ou l'écriture en parallèle des données depuis ou vers les disques qui constituent le volume RAID.

Le RAID peut être matériel ou logiciel. Le RAID matériel utilise un contrôleur matériel permettant de gérer le volume RAID. Ce contrôleur RAID est doté d'un processeur spécifique, une mémoire et un logiciel embarqué (*RAID firmware*). Le contrôleur RAID matériel cache les caractéristiques du volume RAID au système d'exploitation. Ce dernier perçoit le volume RAID comme un disque dur classique.

En RAID logiciel, le contrôleur RAID est un composant du système d'exploitation. Le module MD (*Multiple Disk*) du noyau Linux est un contrôleur RAID logiciel permettant de gérer le volume RAID et offrant aux applications un seul disque dur virtuel.

LVM (*Logical Volume Manager*) est une technique qui crée des systèmes de fichiers sur des volumes logiques, ce qui permet par exemple de redimensionner la taille des partitions en toute transparence du point de vue des applications et sans avoir besoin de redémarrer un serveur en production.

Les sections suivantes décrivent les techniques **RAID** et **LVM** et illustrent la mise en place de ces deux techniques combinées ensemble.

1. RAID

1.1 Concepts généraux

Selon le type d'architecture mis en place, la technique RAID permet d'améliorer :

- soit la performance de lecture et d'écriture, en distribuant les données sur plusieurs disques, ce qui permet au contrôleur de travailler sur plusieurs disques simultanément ;
- soit la tolérance aux pannes, en dupliquant des données sur plusieurs disques, ce qui diminue les risques en cas de défaillance de l'un d'eux ;
- soit les deux.

Il existe plusieurs types de RAID, appelés *niveaux*. Les plus utilisés sont RAID 0, RAID 1, RAID 5 et RAID 10 :

- **RAID 0** : il est utilisé uniquement pour améliorer les performances. Les données sont découpées en blocs (*chunk*) et ces blocs sont répartis sur plusieurs disques, ce qui diminue les temps de lecture et d'écriture (*figure 8*) ;
- **RAID 1** : il offre une redondance des données. En effet elles sont dupliquées sur deux ou plusieurs disques (*figure 8*). La performance de lecture augmente avec le nombre de disques du volume RAID. Ceci peut assurer une meilleure tolérance aux pannes, mais peut nuire à la performance d'écriture, car l'information doit être écrite plusieurs fois ;

Figure 8. RAID 0 et RAID 1[4]

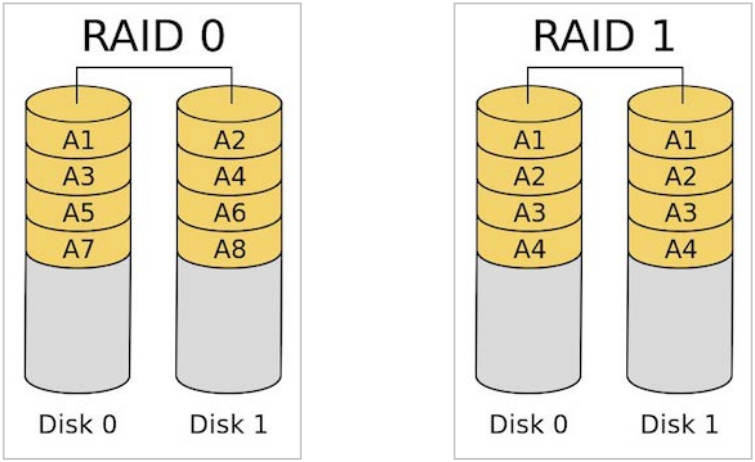
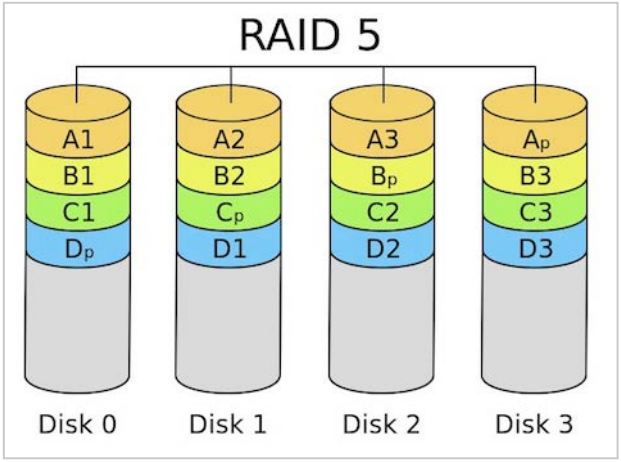


Figure 9. RAID 5[5]

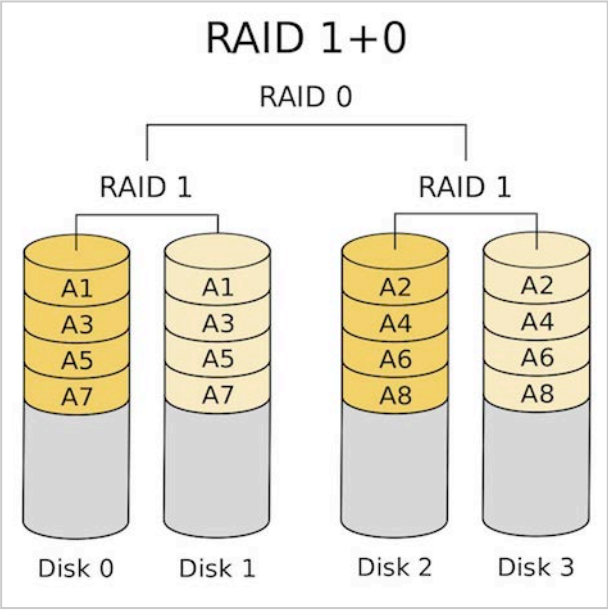


[4] Figure 1 : schémas de Cburnett diffusés sur Wikimedia Commons sous licence Creative Commons BY-SA 3.0 : http://commons.wikimedia.org/wiki/File:RAID_0.svg et http://commons.wikimedia.org/wiki/File:RAID_1.svg

[5] Figure 2 : schéma de Cburnett diffusé sur Wikimedia Commons sous licence Creative Commons BY-SA 3.0 : http://commons.wikimedia.org/wiki/File:RAID_5.svg

- **RAID 4** : les données sont distribuées en petits blocs sur les différents disques durs. Le contrôle de parité est inscrit sur un disque spécifique. Ceci permet de créer une architecture redondante qui améliore le temps d'écriture. Si un disque tombe en panne, les données manquantes seront recalculées à partir des données utiles et des parités. RAID 4 n'est plus utilisé, il est remplacé par RAID 5 ;
- **RAID 5** : à la différence de RAID 4, les données et la parité sont elles aussi distribuées sur l'ensemble des disques (*figure 9*) ;
- **RAID 10** ou (1+0) : combinaison de RAID 1 et de RAID 0. C'est un RAID 0 composé de deux volumes RAID 1 (*figure 10*), ce qui offre les avantages simultanés du RAID 1 et RAID 0 : augmentation de la taille de stockage et des performances de lecture. Il faut au minimum quatre disques de stockage pour créer un volume RAID 10.

Figure 10. RAID 10[6]



1.2 Gestion du RAID logiciel sous Linux

La plupart des distributions Linux utilisent la commande `mdadm` pour générer un volume RAID.

[6] Figure 3 : schéma de Wheart, basé sur un schéma de Cburnett et diffusé sur Wikimedia Commons sous licence Creative Commons BY-SA 3.0 : http://commons.wikimedia.org/wiki/File:RAID_10.svg

Les versions précédentes de la distribution Red Hat utilisaient le paquetage `raidtools`, mais `mdadm` est à la fois plus puissante et plus facile à utiliser que `raidtools`.

La commande `mdadm` permet de créer, contrôler et administrer le volume RAID. Elle possède sept modes de fonctionnement différents, chacun avec ses propres options (*tableau 6*).

Si aucun mode n'est spécifié, la commande `mdadm` est en mode `manage` ou `misc`. Le mode `manage` est sélectionné si le nom de périphérique RAID est spécifié avant toutes les options ou si la première option est `--add`, `--fail`, or `-remove`.

Tableau 6. Les différents modes de fonctionnement du RAID

Mode	Option pour sélectionner le mode	Description
`assemble`	`--assemble ou --A`	Assemble les composants d'un volume RAID déjà créé. Ce mode est utile pour le dépannage ou l'activation d'un volume RAID transféré depuis un autre ordinateur. L'option `-s` ou `--scan` récupère les informations sur le volume RAID et les disques sous-jacents.
`build`	`--build ou -B`	Crée un volume RAID à partir des disques qui n'ont pas de superblocs (des métadonnées).
`create`	`--create ou -C`	Crée un nouveau volume RAID avec des superblocs (des métadonnées) dans chaque disque
`manage`	*(par défaut)*	Peut ajouter ou supprimer des disques à un volume RAID en cours d'exécution. Ceci est utile pour enlever des disques durs en panne, ajouter des disques de rechange ou remplacer des disques.
`misc`	*(par défaut)*	Effectue des opérations spécifiques sur un disque sous-jacent, telles que l'effacement de superblocs ou du paramètre en lecture seule.
`monitor`	`--follow, --monitor ou -F`	Surveille un volume RAID. Peut par exemple automatiser l'envoi des alertes aux administrateurs lorsque le volume RAID rencontre des erreurs. Peut aussi exécuter automatiquement des commandes, comme enlever et réinsérer un disque pour tenter de corriger une défaillance non fatale.
`grow`	`--grow ou -G`	Change le nombre ou la taille des blocs ou des disques sous-jacents.

Les volumes RAID sont configurés dans le fichier `/etc/mdadm.conf`. Ce fichier est utilisé par la commande `mdadm` dans le mode `create` pour créer le volume RAID ou pour l'initialiser après chaque démarrage. Les principaux paramètres de ce fichier sont :

- `DEVICE` : liste des disques et partitions susceptibles d'être utilisés dans le volume RAID ;
- `ARRAY` : nom du volume RAID, par exemple `/dev/md0` ou `/dev/md/00` ;

- `level` : niveau RAID, par exemple `raid0`, `raid1`, `raid4` ou `raid5` ;
- `devices` : liste des disques ou partitions (séparés par des virgules) qui seront utilisés pour assembler le volume RAID ;
- `num-devices` : nombre de disques ou partitions sous-jacents dans le volume RAID ;
- `spare-group` : nom du groupe de secours. Les disques ou partitions appartenant à un groupe de secours peuvent être utilisés par plusieurs volumes RAID.

Les options du mode `create` sont :

- `-n` ou `--raid-devices` : nombre de disques du volume RAID ;
- `-l` ou `--level` : niveau RAID ;
- `-c` ou `--chunk` : taille en kilo-octets du bloc (*chunk*). La valeur indiquée doit être une puissance de 2. La valeur par défaut est 64 ko ;
- `-x` ou `--spare-devices` : nombre de disques de secours ;
- `-z` ou `--size` : taille en kilo-octets de l'espace à utiliser dans chaque disque pour les niveaux RAID 1, 4, 5 et 6. Cette taille doit être multiple de la taille d'un bloc (*chunk size*). Par défaut elle correspond à la taille du plus petit disque (ou partition) sous-jacent du volume RAID ;
- `-p` ou `--parity` : algorithme de parité utilisé. Par défaut c'est l'algorithme *left-symmetric*.

EXEMPLES

- Créer un volume RAID niveau 5, `/dev/md1`, composé des partitions `/dev/sda1`, `/dev/sdb1` et `/dev/sdc1` :

```
# mdadm --create /dev/md1 --raid-devices=3 /dev/sda1 /dev/sdb1 /dev/sdc1 -
level=5
```

- Commande identique à la précédente avec des options courtes :

```
# mdadm -C /dev/md1 -n3 /dev/sda1 /dev/sdb1 /dev/sdc1 -l5
```

- Créer un volume RAID à partir des informations de configuration du fichier `/etc/mdadm.conf`. La commande `mdadm` prend comme argument le nom du volume RAID :

```
# mdadm --create /dev/md0
```

- Créer deux volumes RAID, `/dev/md0` et `/dev/md1`. Le premier volume RAID est créé avec un disque de secours `/dev/sdd1`. Les deux volumes RAID utilisent le même groupe de secours `mongroupe` (défini par l'option `--spare-group`), ce qui leur permet de partager les disques de secours. Cela signifie que si un disque du volume RAID `/dev/md1` tombe en panne, il sera remplacé automatiquement par le disque `/dev/sdd1` :

```
# mdadm --create /dev/md0 --raid-devices=3 /dev/sda1 /dev/sdc1 -x
```

```
/dev/sdd1 --level=1 --spare-group= mongroupe
# mdadm --create /dev/md1 --raid-devices=2 /dev/sda2 /dev/sdc2 --level=1
--spare-group= mongroupe
```

2. LVM

2.1 Concepts généraux du LVM

La gestion des volumes logiques définit une couche d'abstraction de haut niveau sur les partitions physiques du disque dur. Les volumes physiques (*physical volumes* ou **pv**) sont regroupés pour former des groupes de volumes (*volume group*s ou vg). Les volumes physiques peuvent être des disques, des partitions ou même des volumes RAID. Dans un groupe de volumes on peut créer plusieurs volumes logiques qui seront accessibles comme des partitions classiques. Enfin, sur ces volumes logiques, on peut créer des systèmes de fichiers et les monter sur l'arborescence système.

LVM offre ainsi plusieurs avantages. On peut :

- avoir une utilisation et une allocation efficaces de l'espace de stockage, puisque les volumes logiques sont répartis sur plusieurs disques physiques ;
- augmenter et réduire la taille des volumes logiques sans risque d'interrompre des services du système ni de perdre des données ;
- on peut prendre des instantanés (*snapshots*) sur le système de fichiers. Ces instantanés servent à sauvegarder et restaurer les données.

2.2 Gestion du LVM sous Linux

La configuration LVM sous Linux est faite en trois étapes :

- création et initialisation des volumes physiques ;
- ajout des volumes physiques à un groupe de volumes ;
- création des volumes logiques au sein du groupe de volumes.

Les commandes LVM commencent par deux lettres qui reflètent le niveau d'abstraction LVM :

- les commandes `pv` manipulent les volumes physiques ;
- les commandes `vg` manipulent les groupes de volumes ;
- les commandes `lv` manipulent les volumes logiques.

Les principales commandes de gestion des volumes physiques sont :

- `pvcreate` : initialise un périphérique (partition, disque ou volume RAID) comme un volume physique pour une utilisation par LVM ;
- `pvdisplay` : affiche des informations détaillées sur un volume physique, y compris le nom du groupe de volumes auquel il appartient et sa taille ;

- `pvscan` : analyse les partitions de disque à la recherche des périphériques de blocs contenant des volumes physiques ;
- `pvck` : contrôle la cohérence du volume physique ;
- `pvs` : affiche des informations sommaires sur les volumes physiques.

Les principales commandes de gestion des groupes de volumes sont :

- `vgcreate` : crée un groupe de volumes ;
- `vgchange` : modifie certains attributs d'un groupe de volumes, par exemple pour l'activer ou le désactiver ;
- `vgdisplay` : affiche les caractéristiques détaillées d'un volume logique. L'option `-v` permet de visualiser la liste des volumes logiques et des volumes physiques de chaque groupe de volumes ;
- `vgscan` : analyse le système pour rechercher des groupes de volumes ;
- `vgextend` : ajoute des volumes physiques à un groupe de volumes existant ;
- `vgreduce` : supprime un ou plusieurs volumes physiques d'un groupe de volumes ;
- `vgremove` : supprime un groupe de volumes ;
- `vgrename` : renomme un groupe de volumes ;
- `vgs` : affiche des informations sommaires sur les groupes de volumes.

Les principales commandes de gestion des volumes logiques :

- `lvdisplay` : affiche des informations détaillées sur les volumes logiques ;
- `lvcreate` : crée un volume logique ;
- `lvrename` : renomme un volume logique ;
- `lvchange` : change les attributs d'un volume logique ;
- `lvextend` : augmente la taille d'un volume logique ;
- `lvreduce` : réduit la taille d'un volume logique ;
- `lvremove` : supprime des volumes logiques ;
- `lvs` : affiche des informations sommaires sur les volumes logiques ;
- `lvresize` : redimensionne un volume logique (équivalent à la fois à `lvextend` et `lvreduce`) ;
- `lvscan` : analyse le système pour rechercher des volumes logiques.

3. Exemple de configuration avec LVM et RAID logiciel

Dans ce qui suit, on étudie un exemple de mise en place d'une architecture **LVM** et **RAID**. Cette configuration est particulièrement utile pour les serveurs.

L'objectif est de créer un volume **RAID 5** sur trois disques vides, puis de définir sur ce volume deux partitions **LVM,** web1 et web2.

3.1 Création du volume RAID

Le premier disque `/dev/sda` contient les partitions du système. On commence par créer des tables de partitions dans les autres disques non partitionnés `/dev/sdb`, `/dev/sdc` et `/dev/sdd`.

Pour le disque `/dev/sdb` (à répéter ensuite pour les disques `/dev/sdc` et `/dev/sdd`) :

```
# fdisk /dev/sdb

Commande (m pour l'aide): n
Commande d'action
   e   étendue
   p   partition primaire (1-4)
1
Numéro de partition non valide pour le type « 1 »
Commande d'action
   e   étendue
   p   partition primaire (1-4)
p
Numéro de partition (1-4): 1
Premier cylindre (1-509, par défaut 1):
Utilisation de la valeur par défaut 1
Dernier cylindre, +cylindres or +taille{K,M,G} (1-509, par défaut 509):
Utilisation de la valeur par défaut 509

Commande (m pour l'aide): type
Partition sélectionnée 1
Code Hexa (taper L pour lister les codes): fd
Type système de partition modifié de 1 à fd (Linux raid autodetect)
```

Une seule partition est créée dans chaque disque. Le type de chaque partition est représenté par le code **fd** (*Linux Raid Autodetect*). Ce type permet au système RAID de Linux de détecter automatiquement ces partitions.

La commande suivante génère un volume RAID 5 sur les trois partitions `/dev/sdb1`, `/dev/sdc1` et `/dev/sdd1` :

```
# mdadm --create /dev/md0 --level=5 --raid-devices=3 /dev/sdb1 /dev/sdc1
/dev/sdd1
mdadm: array /dev/md/0 started.
```

Le fichier `/proc/mdstat` contient l'état courant du volume RAID :

```
# watch cat /proc/mdstat
     Personalities : [raid6] [raid5] [raid4]
     md0 : active raid5 sdd1[3] sdc1[1] sdb1[0]
           8385664 blocks level 5, 64k chunk, algorithm 2 [3/2] [UU_]
           [=============>.......]  recovery = 68.1% (2859008/4192832)
     finish=0.5min speed=44160K/sec

     unused devices: <none>
```

Il est particulièrement utile de surveiller ce fichier après l'ajout d'un nouveau disque ou le remplacement d'un disque défectueux.

On peut aussi voir des messages de notification sur la création du volume RAID dans le fichier `/var/log/messages`.

La commande `mdadm` sert aussi à activer le volume RAID et à le rendre disponible pour les utilisateurs. Il est également recommandé de créer un fichier de configuration RAID, `/etc/mdadm.conf`, pour servir de secours en cas de problème et simplifier l'activation du volume RAID en dispensant de redonner ses paramètres.

On peut utiliser la commande `mdadm --details --scan` pour écrire la configuration courante du volume RAID dans un fichier de configuration :

```
# echo DEVICE /dev/sdb1 /dev/sdc1 /dev/sdd1 > /etc/mdadm.conf
#  mdadm --detail --scan >> /etc/mdadm.conf
```

La commande `mdadm` peut maintenant lire ce fichier de configuration au démarrage ou à l'arrêt du système.

Pour activer le volume RAID on exécute la commande :

```
# mdadm -As /dev/md0
mdadm: /dev/md/0 has been started with 3 drives.
```

Pour arrêter le volume RAID on exécute la commande :

```
#  mdadm -S /dev/md0
mdadm: stopped /dev/md0
```

On a maintenant réuni les trois disques dans un volume RAID unique. Sur ce volume on va définir des groupes de volumes logiques sur lesquels on pourra créer des systèmes de fichiers.

3.2 Création du LVM

Les versions anciennes du LVM nécessitent de lancer la commande `vgscan` comme étape

initiale, mais ceci n'est plus nécessaire, on commence plutôt par initialiser chaque périphérique physique avec la commande `pvcreate`. Dans l'exemple on utilise le volume physique RAID 5 `/dev/md0` que l'on a créé :

```
# pvcreate /dev/md0
  Physical volume "/dev/md0" successfully created
```

L'opération précédente détruit toutes les données sur les disques et les partitions. Bien qu'on utilise un seul volume physique dans cet exemple, LVM permet d'ajouter plusieurs volumes de types différents dans un groupe de volumes.

On peut maintenant créer le groupe de volumes VG1 :

```
# vgcreate VG1 /dev/md0
  Volume group "VG1" successfully created
```

Pour examiner les caractéristiques du groupe de volume, on utilise la commande `vgdisplay` :

```
# vgdisplay VG1
  --- Volume group ---
  VG Name               VG1
  System ID
  Format                lvm2
  Metadata Areas        1
  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                1
  Act PV                1
  VG Size               8,00 GB
  PE Size               4,00 MB
  Total PE              2047
  Alloc PE / Size       0 / 0
  Free  PE / Size       2047 / 8,00 GB
  VG UUID               1pqn22-KEu1-uJde-0rgB-WhiK-KjV1-7ZTc0G
```

La dernière étape est de créer dans le groupe de volumes VG1 des volumes logiques qui seront accessibles ensuite comme des partitions du disque :

```
# lvcreate -L 2G -n web1 VG1
  Logical volume "web1" created
```

```
# lvcreate -L 2G -n web2 VG1
  Logical volume "web2" created
```

Ces volumes logiques, une fois créés, sont représentés par des fichiers de périphérique situés dans `/dev/mapper/` :

```
$ ls -l /dev/mapper/
total 0
crw-rw----. 1 root root  10, 63 sept.  7 12:56 control
brw-rw----. 1 root disk 253,  1 sept.  7 12:56 VG1-web1
brw-rw----. 1 root disk 253,  0 sept.  7 12:56 VG1-web2
```

Maintenant que les deux volumes logiques web1 et web2 sont créés dans le groupe de volumes VG1, on peut créer et monter les systèmes de fichiers :

```
#  mke2fs -j /dev/VG1/web1
mke2fs 1.41.4 (27-Jan-2009)
Étiquette de système de fichiers=
Type de système d'exploitation : Linux
Taille de bloc=4096 (log=2)
Taille de fragment=4096 (log=2)
131072 i-noeuds, 524288 blocs
26214 blocs (5.00%) réservés pour le super utilisateur
Premier bloc de données=0
Nombre maximum de blocs du système de fichiers=536870912
16 groupes de blocs
32768 blocs par groupe, 32768 fragments par groupe
8192 i-noeuds par groupe
Superblocs de secours stockés sur les blocs :
   32768, 98304, 163840, 229376, 294912
Écriture des tables d'i-noeuds : complété
Création du journal (16384 blocs) : complété
Écriture des superblocs et de l'information de comptabilité du système de
fichiers : complété

# mke2fs -j /dev/VG1/web2
# mkdir /mnt/web1 /mnt/web2

# mount /dev/VG1/web1 /mnt/web1
# mount /dev/VG1/web2 /mnt/web2
```

Finalement les systèmes de fichiers sont prêts à être utilisés. On ajoute les nouveaux systèmes de fichiers dans le fichier `/etc/fstab` et on redémarre le système.

3.3 Simulation d'une panne

En cas de partition corrompue ou de panne de disque, il est important de savoir résoudre le problème rapidement. Le volume RAID 5 qu'on a construit précédemment va continuer de fonctionner en cas de panne d'un disque. En effet il offre une redondance des données. Les utilisateurs ne seront pas nécessairement conscients de ces problèmes.

La commande `mdadm` offre la possibilité de simuler une panne de disque. Dans l'exemple ci-dessous elle simule une panne au niveau de la partition `/dev/sdc1` :

```
# mdadm /dev/md0 -f /dev/sdc1
mdadm: set /dev/sdc1 faulty in /dev/md0
```

Le fichier journal `/var/log/messages` contient les informations sur la panne simulée :

```
# cat /var/log/messages
Apr 26 17:37:21 tabarka kernel: raid5: Disk failure on sdc1, disabling
device.
Apr 26 17:37:21 tabarka kernel: raid5: Operation continuing on 2 devices.
Apr 26 17:37:21 tabarka kernel: RAID5 conf printout:
Apr 26 17:37:21 tabarka kernel: --- rd:3 wd:2
Apr 26 17:37:21 tabarka kernel: disk 0, o:1, dev:sdb1
Apr 26 17:37:21 tabarka kernel: disk 1, o:0, dev:sdc1
Apr 26 17:37:21 tabarka kernel: disk 2, o:1, dev:sdd1
Apr 26 17:37:21 tabarka kernel: RAID5 conf printout:
Apr 26 17:37:21 tabarka kernel: --- rd:3 wd:2
Apr 26 17:37:21 tabarka kernel: disk 0, o:1, dev:sdb1
Apr 26 17:37:21 tabarka kernel: disk 2, o:1, dev:sdd1
Apr 26 17:37:55 tabarka pulseaudio[1932]: ratelimit.c: 27 events suppressed
Apr 26 17:38:00 tabarka pulseaudio[1932]: ratelimit.c: 6437 events
suppressed
Apr 26 17:38:22 tabarka pulseaudio[1932]: ratelimit.c: 55 events suppressed
```

Des informations similaires sont aussi disponibles dans le fichier d'état du volume RAID `/proc/mdstat` :

```
# cat /proc/mdstat

Personalities : [raid6] [raid5] [raid4]
md0 : active raid5 sdb1[0] sdd1[2] sdc1[3](F)
      8385664 blocks level 5, 64k chunk, algorithm 2 [3/2] [U_U]
```

À ce stade l'administrateur doit prendre les mesures suivantes :

– supprimer le disque défectueux du volume RAID avec l'option `-r` de la commande `mdadm` :

```
#mdadm /dev/md0 -r /dev/sdc1
mdadm: hot removed /dev/sdc1
```

- procéder au remplacement du disque (dans l'exemple, la panne est juste une simulation) ;
- enfin, ajouter le périphérique au volume RAID avec la commande suivante :

```
# mdadm /dev/md0  -a /dev/sdc1
mdadm: re-added /dev/sdc1
```

3.4 La réaffectation d'espace de stockage

On suppose que la taille de la partition `/mnt/web1` a augmenté de façon non prévue. Le redimensionnement des partitions LVM est simple et dépend du type de système de fichiers. Les étapes suivantes sont applicables sur un système de fichiers `ext3`.

On a laissé dans le groupe de volumes VG1 un espace supplémentaire qui sera utilisé pour agrandir la taille de la partition `/mnt/web1`.

La commande `vgdisplay` sert à examiner l'espace disponible dans le groupe de volumes et la commande `df` indique l'espace disponible dans une partition :

```
# df /mnt/web1
Sys. de fich.        1K-blocs       Occupé Disponible Capacité Monté sur
/dev/mapper/VG1-web1
                       2064208   1543352    416000  79% /mnt/web1
# df /mnt/web2
Sys. de fich.        1K-blocs       Occupé Disponible Capacité Monté sur
/dev/mapper/VG1-web2
                       2064208     68676   1890676   4% /mnt/web2
```

La commande précédente montre que 79 % de la partition `/mnt/web1` est déjà occupée. Pour augmenter la taille de cette partition, on utilise la commande `lvextend`, qui ajoute de l'espace à un volume logique et ensuite la commande `resize2fs`, qui redimensionne la structure du système de fichiers :

```
# lvextend -L+2G /dev/VG1/web1
  Extending logical volume web1 to 4,00 GB
  Logical volume web1 successfully resized
# resize2fs /dev/VG1/web1
resize2fs 1.41.4 (27-Jan-2009)
Le système de fichiers de /dev/VG1/web1 est monté sur /mnt/web1 ; le
changement de taille doit être effectué en ligne
old desc_blocks = 1, new_desc_blocks = 1
En train d'effectuer un changement de taille en ligne de /dev/VG1/web1 vers
```

```
1048576 (4k) blocs.
Le système de fichiers /dev/VG1/web1 a maintenant une taille de 1048576
blocs.
```

Enfin on peut utiliser la commande `df` pour vérifier le nouvel espace disponible dans `/mnt/web1` :

```
# df /mnt/web1
Sys. de fich.        1K-blocs       Occupé Disponible Capacité Monté sur
/dev/mapper/VG1-web1
                       4128448   1800620   2118140  46% /mnt/web1
```

4. Ajustement des paramètres d'accès aux disques

Les serveurs connectent généralement leurs disques à travers une interface SCSI (*Small Computer Systems Interface*). Sur les ordinateurs de bureau et les ordinateurs portables c'est l'interface IDE (*Integrated Drive Electronics*) qui est utilisée. Il existe plusieurs pilotes pour ces différents types de disques et d'interfaces.

4.1 Interfaces des disques durs

L'interface du disque dur sert à transmettre les données entre le disque et son contrôleur qui les transmet alors au système. Il existe plusieurs interfaces qui se distinguent par des contrôleurs, des connecteurs, des débits et par le nombre de disques gérés.

PATA

L'interface PATA (*Parallel Advanced Technology Attachment*), permet de relier les périphériques de masse (disques, lecteurs de cédéroms ...) à la carte mère par des câbles plats, souples et composés de 40 ou 80 broches.

On peut connecter deux disques (un maître et un esclave) sur le même câble. L'interface PATA est aussi connue sous le nom IDE (IDE) ou *Enhanced IDE* (eIDE).

L'interface PATA utilise les standards ATA (*Advanced Technology Attachment*) et ATAPI (*Parallel Advanced Technology Attachment Packet Interface*)

SATA

L'interface SATA (S*erial ATA*) succède à l'interface PATA. Elle permet un débit de transfert plus élevé. La méthode de transfert des données est en série. Chaque disque dur est relié à son propre contrôleur via son propre câble ainsi chaque disque bénéficie de la totalité de la bande passante. Les câbles utilisés sont beaucoup plus minces que les câbles des disques PATA, ils ne comportent que 7 fils, ce qui encombre moins les boîtiers des ordinateurs.

L'interface SATA supporte le branchement à chaud (*hot plug*). Les périphériques SATA

sont internes à l'ordinateur. Une variante de l'interface, connue sous le nom de eSATA, est utilisée pour les disques durs externes.

SCSI

L'interface SCSI (*Small Computer System Interface*) permet de relier, simultanément et en série, plusieurs périphériques tels que les disques durs, les lecteurs de cédéroms, les graveurs, les scanners. etc. Selon sa version, l'interface SCSI peut prendre en charge de 8 à 16 périphériques par câble. Le contrôleur SCSI est considéré comme un périphérique à part entière, de sorte qu'on peut brancher réellement 7 ou 15 périphériques.

L'interface SCSI se présente sous la forme d'une petite carte comportant un micro-contrôleur indépendant du processeur, qui permet de bien le soulager et d'augmenter les performances systèmes.

Généralement on trouve l'interface SCSI dans les serveurs et les stations de travail haut de gamme. Elle est moins utilisée sur les ordinateurs du bureau et les ordinateurs portables à cause de son coût élevé.

USB

L'interface USB (*Universal Serial Bus*) permet de relier plusieurs types de périphériques externes à un ordinateur, y compris les disques durs et les lecteurs flash. La première et la deuxième génération de l'USB sont peu performantes, mais l'USB 3.0 est nettement plus rapide.

4.2 Ressources utilisés par les disques

Les contrôleurs de disques durs utilisent des ressources matérielles. Généralement ces ressources sont gérées automatiquement par le noyau Linux et ses pilotes.

Une ressource matérielle importante, utilisée par le contrôleur de disque dur, est la demande d'interruption (IRQ – *Interruption Request*).

L'architecture traditionnelle Intel x86 prend en charge 16 interruptions, numérotées de 0 à 15. Les interruptions numéros 14 et 15 sont dédiées respectivement aux contrôleurs PATA primaire et secondaire.

Les architectures modernes des PC prennent en charge plus d'interruptions. Le fichier `/proc/interrupts` regroupe les interruptions en cours d'utilisation :

```
$ cat /proc/interrupts
           CPU0       CPU1
  0:    1081739    1088752   IO-APIC-edge      timer
  1:       2106       1859   IO-APIC-edge      i8042
  8:          0          1   IO-APIC-edge      rtc0
  9:       1143       1113   IO-APIC-fasteoi   acpi
 12:         67         56   IO-APIC-edge      i8042
 16:          1          0   IO-APIC-fasteoi   uhci_hcd:usb3, mmc0,
```

```
firewire_ohci
 18:          0          0   IO-APIC-fasteoi   uhci_hcd:usb8
 19:       2129       2172   IO-APIC-fasteoi   ehci_hcd:usb1,
uhci_hcd:usb5, uhci_hcd:usb7
 21:          0          0   IO-APIC-fasteoi   uhci_hcd:usb4
 23:      26776      24158   IO-APIC-fasteoi   ehci_hcd:usb2, uhci_hcd:usb6
 44:          0          1   PCI-MSI-edge      sky2@pci:0000:03:00.0
 45:      20330      19504   PCI-MSI-edge      ahci
 46:      44632      40806   PCI-MSI-edge      iwlagn
 47:        264        263   PCI-MSI-edge      hda_intel
 48:         28         32   PCI-MSI-edge      hda_intel
 49:      68611      67586   PCI-MSI-edge      fglrx[0]@PCI:1:0:0
NMI:          0          0   Non-maskable interrupts
LOC:     610822     668286   Local timer interrupts
SPU:          0          0   Spurious interrupts
PMI:          0          0   Performance monitoring interrupts
PND:          0          0   Performance pending work
RES:      34364      36349   Rescheduling interrupts
CAL:        272         82   Function call interrupts
TLB:      40172      32896   TLB shootdowns
TRM:          0          0   Thermal event interrupts
THR:          0          0   Threshold APIC interrupts
MCE:          0          0   Machine check exceptions
MCP:          9          9   Machine check polls
ERR:          1
MIS:          0
```

La dernière colonne du résultat de la commande ci-dessus indique le nom du périphérique. On peut constater que les interruptions numéros 47 et 48 sont associées au pilote du disque dur hda_intel et que l'interruption numéro 45 est liée à la méthode d'accès AHCI (*Advanced Host Controller Interface*) qui permet de communiquer avec les disques durs SATA.

Un autre type de ressource matérielle qu'on peut utiliser est le DMA (*Direct Memory Access*). Dans cette configuration le contrôleur de disques durs fait le transfert des données directement vers et à partir d'un espace mémoire, sans passer par le microprocesseur, ce qui permet d'améliorer la performance du système.

Le fichier `/proc/dma` contient la liste des canaux DMA enregistrés et utilisés.

4.3 Modification des paramètres disques

Il existe plusieurs commandes pour ajuster les paramètres disque :

- `hdparm` pour les disques IDE ;
- `sdparm` pour les disques SCSI et SATA.

La commande `hdparm` communique avec le pilote IDE afin d'obtenir et modifier des paramètres du disque tels que mémoire cache, gestion d'énergie APM (*Advanced Power Management*), gestion acoustique et DMA. En réglant ces paramètres on peut améliorer les performances du disque IDE.

Les options de la commande `hdparm` utilisées pour améliorer les performances des disques IDE sont :

- `-d`*numéro* : active ou désactive le paramètre « *using_dma* » du disque ; `-d0` active le mode PIO (*Programmed Input/Output*) ; `-d1` active le mode DMA. Cette option fonctionne maintenant avec la majorité des combinaisons de disques et d'interfaces qui supportent le DMA et sont reconnus par le gestionnaire de périphériques IDE. En général l'option `-d` est utilisée avec l'option `-X` pour s'assurer que le disque lui-même est programmé pour le mode DMA approprié, bien que la plupart des BIOS le fassent au démarrage.

  Activer le DMA donne pratiquement toujours les meilleures performances, avec des entrées-sorties à haut débit et une faible utilisation du processeur. Cependant, il y a quand même quelques configurations de *chipsets* et de disques pour lesquelles le DMA ne fait pas beaucoup de différence, voire ralentit la machine ;
- `-p mode` : définit le mode PIO, qui varie de 0 à 5 dans la plupart des cas. Augmenter le mode PIO correspond à une meilleure performance ;
- `-c mode` : affiche ou définit le mode de transfert 32-bits. Omettre le mode permet d'afficher la valeur courante. La valeur 0 du mode désactive le support d'entrée-sortie 32-bits, la valeur 1 active les transferts de donnés 32-bits, et la valeur 3 active les transferts de donnés 32-bits avec une séquence spéciale *sync* requise par de nombreux contrôleurs ;
- `-S délai` : fixe le délai avant la suspension (*spindown*) du disque. Cette valeur est utilisée par le disque pour déterminer combien de temps attendre (sans activité disque) avant de suspendre la rotation du moteur pour économiser l'énergie. Dans de telles circonstances, le disque peut mettre jusqu'à une trentaine de secondes pour répondre à un accès, bien que la plupart des disques soient bien plus rapides. Le délai est indiqué dans une unité qui varie selon l'intervalle :
  - la valeur zéro signifie « pas de suspension »,
  - les valeurs comprises entre 1 à 240 sont multiples de 5 secondes. Par exemple, 120 signifie 600 secondes, ou 10 minutes,
  - les valeurs comprises entre 241 à 251 spécifient de 1 à 11 unités de 30 minutes,
  - la valeur 252 signifie un délai de 21 minutes,
  - la valeur 253 fixe un délai défini par le constructeur,
  - la valeur 254 est réservée,
  - la valeur 255 est interprétée comme 21 minutes et 15 secondes ;
- `-v` : affiche tous les paramètres du disque. Ceci est aussi le comportement par défaut quand aucune option n'est spécifiée ;

- `-X modeTransfert` : définit le mode de transfert DMA utilisé par un disque. Cette option est habituellement utilisée en combinaison avec `-d1` pour activer le mode DMA. Le `modeTransfert` peut prendre les valeurs `sdmax`, `mdmax`, ou `udmax`. Ces valeurs correspondent respectivement au transfert simple-mot DMA, multimot DMA ou rafales ultra DMA. Le nombre `x` représente la valeur du mode DMA. Les disques modernes sont capables de supporter des modes de transferts tels que `-X udma5` ou `-X udma6`. Cette option doit être utilisée avec prudence, un mode incorrect peut rendre le disque inaccessible.

EXEMPLES

Important : L'utilisation incorrecte de la commande `hdparm` peut détruire les données et, dans certains cas, le disque dur !

- Afficher les paramètres du disque `/dev/sda` :

```
# hdparm -v /dev/sda

/dev/sda:
 multcount     = 128 (on)
 IO_support    =  1 (32-bit)
 readonly      =  0 (off)
 readahead     = 256 (on)
 geometry      = 2088/255/63, sectors = 33554432, start = 0
```

- Interroger le pilote à propos des paramètres modifiables du disque `/dev/sda` :

```
# hdparm -i /dev/sda

/dev/sda:
Model=VBOX, FwRev=1.0, SerialNo=VBc3f20747-1a6d6cf5
 Config={ Fixed }
 RawCHS=16383/16/63, TrkSize=0, SectSize=512, ECCbytes=0
 BuffType=DualPortCache, BuffSize=256kB, MaxMultSect=128, MultSect=128
 CurCHS=16383/16/63, CurSects=16514064, LBA=yes, LBAsects=33554432
 IORDY=yes, tPIO={min:120,w/IORDY:120}, tDMA={min:120,rec:120}
 PIO modes:  pio0 pio3 pio4
 DMA modes:  mdma0 mdma1 mdma2
 UDMA modes: udma0 udma1 udma2 udma3 udma4 udma5 *udma6
 AdvancedPM=no WriteCache=enabled
 Drive conforms to: unknown:  ATA/ATAPI-1,2,3,4,5,6

 * signifies the current active mode
```

- Tester la vitesse de lecture :
  - l'option `-t` affiche la vitesse de lecture à travers la mémoire cache, sur le disque,

sans aucune mise en cache préalable des données ;

- l'option `-T` affiche la vitesse de lecture directement à partir de la mémoire cache sans accès au disque :

```
# hdparm -tT /dev/sda
 /dev/sda:
 Timing cached reads:   2970 MB in  1.97 seconds = 1507.47 MB/sec
 Timing buffered disk reads:  154 MB in  3.02 seconds =  50.98 MB/sec
```

- Activer les transferts des données 32-bits :

```
# hdparm -c1 /dev/sda
/dev/sda:
 setting 32-bit IO_support flag to 1
 IO_support    =  1 (32-bit)
```

Exercices

1. Parmi les commandes suivantes, laquelle regroupe un volume RAID 1 à partir des périphériques composant `/dev/sda1` et `/dev/hda2` ?

- ❑ A. `mdadm --create --level=1 --raid-devices=2 /dev/sda1 /dev/hda2`
- ❑ B. `mdadm --level=5 --raid-devices=2 /dev/sda1 /dev/hda2`
- ❑ C. `mdadm --create /dev/md0 --level=1 --raid-devices=2 /dev/sda1 /dev/hda2`
- ❑ D. `mdadm --level=1 --raid-devices=2 /dev/sda1 /dev/hda2`

2. Que pouvez-vous conclure de la ligne suivante du fichier `/proc/mdstat` ?

`md0 : active raid4 sdd2[2] sdc1[1] sda8[0]`

- ❑ A. `/dev/md0` est un volume RAID 4 construit à partir des partitions `/dev/sda8`, `/dev/sdc1` et `/dev/sdd2`.
- ❑ B. Une partition manque au volume RAID `/dev/md0`. Étant donné son niveau de RAID 4, il devrait avoir quatre partitions, .
- ❑ C. Le volume RAID /dev/md0 est mal configuré.
- ❑ D. Aucune des réponses précédentes.

3. Dans une configuration LVM, quelle commande est utilisée pour initialiser une partition afin qu'elle puisse fonctionner comme un volume physique :

- ❑ A. `lvconvert`

- ❑ B. `lvcreate`
- ❑ C. `pvcreate`
- ❑ D. `pvchange`

4. Où êtes-vous susceptibles de trouver des nœuds d'un périphérique pour le groupe de volumes MonGroupe ?

- ❑ A. `/dev/mongroupe`
- ❑ B. `/dev/mapper`
- ❑ C. `/dev/lvm`
- ❑ D. `/dev/LVM/MonGroupe`

Chapitre 5. Configuration réseau

Objectifs

Configuration élémentaire d'un réseau :
- Configurer des interfaces réseaux afin de se connecter à un réseau local, filaire ou sans fil, et à un réseau étendu ;
- Communiquer entre les différents sous-réseaux d'un même réseau y compris les réseaux IPv4 et IPv6.

Configuration avancée d'un réseau et dépannage :
- Configurer un hôte multiréseau ;
- Configurer un client VPN ;
- Résolution des problèmes de communication.

Dépannage des problèmes réseaux :
- Identifier et corriger les problèmes réseaux les plus fréquents ;
- Reconnaître les emplacements des fichiers de configuration.

Notification des utilisateurs :
- Notifier aux utilisateurs les problèmes relatifs au système.

Points importants

- Utilitaires de configuration et de manipulation des interfaces Ethernet.
- Configuration des réseaux sans fil.
- Utilitaires de manipulation des tables de routage.
- Utilitaires d'analyse de l'état des interfaces réseaux.
- Utilitaires de suivi et d'analyse du trafic TCP/IP
- OpenVPN.
- Localisation et contenu des fichiers de contrôle d'accès.
- Utilitaires de détection et d'affichage de l'état d'un réseau.
- Utilitaires de récupération d'information sur la configuration réseau.
- Méthodes d'information sur les périphériques détectés et utilisés.
- Fichiers et *scripts* d'initialisation du système.
- Automatisation de la communication avec les utilisateurs à travers les messages de connexion.
- Notification aux utilisateurs des opérations de maintenance du système.

Mots clés

- `ifconfig`, `ip`, `arp`, `iwconfig`, `hostname`, `route`, `openvpn`, `ping`, `traceroute`, `dig`, `netstat`, `tcpdump`, `lsof`, `nc`, `nmap`, `wireshark`, `dmesg`, `host`, `wall`, `shutdown`.
- `/etc/network || /etc/sysconfig/network-scripts/`, `/etc/hostname |`

/etc/HOSTNAME, /etc/resolv.conf, /etc/hosts, /var/log/syslog, /var/log/messages, /etc/openvpn/*, /etc/hosts.allow, /etc/hosts.deny, /etc/issue, /etc/issue.net, /etc/motd.

La connexion d'un hôte à un réseau IP nécessite des opérations d'installation et de configuration. Les étapes de la mise en réseau sont :

- installation d'une interface réseau : cette étape consiste à installer une (ou plusieurs) carte réseau et à paramétrer l'interface réseau correspondante. Elle suit la procédure générale d'installation d'un périphérique. Lors de l'installation du système Linux, les cartes réseaux sont automatiquement détectées et configurées ;
- configuration IP : cette étape traite la configuration de niveau réseau ;
- test de la configuration : cette étape consiste à tester la configuration précédemment effectuée et à utiliser les utilitaires de diagnostic en cas de problèmes réseaux.

1. Interface réseau

1.1. Détection des interfaces réseaux

Les interfaces réseaux sont identifiées par des noms de la forme `type-numéro` où `type` spécifie le type de l'interface réseau et `numéro` est l'ordre de l'interface. Les types d'interface sont « eth » pour les interface Ethernet, « wlan » pour les interfaces Wi-Fi, « ppp » pour les interfaces point à point, etc. Par exemple eth0 est la première interface Ethernet, eth1 est la deuxième, etc.

La commande `lshw` permet d'identifier les interfaces réseaux. Elle affiche pour chaque interface l'information sur le bus, le détail du pilote et les fonctionnalités supportées.

EXEMPLE

La commande `lshw` de cet exemple détecte deux interfaces réseaux connectées sur le bus PCI. La première est une interface de type Ethernet, nommée « eth0 » et gérée par le pilote « e1000e ». La deuxième est une interface de type Wi-Fi, nommée « wlan0 » et gérée par le pilote « iwl3945 ».

```
# lshw -class network
  *-network
       description: Ethernet interface
       product: 82562GT 10/100 Network Connection
       vendor: Intel Corporation
       physical id: 19
       bus info: pci@0000:00:19.0
       logical name: eth0
       version: 03
       serial: 00:1a:4b:90:df:b0
```

```
       size: 100MB/s
       capacity: 100MB/s
       width: 32 bits
       clock: 33MHz
       capabilities: pm msi bus_master cap_list ethernet physical tp 10bt
10bt-fd 100bt 100bt-fd autonegotiation
       configuration: autonegotiation=on broadcast=yes driver=e1000e
driverversion=1.0.2-k2 duplex=full firmware=1.1-2 ip=172.16.1.1 latency=0
link=yes multicast=yes port=twisted pair speed=100MB/s
       resources: irq:28 memory:dc500000-dc51ffff memory:dc520000-dc520fff
ioport:5000(size=32)

  *-network
       description: Wireless interface
       product: PRO/Wireless 3945ABG [Golan] Network Connection
       vendor: Intel Corporation
       physical id: 0
       bus info: pci@0000:10:00.0
       logical name: wlan0
       version: 02
       serial: 00:1c:bf:76:e9:2b
       width: 32 bits
       clock: 33MHz
       capabilities: pm msi pciexpress bus_master cap_list ethernet
                     physical wireless
       configuration: broadcast=yes driver=iwl3945 latency=0 multicast=yes
                      wireless=IEEE 802.11abg
       resources:    irq:30 memory:dc000000-dc000fff
```

1.2. Pilotes et noms des interfaces réseaux

Pour les noyaux Linux modulaires, les pilotes des interfaces réseaux se présentent sous la forme de modules noyau. Lors de démarrage du système, le noyau charge ces modules et attribue un nom à chaque interface. Avec les systèmes antérieurs à `udev` le nom d'une interface est attribué comme un alias au nom du pilote dans le fichier `/etc/modprobe.conf`.

EXEMPLE

D'après le fichier `/etc/modprobe.conf` suivant, le nom « eth0 » est attribué à l'interface réseau gérée par le pilote « e1000e » et le nom « wlan0 » à celle gérée par le pilote « iwl3945 ».

```
# cat /etc/modprobe.conf
alias eth0  e1000e
alias wlan0 iwl3945
```

Pour les systèmes se basant sur le gestionnaire de périphériques udev, le nom d'une interface est fixé par le paramètre NAME de l'interface ayant l'adresse MAC correspondante dans le fichier des règles udev.

EXEMPLE

Le fichier des règles udev suivant attribue le nom « eth0 » à l'interface réseau d'adresse MAC « 00:1a:4b:90:df:b0 » et le nom « wlan0 » à l'interface réseau d'adresse MAC « 00:1c:bf:76:e9:2b ».

```
# cat /etc/udev/rules.d/70-persistent-net.rules
  # This file maintains persistent names for network interfaces.
  # PCI device 0x8086:0x10c4 (e1000e)
  SUBSYSTEM=="net", ACTION=="add", DRIVERS=="?*",
  ATTR{address}=="00:1a:4b:90:df:b0", ATTR{dev_id}=="0x0",
  ATTR{type}=="1", KERNEL=="eth*", NAME="eth0"

  # PCI device 0x8086:0x4222 (iwl3945)
  SUBSYSTEM=="net", ACTION=="add", DRIVERS=="?*",
  ATTR{address}=="00:1c:bf:76:e9:2b", ATTR{dev_id}=="0x0",
  ATTR{type}=="1",   KERNEL=="wlan*", NAME="wlan0"
```

1.3. Paramétrage des interfaces Ethernet

L'utilitaire ethtool affiche et modifie les paramètres des interfaces Ethernet tels que auto-negotiation, port speed, duplex mode et Wake-on-LAN.

EXEMPLE

La ligne de commande suivante change le mode de fonctionnement de l'interface « eth0 » à « full duplex », fixe le débit à 100 Mb/s, active la négociation automatique et active le démarrage de l'hôte à la réception du paquet magique (wol g : *Wake-on-LAN* = *Wake on Magic Packet*).

```
# ethtool -s eth0 speed 100 duplex full autoneg on wol g
# ethtool eth0
Settings for eth0:
   Supported ports: [ TP ]
   Supported link modes:   10baseT/Half 10baseT/Full
                           100baseT/Half 100baseT/Full
   Supports auto-negotiation: Yes
   Advertised link modes:  100baseT/Full
   Advertised pause frame use: No
   Advertised auto-negotiation: Yes
   Link partner advertised link modes:  Not reported
   Link partner advertised pause frame use: No
```

```
Link partner advertised auto-negotiation: No
Speed: 100Mb/s
Duplex: Full
Port: Twisted Pair
PHYAD: 1
Transceiver: internal
Auto-negotiation: on
MDI-X: off
Supports Wake-on: pumbag
Wake-on: g
Current message level: 0x00000001 (1)
Link detected: yes
```

Les changements des paramètres des interfaces Ethernet par la commande `ethtool` sont temporaires et seront perdus au prochain redémarrage du système. Afin de les rendre permanents, il faut spécifier ces paramètres dans le fichier de configuration de l'interface. Pour les distributions *Debian* et dérivées, la commande `ethtool` est spécifiée dans l'instruction `pre-up` du fichier de configuration `/etc/network/interfaces` et pour les distributions *Red Hat* et dérivées les paramètres sont attribués à la variable `ETHTOOL_OPTS` dans le fichier de configuration de l'interface correspondante (`/etc/sysconfig/network-scripts/ifcfg-eth0` pour l'interface eth0).

EXEMPLE

Cet exemple illustre le paramétrage permanent de l'interface « eth0 » de l'exemple précédent pour les deux types de distributions.

Pour les distributions Debian et dérivées :

```
Debian-like# cat /etc/network/interfaces
auto eth0
...
pre-up /usr/sbin/ethtool -s eth0 speed 100 duplex full autoneg off wol g
```

et pour les distributions *R*ed Hat et dérivées :

```
RedHat-like# cat /etc/sysconfig/network-scripts/ifcfg-eth0
...
ETHTOOL_OPTS="speed 100 duplex full autoneg off wol g "
```

1.4. Paramétrage des interfaces sans fil

1.4.1. Généralités

La norme la plus utilisée pour les réseaux sans fil est la norme IEEE 802.11 connue, aussi, sous le nom de Wi-Fi. Avant d'aborder les commandes de paramétrage des interfaces

sans fil, il est nécessaire de rappeler les termes fréquents dans ce contexte tels que : « point d'accès », « service set id » et « clé de cryptage ».

Point d'accès

Un point d'accès (*AP : Access Point*) est un équipement jouant le rôle de concentrateur ou commutateur pour les communications sans fil. En général, un réseau local sans fil (*WLAN : Wireless LAN*) est construit par des hôtes équipés d'interfaces sans fil et interconnectés entre eux par un point d'accès formant ainsi un réseau local sans fil. Ce mode de mise en réseau est nommé **mode infrastructure**.

Il est possible que les hôtes communiquent entre eux sans utiliser de point d'accès. Dans ce cas de figure le mode de mise en réseau est nommé « Ad-Hoc ».

Identifiant réseau

Les réseaux sans fil 802.11 a/b partagent la même plage de fréquence permettant ainsi l'écoute du trafic des réseaux voisins. Pour ignorer le trafic inutile, un identifiant réseau (*ESSID : Extended Service Set ID*) est défini. L'identifiant réseau doit être fixé dans la configuration des interfaces réseaux des hôtes ainsi qu'au niveau du point d'accès.

Cryptage

Pour éviter l'écoute du trafic non autorisé, les données transmises dans un réseau sans fil sont cryptées. Le même mécanismes de cryptage doit être utilisé par toutes les entités communicantes (hôtes et point d'accès).

Le mécanisme de cryptage le plus utilisé est WEP (*Wired Equivalent Privacy*). Mais une faille de sécurité a été découverte et des utilitaires tels que « WEP Crack » et « aircrack-ng » sont disponibles et permettent de déchiffrer le cryptage WEP en quelques minutes.

Pour remédier à la défaillance du cryptage WEP, le mécanisme de cryptage WPA (*Wi-Fi Protected Access*) a été défini. Ce mécanisme propose deux modes :

- le « mode personnel » ou PSK (Pre Shared Key) qui utilise une clé de cryptage configurée manuellement et
- le « mode entreprise » qui combine, en général, un mécanisme de cryptage et un mécanisme d'authentification.

1.4.2. Paquetage wireless-tools

Le paquetage `wireless-tools` contient les commandes de gestion des interfaces sans fil au niveau liaison de données qui sont `iwconfig`, `iwlist`, `iwevent`, `iwgetid`, `iwpriv` et `iwspy`. Les commandes `iwconfig` et `iwlist` sont les deux commandes les plus pertinentes de ce paquetage.

1.4.3. Commande iwconfig

La commande `iwconfig` est l'outil principal de paramétrage d'une interface sans fil. Elle permet de changer tous les éléments d'une configuration tels que le ESSID, le canal, la fréquence et les clés.

SYNTAXE

La commande `iwconfig`, exécutée sans paramètre, affiche les paramètres des interfaces sans fil ainsi que des statistiques sur la liaison. La syntaxe pour modifier un paramètre est la suivante :

```
iwconfig interface paramètre valeur
```

Les paramètres les plus utilisés sont :

- `essid` : spécifie l'identifiant du réseau ;
- `mode` : spécifie le mode de fonctionnement de l'interface qui dépend de la topologie du réseau. La valeur `Managed` désigne le mode infrastructure, la valeur `Ad-Hoc` désigne le mode Ad-Hoc, etc. ;
- `ap` : force la carte à s'associer à un point d'accès donné ;
- `key/enc` : spécifie la clé de cryptage.

EXEMPLES

Dans le premier exemple, la commande `iwconfig`, exécutée sans paramètre, affiche les valeurs des paramètres des interfaces sans fil.

```
$ iwconfig
lo      no wireless extensions.
eth0    no wireless extensions.
wlan0   IEEE 802.11abg  ESSID:"Hedi-HOME"
        Mode:Managed  Frequency:2.412 GHz  Access Point: 00:90:D0:E0:6B:5A
        Bit Rate=54 Mb/s   Tx-Power=15 dBm
        Retry  long limit:7   RTS thr:off   Fragment thr:off
        Power Management:off
        Link Quality=42/70  Signal level=-68 dBm
        Rx invalid nwid:0  Rx invalid crypt:0  Rx invalid frag:0
        Tx excessive retries:0  Invalid misc:0   Missed beacon:0
```

Le deuxième exemple configure l'interface « wlan0 » afin de joindre le réseau sans fil en mode infrastructure (`managed`) d'identifiant « Hedi-Work » avec la clé de cryptage WEP « 1234567890 ».

```
# iwconfig wlan0 mode Managed
# iwconfig wlan0 essid Hedi-Work
# iwconfig wlan0 key 1234567890
```

Pour que cette configuration soit permanente, ces paramètres doivent être spécifiés et enregistrés dans les fichiers de configuration correspondants. Si la distribution est de type *Debian* ou dérivées, le fichier `/etc/network/interfaces` doit contenir ce qui suit :

```
auto wlan0
```

```
...
wireless-mode  Managed
wireless-essid Hedi-Work
wireless-key   1234567890
```

Si la distribution est de type *Red Hat* ou dérivées le fichier `/etc/sysconfig/network-scripts/ifcfg-wlan0` doit contenir ce qui suit :

```
...
TYPE=Wireless
MODE=Managed
ESSID=Hedi-Work
KEY=1234567890
```

1.4.4. Commande iwlist

La commande `iwlist` liste les canaux, les fréquences, les débits et d'autres informations disponibles pour une interface sans fil.

SYNTAXE

La syntaxe de la commande `iwlist` est :

```
iwlist [interface] type
```

Si l'interface n'est pas spécifiée alors les informations disponibles pour toutes les interfaces sans fil sont listées.

Le `type` correspond au type d'information à lister. Les valeurs possibles sont :

- scan ou scanning : affiche les paramètres des points d'accès et des cellules Ad-Hoc à la portée de l'interface ;
- freq, frequency ou channel : liste les fréquences et le nombre des canaux disponibles au niveau de l'interface ;
- rate, bit ou bitrate : liste les débits supportés par l'interface ;
- keys, enc ou encryption : affiche les tailles des clés de cryptage supportées et liste toutes les clés installées dans l'interface ;
- etc.

EXEMPLE

La commande `iwlist` suivante détecte un seul point d'accès à la portée de l'interface « wlan0 ». Elle liste toutes les informations de ce point d'accès telles que l'adresse MAC, l'identifiant réseau, le mode, etc.

```
# iwlist wlan0 scan
wlan0     Scan completed :
```

```
          Cell 01 - Address: 00:90:D0:E0:6B:5A
                    Channel:1
                    Frequency:2.412 GHz (Channel 1)
                    Quality=43/70  Signal level=-67 dBm
                    Encryption key:on
                    ESSID:"Hedi-Home"
                    Bit Rates:1 Mb/s; 2 Mb/s; 5.5 Mb/s; 11 Mb/s; 18 Mb/s
                              24 Mb/s; 36 Mb/s; 54 Mb/s
                    Bit Rates:6 Mb/s; 9 Mb/s; 12 Mb/s; 48 Mb/s
                    Mode:Master
                    ...
#
```

1.5. Paramétrage des interfaces point-à-point

Le protocole PPP est utilisé pour établir des liens internet au dessus d'une connexion RTC par modem, connexion DSL, etc.

Le démon `pppd` fonctionne avec le pilote `ppp` du noyau pour établir et maintenir une liaison ppp avec un hôte nommé « peer » et pour négocier les adresses IP pour chaque extrémité du lien. Il peut aussi gérer la procédure d'authentification avec l'hôte *peer*.

Les principaux fichiers de configuration du démon `pppd` sont :

- /etc/ppp/options : contient les paramètres généraux d'exécution de pppd,
- /etc/ppp/pap-secret : contient les données d'authentification du protocole PAP (risque de sécurité) et
- /etc/ppp/`chap-secret` : contient les données d'authentification du protocole CHAP (plus sécurisé).

Pour simplifier le paramétrage d'une interface ppp, des utilitaires de génération de fichiers de configuration sont utilisés : utilitaires `pppconfig` et `wvdialconf` pour les connexions PPP et `pppoeconf` pour les liaisons PPPoE (*PPP over Ethernet*).

Les fichiers de configuration de ces utilitaires sont :

- /etc/ppp/peers/<fournisseur-accès-Internet> : fichier de configuration spécifique à <fournisseur-accès-Internet> créé par pppconfig pour le démon pppd;
- /etc/chatscripts/<fournisseur-accès-Internet> : fichier de configuration spécifique à <fournisseur-accès-Internet> créé par pppconfig pour le protocole chat ;
- /etc/ppp/peers/wvdial : fichier de configuration spécifique à wvdial créé par wvdialconf pour le démon pppd ;
- /etc/wvdial.conf : fichier de configuration créé par wvdialconf ;
- /etc/ppp/peers/`<fournisseur-accès-adsl>` : fichier de configuration spécifique à PPPoE créé par `pppoeconf` pour le démon `pppd`

PROCEDURE

Cette procédure crée une configuration pour une connexion RTC nommée FAI en utilisant l'utilitaire `pppconfig` qui génère les fichiers de paramétrage de la liaison PPP correspondante. Ensuite elle démarre le démon `pppd` avec cette configuration.

– Lancement de l'utilitaire `pppconfig` et paramétrage de la connexion :

```
# pppconfig
  "Utilitaire de configuration de PPP pour GNU/Linux"

                    Menu principal

  Cet outil est destiné à la configuration de PPP. Il n'ouvrira
  aucune connexion chez votre fournisseur d'accès (FAI) mais
  configure simplement PPP pour que vous puissiez ensuite établir
  la connexion. Plusieurs informations vous seront demandées :
  identifiant, mot de passe et numéro de téléphone de votre FAI.
  Ces informations sont fournies par votre FAI. Si celui-ci utilise
  PAP ou CHAP, elles seront suffisantes. ...

                    Create Créer une connexion
                    Change Modifier une connexion
                    Delete Supprimer une connexion
                    Quit   Sortir de ce programme

                          <Ok>
```

– Validation et sauvegarde des paramètres :

```
                  "Réglages pour FAI"

  Veuillez choisir le réglage à modifier. Choisissez « Annuler » pour
  revenir au menu principal

    Number   21673123456        Numéro de téléphone
    User     hedi               Identifiant chez le FAI
    Password magroun            Mot de passe chez le FAI
    Speed    115200             Vitesse du port
    Com      /dev/ttyS1         Port de communication du modem
    Method   CHAP               Méthode d'authentification

    Advanced Options avancées

    Finished Sauvegarder la configuration et revenir au menu principal
```

```
    Previous Retour au menu précédent
    Quit     Sortir de ce programme

                    <Ok>                    <Annuler>
```

- Visualisation du fichier de configuration `/etc/ppp/peers/FAI` généré par l'utilitaire `pppconfig` :

```
# cat /etc/ppp/peers/FAI
### This optionfile was generated by pppconfig 2.3.18.
hide-password
noauth
connect "/usr/sbin/chat -v -f /etc/chatscripts/FAI"
debug
/dev/ttyS1
115200
defaultroute
noipdefault
user "hedi"
remotename FAI
ipparam FAI
```

- Visualisation du fichier de configuration `/etc/chatscripts/FAI` généré par l'utilitaire `pppconfig` :

```
# cat /etc/chatscripts/FAI
### This chatfile was generated by pppconfig 2.3.18.
### Please do not delete any of the comments.  Pppconfig needs them.
#
# ispauth chat
# abortstring
ABORT BUSY ABORT 'NO CARRIER' ABORT VOICE ABORT 'NO DIALTONE' ABORT 'NO
DIAL TONE' ABORT 'NO ANSWER' ABORT DELAYED
# modeminit
'' ATZ
# ispnumber
OK-AT-OK "ATDT21673123456"
# ispconnect
CONNECT ''
# prelogin

# ispname
ogin: "hedi"
# isppassword
```

```
ssword: "\qhedi mot de passe"
# postlogin
'' \d\c
# end of pppconfig stuff
```

– Lancement du démon `pppd` avec la configuration FAI :

```
# pppd call FAI
```

2. Configuration IP

La configuration réseau consiste à :

– attribuer, pour chaque interface réseau, une adresse IP et un masque de sous-réseau ;

– définir la passerelle par défaut ;

– configurer le service de résolution de noms d'hôtes.

2.1. Commandes de configuration IP

Le paquetage `net-tools` inclut les utilitaires de contrôle du sous-système réseau : `arp`, `ifconfig`, `route`, `netstat`, etc. Les commandes de configuration réseau `ifconfig` et `route` de ce paquetage n'exploitent pas les nouvelles fonctionnalités avancées du noyau Linux. C'est dans ce cadre que le paquetage `iproute` (nommé aussi `iproute2`) a été développé. Les commandes `ip` et `tc` sont les commandes les plus importantes de ce paquetage. La plupart des distributions Linux installent, par défaut, ces deux paquetages.

2.1.1 Attribution d'adresse IP

Les commandes `ifconfig` et `ip` (avec le paramètre `addr`) sont utilisées pour la configuration des interfaces réseaux.

SYNTAXES

Les syntaxes usuelles des commandes `ifconfig` et `ip` pour l'attribution d'adresse IPv4 à une interface réseau sont :

```
ifconfig interface adresse-IPv4 \
         [netmask masque] [broadcast adresse-diffusion]
ip addr action adresse_IPv4/masque [broadcast adresse-diffusion] \
         dev interface
```

et celles pour l'attribution d'adresse IPv6 sont :

```
ifconfig interface inet6 action adresse-IPv6/masque
ip -6 addr action adresse_IPv6/masque dev interface
```

avec :

- interface : désigne l'interface réseau à configurer ;
- adresse-IPv4 : correspond à l'adresse IP version 4 à attribuer ;
- adresse-IPv6 : correspond à l'adresse IP version 6 à attribuer ;
- masque : représente le masque de sous réseau ;
- adresse-diffusion : représente l'adresse de diffusion correspondante ;
- action : représente l'action à effectuer telle que `add` ou `del`.

EXEMPLES

Les lignes de commandes suivantes sont toutes équivalentes et elles configurent l'interface « eth0 » avec l'adresse IP « 192.168.1.1 », le masque « 255.255.255.0 » (le masque par défaut des adresses IP de classe C) et l'adresse de diffusion « 192.168.1.255 ».

```
# ifconfig eth0 192.168.1.1 netmask 255.255.255.0 broadcast 192.168.1.255
# ifconfig eth0 192.168.1.1
# ip addr add 192.168.1.1/24 broadcast 192.168.1.255 dev eth0
# ip addr add 192.168.1.1/24 dev eth0
```

Le deuxième exemple configure l'interface « eth1 » avec l'adresse « 192.168.2.1 » et un masque de sous-réseau égal à « 255.255.255.128 » (ou « /25 » en notation décimale). Toutes les commandes sont équivalentes.

```
# ifconfig eth1 192.168.2.1 netmask 255.255.255.128 \
          broadcast 192.168.2.127
# ifconfig eth1 192.168.2.1 netmask 255.255.255.128
# ip addr add 192.168.2.1/25 broadcast 192.168.2.127 dev eth1
```

Le troisième exemple configure l'interface « eth0 » avec l'adresse IPv6 « 2011:ABCD::1 » et le masque « /64 ». Les deux commandes sont équivalentes.

```
# ifconfig eth0 inet6 add 2011:ABCD::1/64
# ip -6 addr add 2011:ABCD::1/64 dev eth0
```

NOTE

Dans certaines circonstances, il est nécessaire d'attribuer plusieurs adresses IP à un même hôte. Ceci est possible en utilisant la notion d'alias IP qui consiste à attribuer des adresses IP à des instances d'une interface réseau. Par exemple l'interface « eth0 » désigne l'interface Ethernet principale, « eth0:0 » est un alias de celle-ci, « eth0:1 » est un autre alias, etc. La configuration d'une interface de type alias est effectuée de la même manière que celle de l'interface principale.

2.1.2. Passerelle par défaut

Les commandes `route` et `ip` (avec le paramètre `route`) permettent de gérer la table de routage d'un hôte et en particulier de fixer sa passerelle par défaut.

SYNTAXE

Les syntaxes usuelles des commandes `route` et `ip` pour la gestion de la table de routage d'un hôte sont :

```
route action type destination [netmask masque] gw passerelle \
      dev interface
ip route action destination[/masque] via passerelle dev interface
```

avec :

- action : représente l'action à effectuer telle que add ou del pour l'ajout ou la suppression d'une route ;
- type : correspond à -net si la destination est un réseau et -host si la destination est un hôte ;
- destination : correspond à l'adresse de la destination. La valeur default correspond à la route par défaut ;
- masque : correspond au masque de sous réseau de la destination ;
- interface : correspond à l'interface de sortie des paquets envoyés vers la destination.

NOTES

- L'exécution de la commande route sans paramètre permet d'afficher la table de routage et l'option -n affiche les adresses numériques plutôt que des noms de domaines.
- Les routes vers les réseaux adjacents (réseaux auxquels sont connectés les interfaces de l'hôte) sont ajoutées automatiquement.

EXEMPLE

Les lignes de commande suivantes sont équivalentes et permettent de fixer à « 172.16.0.254 » l'adresse de la passerelle par défaut, qui est joignable via l'interface « eth0 ».

```
# route add -net default gw 172.16.0.254 dev eth0
# ip route add default via 172.16.0.254 dev eth0
```

La séquence suivante affiche la table de routage avec les deux commandes `route` et `ip route`.

```
# route -n
Table de routage IP du noyau
Destination   Passerelle      Genmask         Indic Metric Ref    Use Iface
172.16.0.0    0.0.0.0         255.255.0.0     U     1      0        0 eth0
```

```
0.0.0.0         172.16.1.254    0.0.0.0         UG    0      0        0 eth0
# ip route
172.16.0.0/16 dev eth0  proto kernel  scope link  src 172.16.1.1  metric 1
default via 172.16.1.254 dev eth0
```

2.2. Fichiers de configuration réseau

Les commandes `ifconfig`, `ip` et `route` manipulent les paramètres réseau du noyau. Ceci implique que la configuration réseau, fixée par ces commandes, n'est valable que pour la session système en cours. Pour avoir une configuration réseau permanente, les paramètres réseaux doivent être sauvegardés dans des fichiers de configuration dont les noms, les emplacements et les formats diffèrent selon les distributions.

2.2.1. Cas des distributions Debian *et dérivées*

Les distributions *Debian* et dérivées utilisent le fichier `/etc/network/interfaces` pour la configuration permanente des interfaces réseaux. Ce fichier se compose de zéro ou plusieurs entrées `iface`, `mapping`, `auto` et `allow-` :

- auto (ou allow-auto) : active une interface lors du démarrage du système ;
- allow-hotplug : active une interface lorsque le noyau détecte un événement depuis celle ci ;
- mapping : détermine le nom d'interface logique pour une interface physique ;
- iface : permet de fixer les paramètres de configuration IP d'une interface.

Le format de l'entrée `iface` est :

```
iface interface famille méthode
  paramètre1 valeur1
  paramètre2 valeur2
  ...
```

avec :

- `interface` : spécifie le nom de l'interface à configurer ;
- `famille` : spécifie la famille d'adresses utilisée. Les valeurs fréquentes sont `inet` pour la famille d'adresses IPv4 et `inet6` pour la famille d'adresses IPv6 ;
- `méthode` : spécifie la méthode utilisée pour l'attribution d'adresses. Les méthodes les plus utilisées sont :
  - `lo` : c'est la méthode d'attribution d'adresses de boucle locale. Elle n'admet pas de paramètres ;
  - `dhcp` : c'est la méthode d'attribution d'adresses par le service DHCP. Elle admet des paramètres tels que `hwaddress`, `leasetime`, etc. En général, la méthode `dhcp` est utilisée sans paramètres ;

- `static` : c'est la méthode pour attribuer des adresses manuellement. Les paramètres les plus fréquents sont :
  - `address` : spécifie l'adresse IP à affecter ;
  - `netmask` : spécifie le masque de sous réseau ;
  - `broadcast` : spécifie l'adresse de diffusion ;
  - `gateway` : spécifie l'adresse de la passerelle par défaut.

EXEMPLE

Le fichier `/etc/network/interfaces` suivant illustre la configuration réseau d'un hôte ayant deux interfaces Ethernet : « eth0 » configurée statiquement et « eth1 » configurée dynamiquement par le le service DHCP. Il illustre aussi les lignes correspondantes à l'interface de boucle locale (*loopback*) « lo ».

```
# cat /etc/network/interfaces
### The loopback network interface
auto lo
iface lo inet loopback

### The first network interface eth0
auto eth0
iface eth0 inet static
          address 192.168.1.45
          netmask 255.255.255.0
          gateway 192.168.1.1

### The second network interface eth1
auto eth1
iface eth1 inet dhcp
```

NOTE

À chaque modification du fichier de configuration des interfaces, le *script* de contrôle du service réseau est relancé pour recharger les paramètres dans le noyau. Les lignes de commandes suivantes sont équivalentes et assurent le redémarrage du service réseau :

```
# /etc/init.d/networking restart
# invoke-rc.d networking restart
```

2.2.2. Cas des distributions Red Hat *et dérivées*

Pour une configuration réseau permanente, Les distributions *Red Hat* et dérivées utilisent un fichier de configuration pour chaque interface. Ce fichier a pour nom `ifcfg-interface` et se situe sous le répertoire `/etc/sysconfig/network-scripts/`. Par exemple, le fichier `/etc/sysconfig/network-scripts/ifcfg-eth0` est le fichier de configuration de l'interface

« eth0 ».

Chaque ligne des fichiers de configuration est sous la forme : « `paramètre=valeur` ». Il existe des paramètres communs à tout type d'interface (Ethernet, PPP, etc.) et d'autres spécifiques à un type donné. Ce qui suit est la description des paramètres fréquents dans le cas de la configuration d'une interface Ethernet :

- `DEVICE` : correspond au nom de l'interface réseau ;
- `BOOTPROTO` : correspond au protocole utilisé pour la configuration de l'interface. Les valeurs possibles sont :
- `none` : indique qu'aucun protocole ne devrait être utilisé,
- `bootp` : indique que le protocole BOOTP devrait être utilisé et
- `dhcp` : indique que le protocole DHCP devrait être utilisé ;
- `ONBOOT` : correspond à l'état de l'interface indiquant si celle-ci devrait être activée ou non lors du démarrage du système. Les valeurs possibles sont `yes` ou `no` ;
- `IPADDR` : correspond à l'adresse IP de l'interface ;
- `NETMASK` : correspond au masque du sous réseau ;
- `BROADCAST` : correspond à l'adresse de diffusion. Cette directive a été abandonnée car la valeur est calculée automatiquement avec `ifcalc` ;
- `GATEWAY` : correspond à l'adresse IP de la passerelle réseau ;
- `USERCTL` : correspond à la possibilité de contrôler l'interface par les utilisateurs autres que l'utilisateur `root`. Les valeurs possibles sont `yes` et `no`.

EXEMPLES

Le premier exemple illustre une configuration statique de l'interface « eth0 ».

```
# cat /etc/sysconfig/network-scripts/ifcfg-eth0
DEVICE=eth0
BOOTPROTO=none
ONBOOT=yes
IPADDR=192.16.1.1
NETMASK=255.255.255.0
BROADCAST=192.168.1.255
USERCTL=no
```

Le deuxième exemple illustre une configuration dynamique de l'interface « eth1 ».

```
# cat /etc/sysconfig/network-scripts/ifcfg-eth1
DEVICE=eth1
BOOTPROTO=dhcp
ONBOOT=yes
```

NOTE

À chaque modification des fichiers de configuration des interfaces, le *script* de contrôle du service réseau est relancé pour recharger les paramètres dans le noyau. Les lignes de commandes suivantes sont équivalentes et assurent le redémarrage du service réseau :

```
# /etc/rc.d/init.d/network restart
# service network restart
```

2.3. Résolution de noms d'hôtes

Les deux méthodes les plus utilisées pour la résolution des noms d'hôtes sont la méthode `files` et la méthode `dns`.

La méthode `files` utilise le fichier `/etc/hosts` pour associer les adresses IP avec les noms d'hôtes. Chaque ligne de ce fichier correspond à une seule adresse IP et elle respecte le format suivant :

```
Adresse_IP nom_canonique [alias ...]
```

Cette première méthode n'est plus exploitée que dans le cas d'un réseau de quelques hôtes et le fichier `/etc/hosts` est réduit, en général, au contenu suivant :

```
127.0.0.1 localhost.localdomain localhost
127.0.0.1 nom_hôte.nom_domaine nom_hôte
```

où `nom_hôte` correspond au nom de l'hôte défini dans le fichier `/etc/hostname` et `nom_domaine` est le nom du domaine pleinement qualifié (FQDN : *Fully qualified domain name*) de ce dernier.

La deuxième méthode consiste à demander au client DNS (*resolver*) de résoudre le nom d'hôte par l'envoi de requêtes aux serveur DNS précisés dans le fichier de configuration `/etc/resolv.conf`. Le format général de ce fichier est comme suit :

```
domain domaine_local
search domaines_de_recherche
nameserver serveur_DNS
```

avec :

- `domain` : spécifie le domaine de l'hôte ;
- `search` : spécifie la liste des domaines de recherche ;
- `nameserver` : spécifie l'adresse IP d'un serveur DNS.

NOTES

- Les directives `domain` et `search` sont utilisées pour former les noms complets (FQDN) dans le cas où des noms raccourcis sont recherchés.

– La directive `nameserver` peut être utilisée plusieurs fois pour préciser le serveur DNS secondaire, tertiaire, etc.

Le service NSS (*Name Service Switch*) détermine l'ordre d'application des méthodes de résolution des noms d'hôtes. Ceci est spécifié par l'entrée `hosts` de son fichier de configuration `/etc/nsswitch.conf`.

EXEMPLE

La configuration suivante précise que la méthode `files` est appelée en premier pour résoudre un nom d'hôte. S'il est trouvé dans le fichier `/etc/hosts`, elle retourne l'adresse IP correspondante. Sinon la méthode `dns` est appelée.

```
$ grep hosts /etc/nsswitch.conf
hosts:          files dns
```

3. Configurations IP avancées

3.1. Configuration multiréseau

Un hôte multiréseau (*multihomed host*) est un hôte équipé de plusieurs interfaces réseaux connectées chacune à un réseau différent. Cette configuration est utilisée dans le cas où un hôte Linux joue le rôle d'un routeur, d'une passerelle réseau ou d'un pare-feu.

Par défaut, la fonction de routage du noyau Linux est désactivée. C'est à dire que les paquets IP reçus dont l'adresse IP de destination correspond à un autre hôte sont rejetés. Dans le cas d'un hôte multiréseau, la fonction de routage doit être activée. Ceci est assuré par l'affectation de la valeur 1 au contenu du fichier `/proc/sys/net/ipv4/ip_forward` qui peut s'effectuer comme suit :

```
# echo "1" > /proc/sys/net/ipv4/ip_forward
```

ou

```
# sysctl -w net.ipv4.ip_forward=1
```

Le fichier `/proc/sys/net/ipv4/ip_forward` correspond au paramètre « fonction de routage » de la session en cours du système. Les deux valeurs possibles sont :

– 0 : fonction de routage désactivée et

– 1 : fonction de routage activée.

Pour rendre l'activation de la fonction de routage permanente (même en cas de redémarrage du système), le paramètre `net.ipv4.ip_forward` du fichier de configuration `/etc/sysctl.conf` doit être fixé manuellement à 1. La vérification de l'activation permanente de la fonction de routage peut être effectuée comme suit :

```
# grep ip_forward /etc/sysctl.conf
net.ipv4.ip_forward=1
```

EXEMPLE

L'exemple suivant configure (pour la session du système en cours) un hôte multiréseau connecté à un premier réseau local contenant les hôtes de type poste de travail et à un deuxième réseau contenant les hôtes de type serveurs ainsi que le routeur assurant la connexion vers Internet. Il est connecté au premier réseau d'adresse IP « 172.16.0.0/16 » à travers « eth0 » d'adresse « 172.16.255.254 » et au deuxième réseau d'adresse IP « 192.168.1.0/24 » à travers « eth1 » d'adresse « 192.168.1.253 ». Le routeur assurant la connexion vers Internet et jouant le rôle de passerelle par défaut de l'hôte multiréseau est d'adresse « 192.168.1.254 ».

```
# ifconfig eth0 172.16.255.254 netmask 255.255.0.0
# ifconfig eth1 192.168.1.253 netmask 255.255.255.0
# route add -net default gw 192.168.1.254 dev eth1
# route -n
Table de routage IP du noyau
Destination  Passerelle     Genmask         Indic Metric Ref    Use Iface
172.16.0.0   0.0.0.0        255.255.0.0     U     0      0        0 eth0
192.168.1.0  0.0.0.0        255.255.255.0   U     0      0        0 eth1
0.0.0.0      192.168.1.254  0.0.0.0         UG    0      0        0 eth1
# sysctl -w net.ipv4.ip_forward=1
```

3.2. OpenVPN

Un VPN (*Virtual Private Network*) est un réseau privé virtuel. Il est constitué d'un ensemble de réseaux locaux interconnectés par un réseau public à travers des canaux sécurisés appelés tunnels VPN. Les protocoles IPSec, L2TP, OpenVPN et PPP+SSH sont des exemples de protocoles pour la mise en place de VPN.

OpenVPN permet d'établir des tunnels de niveau applicatif en se basant sur le protocole TCP ou UDP. Le chiffrement est assuré par SSL/TLS (*Secure Sockets Layer / Transport Layer Security*) et l'authentification est réalisée par une clé secrète, un certificat x509 ou un couple (nom, mot de passe).

OpenVPN est aussi le nom du logiciel libre qui implémente le protocole OpenVPN. Il fonctionne selon le modèle client/serveur.

PROCEDURE

La procédure suivante permet de mettre en place un tunnel VPN entre deux hôtes : « poste1 » (d'adresse « 172.16.1.1 ») et « poste2 » (d'adresse « 172.16.1.2 ») avec OpenVPN. L'hôte « poste1 » joue le rôle de serveur et « poste2 » celui de client.

– Installation de `openvpn` sur les deux hôtes :

```
poste[12] # apt-get install openvpn
```

– Génération et partage d'une clé secrète :

```
poste1    # openvpn --genkey --secret /etc/openvpn/secret.key
poste1    # scp /etc/openvpn/secret.key poste2:/etc/openvpn/
...
secret.key
100%  636     0.6KB/s   00:00
```

– Configuration de la liaison sur « poste1 » (serveur). Le paramètre `ifconfig` est suivi de l'adresse IP associée à la carte virtuelle locale « tun0 » suivie de l'adresse IP de la carte distante :

```
poste1    # nano /etc/openvpn/openvpn.cfg
dev tun
ifconfig 192.168.1.1  192.168.1.2
secret /etc/openvpn/secret.key
```

– Configuration de la liaison sur « poste2 » (client) :

```
poste2    # nano /etc/openvpn/openvpn.cfg
remote poste1
dev tun
ifconfig 192.168.1.2 192.168.1.1
secret /etc/openvpn/secret.key
```

– Activation de la liaison sur « poste1 » (serveur) :

```
poste1   # openvpn --config /etc/openvpn/openvpn.cfg --verb 1
... OpenVPN 2.1.0 i486-pc-linux-gnu [SSL] ...
... IMPORTANT: OpenVPN's default port number is now 1194, ...
... /usr/sbin/openvpn-vulnkey -q /etc/openvpn/secret.key ...
... TUN/TAP device tun0 opened
... /sbin/ifconfig tun0 192.168.1.1 pointopoint 192.168.1.2 mtu 1500
... UDPv4 link local (bound): [undef]
... UDPv4 link remote: [undef]
```

– Activation de la liaison sur « poste2 » (client) :

```
poste2   # openvpn --config /etc/openvpn/openvpn.cfg --verb 1
... OpenVPN 2.1.0 i486-pc-linux-gnu [SSL] ...
... IMPORTANT: OpenVPN's default port number is now 1194, ...
... /usr/sbin/openvpn-vulnkey -q /etc/openvpn/secret.key ...
```

```
... TUN/TAP device tun0 opened
... /sbin/ifconfig tun0 192.168.1.2 pointopoint 192.168.1.1 mtu 1500
... UDPv4 link local (bound): [undef]
... UDPv4 link remote: [AF_INET]172.16.1.1:1194
---> après quellesque secondes :
... Peer Connection Initiated with [AF_INET]172.16.1.1:1194
... Initialization Sequence Completed
```

– Vérification des interfaces et des routes :

```
poste2   # ifconfig tun0
tun0      Link encap:UNSPEC HWaddr 00-00-00-00-00-00-00-00-00-00-00-00-00-00-
00-00-00
          inet adr:192.168.1.2  P-t-P:192.168.1.1  Masque:255.255.255.255
          UP POINTOPOINT RUNNING NOARP MULTICAST  MTU:1500  Metric:1
          ...
poste2   # route -n
Table de routage IP du noyau
Destination   Passerelle      Genmask         Indic Metric Ref    Use Iface
192.168.1.1   0.0.0.0         255.255.255.255 UH    0      0        0 tun0
...

poste1   # ifconfig tun0
tun0      Link encap:UNSPEC  HWaddr 00-00-00-00-00-00-00-00-00-00-00-00-00-00-
00-00-00
          inet adr:192.168.1.1  P-t-P:192.168.1.2  Masque:255.255.255.255
          UP POINTOPOINT RUNNING NOARP MULTICAST  MTU:1500  Metric:1
          ...
poste1   # route -n
Table de routage IP du noyau
Destination   Passerelle      Genmask         Indic Metric Ref    Use Iface
192.168.1.2   0.0.0.0         255.255.255.255 UH    0      0        0 tun0
...
```

– Test de la liaison :

```
poste2   # ping 192.168.1.1
PING 192.168.1.1 (192.168.1.1) 56(84) bytes of data.
64 bytes from 192.168.1.1: icmp_seq=1 ttl=64 time=4.84 ms
64 bytes from 192.168.1.1: icmp_seq=2 ttl=64 time=0.000 ms
...
```

4. Diagnostic réseau

On est souvent confronté à des problèmes réseaux tels que le ralentissement du trafic, la déconnexion d'un hôte ou d'un réseau, la non disponibilité d'un service réseau, etc. Pour identifier ces problèmes, les opérations de diagnostic suivantes peuvent être effectuées :

- l'utilisation des commandes de paramétrage des interfaces et de configuration réseau précédemment traitées ;
- la vérification des fichiers de configuration ;
- le suivi et l'analyse des journaux (les fichiers log) ;
- l'utilisation d'outils dédiés aux tests et aux diagnostics réseau dont les plus utilisés seront traités dans cette section.

4.1. Test de la connectivité réseau avec ping

La commande `ping` est l'utilitaire de base pour tester la connectivité réseau entre deux hôtes. Elle se base sur les messages « *echo-request* » et « *echo-reply* » du protocole ICMP. Si les résultats de la commande `ping` comporte une réponse positive, alors l'hôte local et l'hôte distant sont correctement configurés ainsi que le chemin réseau entre les deux. Dans le cas où toutes les réponses sont négatives, les déductions suivantes sont possibles :

- la configuration de l'hôte local ou/et de l'hôte distant n'est pas correcte ;
- l'hôte distant est non disponible (éteint par exemple) ;
- il existe une erreur dans la configuration de l'un des routeurs du chemin réseau entre les deux hôtes ;
- l'hôte distant est configuré afin de ne pas répondre aux requêtes ping pour des raisons de sécurité ;
- un routeur ou un pare-feu intermédiaire bloque le trafic ICMP.

En plus du test de connectivité, la commande `ping` affiche des informations utiles sur l'état du réseau telles que le taux de perte, le temps de transit, etc.

Par défaut, la commande `ping` de Linux envoie une demande en continu, toutes les secondes, jusqu'à son interruption avec la combinaison de touches `ctrl C`.

SYNTAXE

```
ping [option …] hôte
```

OPTIONS

Les options les plus utilisées sont :

- `-b` : autorise l'utilisation d'adresses de diffusion ;
- `-c nb` : envoie seulement `nb` paquets ;

- `-i intervalle` : spécifie le délai d'attente entre les envois des paquets. Par défaut il est fixé à une seconde ;
- `-f` : fonctionne en mode *flood*. Si le délai d'attente entre les paquets n'est pas spécifié, les paquets sont envoyés aussi rapidement que possible (délai d'attente = 0). Ce mode est réservé à l'utilisateur `root` ;
- `-q` : mode silencieux. Seules les lignes résumés au début et à la fin de l'exécution de la commande sont affichées ;
- `-s taille` : spécifie, en octets, la taille des données à envoyer. Par défaut c'est 56 octets ;
- `-t ttl` : spécifie la valeur TTL du paquet IP ;
- `-w délai` : spécifie, en secondes, le délai de fonctionnement de la commande `ping` indépendamment du nombre de paquets envoyés ou reçus.

EXEMPLES

Le premier exemple teste la connectivité d'un hôte non présent dans le réseau. Dans cet exemple, la commande `ping` envoie des messages en continu jusqu'à la réception de la combinaison de touches `Ctrl C`.

```
$ ping  172.16.100.250
PING 172.16.100.250 (172.16.100.250) 56(84) bytes of data.
From 172.16.100.200 icmp_seq=1 Destination Host Unreachable
From 172.16.100.200 icmp_seq=2 Destination Host Unreachable
From 172.16.100.200 icmp_seq=3 Destination Host Unreachable
^C
--- 172.16.100.250 ping statistics ---
5 packets transmitted, 0 received, +3 errors, 100% packet loss, ...
```

Le deuxième exemple teste la connectivité de la passerelle. C'est le premier test à faire si la connexion vers l'extérieur ne peut pas s'établir. Dans cet exemple deux messages seulement sont envoyés.

```
$ ping -c 2 172.16.100.254
PING 172.16.100.254 (172.16.100.254) 56(84) bytes of data.
64 bytes from 172.16.100.254: icmp_seq=1 ttl=64 time=0.565 ms
64 bytes from 172.16.100.254: icmp_seq=2 ttl=64 time=0.539 ms

--- 172.16.100.254 ping statistics ---
2 packets transmitted, 2 received, 0% packet loss, time 999ms
rtt min/avg/max/mdev = 0.539/0.552/0.565/0.013 ms
```

Le troisième exemple teste la connectivité de l'hôte distant « www.auf.org ». Dans ce cas, les statistiques sont plus significatives. La commande `ping` est lancée en mode silencieux et sa durée d'exécution est limitée à 5 secondes.

```
$ ping -w 5 -q www.auf.org
PING www.auf.org (199.84.140.19) 56(84) bytes of data.

--- www.auf.org ping statistics ---
5 packets transmitted, 5 received, 0% packet loss, time 4006ms
rtt min/avg/max/mdev = 139.675/152.673/189.220/18.675 ms
```

4.2. Test de la résolution de noms avec nslookup et dig

Les commandes `nslookup` et `dig` sont utilisées pour diagnostiquer la résolution de noms avec la méthode DNS. Leur utilisation est détaillée dans le chapitre 7 « Service DNS ».

EXEMPLE

La commande `dig` suivante interroge le serveur DNS « 172.16.100.254 » (serveur DNS défini dans le fichier `/etc/resolv.conf`) pour la résolution du nom de domaine « www.auf.org ».

```
$ cat /etc/resolv.conf
nameserver 172.16.100.254
$ dig www.auf.org
...
;; ANSWER SECTION:
www.auf.org.        83431   IN  A   199.84.140.19
;; AUTHORITY SECTION:
auf.org.        83431   IN  NS  ns1.refer.org.
auf.org.        83431   IN  NS  ns1.ca.auf.org.
...
;; Query time: 22 msec
;; SERVER: 172.16.100.254#53(172.16.100.254)
...
```

4.3. Test du chemin réseau avec traceroute

La commande `traceroute` affiche le chemin réseau emprunté pour atteindre un hôte distant. Elle envoie des paquets avec des durées de vie (*TTL : Time To Live*) incrémentales et exploite les messages ICMP « *time exceeded* » générés par les routeurs lorsque la valeur TTL du paquet IP devient égale à 0. La valeur TTL d'un paquet est décrémentée à chaque routeur traversé.

La liste des routeurs intermédiaires est affichée avec les temps d'aller-retour qui peuvent donner une idée de la fluidité du trafic dans les différentes portions du chemin.

Les cas où la commande `traceroute` échoue et n'atteint pas la destination peuvent être dus à l'une des raisons suivantes :

- les paquets traceroute sont bloqués ou rejetés par l'un des routeurs intermédiaires ;
- l'hôte distant n'est pas présent dans le réseau ;
- le réseau où réside l'hôte distant n'est pas joignable à cause d'une erreur dans les tables de routage de l'ensemble des routeurs.

EXEMPLE

L'exemple suivant trace la route vers l'hôte « www.google.com ». Le caractère « * » indique qu'aucune réponse n'est reçue pendant 5 secondes (c'est la valeur par défaut du temps d'attente). Le premier nœud du chemin est toujours la passerelle par défaut de l'hôte source.

```
$ traceroute -n www.google.com
traceroute to www.google.com (209.85.148.147), 30 hops max, 60 byte packets
 1  172.16.100.254  1.139 ms  1.093 ms  1.062 ms
 2  * * *
 3  * * *
 4  * * *
 5  * * *
 6  * 196.203.78.17  21.172 ms  25.331 ms
 7  * 196.203.78.50  24.322 ms  40.963 ms
 8  193.95.96.181  23.310 ms  25.252 ms  22.648 ms
 9  193.95.96.65  24.879 ms  21.768 ms  26.217 ms
10  * 193.95.0.50  24.602 ms  21.560 ms
11  72.14.214.141  51.802 ms  49.963 ms  48.962 ms
12  216.239.43.156  50.471 ms  47.080 ms  50.038 ms
13  209.85.253.8 53.090 ms 209.85.253.10  49.857 ms 209.85.253.8  54.086 ms
14  72.14.232.78  60.386 ms  58.346 ms  58.372 ms
15  72.14.239.63  59.318 ms  65.849 ms 72.14.236.21  66.354 ms
16  209.85.254.41  60.772 ms 209.85.254.57  69.770 ms  71.274 ms
17  209.85.148.147  58.484 ms  60.837 ms  59.805 ms
$
```

4.4. Test de la connectivité applicative avec telnet

La commande `telnet` peut être utilisée pour tester la connectivité réseau d'une application serveur. Par défaut elle se connecte sur le port TCP / 23 (port de l'application serveur telnet), mais il est possible de spécifier un autre port pour interroger une autre application réseau selon la syntaxe suivante :

```
telnet hôte port
```

où `hôte` est le nom ou l'adresse IP de l'hôte distant et `port` le numéro de port sur lequel l'application serveur écoute les demandes de connexion.

Pour isoler ou localiser le problème de connectivité d'une application réseau, la démarche suivante peut être appliquée :

- tester la connectivité à partir de l'hôte local en utilisant l'adresse de l'interface de boucle locale puis l'adresse de l'interface physique. Ceci permet d'éliminer l'influence du pare-feu protégeant l'hôte lui-même ;
- Vérifier les règles de filtrage Netfilter avec la commande iptables -L et les listes de contrôle d'accès (ACL : Access Control List) des fichiers /etc/hosts.allow et /etc/hosts.deny si l'exécution de l'application utilise la technique TCP Wrapper.
- tester la connectivité à partir d'un hôte appartenant au même réseau que l'hôte en question. Ceci permet d'éliminer l'influence du pare-feu protégeant le réseau ;
- tester la connectivité à partir d'un hôte distant n'appartenant pas au même réseau que l'hôte hébergeant l'application.

EXEMPLES

Le premier exemple illustre une connexion réussie avec un serveur web (port 80) s'exécutant sur l'hôte d'adresse IP « 192.168.1.1 ». L'interruption de la connexion est effectuée par la combinaison de touches `ctrl ]`.

```
$ telnet 172.16.100.12 80
Trying 172.16.100.12...
Connected to 172.16.100.12.
Escape character is '^]'.
^]

telnet> quit
Connection closed.
$
```

Le deuxième exemple illustre une connexion refusée (*connection refused*) dont la cause peut être :

- l'application n'est pas démarrée ou
- un pare-feu bloque et rejette la connexion.

```
$ telnet 172.16.100.254 80
Trying 172.16.100.254...
telnet: Unable to connect to remote host: Connection refused
$
```

Le troisième exemple illustre une connexion qui ne peut s'établir suite au dépassement du délai d'attente (*connection timed out*). La cause peut être :

- l'hôte n'est pas présent dans le réseau ;
- un pare-feu bloque et ne rejette pas la connexion.

```
$ telnet 125.2.1.68 80
Trying 125.2.1.68...
telnet: Unable to connect to remote host: Connection timed out
$
```

4.5. Test de la connectivité applicative avec nc

La commande nc permet d'ouvrir des connexions TCP, d'envoyer des datagrammes UDP, d'écouter (en tant que serveur) sur des ports TCP ou UDP et de scanner les ports réseaux. Elle est très utile pour établir des connexions réseaux en mode client/serveur pour des fins de diagnostic réseau.

SYNTAXE

La syntaxe générale de la commande nc est la suivante :

```
nc [option ...] [hôte] [port]
```

OPTIONS

Les options les plus usuelles de la commande nc sont :

- -l : spécifie que la commande nc doit écouter les connexions entrantes au lieu d'initialiser une connexion vers un hôte distant ;
- -p : spécifie le port source à utiliser ;
- -u : utilise le protocole UDP à la place du protocole TCP (par défaut) ;
- -z : spécifie que nc doit scanner les ports en état d'écoute.

EXEMPLE

La commande nc suivante établit une connexion réseau avec le serveur Web de l'hôte « monServeur », avec 1234 comme port source. Dans la séquence, le serveur Web répond au message GET par l'envoi de la page Web par défaut contenant des informations utiles.

```
$ nc  -p 1234 monServeur 80
GET  <--- à saisir
<!DOCTYPE HTML PUBLIC "-//IETF//DTD HTML 2.0//EN">
<html><head> ...</head><body> ...
<address>Apache/2.2.9 (Debian) PHP/5.2.6-1+lenny13 with Suhosin-Patch
mod_scgi/1.12 mod_ssl/2.2.9 OpenSSL/0.9.8g mod_wsgi/2.5 Python/2.5.2 Server
at monServeur Port 80</address>
</body></html>
$
```

4.6. Diagnostic réseau avec netstat

La commande `netstat` est un utilitaire très utilisé dans les diagnostics réseaux. Elle fournit des informations sur les connexions réseaux, les tables de routage et un certain nombre de statistiques de l'hôte sur lequel elle s'exécute. Parmi les utilisations fréquentes de la commande `netstat` :

- la détermination des services réseaux actifs et
- la liste de toutes les connexions réseaux disponibles.

SYNTAXE

La syntaxe de la commande `netstat` est :

```
netstat [option ...]
```

OPTIONS

Sans option, la commande `netstat` liste les *sockets* (fichiers réseaux) ouvertes.

Les options les plus utiles sont :

- -a : affiche toutes les sockets, y compris les sockets en état d'écoute ;
- -l : affiche seulement les sockets en état d'écoute ;
- -t : se limite aux connexions TCP ;
- -u : se limite aux connexions UDP ;
- -i : affiche des statistiques par interface ;
- -s : affiche des statistiques par protocole ;
- -p : affiche le PID et le nom des processus des sockets listées. Cette option est utilisée avec le privilège root ;
- -r : affiche la table de routage ;
- -n : affiche les adresses en format numérique au lieu d'essayer de déterminer le nom symbolique d'hôte, de port ou d'utilisateur.

EXEMPLES

Le premier exemple liste les *sockets* TCP en état d'écoute ainsi que les programmes associés.

```
# netstat -ltp
Active Internet connections (only servers)
Proto Recv-Q Send-Q Local Add.      Foreign Add. State   PID/Program name
tcp        0      0 localhost:mysql *:*          LISTEN  939/mysqld
tcp        0      0 *:www           *:*          LISTEN  9439/apache2
tcp        0      0 *:ssh           *:*          LISTEN  906/sshd
tcp        0      0 *:https         *:*          LISTEN  9439/apache2
...
```

Le deuxième exemple permet de calculer le nombre de connexions en état `ESTABLISHED` ou `TIME_WAIT`.

```
# netstat -at | egrep 'ESTABLISHED|TIME_WAIT' | wc -l
13
```

4.7. Diagnostic réseau avec lsof

La commande `lsof` liste des informations à propos des fichiers ouverts par les processus. Parmi ces fichiers on trouve les fichiers réseaux (*sockets*) tels que les fichiers réseaux Internet, les fichiers NFS et les fichiers réseaux du domaine UNIX. L'option `-i` limite la liste aux *sockets* Internet.

SYNTAXE

La syntaxe générale pour lister les *sockets* Internet est la suivante :

```
lsof -i [46][protocole][@hôte][:port]
```

avec :

- 46 : spécifie la version du protocole IP: 4 pour IPv4 et 6 pour IPv6 ;
- protocole: désigne le nom du protocole TCP ou UDP ;
- hôte : spécifie l'adresse IP ou le nom d'un hôte ;
- port : spécifie le nom, le numéro ou une liste de ports.

EXEMPLE

La commande suivante liste les *sockets* Internet en se limitant aux connexions TCP utilisant un port de numéro compris entre 1 et 1024.

```
# lsof -i TCP:1-1024
COMMAND PID USER FD TYPE DEVICE SIZE/OFF NODE NAME
sshd 906 root 3u IPv4 4024 0t0 TCP *:ssh (LISTEN)
sshd 906 root 4u IPv6 4026 0t0 TCP *:ssh (LISTEN)
master 1083 root 12u IPv4 4707 0t0 TCP localhost:smtp (LISTEN)
sshd 3292 root 3r IPv4 778665 0t0 TCP 172.16.100.12:ssh-
>172.16.100.200:42791 (ESTABLISHED)
apache2 3400 root 3u IPv4 616612 0t0 TCP *:www (LISTEN)
apache2 3400 root 4u IPv4 616614 0t0 TCP *:https (LISTEN)
apache2 9439 www-data 3u IPv4 616612 0t0 TCP *:www (LISTEN)
apache2 9439 www-data 4u IPv4 616614 0t0 TCP *:https (LISTEN)
...
#
```

4.8. Scanner de ports nmap

La commande `nmap` est un utilitaire de type scanner de ports réseaux. Elle permet de chercher les ports en état d'écoute, de déterminer les services hébergés et de collecter des informations sur le système d'exploitation d'un hôte distant. Ce type d'utilitaire est utilisé, principalement, pour détecter les vulnérabilités dans un réseau telles que l'exécution de services non autorisés.

SYNTAXE

La syntaxe de la commande `nmap` est :

```
nmap [option ...] [cible]
```

où `cible` est l'hôte, le réseau ou l'ensemble des hôtes à scanner.

OPTIONS

Ci-dessus quelques options utilisées pour scanner les hôtes présents et les ports ouverts.

- `-sP` : scanne les hôtes présents dans le réseau ;
- `-sS`, `-sT`, `-sA`, `-sW`, ou `-sM` : scanne les ports TCP en état d'écoute de différentes méthodes ;
- `-sU` : scanne les ports UDP en état d'écoute ;
- `-p port` : spécifie le port ou la plage des ports à scanner ;

EXEMPLES

Dans le premier exemple, la commande `nmap` scanne les hôtes présents dans le réseau d'adresse IP « 172.16.100.0/24 ».

```
$ nmap -sP 172.16.100.0/24
Starting Nmap 5.00 ( http://nmap.org ) at 2011-11-17 11:02 CET
Host 172.16.100.1 is up (0.0022s latency).
Host 172.16.100.5 is up (0.0020s latency).
Host 172.16.100.6 is up (0.0020s latency).
Host 172.16.100.12 is up (0.0010s latency).
Host 172.16.100.200 is up (0.00016s latency).
Host 172.16.100.204 is up (0.0015s latency).
Host 172.16.100.254 is up (0.0011s latency).
Nmap done: 256 IP addresses (7 hosts up) scanned in 2.56 seconds
```

Dans le deuxième exemple, la commande `nmap` scanne les ports de l'hôte local et détermine les services actifs.

```
$ nmap -p 1-1024 -sT 127.0.0.1
...
Interesting ports on localhost (127.0.0.1):
```

```
Not shown: 1017 closed ports
PORT    STATE SERVICE
21/tcp  open  ftp
22/tcp  open  ssh
25/tcp  open  smtp
80/tcp  open  http
110/tcp open  pop3
143/tcp open  imap
389/tcp open  ldap
Nmap done: 1 IP address (1 host up) scanned in 0.11 seconds
```

4.9. Analyse du trafic avec tcpdump

La commande `tcpdump` est l'un des utilitaires les plus utilisés pour la capture et l'analyse des paquets réseaux. Elle permet d'afficher le trafic réseau sur la sortie standard ou de l'enregistrer dans un fichier, sous une forme binaire, pour une analyse ultérieure. Elle utilise, comme la majorité des analyseurs sous Linux, la librairie `libpacap`.

SYNTAXE

La syntaxe générale de la commande `tcpdump` est la suivante :

```
tcpdump [option …] [expression]
```

EXPRESSION

Les expressions de `tcpdump` sont utilisées pour filtrer le trafic à analyser. Par défaut (sans expression) tout le trafic est analysé. La syntaxe des expressions est celle de `pcap-filter` qui consiste en une ou plusieurs primitives. Chaque primitive est composée d'un identifiant (nom ou numéro) précédé par un ou plusieurs qualificateurs. Il existe trois sortes de qualificateurs :

- type : spécifie le type d'identifiant. Les valeurs possibles sont host (valeur par défaut), net et port ;
- dir : spécifie la direction du paquet. Parmi les valeurs possibles : src, dst, src or dst (valeur par défaut) et src and dst ;
- proto : spécifie le protocole à traiter. Parmi les valeurs possibles : `ether`, `wlan`, `ip`, `ip6`, `arp`, `rarp`, `tcp` et `udp`. La valeur par défaut correspond à tous les protocoles compatibles avec le `type`.

Des expressions complexes peuvent être construites en utilisant les opérateurs logiques `not` (ou « `!` »), `and` (ou « `&&` ») et `or` (ou « `||` ») ainsi que les parenthèses.

Une expression peut être une comparaison (utilisation des opérateurs : <, >, <=, >=, = et !=) entre des expressions arithmétiques utilisant les opérateurs binaires : +, –, *, /, &, |, << et >>.

OPTIONS

Les options les plus utilisées sont :

- `-i interface` : spécifie l'interface de capture ;
- `-w fichier` : enregistre les données de la capture dans un fichier ;
- `-r fichier` : analyse les données à partir d'un fichier ;
- `-n` : ne convertit pas les adresses en nom ;
- `-c nb` : capture nb paquets et s'arrête ;
- `-p` : ne met pas l'interface réseau dans le mode « promiscuous » où l'interface récupère toutes les trames même celles qui ne lui sont pas destinées ;
- `-t` : n'affiche pas l'horodatage au début de chaque ligne.

EXEMPLES

Dans le premier exemple, la commande `tcpdump` intercepte les trames à partir de l'interface « eth0 » et filtre les paquets à afficher pour ne laisser que ceux dont le port source ou destination est 53. La commande se limite à analyser deux paquets.

```
# tcpdump -i eth0 -t -n -c2 port 53
tcpdump: verbose output suppressed, use -v or -vv for full protocol decode
listening on eth0, link-type EN10MB (Ethernet), capture size 96 bytes
IP 172.16.100.200.44396 > 172.16.100.254.53: 28303+ A? www.auf.org. (29)
IP 172.16.100.254.53 > 172.16.100.200.44396: 28303 1/0/0 A 199.84.140.19
(45)
2 packets captured
2 packets received by filter
0 packets dropped by kernel
#
```

Dans le deuxième exemple, la commande `tcpdump` enregistre les paquets capturés dans le fichier `auf.tcpdump` puis visualise ce fichier binaire sur l'écran.

```
# tcpdump -w auf.tcpdump host www.auf.org and \( port 80 or port 443 \)
tcpdump: listening on eth0, link-type EN10MB (Ethernet), capture size 96
bytes
^C
2999 packets captured
3005 packets received by filter
0 packets dropped by kernel

# tcpdump -r auf.tcpdump
17:26:54.337581 IP hedi-laptop.local.39499 > www.auf.org.www: Flags [S],
seq 2913060691, win 5840, options [mss 1460,sackOK,TS val 4509747 ecr
```

```
0,nop,wscale 6], length 0
...
```

4.10. Analyse du trafic avec wireshark

Wireshark est un analyseur de protocole réseau avec une interface graphique. Il permet d'afficher les données à partir du trafic réseau ou à partir des données précédemment sauvegardées dans un fichier. Wireshark utilise la même syntaxe des expressions filtres que la commande tcpdump et peut lire ou importer le format de la même commande ainsi que plusieurs autres formats.

NOTE

La commande tshark est la version textuelle de Wireshark conçue pour la capture et l'affichage des paquets lorsqu'une interface utilisateur graphique n'est pas nécessaire ou n'est pas disponible. Elle supporte les mêmes options que wireshark.

5. Notification des utilisateurs

Cette section traite de la notification des utilisateurs à propos des interventions de maintenance et des pannes systèmes à travers les messages de connexion et les commandes de notification.

Les messages de connexion sont un moyen de notification pour les utilisateurs et ne sont visibles qu'avec les sessions textuelles. Ces messages sont enregistrés dans les trois fichiers suivants :

- /etc/issue pour les messages des connexions locales ;
- /etc/issue.net pour les messages des connexions distantes ;
- /etc/motd pour les messages du jour (*motd : message of th day*).

Le contenu des fichiers /etc/issue ou /etc/issue.net est affiché avant l'invite de connexion, celui du fichier /etc/motd est affiché après la connexion et avant le lancement du shell.

Les fichiers /etc/issue et /etc/issue.net supportent les séquences de contrôle des commandes de type getty telles que « \n » pour le nom de l'hôte, « \l » pour le terminal de connexion, etc.

EXEMPLE

L'exemple suivant illustre l'exploitation des fichiers /etc/issue et /etc/motd pour l'affichage des messages de notification.

```
Debian Squeeze 6.0.2 Serveur1 tty1

Serveur1 login : hedi
```

```
password :

A NOTER!
Tous les systèmes seront arrêtés ce weekend pour des opérations de
maintenance.
Sauvegardez vos travaux!

$ cat /etc/issue
Debian Squeeze 6.0.2 \n \l

$ cat /etc/motd

A NOTER!
Tous les systèmes seront arrêtés ce weekend pour des opérations de
maintenance.
Sauvegardez vos travaux!
$
```

NOTE

Il existe d'autres commandes de notification des utilisateurs telles que `wall` et `shutdown`. La commande `wall` (*write all*) permet à l'utilisateur `root` d'afficher les données du fichier donné comme paramètre ou de l'entrée standard sur tous les terminaux des utilisateurs connectés. La commande `shutdown` notifie les utilisateurs connectés lors de l'arrêt du système.

6. Exercices

1. **Quelles sont les trois commandes utilisées pour déterminer les ports TCP ouverts de l'hôte local ?**

   - ❑ A. `lsof`
   - ❑ B. `netstat`
   - ❑ C. `nmap`
   - ❑ D. `tcpdump`
   - ❑ E. `tcpd`

2. **Un hôte d'adresse IP 172.16.1.22 vient d'être connecté au réseau dont la connexion vers Internet est assurée par un routeur ayant 172.16.1.254 comme adresse IP. Quelle est la commande permettant au nouvel hôte de se connecter à Internet ?**

- ❑ A. `route add -net 172.16.1.0 netmask 255.255.255.0 gw 172.16.1.254`
- ❑ B. `route add -net 172.16.0.0 netmask 255.255.0.0 gw 172.16.1.254`
- ❑ C. `route add -net 0.0.0.0 netmask 0.0.0.0 gw 172.16.1.254`
- ❑ D. `route add -host 172.16.1.254 netmask 255.255.255.255 gw 172.16.1.22`
- ❑ E. `route add 172.16.0.0 gw 172.16.1.254`

3. Quels sont les fichiers en relation avec la résolution des noms ? Trois propositions sont correctes.

- ❑ A. `/etc/hosts`
- ❑ B. `/etc/hosts.allow`
- ❑ C. `/etc/hosts.deny`
- ❑ D. `/etc/hosts.conf`
- ❑ E. `/etc/resolv.conf`
- ❑ F. `/etc/nsswitch.conf`

4. Quelles sont les deux commandes qui ont généré le trafic capturé par la commande tcpdump suivante ?

```
# tcpdump -nli eth0
...
... IP 192.168.1.66 > 192.168.1.254: ICMP echo request, id 24091, ...
... IP 192.168.1.254 > 192.168.1.66: ICMP echo reply, id 24091, ...
...
... IP 192.168.1.66.40978 > 192.168.1.254.53: 26331+ A? www.lpi.org. ...
... IP 192.168.1.254.53 > 192.168.1.66.40978: 26331 1/2/0 A 69.90.69.231...
...
#
```

- ❑ A. `traceroute 192.168.1.66`
- ❑ B. `traceroute 192.168.1.254`
- ❑ C. `ping 192.168.1.66`
- ❑ D. `ping 192.168.1.254`
- ❑ E. `nslookup 192.168.1.254`
- ❑ F. `nslookup 192.168.1.66`
- ❑ G. `nslookup 69.90.69.231`
- ❑ H. `nslookup www.lpi.org`

Chapitre 6. Maintenance système

Objectifs	**Installation à partir des codes sources** : – dépaqueter un fichier de code source ; – construire et installer des programmes à partir du code source. **Opérations de sauvegarde** : – sauvegarder les données importantes du système ; – vérifier l'intégrité des fichiers de sauvegarde ; – restaurer les sauvegardes.
Points importants	**Installation à partir des codes sources** : – utilitaires usuels de compression et d'archivage ; – utilisation de `make` pour compiler des programmes ; – paramétrage du script de configuration ; – emplacement des répertoires de code source. **Opérations de sauvegarde** : – supports de sauvegarde ; – utilitaires de sauvegarde ; – solutions de sauvegarde réseau.
Mots clés	Installation à partir des codes sources : `/usr/src/`, `gunzip`, `gzip`, `bzip2`, `tar`, `configure`, `make`, `uname`, `install` Sauvegarde : `cpio`, `dd`, `tar`, `/dev/st*`, `/dev/nst*`, `mt`, `rsync`, `dump`, `restore`

1. Installation à partir des codes sources

1.1 Introduction

Il existe deux méthodes pour installer un logiciel sous Linux :

- à partir des paquetages binaires qui sont déjà compilés et construits par les éditeurs de logiciels ou de distributions ;

- à partir des paquetages comportant les codes sources.

La première méthode est celle qui est préférée mais la deuxième, qui est traitée dans cette section, peut s'avérer indispensable dans des cas particuliers tels que :

- absence de paquetage binaire ;
- besoin d'une version plus récente que les versions binaires disponibles ou
- besoin de refaire la compilation d'un logiciel pour des raisons d'optimisation ou d'adaptation.

Les paquetages comportant les codes sources des logiciels sont distribués sous forme de fichiers archives. Ces fichiers archives contiennent notamment `README` et INSTALL, les fichiers qui décrivent les étapes d'installation à suivre.

En général les étapes d'installation d'un logiciel à partir de ses codes sources sont :

- récupération du fichier archive ;
- dépaquetage de l'archive ;
- préparation ou configuration de la compilation : pré-compilation ;
- compilation ;
- installation des fichiers binaires : post-compilation.

1.2 Récupération des codes sources

L'archive des codes sources d'un programme ou d'un logiciel peut être récupérée à partir du site hébergeant le projet correspondant. Les sites tels que « www.gnu.org » et « sourceforge.net » sont des plates-formes d'hébergement pour de nombreux projets open source. Les grands projets disposent de leurs propres plates-formes d'hébergement, par exemple le projet serveur web Apache est hébergé par le site « www.apache.org ».

EXEMPLE WGET

Les exemples de cette section traitent l'installation pas à pas, à partir de codes sources, de l'utilitaire *GNU Wget*.

La première étape consiste à récupérer la dernière version de l'utilitaire `wget` à partir du site FTP du projet GNU, « http://ftp.gnu.org/gnu/wget/ ». Il s'agit du fichier `wget-1.12.tar.gz`.

1.3 Dépaquetage

Les codes sources d'un programme sont disponibles sous la forme d'une archive `tar` compressée en `gzip` ou en `bzip2`. Pour la manipulation de ces archives les utilitaires `tar`, `gzip`, `gunzip`, `bzip2` et `bunzip2` peuvent être utilisés.

Une archive au format `.tar.gz` peut être décompressée et désarchivée comme suit :

```
$ tar -xzvf application-version.tar.gz
```

ou

```
$ gunzip application-version.tar.gz
$ tar -xvf application-version.tar
```

ou

```
$ gzip -d application-version.tar.gz
$ tar -xvf application-version.tar
```

Une archive au format `.tar.bz2` peut être décompressée et désarchivée comme suit :

```
tar -xjvf application-version.tar.bz2
```

ou

```
bunzip2 application-version.tar.bz2
tar -xvf application-version.tar
```

ou

```
bzip2 -d application-version.tar.bz2
tar -xvf application-version.tar
```

La commande tar est traitée en détail plus loin ans ce chapitre, dans la section « Commande tar ».

EXEMPLE WGET

D'après l'extension du fichier récupéré, il s'agit d'une archive compressée par `gzip` qui peut être décompressé et désarchivée comme suit :

```
$ tar -xzvf wget-1.12.tar.gz
```

L'affichage du contenu du répertoire `wget-1.12` (voir ci-dessous) montre l'existence des fichiers `README` et `INSTALL` ainsi que celle du script de configuration `configure` qui sera utilisé dans l'étape qui suit.

```
$ cd wget-1.12/
$ ls
ABOUT-NLS   ChangeLog.README  COPYING      MAILING-LIST  msdos    util
aclocal.m4  config.log        doc          maint.mk      NEWS     windows
AUTHORS     config.status     GNUmakefile  Makefile      po
autogen.sh  configure         INSTALL      Makefile.am   README
build-aux   configure.ac      lib          Makefile.in   src
ChangeLog   configure.bat     m4           md5           tests
$
```

1.4 Pré-compilation

La première fonction du script de configuration `./configure` est de détecter des informations à propos du système et de configurer le code source en tenant compte de ces informations. La deuxième fonction et de permettre d'activer ou désactiver des fonctionnalités optionnelles du logiciel.

L'option `--help` affiche la syntaxe d'utilisation du script et précise les arguments pour activer ou désactiver les fonctionnalités optionnelles. Si aucun argument n'est précisé le script de configuration prend les valeurs par défaut qui sont, dans la plupart des cas, les plus adéquats.

L'option `--prefix` est commune à tous les scripts de configuration des logiciels. Elle précise le répertoire d'installation. Par défaut, c'est le répertoire `/usr/local`.

EXEMPLE WGET

Ce qui suit est un extrait de l'aide du script de configuration du paquetage GNU Wget qui montre l'utilisation de l'option `--prefix` et liste ses fonctionnalités optionnelles telles que `digest`, `ntlm`, `debug`, `largefile`, `ipv6`, etc.

```
$ ./configure --help
...
--prefix=PREFIX      install architecture-independent files in PREFIX
                     [/usr/local]

By default, `make install' will install all the files in
`/usr/local/bin', `/usr/local/lib' etc.  You can specify
an installation prefix other than `/usr/local' using `--prefix',
for instance `--prefix=$HOME'.
...
Optional Features:
--disable-FEATURE       do not include FEATURE
                        (same as --enable-FEATURE=no)
--enable-FEATURE[=ARG]  include FEATURE [ARG=yes]
--disable-digest        disable support for HTTP digest authorization
--disable-ntlm          disable support for NTLM authorization
--disable-debug         disable support for debugging output
--disable-largefile     omit support for large files
--disable-ipv6          disable IPv6 support
...
$
```

Si on dispose du privilège *root*, on exécute le script `./configure` sans préciser une valeur pour l'option `--prefix`. Par défaut le répertoire d'installation est `/usr/local`.

```
# ./configure
```

Par contre, si on ne dispose pas du privilège *root*, on doit exécuter le script `./configure` en précisant un répertoire d'installation. Dans la plupart des cas c'est le répertoire personnel qui est spécifié comme répertoire d'installation :

```
$ ./configure --prefix=$HOME
```

1.5 Compilation

L'utilitaire `make` est utilisé pour la construction des programmes binaires à partir des fichiers sources. Il utilise le fichier `Makefile` contenant les instructions de compilation à exécuter pour générer les codes binaires. La durée de l'étape de compilation dépend de la taille du programme source à compiler.

L'un des problèmes rencontrés lors de l'installation à partir des sources est l'absence des bibliothèques et des fichiers en-têtes. Si une bibliothèque `biblioX` est manquante, alors il faut installer le paquetage `lib<biblioX>` ainsi que le paquetage `lib<biblioX>-devel` ou `lib<biblioX>-dev` correspondant.

EXEMPLE WGET

La compilation du paquetage Wget est lancée par :

```
$ make
```

Si la compilation rencontre des erreurs, `make` se termine. En général se sont des erreurs d'absence de fichier en-tête ou de bibliothèque. Pour résoudre ces erreurs, il suffit d'installer les paquetages correspondants et de recompiler l'application.

1.6 Post-compilation

Une fois que la compilation s'est déroulée sans erreurs, il faut copier les différents fichiers dans les répertoires adéquats : les programmes binaires sous le répertoire `bin`, les bibliothèques sous le répertoire `lib`, les fichiers de configuration sous le répertoire `/etc`, les pages manuels sous le répertoire `man`, etc. Cette copie peut être automatisée par la commande :

```
# make install
```

Il faut disposer du privilège root pour exécuter cette commande si, dans l'étape de pré-compilation, le script `./configure` est exécuté sans préciser une valeur pour l'option `--prefix`.

Pour nettoyer l'arborescence du répertoire source des fichiers générés par la compilation avant une éventuelle deuxième compilation, la commande suivante est utilisée :

```
$ make clean
```

EXEMPLE WGET

L'installation des différents fichiers, résultats de l'étape de compilation, dans les répertoires correspondants est effectuée par :

```
# make install
```

L'installation place la nouvelle version de `wget` dans le répertoire `/usr/local/bin`. Maintenant deux versions de `wget` sont disponibles.

```
$ whereis wget
wget: /usr/bin/wget /usr/local/bin/wget
$ /usr/bin/wget --version
GNU Wget 1.10 compilé sur linux-gnu.
...
$ /usr/local/bin/wget --version
GNU Wget 1.12 compilé sur linux-gnu.
...
$
```

1.7 Désinstallation

Pour désinstaller le logiciel, le paramètre `uninstall` est utilisé.

```
# make uninstall
```

EXEMPLE WGET

Désinstallation de la version de l'utilitaire `wget` nouvellement installé :

```
# make uninstall
```

Il est possible de vérifier la désinstallation en relançant la commande `whereis` :

```
$ whereis wget
wget: /usr/bin/wget
$
```

2. Sauvegarde

2.1 Concepts

La sauvegarde est une tâche importante. Elle doit être effectuée périodiquement sur des machines comportant des données sensibles.

Les opérations de sauvegarde sont réalisées afin de pouvoir :

- restaurer la totalité d'un système en état de fonctionnement suite à un incident (disque en panne, feu, …) ;
- restaurer une partie du système (un fichier ou un répertoire) suite à une fausse manipulation telle qu'une suppression accidentelle d'un fichier de configuration.

2.1.1 Types de sauvegarde

Les sauvegardes sont de trois types :

- **sauvegarde complète :** consiste à sauvegarder la totalité du système. Elle se traduit par la sauvegarde de tous les fichiers d'un disque ou d'une partition. Elle peut prendre des heures pour s'achever ;
- **sauvegarde différentielle :** consiste à ne sauvegarder que les fichiers modifiés en se référant à une sauvegarde complète précédemment effectuée. Dans ce cas les sauvegardes sont de plus en plus volumineuses, mais une restauration complète ne nécessite que la sauvegarde complète et la dernière sauvegarde différentielle ;
- **sauvegarde incrémentale** : consiste à ne sauvegarder que les fichiers modifiés depuis la dernière sauvegarde. La première sauvegarde incrémentale se réfère à une sauvegarde complète et chaque sauvegarde incrémentale joue le rôle de référence pour la suivante. Dans ce cas les sauvegardes sont de petites tailles et de courtes durées mais la restauration complète nécessite la restauration de toutes les archives.

Dans la pratique, on effectue une sauvegarde complète combinée avec des sauvegardes différentielles ou des sauvegardes incrémentales. Par exemple, on effectue une sauvegarde complète toutes les semaines et des sauvegardes incrémentales (ou différentielles) quotidiennes.

2.1.2 Politique de sauvegarde

Chaque système a besoin d'une politique de sauvegarde qui constitue une composante importante du plan de reprise d'activités en cas de sinistre. La politique de sauvegarde la plus simple doit pouvoir répondre aux questions suivantes :

- Qu'est-ce qui est sauvegardé ?

On doit sauvegarder tous les fichiers qui changent après l'installation du système et qui contiennent des informations dont on a besoin. Parmi les répertoires les plus critiques on cite `/etc`, `/home`, `/var`.

- Quelle est la fréquence de sauvegarde ?

La fréquence de sauvegarde doit être spécifiée pour chaque type de sauvegarde. Le coût est un critère de choix important. Il faut évaluer d'un côté le coût de la réalisation d'une sauvegarde, de l'autre le coût du travail qui risque d'être perdu si la sauvegarde n'est pas faite, et trouver le meilleur compromis.

- Quel est le type de sauvegarde ?

En général on effectue une sauvegarde complète hebdomadaire et des sauvegardes

incrémentales ou différentielles quotidiennes.

- Quel est le support de sauvegarde ?

Il existe une multitude de supports de sauvegarde tels que le disque dur, le CD-ROM ou DVD-ROM, la clé USB, la sauvegarde réseau, la sauvegarde sur Internet et la bande magnétique. Les facteurs déterminants pour le choix du média sont : le **coût**, la **capacité** et la **fiabilité**.

Le lieu de stockage des supports doit également être pris en compte, il ne faut pas négliger les risques qu'entraîne le fait de stocker tous les supports dans un même emplacement.

NOTES

- Les sauvegardes sont inutiles si on ne peut pas les restaurer.
- Une sauvegarde non testée représente un risque.

2.2 Utilitaires de sauvegarde

2.2.1 Commande mt

La commande `mt` contrôle une bande magnétique.

SYNTAXE

`mt [-f` *`périphérique`*`] `*`opération`*

avec

- *`périphérique`* : le lecteur de la bande, par défaut c'est `/dev/tape`
- parmi les opérations possibles :
  - `rewind` : rembobiner ;
  - `status` : afficher l'état de la bande ;
  - `erase` : effacer la bande ;
  - `offline` : embobiner et éjecter ;
  - `fsf [`*`n`*`]` : avancer de n fichiers ;
  - `bsf [`*`n`*`]` : reculer de n fichiers.

EXEMPLE

Pour archiver le répertoire `/var/www` sur une bande contenant déjà une archive on exécute :

```
# mt rewind
# mt -f /dev/nst0 fsf 1
# tar cf /dev/nst0 /var/www
# mt -f /dev/nst0 offline
```

Pour désarchiver le deuxième fichier de la bande qui contient l'archive du répertoire `/var/www` on exécute :

```
# mt rewind
# mt -f /dev/nst0 fsf 1
# tar xf /dev/nst0
# mt -f /dev/nst0 offline
```

2.2.2 Commande tar

La commande `tar` sauvegarde des fichiers et des répertoires. Elle utilise les options courtes, les options longues et les clés positionnelles. Dans le cas de l'utilisation des clés positionnelles, ces dernières sont regroupées et leurs arguments respectifs sont regroupés et présentés dans le même ordre que les clés.

EXEMPLE

Les trois lignes de commandes suivantes sont équivalentes et illustrent l'utilisation des options courtes, des options longues et des clés positionnelles.

```
# tar -c -N "2011-01-14 00:00" -f home.tar /home
# tar --create --newer "2011-01-14" --file home.tar /home
# tar cNf "2011-01-14" home.tar /home
```

OPTIONS

Les options les plus courantes de la commande `tar` sont :

- `c`, `-c`, `--create` : crée une archive ;
- `x`, `-x`, `--extract` : extrait une archive ;
- `t`, `-t`, `--list` : liste tous les fichiers de l'archive ;
- `f`, `-f`, `--file` *`Fichier`* : précise le fichier archive qui peut être un fichier ordinaire ou un périphérique ;
- `z`, `-z`, `--gzip` : (dé)compacte l'archive `tar` avec `gzip` ;
- `j`, `-j`, `--bzip2` : (dé)compacte l'archive `tar` avec `bzip2` ;
- `M`, `-M`, `--multi-volume` : crée une archive `tar` multivolumes ;
- `v`, `-v`, `--verbose` : active le mode d'affichage détaillé.

EXEMPLES

Le premier exemple illustre les opérations d'archivage et de restauration du répertoire `/home` en utilisant une bande magnétique comme support de stockage.

- Archivage du répertoire `/home` :

```
# tar cvf /dev/st0 /home
```

- Affichage de la liste des fichiers de l'archive :

```
# tar tvf /dev/st0
...
-rw-r--r-- root/root 38031 2011-07-01 17:11 home/hedi/lpi206.odt
...
```

- Restauration de tous les fichiers de l'archive

```
# cd /
# tar xvf /dev/st0
```

- Restauration seulement du fichier `lpi206.odt` du répertoire personnel de l'utilisateur hedi

```
# cd /
# tar xvf /dev/st0 home/hedi/lpi206.odt
```

Le deuxième exemple montre comment archiver les fichiers du répertoire `/home` dont la date de création ou de dernière modification est plus récente que 14 janvier 2011:

```
# tar cvNf "2011-01-14" home.tar /home
```

2.2.3 Commande cpio

La commande `cpio` sauvegarde sur la sortie standard les fichiers dont les noms sont fournis par l'entrée standard. Elle est toujours utilisée avec les mécanismes de redirection et de tube (*pipe*). Elle peut gérer les sauvegardes réparties sur plusieurs volumes.

OPTIONS

Les options les plus courantes de la commande `cpio` sont:

- `-i` : extrait les fichiers de l'archive ;
- `-o` : crée une archive ;
- `-t` : affiche une table du contenu de l'entrée ;
- `-B` : utilise une taille de bloc d'entrée/sortie de 5120 octets ;
- `-c` : utilise un vieux format d'archive portable ;
- `-v` : active le mode d'affichage détaillé.

EXEMPLES

Le premier exemple sauvegarde et extrait les fichiers du répertoire `/etc` en utilisant le fichier `etc.cpio` comme fichier d'archive.

- Sauvegarde du répertoire `/etc` dans le fichier `etc.cpio` :

```
# find /etc -print | cpio -ov > etc.cpio
```

- Visualisation de la liste des fichiers de l'archive `etc.cpio` :

```
# cpio -itv < etc.cpio
```

- Extraction de tous les fichiers de l'archive :

```
# cpio -iv < etc.cpio
```

- Extraction des fichiers `/etc/apache2/*` de l'archive :

```
# cpio iv '/etc/apache2/*' < etc.cpio
```

Le deuxième exemple sauvegarde les fichiers du répertoire personnel de l'utilisateur hedi dont la date de dernière modification est plus récente que celle du fichier `lpi206.odt`

```
# find /home/hedi -type f -newer lpi206.odt | cpio -ov > hedi.cpio
```

2.2.4 Utilitaire rsync

`rsync` (*remote synchronization*) est un utilitaire de synchronisation de fichiers entre l'hôte local et un hôte distant. Il utilise le protocole de mise à jour à distance rsync pour accélérer le transfert de fichiers. Il est fréquemment utilisé pour mettre en place des solutions de sauvegarde distante.

Il y a deux manières pour `rsync` de communiquer avec l'hôte distant :

- en utilisant un shell distant comme transport (tel que `ssh` ou `rsh`). Le transport par shell distant est utilisé à chaque fois que le chemin source ou destination contient un séparateur «:» après la spécification de l'hôte ;
- en contactant un démon `rsync` directement par TCP. Ceci est utilisé lorsque le chemin source ou destination contient un séparateur «::» après la spécification de l'hôte ou lorsqu'une URL « rsync:// » est spécifiée.

OPTIONS

Les options les plus courantes de la commande `rsync` sont :

- `-v` : active le mode d'affichage détaillé ;
- `-a` : active le mode archive, identique à `-rlptgoD` ;
- `-z` : compresse les données à envoyer ;
- `-e commande` : spécifie un shell distant ;
- `-n` : simule le fonctionnement de la commande ;
- `--delete` : supprime les fichiers qui n'existent pas sur l'hôte émetteur.

EXEMPLES

Les exemples suivants présentent des utilisations pratiques de l'utilitaire `rsync`.

- Synchronisation du répertoire local `/local/monsite` avec le répertoire `/var/www` de l'hôte « monserveur » en utilisant le shell distant `ssh` :

```
$ rsync -a -v -e "ssh -l hedi" --delete monserveur:/var/www/ /local/monsite
```

- Synchronisation du répertoire `/var/www` du hôte « monserveur » avec le répertoire local `/local/monsite` en utilisant le shell distant `ssh`:

```
$ rsync -a -v -e "ssh -l hedi" --delete /local/monsite/ monserveur:/var/www
```

- Synchronisation du répertoire local `/var/www` avec un serveur `rsync` distant

```
$ rsync -a -v --delete unserveur::www /var/www
```

2.2.5 Commandes dump et restore

La commande `dump` sauvegarde un système de fichiers de type Ext2/Ext3. La commande `dump` supporte les sauvegardes incrémentales, elle utilise pour cela la notion de niveau d'archive. Le niveau 0 est réservé pour une sauvegarde complète. Un niveau de sauvegarde supérieur à 0 correspond à une sauvegarde incrémentale relative à la dernière sauvegarde de niveau inférieur. Autrement dit, une archive de niveau n (n>0) sauvegarde tous les changements du système de fichiers depuis la dernière archive de niveau m, m étant le plus grand niveau inférieur à n.

La séquence 0 1 2 3 4 5 6 correspond à une sauvegarde complète suivie de sauvegardes incrémentales. Pour restaurer le système de fichiers il faut commencer par la sauvegarde complète (niveau 0) puis traiter les sauvegardes incrémentales dans l'ordre. Par contre la séquence 0 6 6 6 6 6 6 correspond à une sauvegarde complète suivie de sauvegardes incrémentales relatives à la sauvegarde complète (des sauvegardes différentielles). Pour restaurer le système de fichiers il faut restaurer la sauvegarde complète et seulement la dernière incrémentale.

`dump` utilise le fichier `/var/lib/dumpdates` (ou `/etc/dumpdates`) pour enregistrer les informations sur les archives effectuées : le système de fichiers, le niveau et la date.

La commande `dump` supporte les options courtes, les options longues et les clés positionnelles.

OPTIONS

Les options les plus courantes de la commande `dump` sont :

- `-`*`Niveau`* : spécifie le niveau de l'archive. La valeur 0 précise une sauvegarde complète, une valeur différente de 0 précise une sauvegarde incrémentale ;
- `-u` : mise à jour du fichier `dumpdates` ;
- `-f` *`fichier`* **:** spécifie le fichier archive. Par défaut c'est `/dev/st0`. La valeur « - » est utilisée pour la sortie standard ;
- `-L étiquette` : ajoute une étiquette dans l'en-tête de l'archive ;

– `-A fichier` : crée un fichier contenant la liste des fichiers de l'archive.

La commande `restore` extrait des fichiers, des répertoires ou la totalité du système de fichiers d'une archive créée par `dump`. Elle possède un mode de fonctionnement interactif permettant de naviguer dans l'arborescence de l'archive et de sélectionner les fichiers à extraire.

OPTIONS

Les options les plus courantes de la commande `restore` sont :

– `-i` : lance la commande en mode interactif ;

– `-f fichier` : spécifie le fichier archive ;

– `-r` : extrait le contenu de toute l'archive dans le répertoire courant ;

– `-t` : liste le contenu du fichier archive.

Les commandes les plus courantes du mode interactif sont :

– `ls` : liste le contenu du répertoire ;

– `cd [répertoire]` : change le répertoire courant de l'archive ;

– `pwd` : affiche le répertoire courant de l'archive ;

– `add [fichier]` : ajoute un fichier à la liste des fichiers à extraire ;

– `delete [fichier]` : supprime un fichier de la liste des fichiers à extraire ;

– `extract` : extrait les fichiers précisés par la commande `add` ;

– `quit` : quitte le mode interactif ;

– `what` : liste les informations d'en-tête de la commande `dump` ;

– `help ou ?` : affiche l'aide des commandes du mode interactif.

EXEMPLES

L'exemple suivant suppose que le répertoire `/var/www` se trouve dans une partition à part. Périodiquement (chaque semaine par exemple) on effectue une sauvegarde complète suivie de plusieurs sauvegardes incrémentales.

– Pour faire la sauvegarde complète du système de fichiers `/var/www` :

```
# dump 0uf /dev/st0 /var/www
```

– Pour faire la première sauvegarde incrémentale :

```
# dump 1uf /dev/st0 /var/www
```

– Pour faire la deuxième sauvegarde incrémentale :

```
# dump 2uf /dev/st0 /var/www
```

et ainsi de suite pour les sauvegardes incrémentales suivantes.

- Pour lister le contenu de la sauvegarde :

```
# restore -tf /dev/st0
```

- Pour restaurer tout le système de fichiers `/var/www`
  - on se déplace vers le répertoire `/var/www` :

```
# cd /var/www
```

  - puis on charge la dernière bande contenant une sauvegarde complète et on exécute :

```
# restore -rf /dev/st0
```

Le fichier `./restoresymtable` est créé. Il contient des informations utiles pour la restauration des sauvegardes incrémentales. Il ne faut supprimer ce fichier qu'après avoir terminé la restauration.

Dans le cas où on a besoin de sauvegarder tout le système, il faut sauvegarder tous les systèmes de fichiers qui le composent. Si par exemple le système à sauvegarder est composé des systèmes de fichiers `/`, `/usr`, `/home` et `/var` alors il faut taper successivement les commandes suivantes :

```
# dump 0uf /dev/st0 /
# dump 0uf /dev/st0 /usr
# dump 0uf /dev/st0 /home
# dump 0uf /dev/st0 /var
```

2.2.6 Commande dd

La commande `dd` permet de faire une copie bloc par bloc. Elle est adaptée pour la duplication des périphériques de stockage tels qu'un disque, une partition, etc. Elle n'est pas adaptée pour sauvegarder un système de fichiers car les données peuvent changer lors de la lecture, ce qui causera une incohérence dans la sauvegarde.

SYNTAXE

`dd if=`*`fichier-entrée`* `of=`*`fichier-sortie`* `bs=`*`taille`* `count=`*`nb`*

avec

- `if` : spécifie le fichier source, par défaut c'est le fichier entrée standard ;
- `of` : spécifie le fichier destination, par défaut c'est le fichier sortie standard ;
- `bs` : spécifie la taille des blocs, par défaut c'est 512 octets ;
- `count` : spécifie le nombre de blocs à copier, par défaut c'est autant de blocs qu'en contient *`fichier-entrée`*.

EXEMPLES

Les exemples suivants présentent des utilisations pratiques de la commande dd.

- Sauvegarde de la deuxième partition du premier disque sur la bande magnétique :

```
# dd if=/dev/sda2 of=/dev/st0
```

- Restauration de la deuxième partition du premier disque à partir de l'archive précédemment sauvegardée sur la bande magnétique :

```
# dd if=/dev/st0 of=/dev/sda2
```

- Sauvegarde du MBR (premier secteur) du premier disque dans le fichier `/tmp/mbr.img` :

```
# dd if=/dev/sda of=/tmp/mbr.img bs=512 count=1
```

- Restauration du MBR du premier disque précédemment sauvegardé dans le fichier `/tmp/mbr.img` :

```
# dd if=/tmp/mbr.img of=/dev/sda bs=512 count=1
```

- Création d'un fichier vide d'une certaine taille, par exemple 100 Mo, afin de l'associer à un périphérique loop

```
# dd if=/dev/zero of=/tmp/disk1.img bs=1M count=100
# losetup /dev/loop0 /tmp/disk1.img
# mkfs  -t ext2 /dev/loop0
# mount -t ext2 /dev/loop0 /mnt/disk1
…
```

2.3 Solutions de sauvegarde

Les solutions de sauvegarde gèrent les opérations de sauvegarde, de restauration et de vérification de données de différents systèmes tels que Linux, MacOS, Windows, etc. Elles fonctionnent en réseau selon le modèle client/serveur en utilisant des protocoles de transfert de fichiers tels que NFS, SSH, SMB, etc.

Ces solutions sont relativement faciles à utiliser, efficaces et offrent de nombreuses fonctions avancées de gestion de sauvegarde.

Bacula, Amanda et BackupPC sont des solutions de sauvegarde libres distribuées sous licence GPL. Elles sont considérées comme des alternatives viables aux solutions de sauvegarde propriétaires.

3. Exercices

1. **On vient de compiler une application à partir des codes sources. Quelle est la commande qui permet d'installer cette application via le fichier Makefile ?**

   - ❑ `configure`
   - ❑ `make`
   - ❑ `make depend`
   - ❑ `make install`
   - ❑ `install`

2. **Quel est le problème le plus probable si la compilation d'un programme s'effectue avec succès mais l'installation de ses fichiers binaires présente des erreurs de permission ?**

   - ❑ Les droits d'accès du répertoire `/usr/bin` sont erronés
   - ❑ Un préfixe erroné est utilisé lors de la configuration ou n'est pas définit d'une manière propre lors de l'étape de compilation.
   - ❑ Les fichiers binaires doivent être installés dans un répertoire où seul l'utilisateur root peut écrire

3. **Quel est le type de sauvegarde si chaque semaine on effectue les commandes suivantes :**

```
dimanche : # dump 0uf /dev/st0 /home
lundi    : # dump 1uf /dev/st0 /home
mardi    : # dump 2uf /dev/st0 /home
...
samedi   : # dump 6uf /dev/st0 /home
```

   - ❑ Sauvegarde partielle
   - ❑ Sauvegarde différentielle
   - ❑ Sauvegarde incrémentale
   - ❑ Sauvegarde complète

4. **Quelle est la commande qui permet de désarchiver le fichier archive.tar.gz ?**

   - ❑ `tar tvf archive.tar.gz`
   - ❑ `tar xvf archive.tar.gz`
   - ❑ `tar cjf archive.tar.gz`

- ❑ `tar xjf archive.tar.gz`
- ❑ `tar xzf archive.tar.gz`

5. Quels sont les types de système de fichiers qui n'ont pas besoin d'être sauvegardés ?

- ❑ reiserFS
- ❑ swap
- ❑ ext2
- ❑ ext3
- ❑ /proc

Chapitre 7. Service DNS

Objectifs

Configuration élémentaire d'un serveur DNS :
– configurer BIND pour fonctionner comme un serveur de cache ;
– gérer un serveur en exécution et configurer les fichiers journaux.
Création et maintenance des fichiers de zone :
– créer un fichier de zone pour une résolution directe, une résolution inversée ou un serveur de niveau racine ;
– paramétrer les valeurs appropriées des enregistrements de ressources d'une zone ;
– déléguer des zones à d'autres serveurs DNS.
Sécurisation d'un serveur DNS :
– configurer un serveur DNS afin qu'il s'exécute dans le contexte d'un utilisateur non *root* et dans un environnement enfermé (*chrooted environnement*) ;
– configurer des serveurs DNS pour qu'ils s'échangent les données d'une manière sécurisée.

Points importants

– Fichiers de configuration, vocabulaire et utilitaires de BIND9.
– Localisation des fichiers de zone à partir des fichiers de configuration.
– Rechargement des fichiers de configuration et de zone après modification.
– Utilitaires d'interrogation à partir d'un serveur DNS.
– Différentes méthodes d'ajout de nouveaux hôtes dans les fichiers zones en incluant les zones inversées.
– Configuration de BIND9 pour qu'il s'exécute dans un environnement enfermé.
– Sécurisation des échanges de données.

Mots clés

`named`, `named.conf`, `rndc`, `rndc.conf`, `kill`, `host`, `nslookup`, `dig`, `named-checkconf`, `named-checkzone`, `dnssec-keygen`, `dnssec-signzone`, format des fichiers de zone, formats des enregistrements de ressources, DNSSEC, TSIG, ACL, `chroot`

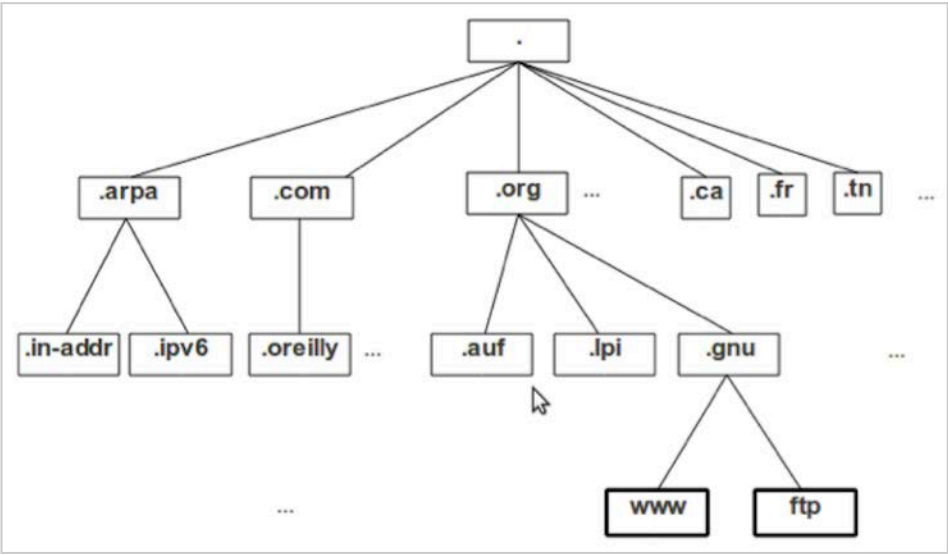
1. Concepts

Le service DNS assure, principalement, la conversion de noms de domaine en adresses IP et la détermination d'un serveur de messagerie pour une adresse électronique. Le service DNS d'Internet est un système distribué constitué de l'ensemble des serveurs DNS.

1.1 Domaine et délégation

Les noms de domaines sont organisés sous la forme d'une arborescence où les nœuds correspondent aux noms des domaines et les feuilles aux noms des hôtes (*figure 11*).

Figure 11. Arborescence des noms de domaine



Chaque domaine appartient à un domaine de niveau supérieur et peut contenir des sous-domaines. Au sommet de l'arbre se trouve le domaine racine (désigné par « . ») et au premier niveau se trouve les domaines de haut niveau (TLD : *Top Level Domains)*.

Les TLD sont de deux types :

- les domaines de haut niveau génériques (gTLD : *generic Top Level Domains*) tels que .com, .org, .net, etc. ;
- les domaines de haut niveau relatifs aux codes des pays (ccTLD : *country code Top Level Domains*) tels que .tn, .fr, .ca, .uk, .us, etc.

Au niveau Internet, les hôtes sont désignés par des noms de domaine complètement qualifiés (FQDN : *Fully qualified domain name*). Par abus de langage « nom de domaine »

est utilisé pour désigner un FQDN.

La gestion de tous les domaines est organisée d'une manière hiérarchique en appliquant le mécanisme de délégation. En effet un organisme autoritaire sur un domaine peut déléguer la gestion et la responsabilité d'un sous domaine à un autre organisme.

L'ICANN (*Internet Corporation for Assigned Numbers and Names*) est l'organisme autoritaire du domaine racine. L'ICANN délègue la gestion des domaines gTLD à des organismes accrédités et les domaines ccTLD aux pays correspondants.

Par exemple, le nom de domaine « www.auf.org » se compose de deux parties « www » et « auf.org ». Le nom de domaine « auf.org » a été délégué à l'Agence universitaire de la Francophonie (AUF) par un organisme accrédité de niveau gTLD gérant « .org ». Ce dernier a été à son tour délégué par l'ICANN. La partie « www » correspond à un nom d'hôte attribué par l'AUF, l'organisme autoritaire du domaine « auf.org ».

1.2 Organisation et structure

Les serveurs DNS d'Internet reflètent la structure de délégation décrite plus haut. En effet il existe un serveur DNS à chaque niveau de la hiérarchie de délégation.

Les serveurs DNS racines (*root DNS*) sont sous la responsabilité de l'ICANN. Il existe plusieurs serveurs racines répartis sur Internet. Ces serveurs doivent être connus par tous les serveurs DNS publics car ils représentent le point de départ des opérations de recherche.

Les serveurs DNS TLD (gTLD ou ccTLD) sont gérés par des agences ou des organismes tels que l'Agence française pour le nommage Internet en coopération (AFNIC) pour le domaine « .fr » et l'Agence tunisienne d'Internet (ATI) pour le domaine « .tn ».

1.3 Fichiers de zone

Les données relatives à un domaine sont formées d'un ensemble d'enregistrements de ressources qui sont stockés dans des fichiers de zone. Ces enregistrements contiennent :

- les données précisant le sommet de la zone et énumérant les propriétés générales de celle-ci. C'est l'enregistrement de type SOA ;
- les données autorité pour tous les nœuds ou hôtes de la zone. Typiquement c'est les enregistrements A (IPv4) ou AAAA (Ipv6) ;
- les données décrivant les informations globales de la zone telles que les enregistrements MX pour les serveurs de messagerie et les enregistrements NS pour les serveurs DNS ;
- dans le cas d'une délégation d'un sous domaine, un ou plusieurs enregistrements de type NS seront recensés permettant d'atteindre les serveurs DNS de ce sous domaine.

Le format des fichiers de zone ainsi que les formats des différents types d'enregistrements des ressources seront détaillés plus loin.

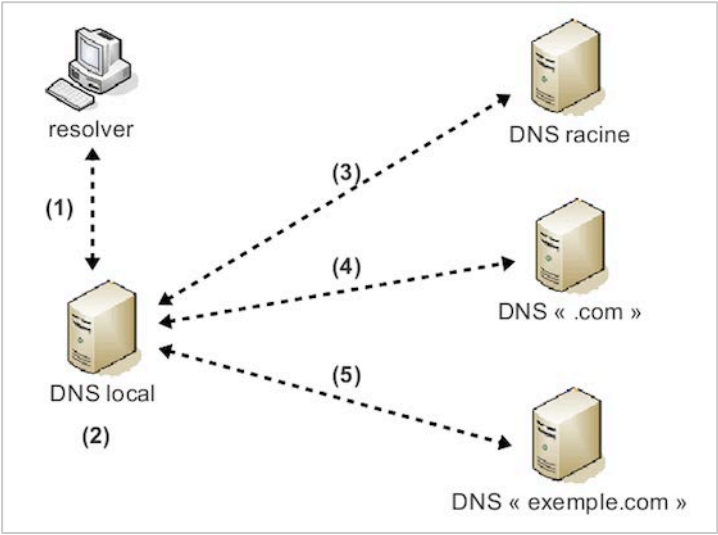
1.4 Résolution de noms

Lorsqu'une application réseau veut communiquer en utilisant un nom de domaine, elle demande la résolution de la correspondance IP à l'application client DNS locale (*resolver*). D'après la configuration (`/etc/resolv.conf`), le *resolver* consulte le DNS adéquat pour assurer cette correspondance.

Les étapes suivantes décrivent le processus de résolution du nom de domaine « www.exemple.com ».

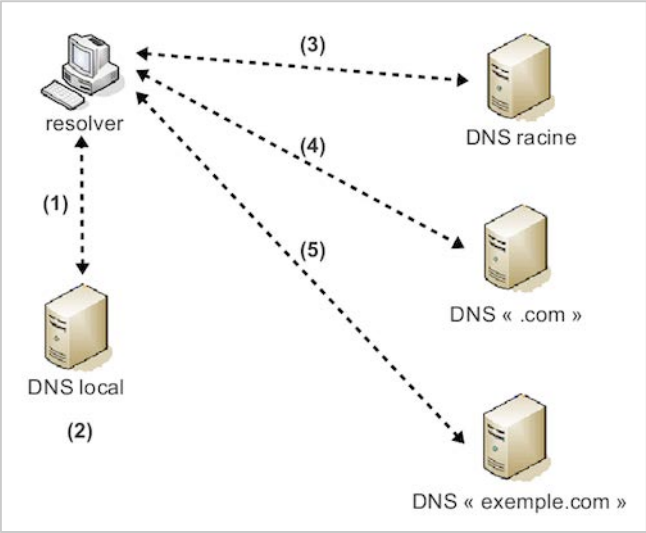
1. Le client DNS de l'hôte envoie la requête de résolution du nom de domaine « www.exemple.com » au serveur DNS local ;
2. Le serveur DNS local cherche l'information dans sa table locale (cache). Si l'information existe, le serveur envoie la réponse vers le client DNS et le traitement de la requête est terminé. Sinon, le processus de résolution de nom se poursuit avec les étapes suivantes :
3. La requête est envoyée vers un serveur racine qui renvoie l'adresse d'un serveur TLD gérant le domaine « .com » ;
4. La requête est envoyée au serveur TLD « .com » qui renvoie l'adresse du serveur DNS gérant le domaine « exemple.com » ;
5. La requête est envoyée au serveur DNS gérant « exemple.com » qui envoie la réponse vers le client DNS.

Figure 12. Résolution récursive



La résolution est dite **récursive** (*figure 12*) si les requêtes des étapes 3, 4 et 5 sont envoyées par le serveur local. Elle est dite **itérative** (*figure 13*) si ces requêtes sont envoyées par le client DNS lui-même.

Figure 13. Résolution itérative



2. Démon named

Le démon `named` est le serveur DNS fourni par le paquetage BIND (*Berkeley Internet Name Domain*). L'Université de Berkeley est à l'origine de BIND mais, actuellement, c'est l'Internet System Consortium (*ISC*) qui assure sa maintenance.

Le fichier de configuration principal de `named` est `named.conf` (sous `/etc/` ou `/etc/bind/`). La structure, le format et le contenu de ce fichier seront détaillés dans la section « Fichier de configuration named.conf »

Une fois lancé, `named` enregistre son numéro de processus (*PID : Process IDentifier*) dans le fichier `/var/run/named.pid`.

2.1 Syntaxe et options

SYNTAXE

La syntaxe de lancement du démon `named` est :

```
named [option …]
```

OPTIONS

Les options les plus courantes de `named` sont :

- `-c fichier` : spécifie un fichier de configuration autre que `named.conf` ;
- `-d niveau-debogage` : précise le niveau de débogage ;
- `-n nombre-CPU` : crée `nombre-CPU` processus légers (*threads*). Par défaut `named` détermine le nombre de CPU et crée autant de *threads*. S'il n'y arrive pas, un seul *thread* est créé ;
- `-p port` : indique le numéro de port d'écoute du serveur, par défaut c'est le port 53 ;
- `-u utilisateur` : spécifie le contexte de l'utilisateur dans lequel sera exécuté le démon. Cette option sera traitée plus en détail dans la section « 8.1 Environnement enfermé » ;
- `-t répertoire` : spécifie le répertoire racine pour une exécution dans un environnement enfermé (*chroot environment*). Cette option sera traitée plus en détail dans la section « 8.1 Environnement enfermé » ;
- `-v` : affiche la version.

2.2 Signaux

Le démon `named` réagit à certains signaux (envoyés par la commande `kill`) d'une manière particulière :

- `SIGHUP` : le démon relit le fichier de configuration ainsi que les fichiers de zone ;
- `SIGINT` et `SIGTERM` : le démon s'arrête normalement.

Dans un fonctionnement normal, l'utilisation de la commande `rndc` (voir plus loin) est préférée aux envois de signaux.

2.3 Démarrage et arrêt

Pour démarrer ou arrêter le démon `named`, la majorité des distributions Linux utilisent les scripts RC ou des commandes équivalentes. Les lignes de commandes qui suivent sont utilisées pour les distributions *Debian* et dérivées :

```
# /etc/init.d/bind9 {start|stop|reload|restart|force-reload|status}
# # ceci est équivalent à :
# invoke-rc.d bind9 {start|stop|reload|restart|force-reload|status}
```

Les commandes équivalentes pour les distributions *Red Hat* et dérivées sont :

```
# /etc/init.d/named {start|stop|reload|restart|force-reload|status}
# # ceci est équivalent à :
# service named {start|stop|reload|restart|force-reload|status}
```

3. Fichier de configuration named.conf

`named.conf` est le fichier de configuration principal du serveur DNS BIND. Il est le premier fichier lu par `named`. La directive `include` permet de répartir la configuration sur plusieurs fichiers pour des fins de clarté ou d'organisation.

3.1 Structure et format

`named.conf` est structuré en clauses regroupant chacune un ensemble d'instructions sous la forme d'un bloc. Il faut impérativement respecter la syntaxe du fichier qui peut être résumée par les règles suivantes :

- une instruction se termine par « ; » ;
- un bloc d'instruction débute par « { » et se termine par « }; » ;
- un commentaire est écrit sous l'une des formes suivantes :

```
/* commentaire au style C */
// commentaire au style C++
#  commentaire au style PERL/SHELL
```

Les clauses les plus usuelles sont :

- `acl` : définit des groupes de machines ou d'utilisateurs identifiés par leurs clés, qui seront référencées dans d'autres clauses ou instructions ;
- `controls` : décrit et contrôle l'administration de `named` par l'utilitaire `rndc` ;
- `key` : définit les clés utilisées pour les opérations de contrôle et d'authentification telles que l'utilisation de l'utilitaire `rndc` ou les mises à jour dynamiques ;
- `logging` : configure l'emplacement, le niveau et le type de journalisation ;
- `options` : regroupe les instructions contrôlant le comportement générique ou global et ayant un effet sur toutes les zones ;
- `server` : définit le comportement et les propriétés du serveur lorsqu'il accède ou répond aux serveurs de noms distants ;
- `trusted-keys` : contient des clés publiques utilisées pour des opérations de DNS sécurisé (DNSSEC) ;
- `view` : contrôle le comportement et les fonctionnalités de `named` en se basant sur les adresses IP des hôtes ;
- `zone` : définit la zone supportée par le serveur.

Le fichier `named.conf` est généralement sous la forme :

```
// définition des ACL
acl "nomACL" {...};
// configuration de la journalisation
logging {...};
```

```
// définition des options globales
options {...};
// déclaration des zones prédéfinies
zone {...};
...
// déclaration des zones à résoudre
zone {...} ;
...
```

3.2 Journalisation

Par défaut, `named` envoie les messages standard au démon `rsyslog`, qui les enregistre dans le fichier `/var/log/messages`.

La clause `logging` permet de paramétrer les opérations de journalisation. Elle comporte plusieurs instructions, dont `channel` et `category` sont les plus importantes. L'instruction `channel` définit la destination des messages de journalisation et l'instruction `category` détermine l'information à journaliser.

EXEMPLE

Le paramétrage de la clause `logging` suivante configure `named` de façon à ce qu'il envoie les messages de débogage relatifs aux requêtes DNS vers le fichier `/var/log/query.log`.

```
logging {
    channel query.log {
        file "/var/log/query.log";
        severity debug 3; // les valeurs de debug sont 1(par défaut),2 et 3
    };
    category queries { query.log; };
};
```

3.3 Zones particulières

Le fichier `named.conf` inclut par défaut la déclaration des zones particulières racine, localhost et « 127.in-addr.arpa » :

- le fichier de la zone racine, désigné par « . » et de type `hint`, contient la liste des serveurs à interroger lorsqu'un serveur de nom n'arrive pas à résoudre une requête ;
- la zone « localhost » permet la résolution du nom « localhost » à l'adresse de boucle locale « 127.0.0.1 » lors de l'utilisation du serveur DNS ;
- la zone « 127.in-addr.arpa » assure la résolution inversée de l'adresse de boucle locale « 127.0.0.1 ».

EXEMPLE

La portion du fichier `named.conf` qui suit illustre la déclaration de ces zones particulières.

```
zone "." {
        type hint;
        file "/etc/bind/db.root";
};
zone "localhost" {
        type master;
        file "/etc/bind/db.local";
};
zone "127.in-addr.arpa" {
        type master;
        file "/etc/bind/db.127";
};
```

4. Configurations types

Il existe plusieurs manières de configurer un serveur DNS. Les configurations types les plus fréquentes sont : un serveur maître (*master*), un serveur esclave (*slave*), un serveur de cache (*caching only*) et un serveur de retransmission (*forwarding*).

Dans la pratique, la configuration d'un serveur DNS est formée d'une combinaison de ces configurations types. En effet un serveur de noms peut être configuré pour fonctionner comme maître pour certaines zones, esclave pour d'autres et fournir le service de cache ou de retransmission pour tout le reste.

NOTE

Le paramètre de zone `type` n'est pas seul à définir le type de configuration d'un serveur DNS. Il existe en effet d'autres paramètre tels que : `forward`, `forwardes`, etc.

4.1 Serveur maître

Un serveur de nom maître définit le fichier de la zone sur laquelle il fait autorité. Cette autorité a été déléguée à ce dernier, à travers l'enregistrement `NS`, par le serveur de niveau supérieur.

EXEMPLE

La portion suivante du fichier `named.conf` configure BIND comme serveur maître (`type master`) pour la zone « exemple.com » dont les enregistrements seront stockés dans le fichier `/etc/bind/db.exemple.com`. L'instruction `notify no` indique que BIND ne doit pas notifier les serveurs esclaves lors du changement du fichier de zone. Par défaut, BIND notifie les serveurs esclaves précisés dans les enregistrements `NS`.

```
zone "exemple.com" {
   type master;
   notify no;
   file "/etc/bind/db.exemple.com";
};
```

4.2 Serveur esclave

Un serveur de nom esclave récupère ses données de zone moyennant l'opération de transfert de zone à partir du serveur maître et répond en tant que serveur autoritaire aux requêtes concernant cette zone. Il est impossible de déterminer si la réponse d'une requête provient du serveur maître de la zone ou de l'un de ses esclaves.

Pour une zone donnée, il y a obligatoirement un serveur maître et éventuellement un ou plusieurs serveur esclaves.

EXEMPLE

La portion suivante du fichier `named.conf` configure BIND comme serveur esclave (`type slave`) pour la zone « exemple.com » dont les données seront récupérées à partir du serveur précisé par l'instruction `masters {…}`. L'instruction `file` est optionnelle, elle permet au serveur esclave de sauvegarder les données de la zone sur le disque afin de les utiliser en cas de redémarrage.

```
zone "exemple.com" {
   type slave;
   file "/var/cache/bind/db.exemple.com";
   masters { 192.168.1.1; }; // adresse IP du serveur maître
};
```

Le fichier de configuration du serveur maître doit être rectifié pour autoriser le transfert de zone vers le(s) serveur(s) esclave(s) en ajoutant l'instruction `allow-transfer {...}`:

```
zone "exemple.com" {
   type master;
   file "/etc/bind/db.exemple.com";
   allow-transfer { 192.168.1.2; }; // liste des adresses IP des
                  // serveurs esclaves
};
```

4.3 Serveur de cache

Un serveur de cache récupère les informations à partir du serveur maître de la zone contenant l'information et sauvegarde (cache) les données localement. À la prochaine requête pour la même information, le serveur de cache répondra en utilisant les données déjà sauvegardées. Si la duré de vie (*TTL : Time To Live*) de l'information expire, le

serveur de cache refait l'opération de récupération de l'information.

En général la configuration d'un serveur de cache est utilisée dans les cas suivants :

- un serveur agissant comme maître ou esclave d'une ou plusieurs zones et comme serveur de cache pour le reste des requêtes ;
- un serveur local de cache seulement : typiquement utilisé pour minimiser le trafic extérieur ou compenser les liens externes de faible débit. Dans ce cas, il est désigné par « serveur proxy DNS ».

NOTE

La réponse d'une requête est dite autoritaire si la réponse est reçue :

- d'un serveur maître de la zone ;
- d'un serveur esclave de la zone avec des informations non encore expirées (*non time-expired*) ;
- par l'intermédiaire d'un serveur de cache qui vient de la récupérer à partir d'un serveur maître ou d'un serveur esclave. Si la réponse est lue à partir du cache, elle n'est pas autoritaire.

EXEMPLE

La portion suivante du fichier `named.conf` configure BIND comme serveur de cache :

```
forwarders {
   10.1.1.1; //par exemple : premier serveur DNS du FAI
   10.1.1.2; //                  deuxième serveur DNS du FAI
};
```

4.4 Serveur de retransmission

Un serveur de retransmission est un serveur qui transfert toutes les requêtes à un autre serveur DNS supportant les résolutions récursives.

Un serveur de retransmission est utilisé :

- en cas de connexion lente au réseau Internet ;
- pour simplifier l'administration locale, en limitant les changements du paramètre « serveurs DNS distants » à un seul hôte (serveur de retransmission) au lieu de le modifier pour tous les hôtes du réseau local ;
- comme un composant dans la configuration d'un service DNS segmenté pour des fins de sécurité.

BIND peut être configuré comme serveur de retransmission en utilisant les instructions `forward` et `forwards` dans la section `options` (niveau global) ou dans la section `zone` (niveau domaine).

EXEMPLES

La portion du fichier de configuration `named.conf` suivante illustre une retransmission globale :

```
options {
        ...
        forwarders {10.0.0.1; 10.0.0.2;};
        forward only;
};
...
```

La portion du fichier de configuration `named.conf` suivante illustre une retransmission de niveau domaine pour la zone « exemple.com »:

```
zone "exemple.com" IN {
        type forward;
        forwarders {192.168.1.1; 192.168.1.2;};
};
```

5. Fichier de zone

Un fichier de zone contient les enregistrements de ressources (RR : *resource records)* d'un espace de noms. Le nom et l'emplacement d'un fichier de zone est spécifié par l'instruction `file` de la clause `zone` du fichier `named.conf`.

5.1 Format d'un fichier de zone

Le format d'un fichier de zone peut se résumer par les points suivants :

- un fichier de zone contient des commentaires, des directives et des enregistrements de ressources ;
- un commentaire commence par « ; » et continue jusqu'à la fin de la ligne ;
- une directive commence par « $ ». Les directives les plus courantes sont :
  - `$ORIGIN` : définit le nom de base qui sera concaténé à tous les enregistrements non totalement qualifiés,
  - `$INCLUDE` : inclut le fichier spécifié à l'endroit où apparaît la directive,
  - `$TTL` : règle la valeur par défaut de la durée de vie (*TTL : Time To Live*) pour la zone. Elle doit être présente et doit figurer avant le premier enregistrement ;
- le premier enregistrement de ressource doit être SOA (*Start Of Authority*) ;
- le format général d'un enregistrement est : « `nom ttl classe type valeur` » où
  - `nom` : nom (ou label) du nœud dans le fichier de zone auquel appartient cet

enregistrement. La valeur @ indique que la valeur de $ORIGIN sera utilisée et un blanc ou une tabulation indique que le dernier nom cité sera utilisé.

- `ttl` : durée de vie (en seconde) de l'enregistrement dans un cache. La valeur 0 indique que l'enregistrement ne doit pas être maintenu dans un cache ;
- `classe` : définit la famille du protocole. La valeur normale est IN (*INternet protocol*) ;
- `type` : type de l'enregistrement de ressource ;
- `valeur` : valeur de l'enregistrement qui dépend du type et de la classe.

5.2 Format des enregistrements de ressources

Les types d'enregistrement de ressources les plus fréquemment utilisés sont : `A`, `CNAME`, `MX`, `NS`, `PTR`, `SOA`. Les formats de ces types sont détaillés dans les paragraphes qui suivent.

`A` (*Address*) : l'enregistrement de type adresse spécifie une adresse IP à associer à un nom d'hôte selon le format « `nom ttl IN A ip` ». Si le nom d'hôte n'est pas spécifié l'adresse IP indiquée sera associée au dernier nom (ou label).

EXEMPLE

Dans l'exemple qui suit, l'adresse « 192.168.1.10 » est associée au nom « serveur1.exemple.com » et les adresses « 192.168.1.20 » et « 192.168.1.21) » au nom « mail.exemple.com ». Pour le nom « mail.exemple.com », `named` répond par défaut : (192.168.1.20, 192.168.1.21) et (192.168.1.21, 192.168.2.20) à tour de rôle.

```
$ORIGIN exemple.com.
;nom     ttl classe type ip
serveur1     IN.    A    192.168.1.10
mail         IN     A    192.168.1.20
             IN     A    192.168.1.21
```

`CNAME` (*Canonical NAME*) : l'enregistrement de type nom canonique (ou alias) permet d'attribuer un deuxième nom au nom réel de l'hôte, qui peut être dans un autre domaine, selon le format « `nom ttl IN CNAME nomRéelle` ».

EXEMPLE

L'exemple qui suit attribue les alias « www.exemple.com » et « irc.exemple.com » à l'hôte « serveur1.exemple.com » et l'alias « ftp.exemple.com » à un hôte de nom appartenant à un autre domaine.

```
$ORIGIN exemple.com.
;nom ttl classe type   nomRéelle
www      IN     CNAME  serveur1.exemple.com
irc      IN     CNAME  serveur1.exemple.com
ftp      IN     CNAME  serveur.un.autre.domaine
```

`MX` (*Mail eXchange*) : l'enregistrement de type `MX` spécifie les noms et les préférences des serveurs de messagerie de la zone. Il est utilisé par les applications agents de messagerie (*mail agents*) pour router les courriers électroniques du domaine. Le format de l'enregistrement de type `MX` est : « `nom ttl IN MX préférence nom` ».

EXEMPLE

D'après l'exemple qui suit, les courriers électroniques du domaine « exemple.com » sont routés vers l'hôte « mail.exemple.com ». Si ce dernier n'est pas disponible (arrêté, dérangé ou en panne) alors les courriers seront routés vers « mail2.exemple.com ».

```
$ORIGIN exemple.com.
;nom ttl       classe type  préfé.  nom
               IN     MX    10      mail  ;forme courte
; ceci est équivalent à
; exemple.com. IN     MX    10      mail.exemple.com.
               IN     MX    20      mail2.exemple.com.
```

`NS` (*Name Server*) : l'enregistrement de type serveur de noms définit les serveurs de noms autoritaires de la zone. En général, Il est situé juste après l'enregistrement `SOA` et il est utilisé aussi pour définir les délégations des sous domaines. Le format de l'enregistrement NS est : « `nomDomaine ttl IN NS nom` ».

EXEMPLE

La configuration qui suit déclare les hôtes « dns.exemple.com » et « dns2.exemple.com » comme les serveurs DNS de la zone « exemple.com » et délègue la gestion du sous-domaine « sd.exemple.com » au serveur DNS « dns.sd.exemple.com ».

```
$ORIGIN exemple.com.
               SOA ...
;nom                ttl classe type nom
                        IN     NS   dns.exemple.com.
                        IN     NS   dns2.exemple.com.
dns                     IN     A    192.168.1.1
dns2                    IN     A    192.168.1.2
;sd.exemple.com est un sous domaine exemple.com
$ORIGIN sd.exemple.com.
                        IN     NS   dns.sd.exemple.com.
...
dns                     IN     A    192.168.2.1
;ou sans utiliser la directive $ORIGIN
;sd.exemple.com.        IN     NS   dns.sd.exemple.com.
;dns.sd.exemple.com.    IN     A    192.168.2.1
```

`PTR` (*PoinTeR*) : l'enregistrement de type `PTR` sert à la résolution inversée des noms selon le format : « `nomARPA ttl IN PTR nom` ».

EXEMPLE

D'après l'exemple qui suit, l'adresse ip « 192.168.1.10 » sera retournée pour une recherche inversée pour l'hôte « serveur1.exemple.com ».

```
$ORIGIN 1.168.192.IN-ADDR.ARPA.
...
;nomARPA ttl classe type nom
10          IN     PTR  serveur1.exemple.com.
...
```

SOA (*Start of Authority*) : l'enregistrement de type SOA définit les paramètres globaux de la zone (domaine). Il existe un seul enregistrement de type SOA par fichier de zone. Ces paramètres sont :

- serveur : nom du serveur de noms ;
- e-mail : courriel du responsable du domaine ;
- nSerie : numéro de série du fichier de zone qui sera incrémenté à chaque modification ;
- raf : période d'envoi des demandes de rafraîchissement par un serveur esclave vers un serveur maître ;
- ret : période de retransmission des demandes de rafraîchissement si le serveur maître ne répond pas ;
- exp : durée au bout de laquelle, si le serveur maître ne répond pas à une demande de rafraîchissement, le serveur esclave cesse de répondre aux requêtes en tant qu'autoritaire ;
- ttl : durée minimale que doit passer une information de cette zone dans un serveur de cache.

Le format d'un enregistrement SOA est : «nom ttl IN SOA serveur e-mail (nSerie raf ret exp ttl) ».

EXEMPLE

La configuration qui suit définit les paramètres globaux de la zone « exemple.com ».

```
$ORIGIN exemple.com.
;name ttl classe type serveur           e-mail (nSerie raf ret exp ttl)
          IN     SOA   dns.exemple.com. root.exemple.com. (
                       2011092800 ; nSerie
                       172800     ; ou 2d ? raf = 2 jours
                       900        ; ou 15m? ret = 15 minutes
                       1209600    ; ou 2w ? exp = 2 semaines
                       3600       ; ou 1h ? ttl = 1 heure
                       )
; Les lignes qui suivent sont aussi équivalentes:
```

```
;@          IN     SOA dns.exemple.com. root.exemple.com. ( ...)
;exemple.com. IN SOA dns.exemple.com. root.exemple.com. ( ...)
```

6. Utilitaire rndc

L'utilitaire `rndc` (*remote name daemon control*) est utilisé pour administrer le démon `named` de l'hôte local ou d'un hôte distant. Il communique avec `named` d'une manière sécurisée à travers une connexion TCP, par défaut sur le port 953.

Le fichier de configuration de `rndc` est `rndc.conf`. Si le fichier n'existe pas, l'utilitaire utilise la clé localisée dans le fichier `rndc.key`.

6.1 Paramétrage de named.conf

Par défaut, l'administration de démon `named` par `rndc` est autorisée pour les connexions locales. Pour les connexions distantes, il faut rajouter la clause `controls` au fichier de configuration `named.conf`.

EXEMPLES

La clause `controls` décrite dans l'exemple qui suit demande à `named` (s'exécutant sur le hôte « 192.168.1.1 ») d'autoriser les commandes `rndc` provenant de l'hôte « 192.168.1.20 », si la clé adéquate est donnée.

```
controls {
   inet 192.168.1.1 allow { 192.168.1.20; } keys {<nomClé-192.168.1.1>;};
};
```

<nomClé> fait référence à la clause `key` du fichier de configuration `named.conf`. L'exemple suivant illustre une clause `key` utilisant l'algorithme HMAC-MD5.

```
key "<nomClé-192.168.1.1>" {
  algorithm hmac-md5;
  secret "<valeurClé-192.168.1.1>";
};
```

6.2 Paramétrage de rndc.conf

Le fichier de configuration `rndc.conf` possède une structure et une syntaxe similaires au fichier `named.conf` mais il est plus simplifié que ce dernier. Il utilise les trois clauses : `options`, `server` et `key`.

La clause `key` représente la clause la plus importante, elle a la même syntaxe que celle de `named.conf`. Le nom de la clé ainsi que sa valeur doivent être identiques à ceux du fichier `named.conf`.

NOTE

La commande `rndc-confgen` génère les fichiers de configuration de l'utilitaire `rndc`. Elle peut être utilisée comme une alternative à l'édition manuelle du fichier `rndc.conf` et aux clauses `controls` et `key` correspondantes dans le fichier `named.conf`.

EXEMPLE

Le fichier de configuration `rndc.conf` suivant permet à l'utilitaire `rndc` de contrôler le démon `named` distant d'adresse 192.168.1.1 en utilisant la clé `<nomClé-192.168.1.1>`.

```
key "<nomClé-192.168.1.1>" {
  algorithm hmac-md5;
  secret "<valeurClé-192.168.1.1>";
};
server 192.168.1.1{
  key  "<nomClé-192.168.1.1>";
};
```

6.3 Syntaxe et options

La syntaxe générale de l'utilitaire `rndc` est :

```
rndc [option ...] commande [option-commande ...]
```

Les options les plus utilisées sont :

- `-c fichier` : spécifie un fichier de configuration autre que `rndc.conf`;
- `-p port` : spécifie un autre numéro de port ;
- `-s serveur` : spécifie un serveur autre que le serveur par défaut (`default-server`) ;
- `-y clé` : spécifie une clé autre que celle par défaut (`default-key`).

Les sous commandes de `rndc` sont :

- `halt` : arrête immédiatement le service `named` ;
- `Querylog` : enregistre toutes les requêtes effectuées auprès du serveur de noms ;
- `Refresh` : rafraîchit la base de données du serveur ;
- `Reload` : recharge les fichiers de zone mais conserve toutes les réponses précédemment mises en cache. Cette commande permet également d'apporter des changements aux fichiers de zone sans perdre toutes les résolutions de noms stockées. Si les changements n'affectent qu'une zone particulière, il est possible de recharger seulement cette zone en ajoutant le nom de la zone après la commande `reload` ;
- `Stats` : vide les statistiques courantes de `named` vers le fichier `/var/named/named.stats` ;
- `stop` : arrête correctement le serveur, en enregistrant préalablement toutes les mises à jour dynamiques et toutes les données des transferts de zone incrémentaux (`IXFR`).

7. Commandes de diagnostic et de configuration

La distribution BIND fournit avec le démon `named`, des commandes de diagnostic (`host`, `nslookup`, `dig`, `etc`), de vérification (`named-checkconf`, `named-checkzone`, etc) et de sécurité (`dnssec-keygen`, `dnssec-signzone`, etc).

7.1 Commande host

`Host` est une commande simple pour effectuer des recherches DNS.

SYNTAXE

L'utilisation générale de la commande `host` est la suivante :

```
host [option …] nom [serveur]
```

- `nom` : le nom d'hôte ou de domaine à résoudre. Il peut être aussi une adresse IP dans le cas d'une résolution inversée ;
- `Serveur` : nom ou adresse IP du serveur DNS à interroger. Par défaut, la commande `host` utilise le serveur DNS recensé dans le fichier `/etc/resolv.conf`.

EXEMPLES

Les lignes de commande suivantes sont équivalentes et assurent la résolution du nom d'hôte « www.exemple.com » à partir du serveur DNS par défaut.

```
host www.exemple.com
host www.exemple.com a
host -t a www.exemple.com
```

Même exemple mais en interrogeant le serveur DNS « dns.exemple.com ».

```
host www.exemple.com dns.exemple.com
```

Les lignes de commande suivantes sont équivalentes et assurent la résolution inversée de l'hôte d'adresse IP « 192.168.1.10 ».

```
host 192.168.1.10
host 10.1.168.192.in-addr.arpa ptr
host -t ptr 10.1.168.192.in-addr.arpa
```

7.2 Commande nslookup

La commande `nslookup` est officiellement abandonnée et remplacée par la commande `dig`. `nslookup` est cependant presque universellement disponible.

SYNTAXE

```
nslookup [option …] nom [serveur]
```

Les paramètres `nom` et `Serveur` ont les mêmes significations que ceux de la commande `host`.

EXEMPLES

Les lignes de commande suivantes sont équivalentes et assurent la résolution du nom d'hôte « www.exemple.com » à partir du serveur DNS par défaut.

```
nslookup www.exemple.com
nslookup www.exemple.com a
nslookup -type=A www.exemple.com
```

Même exemple mais en interrogeant le serveur DNS « dns.exemple.com ».

```
nslookup www.exemple.com dns.exemple.com
```

Les lignes de commande suivantes sont équivalentes et assurent la résolution inversée de l'hôte d'adresse IP 192.168.1.10

```
nslookup 192.168.1.10
nslookup 10.1.168.192.in-addr.arpa ptr
nslookup -type=ptr 10.1.168.192.in-addr.arpa
```

7.3 Commande dig

La commande `dig` est l'outil préféré de diagnostic d'un serveur DNS. Elle est plus puissante et plus riche que la commande `nslookup`.

SYNTAXE

La syntaxe typique de la commande `dig` est :

```
dig [@serveur] nom [type-req] [+option-req ...] [-option-dig ...]
```

avec

- `serveur` : nom ou adresse IP du serveur DNS à interroger. Il doit être précédé par « **@** ». Par défaut c'est le serveur défini dans le fichier `/etc/resolv.conf` ;
- `Nom` : nom d'hôte ou de domaine ou adresse IP à résoudre ;
- `Type-req` : type d'enregistrement à retourner : `a` (par défaut), `mx`, `ns`, `soa`, etc. La valeur particulère `any` indique n'importe quel type et la valeur particulière `axfr` demande la liste complète de tous les enregistrements. La disponibilité de la fonctionnalité `axfr` dépend de l'option `allow-transfer` du serveur DNS à interroger ;
- `Option-req` : précise une option de la requête ou un style d'affichage des résultats. Elle doit être précédé par « **+** » ;
- `option-dig` : option de la commande `dig`.

OPTIONS

Quelques options de la commande `dig` :

- `-P` : lance un *ping* sur le serveur à utiliser ;
- `-p port` : change le port vers lequel seront envoyées les requêtes ;
- `-f fichier` : spécifie un fichier contenant les commandes différées ;
- `-T secondes` : précise le temps entre les exécutions des commandes du fichier différé ;
- `-c` : indique la classe de requête : `in` (par défaut), `hesiod`, `chaos`, `any` ;
- `-t` : indique le type d'enregistrement à récupérer ;
- `-x` : spécifie que la notation inversée sera utilisée.

EXEMPLES

Les lignes de commandes suivantes sont équivalentes et assurent la résolution du nom d'hôte « www.exemple.com » à partir du serveur DNS par défaut.

```
dig www.exemple.com
dig www.exemple.com a
dig -t a www.exemple.com
```

Même exemple mais en interrogeant le serveur DNS « dns.exemple.com ».

```
dig @dns.exemple.com www.exemple.com
```

Les lignes de commande suivantes sont équivalentes et assurent la résolution inversé de l'hôte d'adresse IP 192.168.1.10

```
dig -x 192.168.1.10
dig 10.1.168.192.in-addr.arpa ptr
dig -t ptr 10.1.168.192.in-addr.arpa
```

L'exemple suivant récupère l'enregistrement `MX` du domaine « exemple.com » en précisant les options de requête : « `+vc +time=5 +retry=2` ». La requête utilisera une connexion TCP (`vc` correspond à TCP et `novc` (par défaut) à UDP) avec un délai limité à 5 secondes et 2 tentatives au maximum.

```
dig exemple.com mx +vc +time=5 +retry=2
```

7.4 Commande named-checkconf

La commande `named-checkconf` vérifie la syntaxe d'un fichier de configuration de `named` ainsi que les fichiers inclus à travers l'instruction `include`.

SYNTAXE

```
named-checkconf [option …] [fichier]
```

avec `fichier` : chemin du fichier de configuration à vérifier. Si aucun fichier n'est spécifié, `named.conf` sera traité.

NOTE

Les fichiers lus par `named` et qui ne sont pas spécifiés par l'instruction `include` tels que `rndc.key` et `bind.keys` ne sont pas traités automatiquement par `named-checkconf`. Il faut vérifier ces fichiers explicitement.

OPTIONS

Les options les plus utiles de la commande `named-checkconf` sont :

- `-t répertoire` : spécifie le répertoire racine pour un environnement enfermé ;
- `-p` : affiche le fichier de configuration `named.conf` ainsi que les fichiers inclus sous une forme canonique ;
- `-z` : teste le chargement de toutes les zones maîtres du fichier `named.conf` ;
- `-j` : lit le journal, s'il existe, lors du chargement d'un fichier zone.

EXEMPLE

Les lignes de commandes suivantes vérifient la syntaxe du fichier de configuration `named.conf` (ainsi que les fichiers inclus), `/etc/bind/rndc.key` et `/etc/bind/bind.keys`.

```
Named-checkconf
Named-checkconf /etc/bind/rndc.key
Named-checkconf /etc/bind/bind.keys
```

7.5 Commande named-checkzone

La commande `named-checkzone` vérifie la syntaxe et l'intégrité d'un fichier de zone. Elle effectue les mêmes vérifications que `named`, d'où l'intérêt de l'utiliser avant d'ajouter les fichiers de zone à la configuration d'un serveur de noms.

SYNTAXE

```
named-checkzone [option …] zone fichier-zone
```

avec

- `zone` : nom de domaine de la zone à vérifier ;
- `fichier-zone` : fichier contenant les données de la zone.

OPTIONS

Les options les plus utiles de la commande `named-checkzone` sont :

- `-d` : active le débogage ;
- `-j` : lit le fichier journal (cache) lors du chargement du fichier zone ;

- `-f format` : spécifie le format du fichier zone. Les valeurs possibles de `format` sont `text` (par défaut) pour le format texte et `raw` pour le format binaire ;
- `-t répertoire` : vérifie la syntaxe du fichier dans un environnement enfermé dont le répertoire racine est `répertoire`.

EXEMPLE

L'exemple qui suit permet de vérifier la syntaxe et l'intégrité des enregistrements du fichier `/etc/bind/db.exemple.com` et du fichier journal correspondants à la zone du domaine « exemple.com » tout en affichant les messages de débogage.

```
named-checkzone -j -d exemple.com /etc/bind/db.exemple.com
```

7.6 Commande dnssec-keygen

La commande `dnssec-keygen` génère des clés de cryptage qui seront utilisées par DNSSEC, TSIG et TKEY.

SYNTAXE

La syntaxe générale de la commande `dnssec-keygen` est :

```
dnssec-keygen [option ...] nom-clé
```

avec `nom-clé` : nom de la clé à générer. Le nom de la clé spécifiée pour DNSSEC doit correspondre au nom de la zone pour laquelle la clé est générée.

Après la génération de la clé, `dnssec-keygen` affiche la chaine d'identification de la clé sous la forme `Knom-clé.+aaa+iiiii` où :

- `nom-clé` est le nom de la clé ;
- `aaa` est la représentation numérique de l'algorithme et
- `iiiii` est l'identifiant de la clé.

Les fichiers créés sont `Knom-clé.+aaa+iiiii.key` contenant la clé publique et `Knom-clé.+aaa+iiiii.private` contenant la clé privée.

Le fichier `Knom-clé.+aaa+iiiii.key` contient un enregistrement `DNSKEY` qui peut être inséré dans le fichier de zone (directement ou à travers l'instruction `INCLUDE`).

NOTE

Les deux fichiers `.key` et `.private` sont générés aussi pour les algorithmes de cryptage symétrique tel que HMAC-MD5. Ils sont équivalents.

OPTIONS

Les options fréquemment utilisées de la commande `dnssec-keygen` sont :

- `-a algorithme` : précise l'algorithme cryptographique à choisir. Plusieurs algorithmes peuvent être utilisés pour la génération des clés. Le protocole DNSSEC oblige

l'implémentation de l'algorithme RSASHA1 et recommande l'implémentation de l'algorithme DSA. Pour TSIG, l'implémentation de l'algorithme HMAC-MD5 est obligatoire ;

- `-b taille` : fixe la taille (en bits) de la clé. Le choix de la taille de clé dépend de l'algorithme utilisé. La taille des clés RSASHA1 est [512..4096], celle des clés DSA est [512..1024] et elle est divisible par 64, et pour les clés HMAC-MD5 elle est [1..512] ;
- `-n type` : spécifie le type de la clé. Parmi les valeurs possibles, la valeur `ZONE` est spécifiée pour la clé de signature de zone et `HOST` ou `ENTITY` pour la clé associée à un hôte ;
- `-f flag` : définit la valeur du champ `flag` de l'enregistrement `KEY/DNSKEY`. Les seules valeurs possibles sont `KSK` (*Key Signing Key*) et `REVOKE` ;
- `-K répertoire` : spécifie le répertoire où les fichiers clés seront générés ;
- `-q` : mode silencieux. N'affiche pas les symboles de progression de la génération des clés ;
- `-r fichier-aléatoire` : spécifie le fichier source pour la génération des nombres aléatoires. Si le système d'exploitation ne fournit pas un fichier périphérique `/dev/random` ou un équivalent, le clavier sera utilisé comme source.

EXEMPLES

La première commande `dnssec-keygen` qui suit génère une clé DSA de taille 512 bits pour le domaine « exemple.com ».

Elle affiche la chaine d'identification `Kexemple.com.+003+27270` et crée :

- le fichier `Kexemple.com.+003+27270.key` contenant la clé publique,
- le fichier `Kexemple.com.+003+27270.private` contenant la clé privée.

```
$ dnssec-keygen -a DSA -b 512 -n ZONE exemple.com
Kexemple.com.+003+27270
$ ls Kexemple.com*
Kexemple.com.+003+27270.key
Kexemple.com.+003+27270.private
$
```

Dans le deuxième exemple, la commande `dnssec-keygen` génère une clé HMAC-MD5 de taille 512 bits et de type `HOST`. Les deux fichiers `.key` et `.private` sont équivalents et contiennent la même clé.

```
$ dnssec-keygen -a HMAC-MD5 -b 512 -n HOST tsig-serveur1-serveur2
Ktsig-serveur1-serveur2.+157+65523
$ ls Ktsig-serveur1-serveur2*
Ktsig-serveur1-serveur2.+157+65523.key
Ktsig-serveur1-serveur2.+157+65523.private
```

```
$
```

7.7 Commande dnssec-signzone

La commande `dnssec-sigzone` génère, à partir d'un fichier de zone, une version signée de cette zone. En particulier, elle calcule et rajoute les enregistrements `NSEC` et `RRSIG` pour l'authentification des données dans le protocole DNSSEC.

NOTE

Une sous zone déléguée d'une zone signée peut être signée ou non. Si elle est signée, le fichier `keyset` correspondant doit exister.

SYNTAXE

La syntaxe générale de la commande `dnssec-signzone` est :

```
dnssec-signzone [option …] fichier-zone [clé …]
```

avec

- `fichier-zone` : fichier contenant la zone à signer ;
- `clé` : spécifie les clés à utiliser pour signer la zone. Si aucune clé n'est spécifiée, alors la zone sera examinée pour les enregistrements `DNSKEY` au sommet de la zone. Si les enregistrements sont retrouvés et des clés privées correspondantes existent dans le répertoire courant, alors elles seront utilisées.

OPTIONS

- `-d répertoire` : spécifie le répertoire où chercher les fichiers `dsset-` ou `keyset-` ;
- `-K répertoire` : spécifie le répertoire où chercher les clés DNSSEC. Par défaut c'est le répertoire courant ;
- `-k clé-KSK` : spécifie la clé KSK (*Key Signing Key*) ;
- `-f fichier` : précise le fichier résultat contenant la zone signée. Par défaut elle ajoute `.signed` au nom du fichier de la zone ;
- `-o origine` : spécifie le nom de base qui sera concaténé à tous les enregistrements non totalement qualifiés (*zone origin*). Si elle n'est pas spécifiée, le nom de fichier zone sera utilisé comme origine ;
- `-r fichier-aléatoire` : spécifie la source de génération des nombres aléatoires. Si le système d'exploitation ne fournit pas un fichier périphérique `/dev/random` ou un équivalent, le clavier sera utilisé comme source.

EXEMPLE

Dans l'exemple suivant, la commande `dnssec-signzone` signe la zone « exemple.com » avec la clé ZSK (*Zone Signing KEY*) : `Kexemple.com.+005+27270` et la clé KSK (Key *Signing KEY*) : `Kexemple.com.+005+07082`. Les résultats (zone signée) seront enregistrés dans le fichier `db.exemple.com.signed`.

```
# dnssec-signzone -o exemple.com \
    -f db.exemple.com.signed \
    -k Kexemple.com.+005+07082.private \
    /etc/bind/db.exemple.com\
    Kexemple.com.+005+27270.private
exemple.com.signed
#
```

8. Sécurité

Dans un système informatique ouvert sur Internet, le service DNS est un service critique. En effet :

- la non disponibilité du serveur DNS entraînera la non disponibilité des serveurs web et de messagerie ;
- une intrusion dans le serveur DNS, entraîne la collecte d'informations privées telles que les adresses IP d'autre serveurs, des postes clients, des commutateurs, etc ;
- si les informations du DNS sont empoisonnées (modifiées), un pirate peut rediriger les internautes vers ses serveurs au lieu des serveurs réels. Le problème sera plus grave s'il y a des transactions financières.

Pour renforcer la sécurité de ce service, les opérations suivantes peuvent être appliquées :

- l'exécution du démon correspondant dans un environnement enfermé ;
- l'utilisation des listes de contrôle d'accès (ACL *: Access control List*) ;
- la signature des transactions (TSIG *: Transaction SIGnature*) et
- la signature des informations des zones (DNSSec : *DNS Security*).

8.1 Environnement enfermé

L'une des fonctionnalités utiles de BIND est la possibilité d'exécuter le démon `named` dans le contexte d'un utilisateur non privilégié (option `-u` de `named`). La plupart des distributions utilisent, par défaut, cette fonctionnalité. Dans la distribution Debian, par exemple, le démon `named` est exécuté dans le contexte de l'utilisateur `bind` créé spécialement pour cette fonctionnalité.

Une autre fonctionnalité intéressante de BIND est la possibilité d'exécuter le démon `named` dans un environnement enfermé (*chrooted environnement*). Cette fonctionnalité permet de limiter les dommages dans le cas où le serveur est compromis. L'option `-t répertoire` informe `named` qu'il sera exécuté dans un environnement enfermé. La majorité des distributions n'implémentent pas, par défaut, cette fonctionnalité.

PROCEDURE

La procédure qui suit illustre la mise en place d'un environnement enfermé de BIND sous

la distribution Debian Squeeze. Il est très facile d'adapter cette procédure pour une autre distribution.

- Arrêt du service BIND

```
# invoke-rc.d bind9 stop
```

- Création de l'arborescence. Le répertoire racine de l'environnement enfermé est `/chroot/bind`.

```
# mkdir -p /chroot/bind
# cd /chroot/bind
# mkdir -p dev etc var/run var/cache
```

- Copie des fichiers utiles pour l'exécution de `named`.

```
# cp -a /dev/log /dev/null /dev/random ./dev/
# cp -p /etc/localtime ./etc/
# cp -a /var/cache/bind ./var/cache/
# cp -a /var/run/named ./var/run/
# mv /etc/bind ./etc/
# ln -s /chroot/bind/etc/bind /etc/bind
```

- Changement du répertoire personnel de l'utilisateur bind.

```
# usermod -d /chroot/bind/ bind
```

- Ajout de l'option `-t /chroot/bind` aux options de démarrage de démon `named` précisant que la racine de l'environnement enfermé est `/chroot/bind`.

```
# cat /etc/default/bind9
  ...
  OPTIONS="-u bind"
# nano /etc/default/bind9
# # ajouter -t /chroot/bind aux options de lancement de amed
# cat /etc/default/bind9
  ...
  OPTIONS="-u bind -t /chroot/bind"
```

- Configuration du démon `rsyslogd` pour accepter les messages de journalisation de `named` s'exécutant dans son nouvel environnement enfermé.

```
# echo '$AddUnixListenSocket /chroot/bind/dev/log' > \
       /etc/rsyslog.d/bind-chroot.conf
# invoke-rc.d rsyslog restart # ou: restart rsyslog
```

– Démarrage du service bind9

```
# invoke-rc.d bind9 start
 * Starting domain name service... bind9                              [ OK ]
```

– Vérification de l'exécution de `named` dans l'environnement enfermé de racine `/chroot/bind`

```
# ps -ef | grep named
bind     26855     ...  /usr/sbin/named -u bind -t /chroot/bind
...
# ls -l /proc/26855/root
lrwxrwxrwx 1 bind bind 0 2011-09-19 17:18 /proc/26855/root -> /chroot/bind
#
```

8.2 Listes de contrôle d'accès

L'utilisation des listes de contrôle d'accès (ACL *Access Control List*) permet de simplifier et d'affiner le contrôle d'accès à un serveur DNS. Elles sont utilisées avec les instructions `allow-notify`, `allow-query`, `allow-query-on`, `allow-recursion`, `allow-recursion-on`, `blackhole`, `allow-transfer`, etc.

Les noms ACL suivants sont prédéfinis dans `named` :

– `none` : ne désigne aucun hôte ;

– `any` : désigne tous les hôtes ;

– `localhost` : désigne toutes les adresses IP de l'hôte exécutant `named` ;

– `localnets` : désigne les adresses IP de tous les réseaux IP auxquels l'hôte exécutant `named` appartient.

EXEMPLE

La portion du fichier `named.conf` suivante utilise les ACL pour contrôler l'accès à un serveur DNS de manière à n'autoriser les résolutions récursives que pour les postes du réseau local.

```
acl "lan" { 192.168.1.0/24; };
options {
   ...
   allow-query     { "lan"; };
   allow-recursion { "lan"; };
   ...
};
zone "exemple.com" {
   type master;
```

```
    file "/etc/bind/db.exemple.com";
    allow-query { any; };
};
```

8.3 Sécurisation des transactions avec TSIG

La fonctionnalité TSIG (*Transaction SIGnature*) permet de crypter les données échangées lors d'un transfert de zone entre le serveur DNS maître d'une zone et ses esclaves. Elle est aussi utilisée pour les mises à jour dynamiques (*dynamic updates*) des enregistrements d'une zone.

Les transactions utilisant TSIG sont horodatées. Il est donc nécessaire que les horloges des entités communicantes soient synchronisées. De préférence, elles doivent être synchronisées auprès d'un serveur de temps commun.

PROCEDURE

La procédure qui suit énumère les étapes de mise en place des transactions sécurisées pour le transfert de la zone « exemple.com » entre le serveur DNS maître « dns1 » et un serveur esclave « dns2 ».

- Génération d'un secret : la commande qui suit génère une clé symétrique de nom `tsig-dns1-dns2`, de type `HOST` en utilisant l'algorithme `HMAC-MD5` de taille 512 bits.

```
# dnssec-keygen -a HMAC-MD5 -b 512 -n HOST tsig-dns1-dns2
```

  Cette commande crée les fichiers `Ktsig-dns1-dns2.+157+06761.[key,private]` contenant le même secret. Néanmoins, dans certains cas d'utilisation avec `dig`, la présence de ces deux fichiers est nécessaire.

- Prise en charge de la clé : sur les deux serveurs, le fichier `tsig-dns1-dns2` contenant la clé symétrique doit être créé. Le contenu du fichier est le suivant :

```
key tsig-dns1-dns2 {
        algorithm hmac-md5;
        secret "Dey4YV...yzREUQ==";
};
```

- Re-configuration du maître : le fichier de configuration `named.conf` du « dns1 » doit être modifié pour y inclure le chemin d'accès à la clé symétrique.

```
include "/chemin/tsig-dns1-dns2";
zone "exemple.com" {
        type master;
        file "/etc/bind/db.exemple.com";
        allow-transfer {
            key tsig-dns1-dns2;
        };
```

```
};
```

– Re-configuration de l'esclave : le fichier de configuration `named.conf` du « dns2 » doit être modifié pour y inclure le chemin d'accès à la clé symétrique.

```
include "/chemin/tsig-dns1-dns2";
server 192.168.1.1  {
       keys { tsig-dns1-dns2; };
};
zone "exemple.com" {
       type slave;
       file "/etc/bind/db.exemple.com";
       masters { 192.168.1.1; };
};
```

– Test du transfert : la commande `dig` suivante interroge « dns1 » pour un transfert de zone.

```
$ dig @dns1.exemple.com \
    -k Ktsig-dns1-dns2.+157+06761.private \
    exemple.com AXFR
```

Le message « `NOERROR` » est retourné s'il n'y a pas d'erreur. En cas d'erreur, le message « `Transfer failed. BADSIG` » est retourné pour une erreur de signature et le message « `Transfer failed. BADKEY` » pour une erreur de nom de la clé.

8.4 Sécurisation des données avec DNSSEC

La fonctionnalité DNSSEC (*DNS SECurity*) permet de signer des zones. Ceci permet de vérifier que les informations relatives à une zone spécifique proviennent du serveur DNS qui les a signées avec une clé de zone (*ZSK : Zone Signing Key*).

PROCEDURE

La procédure qui suit décrit les étapes de signature de la zone « exemple.com » :

– Génération des clés : les commandes qui suivent génèrent deux paires de clés asymétriques l'une pour la signature de la zone (ZSK) « exemple.com » et l'autre pour la signature de la clé (*KSK*) de la zone. Le contenu des fichiers `.key` (clés publiques) générés est ajouté à la fin du fichier zone « exemple.com ».

```
# dnssec-keygen -q -a RSASHA1 -b 512 -n ZONE exemple.com
Kexemple.com.+005+50823
# cat Kexemple.com.+005+50823.key >> /etc/bind/db.exemple.com
#
# dnssec-keygen -q -a RSASHA1 -b 512 -f KSK -n ZONE exemple.com
Kexemple.com.+005+53281
```

```
# cat Kexemple.com.+005+53281.key >> /etc/bind/exemple.com
```

- Signature de la zone : la commande `dnssec-signzone` qui suit signe la zone « exemple.com » avec la clé ZSK `Kexemple.com.+005+50823` et la clé KSK `Kexemple.com.+005+53281`. Les résultats (zone signée) seront enregistrés dans le fichier `db.exemple.com.signed`.

```
# dnssec-signzone -o exemple.com \
    -f /etc/bind/db.exemple.com.signed \
    -k Kexemple.com.+005+53281.private \
    /etc/bind/db.exemple.com\
    Kexemple.com.+005+50823.private
```

- Chargement de la zone : le fichier `named.conf` doit être modifié pour remplacer le fichier `/etc/bind/db.exemple.com` par `/etc/bind/db.exemple.com.signed`.

```
zone "exemple.com" {
        type master;
        file "/etc/bind/db.exemple.com.signed";
        ...
};
```

Le démon `named` doit recharger le fichier de configuration `named.conf` puisqu'il vient d'être modifié.

- Test de la zone signée : la commande `dig` suivante interroge « dns1 » en activant le bit DO (*Dnssec OK*) à travers l'option `+dnssec`.

```
$ dig @dns1.exemple.com www.exemple.com a +dnssec
```

9. Exercices

1. **Quelle est l'instruction du fichier named.conf autorisant la mise à jour d'un serveur esclave ?**
   - ❑ A. `allow-update { ADRESSE_IP; };`
   - ❑ B. `allow-access { ADRESSE_IP; };`
   - ❑ C. `allow-transfer { ADRESSE_IP; };`
2. **Qu'est ce qu'il faut faire après l'ajout d'un nouvel hôte à une zone ?**
   - ❑ A. Mettre à jour les serveurs esclaves.
   - ❑ B. Autoriser la résolution de noms pour cet hôte.

- ❑ C. Incrémenter le numéro de série de la zone.

3. Quelle instruction configure un serveur DNS pour qu'il transfère toutes les requêtes à un autre serveur DNS d'adresse IP 172.16.161.254 ?

- ❑ A. `forward { 172.16.1.254; };`
- ❑ B. `forwarder { 172.16.1.254; };`
- ❑ C. `forward-only { 172.16.1.254; };`
- ❑ D. `forward only; forwarders { 172.16.1.254; };`

4. Quelle est la commande qui fournit plus d'information à propos d'une requête DNS ?

- ❑ A. `host`
- ❑ B. `nslookup`
- ❑ C. `dig`
- ❑ D. `named-checkconf`
- ❑ E. `named-checkzone`

5. Quelle est l'instruction qui bloque les requêtes DNS envoyées par un ou plusieurs hôtes spécifiés ?

- ❑ A. `block { ADRESSE_IP; };`
- ❑ B. `deny-req { ADRESSE_IP; };`
- ❑ C. `blackhole { ADRESSE_IP; };`

Index des mots clés

Table des figures et des tableaux

Les auteurs

Zied Bouziri (Tunisie) : enseignant depuis 2003 au département Informatique de l'Institut supérieur des études technologiques de Charguia. Il est ingénieur en informatique, diplômé de l'École nationale des sciences de l'informatique de Tunis (ENSI). Entre 1999 et 2003, il a été ingénieur conception et développement au département recherche et développement chez Alcatel.

Hedi Magroun (Tunisie) : enseignant depuis 1998 au département informatique de l'ISET de Sousse et directeur de son Centre des Ressources Informatiques depuis 2010. Il est ingénieur principal en informatique, diplômé de l'École nationale des sciences de l'informatique de Tunis (ENSI). Il est certifié LPIC2, RHCE et CCNA.

Véronique Pierre (France) : documentaliste et éditrice scientifique multimédia.