Collection Savoirs francophones
Série Technologies de l'information

Préparation à l'examen 102
pour la certification de l'Institut professionnel de Linux, niveau junior (LPIC-1)

Zied Bouziri (2e édition)

Niry H. Andriambelo, Andrei Boyanov, Nicolas Larrousse (1re édition)

Pour citer cet ouvrage

Z. Bouziri, N. H. Andriambelo, A. Boyanov, N. Larrousse (2010). *Préparation à l'examen 102 pour la certification de l'Institut professionnel de Linux, niveau junior (LPIC-1)*. Agence universitaire de la Francophonie, Paris. Disponible sur le Web : www.lpi-francophonie.org/spip.php?article234.

Première édition :

N. H. Andriambelo, A. Boyanov, N. Larrousse (2007). *Institut professionnel de Linux. Support de formation LPIC 102.* Agence universitaire de la Francophonie, Éditions des archives contemporaines, Paris. 199 p. ISBN 978-2-914610-51-3

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 Inc.)* – permet de valider les connaissances et l'expérience des administrateurs systèmes et réseaux qui travaillent avec le système d'exploitation GNU/Linux. Le niveau 1 – *Junior Level Linux Professional* – abrégé en « **LPIC-1** », concerne l'installation d'une machine, sa connexion à un réseau ainsi que les tâches de maintenance simple.

Cet ouvrage permet de préparer l'**examen 102**, qui constitue le second examen à passer pour obtenir la **LPIC-1**.

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*).

Dans son édition 2010 – 1re édition en 2007 – il prend en compte les objectifs détaillés de l'examen 102 mis à jour en avril 2009 :

- version originale sur le site du LPI *Exam 102 : Detailed Objectives* www.lpi.org/eng/certification/the_lpic_program/lpic_1/exam_102_detailed_objectives
- traduction en français sur le site du LPI Francophonie : www.lpi-francophonie.org/spip.php?rubrique19

Chaque chapitre traite d'un sujet du programme de certification. En annexe, un exemple d'examen permet de vérifier ses connaissances et de se familiariser avec le style des questions posées à l'examen.

Pourquoi une certification Linux ?

Les objectifs de la certification créée par le LPI Inc. 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 Inc., un organisme neutre fondé par la communauté du logiciel libre.

Le LPI Inc. 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 Inc. 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 Inc. 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 Inc. 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. Programmation bash

Objectifs	⇒ Connaître Bash. ⇒ Connaître les bases de la programmation du shell. ⇒ Maîtriser le principe des scripts shell pour comprendre les scripts système. ⇒ Savoir programmer des expressions logiques et arithmétiques ainsi que des boucles.
Points importants	Automatiser les tâches administratives avec des scripts bash est indispensable et très efficace. Une bonne utilisation de Linux est indissociable des scripts
Mots clés	$*, $#, $0, $1, $2, $!, $$, $?, ~/.profile, ~/.bashrc, ~/.bash_logout, ~/.inputrc, /bin/bash, /etc/bash_logout, /etc/bashrc, /etc/inputrc, /etc/profile, bash, case, do, done, else, env, esac, export, expr, fi, for, if, PATH, select, set, test, then, unset, while

Bash est le shell de GNU/Linux, un shell étant l'interface utilisateur d'un système d'exploitation. Il est basé sur le *Bourne Shell* d'Unix, d'où son nom, qui est l'acronyme de *Bourne-again shell*.

A. L'environnement bash

a) Les variables en bash

Pour affecter un contenu à une variable, on utilise la commande = de la manière suivante :

```
ma_variable='Ne pas oublier le chat'
```

Attention à ne pas mettre d'espace avant et après le signe « = » !!!

Pour faire référence au contenu d'une variable, on la préfixe par le signe « $ ».

```
echo $ma_variable
   Ne pas oublier le chat
```

Pour effacer le contenu d'une variable, on utilise la commande unset.

```
unset ma_variable
```

Le shell utilise des variables pour tenir compte des paramètres de configuration spécifiques des utilisateurs appelés **variables d'environnement**.

Les variables HOME, DISPLAY, PWD en font partie.

```
echo $HOME
/usr/home/nicolas
```

Lors de l'utilisation d'un programme, le shell utilise la variable d'environnement PATH pour retrouver le chemin d'accès à ce programme. On peut afficher le contenu de cette variable par la commande echo :

```
echo $PATH
/usr/local/bin:/bin:/usr/bin:/usr/X11R6/bin:/usr/games:/usr/X11R
6/bin:/usr/home/nicolas:/:/usr/bsd:/usr/sbin:/usr/local/bin:/usr
/bin/X11
```

Pour qu'une variable soit visible de tous les shells (donc de toutes les commandes lancées depuis le shell courant), il faut l'exporter par la commande export.

```
export MANPATH="/usr/share/docs"
```

Lors du démarrage d'une session shell, la plupart des variables d'environnement sont initialisées et exportées à partir des différents fichiers de configuration, tels que les fichiers bashrc et profile.

La commande env permet de démarrer un shell avec des variables d'environnement déjà positionnées à des valeurs données. Ces variables d'environnement ont une durée de vie égale à celle du shell démarré.

Par exemple, pour lancer la commande `ma_commande` en positionnant la variable d'environnement ma_variable à la valeur « ma_valeur » :

```
env ma_variable=ma_valeur ma_commande
```

La durée de vie et la visibilité de ma_variable sont limitées à la durée d'exécution de `ma_commande`.

b) Les fichiers de configuration

Il y a plusieurs types de fichiers de configuration, ceux qui sont lus au moment de la connexion (*login*) et ceux qui sont lus à chaque lancement d'un shell.

Les fichiers lus au moment de la connexion au système sont :

- /etc/profile, commun à tous les utilisateurs (s'il existe) ;
- ~/.bash_profile ou éventuellement ~/.bash_login ou ~/.profile/, spécifiques à chaque utilisateur.

Ils servent généralement à décrire l'environnement de l'utilisateur.

Les fichiers lus à chaque lancement de shell sont :

- /etc/bashrc commun à tous les utilisateurs (s'il existe) ;
- ~/.bashrc spécifique à chaque utilisateur.

c) La famille bashrc

Les fichiers de la famille « bashrc » sont lus chaque fois qu'un shell est lancé (e.g. commandes `xterm` ou `bash`).

Ils servent généralement à stocker les alias et les fonctions utilisés communément.

On peut démarrer le shell bash avec différentes options lui indiquant les fichiers de configuration à lire au démarrage :

- `bash -login` : force la lecture des fichiers de connexion famille « profile » ;
- `bash -norc` : pas de lecture des fichiers « bashrc » ;
- `bash -noprofile` : pas de lecture des fichiers « profile ».

Attention, toutes les nouvelles sessions de bash, puisqu'elles sont des processus fils, héritent des variables définies dans les fichiers « profile » lors de la connexion.

Le fichier /etc/bash_logout, s'il est présent, est exécuté à la fin de la session bash de tous les utilisateurs.

Le fichier ~/.bash_logout, s'il est présent, est exécuté à la fin de la session bash spécifique à un utilisateur.

Le fichier inputrc permet de reconfigurer le clavier pour ajuster le fonctionnement des touches comme par exemple la touche d'effacement arrière. Le fichier général est dans /etc/inputrc, les fichiers personnels dans ~/.inputrc.

B. Les scripts

a) Script shell

Un script shell est une liste d'instructions contenues dans un fichier.

```
#!/bin/bash
# Un petit script mon_script
echo 'Ne pas oublier le chat'
```

Pour pouvoir exécuter ces instructions, deux conditions doivent être remplies :

- la première ligne doit contenir `#!/bin/bash` (pour un shell script utilisant bash) ;
- le fichier doit être exécutable (e.g. en utilisant la commande `chmod +x`) et lisible (e.g. avec les permissions `755`)

```
chmod +x mon_script
./mon_script
```

Si toutes ces conditions ne sont pas remplies, il est toujours possible de forcer l'exécution du script avec la commande `bash`.

```
bash mon_script
```

b) Passage de paramètres à un script

Les variables passées au script sur la ligne de commande sont accessibles dans le script par les variables réservées $1 pour le premier argument, $2 pour le deuxième et ainsi de suite.

À noter qu'il existe un opérateur **shift** qui décale ces paramètres : la valeur contenue dans $2 passe dans $1, celle contenue dans $3 passe dans $2 et ainsi de suite. Cela permet essentiellement d'utiliser plus de 9 paramètres en entrée.

D'autres variables réservées sont accessibles à l'intérieur d'un script :

- $0 : nom du script ;
- $* : liste des paramètres ;
- $# : nombre de paramètres ;
- $$: numéro du processus en cours d'exécution ;
- $? : valeur de retour de la dernière commande.

```
#!/bin/bash
# un autre script
echo "mon script est $0"
echo "il y a eu $# paramètres en entrée"
echo "le premier paramètre est $1"
```

C. Les expressions logiques

Les expressions logiques sont évaluées à l'aide de la fonction `test`, qui peut également s'écrire `[ ]`.

Le résultat de l'évaluation est stocké dans la variable $? qui contient :

- 0 si le résultat est vrai ;
- une valeur différente de 0 si le résultat est faux.

Pour vérifier si le fichier /bin/bash existe :

```
test -f /bin/bash
```

ou

```
[ -f /bin/bash ]
```

Pour vérifier si le fichier ~/bin/mon_script est exécutable :

```
test -x ~/bin/mon_script
```

ou

```
[ -x /bin/mon_script ]
```

Les expressions logiques peuvent être combinées par les opérateurs logiques `&&` (ET/AND) et `||` (OU/OR). Il est également possible d'utiliser les connecteurs `-a` (ET/AND) et `-o` (OU/OR).

Pour vérifier si les fichiers /etc/profile et /etc/bashrc existent :

```
test -f /etc/profile -a test -f /etc/bashrc
```

ou

```
test -f /etc/profile && test -f /etc/bashrc
```

ou

```
[ -f /etc/profile -a -f /etc/bashrc ]
```

Quelques options de la commande `test` :

- `-f` : vérifie si le fichier est un fichier standard ;
- `-d` : vérifie si le fichier est un répertoire ;
- `-b` : vérifie si le fichier est de type bloc ;
- `-e` : vérifie si le fichier existe indépendamment de son type ;
- `-r` : vérifie si le fichier est lisible ;
- `-w` : vérifie si le fichier est inscriptible ;
- `-x` : vérifie si le fichier est exécutable.

D'autres options seront données pour le traitement spécifique des nombres.

D. Les boucles

a) Test `if`

Cette boucle sert pour les tests et branchements.

Syntaxe (la partie « else... » est optionnelle) :

```
if <condition> then
   <commande1>
   <commande2>
   ...
else
   <commande1>
   <commande2>
   ...
fi
```

Pour tester l'existence du fichier « monfichier.txt » :

```
#!/bin/sh
if test -f monfichier.txt then
   echo " le fichier existe "
fi
```

b) Test `case`

Ce test permet de spécifier les commandes à exécuter pour chacune des valeurs prises par la variable passée en argument.

Syntaxe :

```
case <variable> in
   valeur1) commande1 ;;
   valeur2) commande2 ;;
   valeur3) commande3 ;;
   ...
esac
```

Pour tester la valeur du premier paramètre :

```
#!/bin/sh
case $1 in
   1) echo " un ";;
   2) echo " deux ";;
   3) echo " trois ";;
esac
```

c) Boucle `for`

Cette boucle sert pour répéter les traitements un nombre de fois connu.

Syntaxe :

```
for <variable> in <liste> do
   commande1
   commande2
   commande3
   ...
done
```

Pour afficher les jours de la semaine :

```
for jour in lundi mardi mercredi jeudi vendredi samedi dimanche
do
   echo $jour
done
```

d) Boucles `while`

Cette boucle sert pour répéter les traitements un nombre de fois inconnu *a priori*. Le test de continuité se fait au début de la boucle. La boucle continue tant que la condition est vraie.

Syntaxe :

```
while <condition> do
   <commande1>
   <commande2>
   ...
done
```

Pour faire le test de lecture d'une valeur jusqu'à ce que l'utilisateur entre la valeur « 1 » :

```
#!/bin/sh
i=0
while [ $i -ne "1" ] do
   read i
done
```

Il existe une autre boucle, qui utilise le mot clé `until` à la place de `while`. Elle diffère dans le traitement de la condition : la boucle est exécutée jusqu'à ce que la condition soit vraie. Le test est donc effectué en fin de boucle et la boucle est toujours exécutée au moins une fois.

E. Les paramètres d'entrée d'un script

La lecture d'une valeur peut se faire par le passage de paramètres sur la ligne de commande (*cf.* précédemment l'utilisation des variables $1, $2, etc.) ou en lisant une entrée au clavier depuis le script avec la commande `read`.

Pour lire le nom d'une personne dans la variable « nom » et afficher son contenu :

```
#!/bin/sh
echo 'Entrez votre nom'
read nom
echo '' Vous vous appelez $nom ''
```

L'entrée au clavier lue avec la commande `read` peut ensuite être traitée avec la commande `case` vue précédemment.

```
#!/bin/sh
echo ''entrez votre choix''
read choix
case $choix in
   1) echo " menu un ";;
   2) echo " menu deux ";;
   3) echo " menu trois ";
```

```
esac
```

La commande `select` est utilisée pour demander à un utilisateur de choisir une valeur et une seule dans une liste de valeurs prédéfinies. L'invite à afficher est à indiquer dans la variable prédéfinie PS3. La commande `select` a deux arguments : la liste des valeurs proposées, et une variable dans laquelle sera stockée la valeur choisie. Un numéro séquentiel est automatiquement attribué à chaque valeur proposée. Le numéro de la valeur choisie sera stocké dans la variable prédéfinie REPLY.

Syntaxe :

```
select <variable> in <liste de choix> »
```

On peut sortir de la boucle avec la commande `break`.

Exemple utilisant les commandes `select` et `break` :

```
#/bin/bash
PS3="Entrez le numéro de votre commande -> "
echo "Que désirez-vous boire ?"
select biere in "Rien, merci" "Skoll" "THB" "Dodo"
do
   echo "Vous avez fait le choix numéro $REPLY..."
   if [ "$REPLY" -eq 1 ] then
       echo "Au revoir!"
       break
   else
       echo "Votre $biere est servie."
   fi
   echo
done
```

F. Calculs

Il est possible de comparer des nombres en utilisant la commande `test`, vue précédemment, avec les options suivantes :

- `-lt` pour « inférieur à » (<) ;
- `-gt` pour « supérieur à » (>) ;
- `-le` pour « inférieur ou égal à » (<=) ;
- `-ge` pour « supérieur ou égal à » (>=) ;
- `-eq` pour « égal à » (=) ;
- `-ne` pour « différent de » (!=).

Pour tester le nombre de paramètres en entrée :

```
#!/bin/bash
if [ $# -eq 0 ] then
   echo " Vous n'avez entré aucun paramètre"
fi
```

La commande `expr` permet d'effectuer les quatre opérations arithmétiques de base, avec les opérateurs suivants :

- + pour l'addition ;
- - pour la soustraction ;
- * pour la multiplication ;
- / pour la division.

Par exemple, la commande `expr 1 + 2` renvoie « 3 ».

Pour afficher une table de multiplication :

```
#!/bin/bash
for i in 1 2 3 4 5 6 7 8 9 do
   expr $i \* $1
done
```

Pour compter jusqu'à 100 :

```
#!/bin/bash
i=0
while [ $i -ne 100 ] do
   i=`expr $i + 1`
   echo $i
done
```

On peut également écrire `expr <expression>` sous la forme `$((<expression>))`

```
#!/bin/bash
i=0
while [ $i -ne 100 ] do
   #i=`expr $i + 1`
   i=$(($i+1))
   echo $i
done
```

G. Exercices

1. **Dans le répertoire personnel de l'utilisateur, parmi ces couples de fichiers, lesquels sont utilisés pour configurer l'environnement bash ?**

   - ❑ bash et .bashrc
   - ❑ bashrc et bash_conf
   - ❑ bashrc et bashprofile
   - ❑ bashrc et .bash_profile
   - ❑ bash.conf et .bash_profile

2. **Quel fichier doit modifier l'utilisateur dans son répertoire personnel pour configurer la variable d'environnement PATH ? Donnez seulement le nom du fichier, sans le chemin d'accès.**

Chapitre 2. Gestion des données avec SQL

Objectifs	⇒ Savoir interroger une base de données et manipuler les données en utilisant le langage SQL. ⇒ Savoir écrire des requêtes de jointure sur plusieurs tables et utiliser les sous-requêtes.
Points importants	SQL (*Structured Query Language*) est un langage de gestion de bases de données relationnelles. Il a été conçu par IBM dans les années 70. Il est devenu le langage standard des systèmes de gestion de bases de données relationnelles.
Mots clés	delete, from, group by, insert, join, order by, select, update, where

A. Introduction

Le langage SQL est à la fois :

- un langage de manipulation de données (LMD) qui permet d'insérer, modifier ou supprimer des données ;
- un langage d'interrogation de données ;
- un langage de définition de données (LDD) qui permet de créer, modifier et supprimer des tables dans une base de données relationnelle ;
- un langage de contrôle de l'accès aux données (LCD) qui permet de définir des règles d'accès aux données par les utilisateurs.

Dans ce chapitre, l'accent est mis sur les requêtes SQL permettant de manipuler et d'interroger des bases de données. On utilise pour les exemples une base de données simple nommée « Bibliothèque » et constituée de deux tables, « Livre » et « Editeur ».

Voici leur description, affichée grâce à la commande `describe` :

```
mysql> describe Livre;
+-------------+--------------+------+-----+---------+
| Field       | Type         | Null | Key | Default |
+-------------+--------------+------+-----+---------+
| ISBN        | varchar(50)  | NO   | PR  | NULL    |
| Titre       | varchar(100) | NO   |     | NULL    |
| Prix        | float        | YES  |     | NULL    |
| Id_Editeur  | int(11)      | NO   | MUL | NULL    |
+-------------+--------------+------+-----+---------+
mysql> describe Editeur;
+-------------+--------------+------+-----+---------+
| Field       | Type         | Null | Key | Default |
+-------------+--------------+------+-----+---------+
| Id_Editeur  | int(11)      | NO   | PRI | NULL    |
| Nom         | varchar(100) | NO   |     | NULL    |
| Telephone   | varchar(30)  | YES  |     | NULL    |
+-------------+--------------+------+-----+---------+
```

B. Manipulation des données

a) Insertion des données

L'ordre `INSERT` permet d'ajouter une ou plusieurs lignes à une table.

Syntaxe :

```
INSERT INTO NomTable
[(Colonne1,Colonne2,Colonne3,...)]
VALUES (Valeur1,Valeur2,Valeur3,...),
(Valeur1,Valeur2,Valeur3,...), ... ;
```

Lorsque l'ajout de lignes concerne toutes les colonnes de la table, l'énumération des colonnes est facultative.

Pour ajouter l'éditeur VUIBERT dans la table Editeur :

```
INSERT INTO Editeur
(Id_Editeur,Nom)
VALUES (6,'VUIBERT');
```

Pour ajouter plusieurs lignes dans les tables Editeur puis Livre :

```
INSERT INTO Editeur
VALUES (1,'PEARSON EDUCATION','6666-333'),
```

```
        (2,'CAMPUS-DUNOD','999-666'),
        (3,'O\'REILLEY','6666-3333'),
        (4,'EYROLLES','5555-999'),
        (5,'DUNOD','3333-999');

INSERT INTO Livre
VALUES ('9782744071768','ARCHITECTURE DE L\'ORDINATEUR',50.6,1),
        ('9782100518067','TECHNOLOGIE DES ORDINATEURS ET DES
   RESEAUX',100.9,2),
        ('9782841772513','MAC OS X',70.9,3),
        ('9782212122732','SHELLS LINUX ET UNIX',60.3,4),
        ('9780596005283','LPI Linux certification in a
    Nutshell',100.3,3),
        ('9782744071782','ALGORITHMIQUE EN JAVA 5',90.3,1),
        ('9782100518319','EJB 3 - DES CONCEPTS: L\'ECRITURE DU
   CODE',150.3,5)
;
```

b) Mise à jour des données

L'ordre `UPDATE` permet de modifier des lignes dans une table, la clause `SET` précise la modification à effectuer. Il s'agit d'une affectation d'une valeur à une colonne grâce à l'opérateur =, suivi d'une expression algébrique, d'une constante ou du résultat provenant d'un ordre `SELECT`.

Les lignes sur lesquelles la mise à jour a lieu sont définies grâce à la clause `WHERE`.

Syntaxe :

```
UPDATE NomTable
SET NomColonne = Valeur_Ou_Expression
[WHERE qualification] ;
```

Pour doubler les prix des livres de l'éditeur O'REILLY, dont l'identifiant est « 3 » :

```
UPDATE Livre SET prix=prix*2
WHERE Id_Editeur=3;
```

c) Suppression des données

L'ordre `DELETE` permet de supprimer des données dans une table. La clause `FROM` précise la table à traiter et la clause `WHERE` les lignes à supprimer.

Syntaxe :

```
DELETE
```

```
FROM NomTable
[WHERE qualification];
```

Pour supprimer toutes les lignes de la table Livre :

```
DELETE FROM Livre;
```

C. Interrogation

L'ordre `SELECT` permet d'extraire des données d'une base.

Syntaxe :

```
SELECT [DISTINCT] coli, colj, ...
FROM table1, table2, ...
[WHERE critères de sélection]
[GROUP BY coli, colj, ... HAVING prédicat]
[ORDER BY coli [DESC], colj[DESC], ...];
```

Dans une première étape, on ne conserve que les lignes qui répondent aux `critères de sélection`.

Puis on ne conserve dans le résultat obtenu que les colonnes dont la liste est donnée par `coli, colj, ...` Pour sélectionner l'ensemble des colonnes d'une table il suffit de remplacer la liste de ces colonnes par `*`.

L'option `DISTINCT` est utilisée afin de ne conserver que des lignes distinctes.

La clause `GROUP BY exp1, exp2, ...` groupe en une seule ligne toutes les lignes pour lesquelles `exp1, exp2,...` ont la même valeur (voir détails plus loin).

On peut aussi ordonner les lignes en les triant en fonction de la valeur d'une colonne : `ORDER BY coli`. Par défaut, le tri est ascendant. L'option `DESC` le rend descendant. Il est aussi possible d'appliquer un deuxième critère de tri `colj`, puis un troisième etc.

Pour afficher toutes les colonnes de la table Editeur :

```
SELECT * FROM Editeur;
```

a) Les fonctions d'agrégation

Avec `SELECT` et `HAVING` on peut utiliser les fonctions d'agrégation qui permettent de faire des statistiques sur les colonnes (*tableau 1*).

Tableau 1. Fonctions d'agrégation

Fonction	Description
AVG	moyenne
SUM	somme
MIN	plus petite des valeurs
MAX	plus grande des valeurs
VARIANCE	variance
STDDEV	écart type
COUNT(*)	nombre de lignes
COUNT(col)	nombre de valeurs non nulles de la colonne

Pour déterminer le nombre de lignes dans la table Livre :

```
SELECT COUNT(*) FROM Livre;
```

Pour déterminer le prix du livre le plus cher :

```
SELECT MAX(Prix) FROM Livre;
```

Pour afficher le nombre de livres par maison d'édition :

```
SELECT Livre.Id_Editeur, nom AS 'Maison d\'édition', COUNT(*)
FROM Livre, Editeur
WHERE Livre.id_Editeur=Editeur.Id_Editeur
GROUP BY Id_Editeur;
+------------+-------------------+----------+
| Id_Editeur | Maison d'édition  | COUNT(*) |
+------------+-------------------+----------+
| 1          | PEARSON EDUCATION | 2        |
| 2          | CAMPUS-DUNOD      | 1        |
| 3          | O'REILLEY         | 3        |
| 4          | EYROLLES          | 1        |
| 5          | DUNOD             | 1        |
+------------+-------------------+----------+
```

Dans l'exemple précédent on a utilisé l'opérateur AS pour donner un alias à la colonne nom.

En utilisant la clause HAVING, on peut appliquer une restriction sur les groupes créés grâce à la clause GROUP BY.

Pour identifier les maisons d'édition pour lesquelles la bibliothèque détient plus de trois livres :

```
SELECT Livre.Id_Editeur, nom AS 'Maison d\'édition', COUNT(*)
FROM Livre, Editeur
WHERE Livre.id_Editeur=Editeur.Id_Editeur
GROUP BY Id_Editeur HAVING COUNT(*) >= 3;
```

b) Définir des critères de sélection avec la clause WHERE

La clause WHERE définit les critères de sélection à appliquer pour sélectionner un sous-ensemble de lignes.

Elle est suivie d'une expression logique (ayant la valeur vrai ou faux) qui sera évaluée pour chaque ligne.

La clause WHERE est utilisée avec les ordres SELECT, UPDATE et DELETE avec la même syntaxe.

L'expression logique peut être exprimée par la comparaison de deux expressions ou plus au moyen d'un opérateur logique. Les trois types d'expressions à savoir arithmétique, caractère ou date, peuvent être comparés au moyen des opérateurs d'égalité ou d'ordre (=, !=, <, >, <=,>=).

Pour afficher par ordre de prix décroissant les informations sur les livres dont le prix est supérieur à 70 :

```
SELECT * FROM Livre
WHERE Prix >=70
ORDER BY Prix DESC;
```

L'opérateur LIKE sert à tester l'égalité de deux chaînes de caractères. On peut utiliser des caractères de remplacement :

- « _ » remplace exactement un seul caractère ;
- « % » remplace zéro à n caractères.

Pour afficher les informations sur les livres dont le titre contient « a » en deuxième position :

```
SELECT *
FROM Livre
WHERE Titre LIKE "_A%";
```

Pour vérifier qu'une valeur appartient à un intervalle, on peut utiliser l'opérateur BETWEEN.

Pour afficher les informations sur les livres dont le prix est compris entre 50 et 100 :

```
SELECT *
```

```
FROM Livre
WHERE Prix BETWEEN 50 AND 100;
```

Pour vérifier si une donnée appartient à une liste de valeurs on peut utiliser l'opérateur IN.

Pour vérifier si les livres *Guide des shells Unix*, *Shells Linux et Unix* et *Comprendre les shells Unix* existent dans la table Livres :

```
SELECT *
FROM Livre
WHERE Titre IN ('Guide des SHELLS UNIX', 'SHELLS LINUX et
UNIX','Comprendre les SHELLS Unix');
```

Les opérateurs logiques AND, OR et NOT peuvent être utilisés pour combiner plusieurs expressions logiques.

L'opérateur AND est prioritaire par rapport à OR.

Pour afficher les informations sur les livres dont le titre contient « Unix » et « Linux » :

```
SELECT *
FROM Livre
WHERE Titre LIKE '%Linux%' AND Titre LIKE '%Unix%' ;
```

D. Jointure

Afin de regrouper plusieurs informations issues de plusieurs tables, on utilise les jointures.

Les tables impliquées sont associées au moyen de **clés étrangères**. Dans l'exemple de la base Bibliothèque, la table Livre est associée à la table Editeur au moyen de l'attribut Id_Editeur (clé étrangère) qui se réfère à la clé primaire de la table Editeur.

Pour afficher les titres et les prix des livres (issus de la table Livres) et les noms des éditeurs associés (issus de la table Éditeurs) :

```
SELECT Titre, Prix, Nom AS Editeur
FROM Livre
JOIN Editeur ON
Editeur.id_Editeur=Livre.Id_Editeur;
```

E. Exercices

3. Quel est l'effet de la requête SQL suivante ?

Update Livre Set Prix=100 Where Titre Like '%Linux%'

- ❑ Afficher la liste des livres dont le prix est 100 et le titre contient le mot Linux.
- ❑ Affecter la valeur 100 au champ Prix et la valeur %Linux% au champ Titre des lignes de la table Livre.
- ❑ Affecter la valeur 100 au champ Prix des lignes dont le champ Titre contient le mot Linux.
- ❑ Affecter la valeur 100 au champ Prix des lignes dont le champ Titre contient le mot %Linux%.

4. Comment supprimer une table appelée « livre » à partir d'une base de données SQL ?

- ❑ rm livre
- ❑ delete table livre
- ❑ drop table livre
- ❑ delete livre

Chapitre 3. X Window

Objectifs	⇒ Comprendre les différents composants du système d'affichage X Window sous Linux. ⇒ Comprendre la logique du client-serveur utilisée pour l'affichage, la notion de « *display manager* » et de gestionnaire de fenêtres. ⇒ Connaître les fichiers de configuration des composants du système X Window et être capable de les modifier. ⇒ Connaître des outils d'accessibilité dédiés aux personnes en situation de handicap.
Points importants	Le système X Window est parfois délicat à configurer et à utiliser. Il est utile de comprendre son fonctionnement et ses divers composants pour être capable de résoudre les problèmes. Même si les versions évoluent (X, XFree, Xorg…) les bases restent les mêmes.
Mots clés	/etc/X11/xorg.conf, assistance sonore, clavier d'écran, DISPLAY, emacspeak, gdm (fichier de commande), Gestures, GOK, lecteur d'écran, kdm (fichier de commande), logiciel Braille, logiciel Daltonisme, loupes d'écran, Orca, simuler la souris avec les touches du clavier, X, xdpyinfo, xhost, xdm (fichier de commande), xwininfo.

A. Présentation

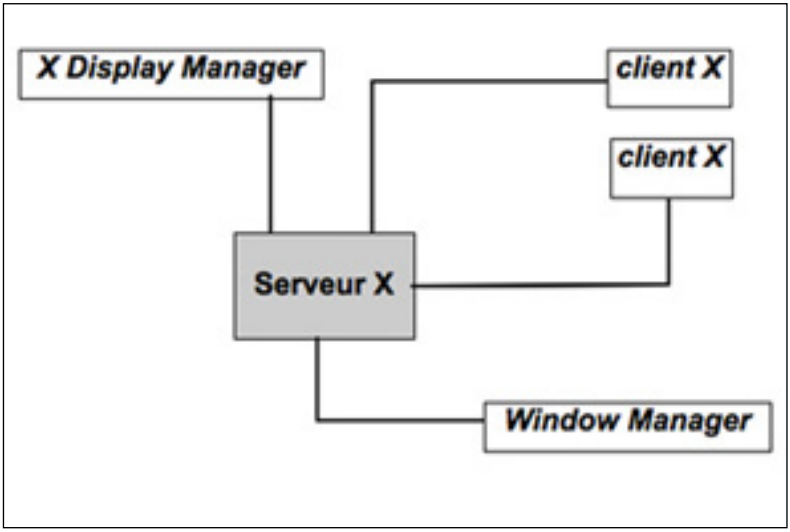
Le système X Window a été développé au *Massachusetts Institute of Technology* (MIT) dans le cadre du projet « Athena ».

Sous Linux, il existe une implémentation libre du système X Window version 11 release 6 (**X11R6**) pour les machines à base de processeurs Intel (x86) appelée **XFree86**.

Mais suite à un désaccord de licence de XFree86, un nouveau projet baptisé **Xorg** a été développé en 2004. Xorg est maintenant un projet à part entière qui évolue très rapidement par rapport à XFree. Il a été intégré par défaut à la place de XFree dans la majorité des distributions actuelles.

Comme le montre la *figure 1*, le système est constitué d'un **serveur X** auquel se connectent des clients localement ou en réseau. Il peut s'agir de clients simples, comme **xterm**, ou plus élaborés, comme les ***Window Manager*** (WM), interfaces graphiques qui apportent les fonctionnalités classiques d'une interface comme le déplacement de fenêtre, ou les ***X Display Manager*** (XDM), gestionnaires qui assurent l'authentification de l'utilisateur et le choix de l'interface graphique.

Figure 1. Le modèle client/serveur X



Le serveur X gère le matériel d'affichage, à savoir l'écran graphique, le clavier, la souris, la tablette graphique, etc. Il s'exécute sur l'ordinateur auquel est connecté ce matériel.

B. Configuration du serveur

Le fichier de configuration `/etc/X11/xorg.conf` est en général généré par des utilitaires de configuration comme `xorgconfig` ou `Xorg -configure`.

L'utilitaire `xvidtune` permet d'ajuster de façon interactive les différents modes vidéos et de générer un format de données utilisable dans le fichier /etc/X11/xorg.conf.

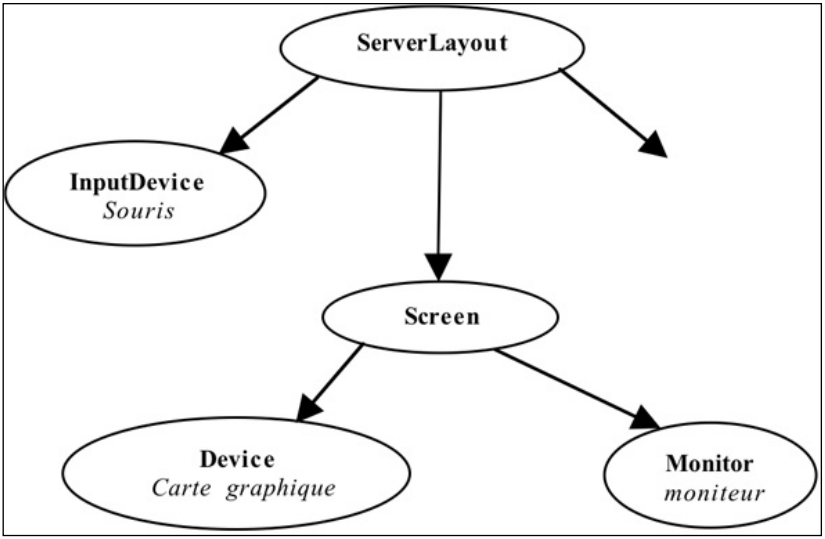
Les distributions Linux proposent des utilitaires spécifiques :

- `dpkg-reconfigure xserver-xorg` (Debian) ;
- `system-config-display` (Fedora) ;
- `sax2` (Suse).

Le fichier /etc/X11/xorg.conf est composé de plusieurs sections. Beaucoup de sections sont optionnelles, cependant quelques unes sont indispensables au bon fonctionnement du système.

La *figure 2* illustre la hiérarchie des sections du fichier xorg.conf.

Figure 2. Relations entre les sections du fichier xorg.conf



L'écran **Screen** est défini par une carte graphique, **Device**, et un moniteur, **Monitor**. Dans certains cas, une section Monitor peut dépendre d'une section **Mode**. Le clavier et la souris sont chacun définis par une section **InputDevice**. La combinaison de la section Screen et des InputDevice forme un **ServerLayout**.

S'y ajoute la section **Files** qui indique les chemins et/ou le serveur des polices de caractères.

Voici un exemple de quelques sections typiques d'un fichier xorg.conf :

- les sections **InputDevice** permettent de décrire tous les types de périphériques d'entrée. En pratique, il s'agit souvent de claviers et de souris, mais il est également possible de connecter des périphériques plus exotiques tels que les *joysticks* et les tablettes de dessin :

```
Section "InputDevice"
   Identifier "Keyboard1"
   Driver "keyboard"
   Option "XkbModel" "pc105"
   Option "XkbLayout" "fr"
   Option "XkbOptions" ""
EndSection
```

```
Section "InputDevice"
   Identifier "Mouse1"
   Driver "mouse"
   Option "Protocol" "ExplorerPS/2"
   Option "Device" "/dev/mouse"
   Option "ZAxisMapping" "6 7"
EndSection
```

- la section **Device** permet de décrire la carte vidéo et le module qui lui est associé :

```
Section "Device"
   Identifier "device1"
   VendorName "S3 Incorporated"
   BoardName "S3 Savage4"
   Driver "savage"
   VideoRam 4096
   Option "DPMS"
   # Option "no_accel" # You may enable this if there are
   timeouts when starting X
EndSection
```

- la section **Monitor** permet de décrire les caractéristiques du moniteur :

```
Section "Monitor"
   Identifier "monitor1"
   VendorName "Generic"
   ModelName "1024x768 @ 70 Hz"
   HorizSync 31.5-57.0
   VertRefresh 50-70
```

```
EndSection
```

- la section **Screen** permet de définir les paramètres d'affichage (résolution, couleurs...) :

```
Section "Screen"
   Identifier "screen1"
   Device "device1"
   Monitor "monitor1"
   DefaultColorDepth 24
   Subsection "Display"
       Depth 24
       Virtual 1024 768
   EndSubsection
EndSection
```

- la section **ServerLayout** est la combinaison des sections **Screen** et **InputDevice** :

```
Section "ServerLayout"
   Identifier "Default Layout"
   Screen "Screen1"
   InputDevice "Mouse1" "CorePointer"
   InputDevice "Keyboard1" "CoreKeyboard"
EndSection
```

- la section **Files** contient les chemins vers les ressources utilisés par le serveur X. Cela peut correspondre aux répertoires de polices de caractères, aux répertoires d'installation des modules du serveur X, ou encore aux chemins indiquant l'adresse et le port de serveurs de polices sur un réseau :

```
Section "Files"
   RgbPath "/usr/X11R6/lib/X11/rgb"
   FontPath "/usr/X11R6/lib/X11/fonts/local"
   FontPath "/usr/X11R6/lib/X11/fonts/misc"
   FontPath "/usr/X11R6/lib/X11/fonts/75dpi"
   FontPath "/usr/X11R6/lib/X11/fonts/100dpi"
   ModulePath "/usr/X11R6/lib/modules"
EndSection
```

La commande `xhost` permet de spécifier la liste des machines ou des utilisateurs autorisés à se connecter au serveur X :

```
xhost + la_belle_machine.auf.org
```

C. Les applications clientes

Les applications clientes utilisent des fichiers de configuration spécifiques nommés **Xresources**. Au lancement, le client vérifie qu'il existe un fichier .Xresource dans le répertoire *home* de l'utilisateur. Si ce n'est pas le cas, il utilise le fichier standard du système /etc/X11/Xresources. Les fichiers Xresources sont également nommés **Xdefaults** ou **.Xsession** selon les versions des systèmes et serveurs X.

En voici un extrait qui décrit la manière dont vont s'afficher les clients xterm :

```
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
! xterm (and friends)
XTerm*highlightSelection: true
! Uncomment this to use color for the bold attribute
XTerm*VT100*colorBDMode: on
XTerm*VT100*colorBD: blue
! Uncomment this to use color for underline attribute
XTerm.VT100*colorULMode: on
XTerm.VT100*underLine: off
XTerm*VT100*colorUL: magenta
! Uncomment this to display the scrollbar
XTerm*scrollBar: true
```

La variable d'environnement `DISPLAY` permet de spécifier à un client X la machine serveur à utiliser ainsi que, si plusieurs serveurs tournent sur la machine, l'instance du serveur à utiliser :

```
export DISPLAY:la_belle_machine.auf.org :0
```

Le client doit avoir le droit de se connecter sur le serveur pour effectuer cette opération (*cf.* la commande `xhosts` vue précédemment).

D. Lancer le serveur X

Une session X peut être lancée de deux manières différentes :

– à partir de la ligne de commande, avec le script `startx`. Par exemple :

```
startx
```

Ce script lance le script `xinit`, qui lancera le serveur X et le script `xinitrc`, qui lira le fichier Xresource, qui permettra le lancement de l'interface (Window Manager) ;

- en utilisant un XDM qui est lancé automatiquement au niveau d'exécution 5 (*cf.* `/etc/inittab`, le fichier de configuration du processus init, présenté dans le manuel de la même collection consacré à l'examen 101, chapitre 2 *Le démarrage de Linux*).

E. Le gestionnaire d'affichage : Display Manager

Trois *Display Manager* sont communément utilisés avec Linux :

- xdm, l'original, configuré dans /etc/X11/xdm ;
- gdm, de Gnome, configuré dans /etc/gdm ;
- kdm, de KDE, configuré dans /etc/kde/kdm.

Ils sont généralement utilisés au niveau d'exécution 5 (graphique) pour permettre à l'utilisateur de se connecter et de choisir un Window Manager.

Ils peuvent être utilisés en réseau pour offrir une interface de connexion graphique à distance en utilisant le protocole XDMCP qui n'est pas activé par défaut pour des raisons de sécurité.

Les fichiers de configuration généraux sont situés dans le répertoire /etc/X11/xdm. Le fichier de configuration de base est xdm-config, il définit les chemins des autres fichiers de configuration de XDM.

Le fichier Xressources configure l'aspect graphique de l'écran de connexion et le fichier Xsetup permet de personnaliser l'interface XDM de login, par exemple en y ajoutant des applications graphiques.

Quand l'utilisateur démarre sa session, le script `Xsession` s'exécute afin de configurer et démarrer l'environnement graphique de l'utilisateur. Ce script fait appel au script `.xsession` dans le répertoire personnel de l'utilisateur. Si ce script existe, il permet de lancer le gestionnaire de fenêtres, des barres de tâches, des *applets* et d'autres programmes propres à la session de l'utilisateur.

Exemple de fichier /etc/X11/xdm/xdm-config :

```
! $XFree86: xc/programs/xdm/config/xdm-conf.cpp,v 1.1.1.2.4.2
1999/10/12 18:33:29 hohndel Exp $
DisplayManager.servers: /etc/X11/xdm/Xservers
DisplayManager.accessFile: /etc/X11/xdm/Xaccess
! All displays should use authorization, but we cannot be sure
```

```
! X terminals will be configured that way, so by default
! use authorization only for local displays :0, :1, etc.
DisplayManager._0.authorize: true
DisplayManager._1.authorize: true
DisplayManager*resources:
/etc/X11/xdm/XresourcesDisplayManager*session:
/etc/X11/xdm/Xsession
DisplayManager*authComplain: false
! SECURITY: do not listen for XDMCP or Chooser requests
! Comment out this line if you want to manage X terminals with
xdm
DisplayManager.requestPort: 0
Le fichier /etc/X11/xdm/Xservers détermine la liste des serveurs
X géré par XDM :
# $XConsortium: Xserv.ws.cpp,v 1.3 93/09/28 14:30:30 gildea Exp
$
# $XFree86: xc/programs/xdm/config/Xserv.ws.cpp,v 1.1.1.1.12.2
1998/10/04 15:23:14 hohndel Exp $
# Xservers file, workstation prototype
# This file should contain an entry to start the server on the
# local display; if you have more than one display (not screen),
# you can add entries to the list (one per line). If you also
# have some X terminals connected which do not support XDMCP,
# you can add them here as well. Each X terminal line should
# look like:
# XTerminalName:0 foreign
:0 local /usr/X11R6/bin/X
```

Le fichier /etc/X11/xdm/Xaccess gère l'accès XDMCP, permettant aux machines distantes de se connecter à la machine locale via XDMCP, de façon à obtenir une invite d'authentification. Voici un extrait de ce fichier :

```
# La première ligne pour les requêtes directes
*
# Les lignes suivantes pour les requêtes indirectes
* CHOOSER BROADCAST
```

Le premier « * » signifie que n'importe quel hôte peut demander une invite d'authentification à XDM.

La ligne « CHOOSER » spécifie les hôtes qui peuvent se connecter à XDM en utilisant des requêtes indirectes. Dans ce cas, n'importe quel hôte peut demander à la machine une liste d'hôtes potentiels auxquels se connecter (la seconde ligne « * »). L'utilisateur obtiendra alors, à la place de l'invite

d'authentification, une application « *chooser* », qui lui fournira une liste des hôtes détectés sur le réseau et acceptant des connexions XDMCP.

F. Les gestionnaires de fenêtres : Window Manager

Il est pratiquement impossible d'utiliser X sans un gestionnaire de fenêtres, qui offre des menus, des barres de titres pour les fenêtres et tout ce qui est nécessaire pour disposer d'une interface graphique agréable.

La fenêtre « racine » (*root*) du gestionnaire couvre l'écran complet du moniteur et représente le « bureau ».

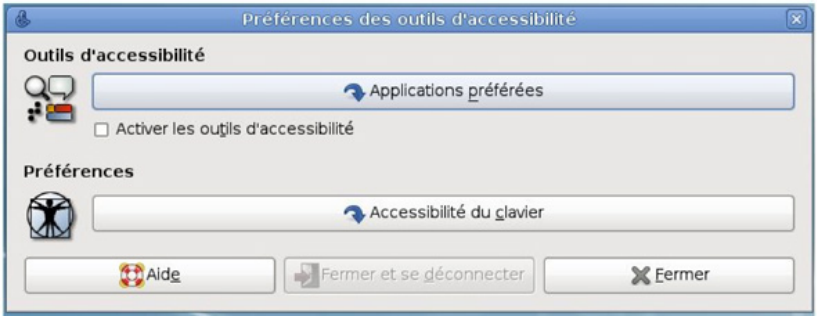
À part KDE et Gnome qui sont les plus courants, on peut citer les gestionnaires WindowMaker, fvwm, icewm, twm, et beaucoup d'autres !

G. Accessibilité

Linux offre un panel d'outils d'accessibilité destinés aux personnes en situation de handicap. L'objectif est de garantir une indépendance la plus grande possible de ces personnes.

Les options d'accessibilité peuvent être activées à travers certains paramètres du serveur X. Les environnements Gnome et KDE proposent aussi des programmes d'accessibilité, dans le menu Système → Préférences → Outils d'accessibilité (*figure 3*).

Figure 3. Outils d'accessibilité sous Gnome



a) Ajuster les options du clavier et de la souris

Plusieurs options d'utilisation du clavier et de la souris sont offertes :

- régler la fréquence de répétition des caractères, soit pour désactiver la répétition d'un caractère lorsque la touche du clavier est enfoncée, soit pour fixer un délai très long de répétition des caractères. Le réglage de ces paramètres peut aider les personnes souffrant de troubles moteurs ;
- modifier le comportement de certaines touches telles que <Ctrl>, <Alt> et <Shift> afin qu'elles restent actives même après leur libération (touches *rémanentes*). Ceci peut être utile pour les personnes qui rencontrent des difficultés pour appuyer simultanément sur plusieurs touches ;
- inverser la fonction des boutons droit et gauche de la souris ;
- modifier la taille du pointeur de la souris, pour l'agrandir afin d'améliorer sa visibilité ;
- contrôler le pointeur de la souris avec des touches du clavier. Cette fonctionnalité est conçue pour les personnes ayant des difficultés à utiliser la souris.

b) Clavier visuel

Le clavier visuel est utile pour les personnes ayant des difficultés à utiliser un clavier ordinaire, mais qui peuvent utiliser la souris.

GOK (*Gnome On-Screen Keyboard*) est un clavier d'écran puissant qui offre, en complément des fonctionnalités ordinaires d'un clavier, des raccourcis pour les menus et les barres d'outils des programmes.

c) Ajuster les options d'affichage

Les utilisateurs malvoyants peuvent bénéficier d'ajustements des paramètres d'affichage, notamment des options de polices, de contraste et de grossissement d'écran.

Gnome offre la boîte de dialogue « Apparence » (Système → Préférences → Apparence) qui donne la possibilité de modifier la police (Time, Arial, Courier, etc…), le style de la police (gras, italique, …) et sa taille.

Sous KDE une boîte de dialogue similaire est accessible à partir de l'élément « Apparence » du panel « Configuration du Système ».

KDE et Gnome offrent aussi plusieurs thèmes, certains thèmes sont meilleurs que d'autres en terme de lisibilité. Par exemple certains sont très faibles en contraste, d'autres sont riches en contraste.

Une loupe d'écran agrandit une partie de l'écran, généralement la zone située immédiatement autour de la souris. La loupe d'écran Kmag est intégrée à KDE.

d) Autres outils d'accessibilité

Des programmes supplémentaires sont offerts pour aider les personnes ayant des besoins spéciaux :

- Gnopernicus offre aux personnes non-voyantes la possibilité de lire l'écran et d'envoyer ces informations vers différents périphériques : Braille, sortie audio … ;
- Orca est un autre lecteur d'écran fournit par Gnome ;
- Emacspeak est un lecteur d'écran pour l'éditeur emacs ;
- BRLTTY fournit un accès à la console Linux/Unix pour une personne aveugle.

H. Exercices

1) À quoi sert un fichier Xressources ?

- ❑ À contrôler l'apparence des clients X.
- ❑ À affecter des ressources pour le serveur X.
- ❑ À définir les droits d'accès au serveur X en réseau.

2) Lorsque vous utilisez xdm, lequel des fichiers suivants peut être utilisé pour démarrer un gestionnaire de fenêtres ?

- ❑ Xservers
- ❑ Xaccess
- ❑ xdm-config
- ❑ Xsession
- ❑ Xsetup_0

3) Quelle option faut-il utiliser afin d'afficher à distance une application X Window ?

- ❑ -display
- ❑ -connect
- ❑ -remote
- ❑ -xhost

4) Dans le fichier xorg.conf, une section InputDevice ne peut pas décrire :

- un clavier
- une souris
- une tablette de dessin
- un moniteur

Chapitre 4. Impression

Objectifs	⇒ Comprendre les filtres et les queues d'impression. ⇒ Savoir gérer les queues d'impression. ⇒ Connaître les outils d'impression. ⇒ Connaître les fichiers de configuration CUPS.
Points importants	Sous Linux, les logiciels d'impression évoluent. Du démon LPD, hérité d'Unix, en passant par son amélioration avec LPRNG puis l'utilisation de CUPS, la philosophie a profondément changé. Mais la compatibilité des commandes de base est en général assurée.
Mots clés	/usr/bin/lpr, /usr/bin/lprm, /usr/bin/lpq, ghostscript, fichiers de configuration et utilitaires du serveur CUPS

A. Terminologies

La plupart des serveurs d'impression sous Linux sont compatibles avec le système d'impression traditionnel BSD (*Berkeley Software Distribution*).

On appelle travaux d'impression, ou *jobs*, l'ensemble des fichiers soumis à l'impression.

Le serveur d'impression, ou *spooler*, est le programme responsable de la réception, de la mise en queue d'impression et de l'envoi vers l'imprimante des travaux d'impression. Le *spooler* offre aussi les moyens pour consulter ou annuler les impressions en cours.

Exemples de serveurs d'impression :

- **LPR** : Le système d'impression historique du BSD ;
- **LPRng** est une version améliorée du LPR, il fusionne les fonctions d'impression du système V avec celle du système Berkeley ;

- **CUPS** (*Common Unix Printing System*) utilise le protocole IPP (*Internet Printing Protocol*) pour la gestion des travaux et des queues d'impression ;
- **PDL** (*Page Description Langage*) est le langage qui permet la description d'une mise en page de façon indépendante du périphérique. Les langages PostScript et PCL sont deux exemples du langage PDL.

Le **filtre** est un programme qui traite les *jobs* avant leur envoi vers l'imprimante. Le serveur d'impression envoie les *jobs* vers le filtre, ce dernier transforme ces jobs en format PDL supporté par l'imprimante cible.

Unix/Linux imprime directement les fichiers en format texte. Pour ceux qui sont dans un format différent, on utilise des filtres qui les transforment au format PostScript. Cela permet de les envoyer directement à une imprimante PostScript.

Comme toutes les imprimantes ne supportent pas le langage PostScript, on utilise une imprimante PostScript virtuelle, **ghostscript**, qui finalement traduit le PostScript en langage PDL de l'imprimante.

Exemples de programmes Ghostscript : Aladdin Ghostscript (version commerciale), GNU Ghostscript et ESP Ghostscript (CUPS).

B. Outils d'impression

Dans cette partie nous allons voir les commandes utilisées pour imprimer des fichiers et pour gérer des queues d'impression.

a) `lpr`

La commande `/usr/bin/lpr` est utilisée pour envoyer une demande d'impression à une imprimante. C'est une version modernisée de la commande Unix `lp` (*line print*) utilisée auparavant.

Il faut bien comprendre que l'on peut associer différentes queues d'impression à une même imprimante.

Supposons que l'on veuille imprimer un fichier appelé « ma_lettre », voici deux manières de l'imprimer :

- envoyer la demande à l'imprimante par défaut,

```
lpr ma_lettre
```

- envoyer la demande à l'imprimante « imp »,

```
lpr -Pimp ma_lettre
```

Voici quelques options de la commande `lpr` :

- -P<queue> : envoie la demande à la queue <queue> ;
- -#<nombre> : imprime <nombre> copies ;
- -o <option> : permet de passer une ou plusieurs options, comme le mode paysage etc.

La commande mpage permet de créer un fichier PostScript, prêt à être imprimé, à partir de plusieurs fichiers en format texte ou PostScript. Elle offre la possibilité de réduire le texte pour mettre plusieurs pages par feuille imprimée.

```
mpage -4 le_fichier.txt l_autre_fichier.ps
```

b) lpq

La commande /usr/bin/lpq permet d'afficher le contenu des queues d'impression.

Pour afficher les demandes de la queue par défaut :

```
lpq
```

Pour afficher les demandes de toutes les queues :

```
lpq -a
```

Pour afficher les demandes de la queue « imp » :

```
lpq -Pimp
```

c) lprm

La commande /usr/bin/lprm permet de détruire des demandes dans les queues d'impression.

Les autorisations de destruction par utilisateur sont décrites dans le fichier /etc/lpd.perms (*cf.* une description plus précise plus loin).

Pour détruire la dernière demande d'impression :

```
lprm
```

Pour détruire la dernière demande d'impression de l'utilisateur « mejdi » :

```
lprm mejdi
```

Pour détruire toutes les demandes :

```
lprm -a
```

ou

```
lprm -
```

On peut également faire référence à la demande par son numéro que l'on récupère avec la commande `lpq` vue précédemment.

De même que pour les commandes précédentes, `lprm` supporte l'option « `-P` » qui permet de faire référence à une autre queue que la queue par défaut.

C. Fichiers de configuration

Le fichier de configuration principal de CUPS est /etc/cups/cupsd.conf.

Chaque ligne de ce fichier est soit une directive de configuration, soit une ligne vide, soit un commentaire.

Les directives de configuration sont volontairement similaires à celles utilisées par le serveur web Apache.

Quelques directives intéressantes utilisées dans le fichier cupsd.conf :

- `Browsing` définit si la récupération des informations des imprimantes distantes doit être activée ;
- `BrowseAddress` définit une adresse où diffuser les informations sur les imprimantes ;
- `BrowseAllow` accepte les paquets arrivant de machines nommées ou d'adresses IP ;
- `BrowseDeny` refuse les paquets de demande d'informations sur les imprimantes arrivant de machines nommées ou d'adresses IP ;
- `BrowseInterval` définit l'intervalle maximum entre les demandes d'information sur les imprimantes ;
- `BrowseOrder` définit le contrôle d'accès aux informations des imprimantes (allow,deny ou deny,allow) ;
- `ServerName` définit le nom complet du serveur ;
- `Listen` définit le port IPP d'écoute du serveur CUPS ;
- `<Location /chemin> …. </Location>` définit les contrôles d'accès à l'interface web du serveur CUPS.

Exemple de fichier /etc/cupsd.conf :

```
# Autoriser le partage sur le réseau local
listen 192.168.1.*:631
Browsing On
```

```
BrowseAddress 192.168.210.255
BrowseDeny All
BrowseAllow 192.168.210.*
BrowseOrder deny,allow
#Autoriser les stations du réseau local à accéder à #notre
serveur « ServeurCUPS » via l'adresse
#http://ServeurCUPS :631
<Location />
   Order Allow, Deny
   Allow 192.168.210.*
</Location >
```

D. Exercices

5. Quelle commande utilisez-vous pour suspendre ou mettre en attente une queue d'impression ?

- ❑ lpr
- ❑ lpq
- ❑ lpc
- ❑ lpd
- ❑ prm

6. Que va faire la commande suivante : cat hosts | lpr -#2

- ❑ Imprimer le fichier hosts sur l'imprimante par défaut deux fois.
- ❑ Classer hosts et imprimer le classement comme tâche #2.
- ❑ Envoyer le fichier hosts à l'imprimante et le mettre dans la queue numéro 2.
- ❑ Envoyer le fichier hosts sur la sortie standard puis envoyer la tâche en cours à l'imprimante 2.

Chapitre 5. Gestion des utilisateurs et des groupes d'utilisateurs

Objectifs	⇒ Savoir créer des utilisateurs. ⇒ Savoir gérer les groupes et la participation des utilisateurs dans différents groupes. ⇒ Connaître les fichiers de configuration. ⇒ Modifier les comptes des utilisateurs et les informations de configuration par défaut.
Points importants	Le système de gestion des utilisateurs sous Linux est simple mais efficace. Cependant, il a quelques limitations.
Mots clés	/bin/false, /etc/default/useradd, /etc/group, /etc/gshadow, /etc/passwd, /etc/shadow, /etc/skel, groupadd, groupdel, groupe, groups, grpconv, grpunconv, id, newgrp, passwd, pwconv, pwunconv, useradd, userdel, usermod, utilisateur

A. Les utilisateurs

Pour la création d'un utilisateur, on utilise la commande `/usr/sbin/useradd`, ou son alias `/usr/sbin/adduser` qui est un lien symbolique vers la commande précédente pour des raisons de compatibilité historique.

Syntaxe :

```
/usr/sbin/useradd [options] nom-utilisateur
```

Quelques options utiles :

- `-c` : commentaire ;

- `-g` : groupe ;
- `-s` : shell.

Pour ajouter un utilisateur « mejdi » dans le groupe « chefs » avec le shell « tcsh » :

```
/usr/sbin/adduser -c 'Mejdi le chef' -g 'chefs' -s '/bin/tcsh'
mejdi
```

Les options par défaut se trouvent dans le fichier /etc/default/useradd, ou bien sont listées par l'option `-D` de la commande `useradd`.

```
# useradd defaults file
GROUP=100
HOME=/home
INACTIVE=-1
EXPIRE=
SHELL=/bin/bash
SKEL=/etc/skel
```

Chaque utilisateur possède un identifiant ou UID (*user identifier*), numéro généré automatiquement et compris entre 500 et 60 000. Un autre intervalle de valeurs peut si nécessaire être imposé. Il doit dans ce cas être spécifié dans le fichier `/etc/login.defs`.

Pour activer le compte, l'administrateur doit définir un mot de passe pour le compte par la commande `/usr/bin/passwd` :

Syntaxe :

```
/usr/bin/passwd nom-utilisateur
```

Exemple :

```
/usr/bin/passwd mejdi
```

Cette commande permet également à l'utilisateur de changer lui-même son mot de passe.

B. Les groupes

Un utilisateur appartient toujours au moins à un groupe dit **groupe primaire** (*primary group*).

Si le groupe n'est pas spécifié au moment de la création du compte deux stratégies générales sont employées pour assigner un groupe par défaut :

- le groupe par défaut est le même pour tous. Il s'appelle par exemple « **users** » ;
- la distribution Red Hat a introduit la notion de groupe privé par utilisateur ou **UPG** (*User Private Group*). Le nom du groupe est identique à celui du login.

Selon la stratégie employée, le masque par défaut (*umask*) de création est initialisé à 022 dans le premier cas (classique) et à 002 dans le deuxième cas (UPG).

Si un utilisateur crée un fichier, celui-ci appartiendra par défaut au groupe primaire de l'utilisateur.

Un utilisateur peut appartenir à d'autres groupes, ce sont les groupes secondaires.

Pour connaître la liste des groupes auxquels l'utilisateur appartient, on utilise la commande `id`.

Dans l'exemple qui suit, l'utilisateur « moi » appartient au groupe primaire « normal » et aux groupes secondaires « compta » et « chefs ».

```
id
uid=1421(moi) gid=1664(normal)
groupes=1664(normal),2010(compta),2008(chefs)
```

La commande `newgrp` permet de changer temporairement de groupe primaire, à condition que le nouveau groupe soit un groupe secondaire de l'utilisateur ou que l'utilisateur en connaisse le mot de passe.

```
newgrp chefs
```

La commande `id` donne alors :

```
id
uid=1421(moi) gid=2008(chefs)
groupes=1664(normal),2010(compta),2008(chefs)
```

La commange `groups` permet elle aussi d'afficher les groupes auxquels appartient un utilisateur.

```
groups
normal compta chefs
```

Pour ajouter un groupe, on utilise la commande `groupadd` :

```
groupadd forcats
```

Pour supprimer un groupe, on utilise la commande `groupdel` :

```
groupdel forcats
```

Ces commandes mettent à jour le fichier /etc/group.

Pour gérer les utilisateurs d'un groupe, on utilise la commande `gpasswd`.

Les options sont les suivantes :

- `-a` : ajout d'un utilisateur ;
- `-d` : retrait d'un utilisateur ;
- `-A` : affectation d'un administrateur au groupe.

```
gpasswd -a nicolas forcats
```

La commande était prévue à l'origine pour ajouter un mot de passe commun au groupe et permettre aux utilisateurs appartenant à un même groupe de se connecter avec le même mot de passe, ce qui explique le nom de la commande. Cette possibilité n'existe plus pour des raisons de sécurité évidentes.

C. Les fichiers de configuration

a) Gestion des utilisateurs

Le fichier /etc/passwd contient les informations sur les utilisateurs, structurées en sept champs :

- login ;
- UID ;
- GID ;
- mot de passe ou « x » s'il existe un fichier /etc/shadow ;
- description de l'utilisateur ;
- répertoire par défaut de l'utilisateur ;
- shell.

Les sept champs sont présentés sur une ligne et séparés par le caractère « : ».

Exemple de ligne extraite d'un fichier /etc/passwd avec utilisation d'un fichier /etc/shadow :

```
nicolas:x:502:502:Nicolas L:/home/nicolas:/bin/tcsh
```

Depuis quasiment l'origine, la majorité des distributions Linux utilise un fichier /etc/shadow pour stocker les mots de passe. La sécurité est bien meilleure car il est protégé en lecture. Le fichier /etc/passwd est, lui, lisible par toutes les applications.

Pour créer un fichier « /etc/shadow » à partir d'un fichier « /etc/passwd » on utilise la commande /usr/sbin/pwconv.

Pour revenir à la configuration précédente (*i.e.* stockage des mots de passe dans le fichier /etc/passwd), on utilise la commande /usr/sbin/pwunconv.

Attention à fixer correctement les droits sur ces fichiers : 600 ou même 400 pour /etc/shadow et 644 pour /etc/passwd.

Ne pas oublier de vérifier, lors de l'utilisation de la commande pwunconv, de remettre les mêmes droits sur le fichier /etc/passwd.

b) Gestion des groupes

Le fichier /etc/group contient les informations sur les groupes, structurées en quatre champs :

- nom du groupe ;
- mot de passe du groupe ou « x » s'il existe un fichier /etc/gshadow ;
- GID ;
- liste des utilisateurs du groupe.

Les quatre champs sont présentés sur une ligne et séparés par le caractère « : ».

Ligne de fichier /etc/group avec utilisation d'un fichier /etc/gshadow :

```
normal:x:555:niry,andrei,kader,nicolas
```

De même que pour le fichier /etc/passwd, pour créer un fichier /etc/gshadow à partir d'un fichier /etc/group on utilise la commande :

/usr/sbin/grpconv

Pour revenir à la configuration précédente (i.e. stockage des mots de passe dans le fichier /etc/group et destruction de /etc/gshadow) :

/usr/sbin/grpunconv

c) Fichiers de configuration par défaut

Le fichier /etc/login.defs contient les informations par défaut sur la validité des comptes et des mots de passe des utilisateurs. Ces informations sont stockées dans le fichier /etc/shadow lors de la création du compte :

- `MAIL_DIR` : répertoire mail par défaut (e.g. /var/spool/mail) ;
- `PASS_MAX_DAYS`, `PASS_MIN_DAYS`, `PASS_MIN_LEN`, `PASS_WARN_AGE` : informations concernant la validité du mot de passe ;
- `UID_MIN`, `UID_MAX` : plage des numéros identifiant des utilisateurs (UID) lors de l'utilisation de `useradd` ;
- `GID_MIN`, `GID_MAX` : plage des numéros identifiants des groupes (GID) lors de l'utilisation de `groupadd` ;
- `CREATE_HOME` : création automatique du répertoire *home* lors de l'utilisation de `useradd` ;
- PASS_MAX_DAYS : nombre maximum de jours d'utilisation d'un mot de passe ;
- PASS_MIN_DAYS : nombre minimum de jours entre deux changements de mot de passe ;
- PASS_MIN_LEN : taille minimum d'un mot de passe ;
- PASS_WARN_AGE : nombre de jours d'envoi d'un avertissement avant que le mot de passe n'expire.

D. Gestion des comptes et des options de création par défaut

Les options de configuration d'un compte peuvent être modifiées par la commande `usermod` :

- `-l` : nouveau nom d'utilisateur ;
- `-c` : commentaire ;
- `-g` : groupe (il doit exister au préalable) ;
- `-s` : shell ;
- `-d` : chemin du répertoire *home* ;
- `-u` : identifiant utilisateur (UID) ;
- `-p` : mot de passe à entrer en format md5 ;
- `-e` : informations d'expiration du compte.

Les options de configuration d'un groupe peuvent être modifiées par la commande `groupmod` :

- `-n` : nouveau nom du groupe ;
- `-g` : identifiant du groupe (GID).

a) Comment bloquer un compte

Un moyen simple est de faire précéder le mot de passe par un « ! » dans les fichiers de configuration. Lors de l'utilisation d'un fichier /etc/shadow, on peut remplacer également le « x » dans le fichier /etc/passwd par un « * ».

Une autre méthode consiste à utiliser les commandes passwd et usermod :

```
passwd -l
usermod -L
```

Pour débloquer le compte en utilisant les mêmes commandes :

```
passwd -u
usermod -U
```

On peut aussi détruire le mot de passe :

```
passwd -d
```

Enfin, on peut affecter à un utilisateur le shell par défaut /bin/false, ce qui l'empêche de se connecter.

b) Gestion des informations d'expiration du compte

Pour modifier les informations par défaut (/etc/login.defs) et les informations d'expiration, on utilise la commande /usr/bin/chage :

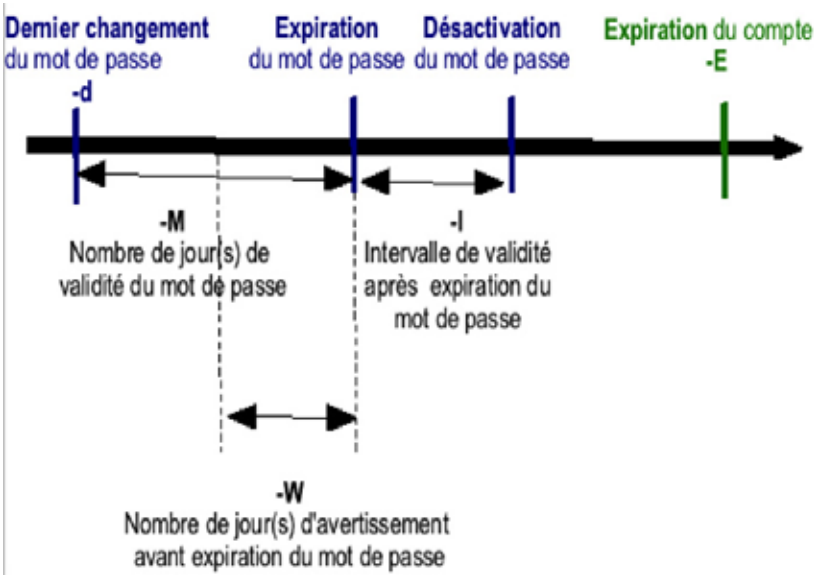
```
chage [ -l ] [ -m min_days ] [ -M max_days ] [ -W warn ] [ -I
inactive ] [ -E expire ] [ -d last_day ] user
```

Options :

- -l donne les valeurs actuelles du compte ;
- -E permet de fixer une date d'expiration sous la forme Unix standard ou sous la forme YYYY/MM/DD ;
- -M permet de changer la valeur de PASS_MAX_DAYS contenue dans le fichier /etc/login.defs ;
- -m permet de changer la valeur de PASS_MIN_DAYS contenue dans le fichier /etc/login.defs ;
- -w permet de changer la valeur de PASS_WARN_AGE contenue dans le fichier /etc/login.defs ;
- -d permet de changer la date de dernier changement de mot de passe sous la forme Unix standard ou sous la forme YYYY/MM/DD.

La *figure 4* récapitule les différentes informations associées à la « vie » du compte.

Figure 4. Informations d'expiration d'un compte associées à la commande en ligne



c) Destruction d'un compte

On utilise la commande `/usr/sbin/userdel`. L'option `-r` permet de détruire également le contenu du répertoire home.

```
/usr/sbin/userdel -r mejdi
```

E. Exercices

1. **Quelle est la commande Unix qui permet de créer un utilisateur user1 qui appartient au groupe auf ?**
   - ❑ useradd -m -g user1 auf
   - ❑ useradd -m user1 -group auf
   - ❑ add -m -g auf user1
   - ❑ useradd -m -g auf user1

2. **L'utilisateur mejdi a été déplacé dans le département BECO. Vous voulez changer son groupe principal en beco. Quelle est la commande la plus simple pour réaliser cela ?**

3. **Quelle commande allez-vous utiliser pour verrouiller le compte de l'utilisateur nicolas ?**

4. **L'utilisateur nicolas de votre serveur a oublié son mot de passe. Quelle commande allez-vous utiliser pour changer son mot de passe en supposant que vous avez ouvert une session en tant que root ?**

5. **Le répertoire suivant contient les fichiers qui sont copiés dans le répertoire de l'utilisateur au moment de la création de son compte :**

   - ❑ /etc/skel
   - ❑ /etc/users
   - ❑ /etc/passwd
   - ❑ /etc/hosts

Chapitre 6. Administration du système GNU/Linux

Objectifs	⇒ Réviser les principales techniques d'administration du système. ⇒ Connaître la gestion du système de *logs*, la programmation des tâches périodiques. ⇒ Être capable de localiser le système en plusieurs langues autres que l'anglais.
Mots clés	/etc/localtime, /etc/timezone, locale, /usr/share/zoneinfo, /var/log, anacron, anacrontab, at, cron, crontab, date, iconv, ISO-8859, LANG, LC_*, LC_ALL, logrotate, tzselect, Unicode, UTF-8

A. Les traces du système

a) Les journaux (*logs*)

Tous les événements qui affectent les processus sont enregistrés dans des fichiers appelés **fichiers journaux** ou **journaux** (*logs files* ou *logs*). Par défaut les journaux sont placés dans l'arborescence /var/log. Certains démons (*daemons*) génèrent leurs journaux dans des arborescences spécifiques. Par exemple les logiciels Samba et Apache enregistrent leurs journaux dans /var/log/samba et /var/log/apache.

La plupart des messages de logs sont gérés par le démon système `syslogd`.

Voici quelques fichiers journaux importants :

- cron : contient les messages concernant cron ;
- maillog : messages relatifs au système de courrier ;

- messages : presque tous les messages du système sauf les messages d'authentification qui se trouvent dans d'autres fichiers spécifiques ;
- secure ou auth : messages relatifs à l'authentification des utilisateurs, la gestion de la base de données d'utilisateurs, etc. ;
- dmesg : messages provenant du noyau.

Le fichier journal le plus utile et le plus consulté est /var/log/messages car la plupart des messages produits par le système y sont enregistrés.

La commande `tail -f` permet d'afficher dynamiquement le contenu des dernières lignes d'un fichier journal ce qui permet d'en suivre l'évolution et de visualiser immédiatement les nouveaux événements.

```
tail -f /var/log/messages
```

b) La configuration

La configuration du démon `syslogd`, dont la tâche est l'enregistrement des messages de *logs*, se fait dans le fichier /etc/syslog.conf.

Les messages de *logs* sont divisés en groupes. Dans chaque groupe ils sont classés par priorité.

Les groupes de messages de *logs* sont les suivants :

- auth et authpriv : authentification ;
- cron : messages de l'utilitaire cron ;
- kern : messages du noyau ;
- mail : le système de courrier ;
- user : messages des processus utilisateurs.

Les différentes priorités des messages, triées par ordre croissant, sont les suivantes :

- emerg
- alert
- crit
- err
- warning
- notice
- info
- debug
- *
- none.

Chaque ligne de configuration de `/etc/syslog.conf` contient une liste de groupes de messages avec la priorité correspondante et le nom du fichier journal dans lequel seront enregistrés ces messages :

```
groupe1.priorité1 ; groupe2.priorité2 ; …/var/log/fichier_de_log
```

Il faut remarquer que la priorité spécifiée de cette manière représente la priorité minimale. Chaque message qui possède une priorité égale ou supérieure à la priorité donnée sera enregistré. Pour spécifier des messages ayant une priorité unique on fait précéder la priorité du signe « = ».

Voici quelques exemples de configuration :

```
authpriv.*   /var/log/secure
mail.*       /var/log/mail.log
cron.*       /var/log/cron
news.=crit   /var/log/news/news.crit
news.=err    /var/log/news/news.err
news.notice  /var/log/news/news.notice
```

Il est également possible d'envoyer les messages de *logs* sur une autre machine (serveur de *logs*) :

```
.emerg                                          @10.1.1.88
```

Cela permet de centraliser tous les logs des machines d'un réseau sur une seule machine pour en faciliter la consultation.

c) La rotation des *logs*

Comme la taille des fichiers journaux augmente en permanence il est nécessaire de les nettoyer régulièrement. La technique utilisée en général est la **rotation des *logs*** qui permet de conserver différentes versions d'un même fichier. Avant d'être vidé le fichier journal est copié dans un autre fichier avec un suffixe .0. Le précédent fichier « .0 » devient « .1 » et ainsi de suite. De cette manière on peut par exemple garder les quatre dernières versions d'un fichier journal ; si l'on fait une rotation par semaine cela correspond à la sauvegarde des messages de logs du mois précédent.

Il est également possible de compresser ces fichiers :

```
messages
messages.0
messages.1.gz
messages.2.gz
messages.3.gz
```

L'outil qui fait la rotation des logs s'appelle `logrotate`. Son fichier de configuration est /etc/logrorate.conf. En plus du fichier général nous pouvons utiliser un fichier spécifique par service dans le répertoire /etc/logrotate.d.

Voilà un exemple du fichier logrotate.conf :

```
# faire la rotation chaque semaine
weekly
# garder les 4 dernier fichiers
rotate 4
# envoyer les erreurs à root
errors root
# après la rotation créer le nouveau fichier de log
create
# compresser les fichiers sauvegardés
compress
# inclure les fichiers de /etc/logrotate.d
include /etc/logrotate.d
# configuration des logs de lastlog et utmp
/var/log/wtmp {
monthly
   create 0664 root utmp
   rotate 1
}
```

B. Exécution automatique de tâches

Deux outils permettent de programmer des tâches périodiques. Ce sont les commandes `cron` et `at`.

a) Cron

Le système `cron` permet de gérer la configuration de tables d'exécution de tâches périodiques (*crontabs*) : une table pour chaque utilisateur et une table pour le système. Les tables des utilisateurs sont enregistrées dans les fichiers /var/spool/cron/<nom_d_utilisateur> gérés par les utilisateurs à l'aide de l'outil `crontab`. La table cron du système se trouve dans le fichier /etc/crontab.

Options de la commande `crontab` :

- `crontab -l` : affiche la liste des tâches programmées ;
- `crontab -e` : ouvre l'éditeur par défaut avec la table cron ;

- `crontab -u` : permet à l'administrateur de gérer la table cron d'un certaine utilisateur ;
- `crontab -r` : supprime la table cron.

Le format des lignes dans la table cron est le suivant :

```
minutes(0-59) heures(0-23) jour du mois(1-31) mois(1-12) jour de
la semaine(0-6) commande
```

Dans la table cron du système (/etc/crontab) il faut également indiquer le nom de l'utilisateur qui va lancer la commande :

```
minutes (0-59) heures(0-23) jour du mois (1-31) mois (1-12) jour
de la semaine (0-6) utilisateur commande
```

Dans la ligne de la table /etc/crontab de l'exemple ci-dessous, l'utilisateur *root* exécute à 5h00 (= « 0 5 »), chaque dimanche (« 0 »), la commande mentionnée :

```
0 5 * * 0 root /usr/bin/find /home/ -name core -exec rm {} \;
```

Par défaut chaque utilisateur peut utiliser la commande `crontab`. Pour restreindre cette possibilité, on peut spécifier des droits dans les fichiers /etc/cron.allow et /etc/cron.deny.

Il est possible d'utiliser une autre version de cron qui se nomme `anacron` pour laquelle le fichier de configuration est /etc/anacrontab. Cette version est utilisée pour lancer des commandes avec une périodicité donnée (par exemple en nombre de jours) et ne nécessite pas que la machine soit en permanence en route. Elle se base sur des fichiers générés dès la fin d'une tâche.

b) `at`

La commande `at` peut être utilisée pour programmer l'exécution d'une tâche à un moment donné. La commande à exécuter est celle passée en argument.

Syntaxe :

```
at [spécification du temps]
```

La spécification du temps peut être de la forme « `now` », « `3am + 2days` », « `midnight` », « `10:15 Apr 12` », « `teatime` », etc.

Pour programmer l'exécution de la commande `/usr/local/updatemirror` à minuit :

```
echo '/usr/local/updatemirror' | at midnight
```

Les commandes programmées par `at` sont enregistrées dans /var/spool/at/ et sont exécutées par le démon `atd`.

Les options suivantes permettent de gérer la queue des tâches `at` :

- `at -l` # donne la liste des tâches programmées ;
- `atq` # est la même commande que `at -l` ;
- `at -d <numéro de la tâche>` # supprime la tâche spécifiée ;
- `atrm` # est la même commande que `at -d`.

Par défaut seul l'utilisateur *root* peut utiliser la commande `at`. Cette configuration peut être modifiée dans les fichiers /etc/at.deny et /etc/at.allow.

C. Localisation et internationalisation

Linux est un logiciel **internationalisé**, c'est-à-dire qu'il peut être adapté aux contextes de chaque pays. Ce travail d'adaptation est appelé **régionalisation** ou **localisation**. Il concerne bien sûr les messages utilisateurs, pour qu'ils soient rédigés dans la langue appropriée, mais également bien d'autres éléments tels le format des dates et des nombres, les couleurs...

N.B. : à la place des termes internationalisation (*internationalization*) et localisation (*localization*) on utilise souvent les abréviations i18n et l10n, construites en ne conservant que les première et dernière lettre de chaque terme et en remplaçant les lettres supprimées par leur nombre.

Linux supporte ainsi une large variété de jeux de caractères, types de claviers, formats d'affichage de dates et d'autres caractéristiques qui peuvent varier d'une région à une autre.

Beaucoup de ces paramètres sont définis lors de l'installation du système, et sont modifiables pour les besoins de personnalisation.

a) Définition du fuseau horaire

Le fuseau horaire est configuré à travers le fichier /etc/localtime qui est le plus souvent un lien symbolique vers un fichier du répertoire /usr/share/zoneinfo.

La commande `date` permet d'afficher le fuseau horaire utilisé au sein du système. Le résultat affiche le code standard du fuseau horaire : CET (*Central European Time*) :

```
$ date
jeu. déc. 17 00:28:32 CET 2009
```

La variable d'environnement TZ permet de modifier temporairement le fuseau horaire. On peut donc utiliser la commande `tzselect` qui demande à l'utilisateur des informations sur son emplacement et fournit en sortie la

description du fuseau horaire. Ce résultat peut être affecté à la variable d'environnement TZ.

```
$ TZ='Africa/Casablanca'; export TZ
$ date
mer. déc. 16 23:25:26 WET 2009
```

On peut changer de façon permanente le fuseau horaire en créant un lien symbolique vers un fichier du répertoire /usr/share/zoneinfo :

```
# ln -s /usr/share/zoneinfo/Africa/Cairo localtime
# date
jeu. déc. 17 01:33:34 EET 2009
```

Certaines distributions utilisent un autre fichier de configuration secondaire contenant le nom du fuseau horaire. Ce fichier doit être mis à jour en cas de changement de fuseau horaire.

Ce fichier est :

- /etc/timezone sur Debian et ses dérivés ;

```
$ cat /etc/timezone
Africa/Tunis
```

- /etc/sysconfig/clock sur Fedora.

```
$ cat /etc/sysconfig/clock
ZONE="Africa/Tunis"
```

b) Les variables de localisation

Des variables d'environnement de localisation, ou paramètres régionaux (*locales* en anglais), permettent de définir la façon dont les données sont présentées à l'utilisateur ainsi que la façon dont les entrées de l'utilisateur sont traitées.

La commande `locale` permet d'afficher le contenu de ces variables :

```
# locale
LANG=fr_FR.UTF-8
LC_CTYPE="fr_FR.UTF-8"
LC_NUMERIC="fr_FR.UTF-8"
LC_TIME="fr_FR.UTF-8"
LC_COLLATE="fr_FR.UTF-8"
LC_MONETARY="fr_FR.UTF-8"
LC_MESSAGES="fr_FR.UTF-8"
LC_PAPER="fr_FR.UTF-8"
```

```
LC_NAME="fr_FR.UTF-8"
LC_ADDRESS="fr_FR.UTF-8"
LC_TELEPHONE="fr_FR.UTF-8"
LC_MEASUREMENT="fr_FR.UTF-8"
LC_IDENTIFICATION="fr_FR.UTF-8"
LC_ALL=
```

Les valeurs contenues par les variables LANG et LC_* et LANG respectent la syntaxe suivante :

```
langue[_pays[.encodage]][@modifier]
```

où :

- `langue` est un code langue (en minuscules) ;
- `pays` est un code pays (en majuscules) ;
- `encodage` désigne une table de caractères (par exemple UNICODE, UTF-8, ISO-8859-1 à ISO-8859-15) ;
- `@modifier` désigne d'autres attributs particuliers, par exemple le dialecte particulier d'une langue, ou une orthographe non standard.

Voici les significations de quelques variables de localisation :

- LANG définit les préférences globales, qui peuvent ensuite être redéfinies au cas par cas par une variable LC_* ;
- LC_COLLATE définit l'ordre alphabétique des chaînes de caractères ;
- LC_MESSAGES définit la localisation des messages du système ;
- LC_MONETARY définit les unités monétaires et le format des valeurs numériques financières ;
- LC_NUMERIC définit le format de valeurs numériques qui ne sont pas monétaires. Cela définit notamment les caractères utilisés comme séparateur de décimale et de millier.
- LC_ALL définit la valeur par défaut des variables précédentes : si une LC_ n'est pas définie, c'est LC_ALL qui est prise en compte.

c) Modifier l'encodage des fichiers textes

Parfois notre système traite des données textuelles provenant d'un système qui utilise un encodage spécifique, mais avec un programme qui ne supporte pas cet encodage.

Par exemple, notre éditeur supporte l'encodage UTF-8 et non ISO-8859, et nous recevons un fichier encodé avec ISO-8859-1 contenant des accents. L'éditeur va afficher des caractères étranges à la place de ces accents.

Afin de résoudre ce problème, il faut encoder le fichier en UTF-8. La modification de l'encodage d'un fichier se fait avec la commande `iconv`.

Syntaxe :

```
iconv -f encodage [-t encodage] [fichier]
```

Le résultat de la commande est écrit sur la sortie standard, les options `-f` et `-t` définissent l'encodage source et l'encodage cible. Si l'encodage cible est absent, l'encodage utilisé sera celui défini par la *locale* du système.

Exemple :

```
iconv -f iso-8859-1 -t UTF-8 monfichier > monfichier-UTF-8
```

D. Exercices

1. **Parmi les niveaux de sévérité syslog, quel est le plus haut niveau de sévérité ?**
   - ❑ emerg
   - ❑ warning
   - ❑ crit
   - ❑ debug

2. **La ligne correcte de la table cron qui permet d'exécuter le script `/usr/local/sbin/chklog` une fois par heure entre trois heures et cinq heures de l'après-midi chaque lundi et jeudi est :**
   - ❑ 0 3,4,5 * * 2,5 /usr/local/sbin/chklog
   - ❑ 0 3,4,5 * * 1,4 /usr/local/sbin/chklog
   - ❑ * 15,16,17 * * 1,4 /usr/local/sbin/chklog
   - ❑ 0 15,16,17 * * 1,4 /usr/local/sbin/chklog
   - ❑ 0 15,16,17 1,4 * * /usr/local/sbin/chklog

3. **Donner la commande qui permet de convertir le fichier monfichier de l'encodage ISO-8859-1 vers l'encodage UTF-8 :**
   - ❑ locale –f iso-8859-1 –t UTF-8 monfichier > monfichier-UTF-8
   - ❑ iconv –f iso-8859-1 –t UTF-8 monfichier > monfichier-UTF-8
   - ❑ i10n –f iso-8859-1 –t UTF-8 monfichier > monfichier-UTF-8
   - ❑ conv –f iso-8859-1 –t UTF-8 monfichier > monfichier-UTF-8

4. **Créer un script bash qui supprime tous les fichiers « *.bak » dans le répertoire « HOME » de l'utilisateur. Programmer l'exécution de ce script tous les matins à 5h05 par la table cron de l'utilisateur.**

5. **Programmer par `at` l'archivage du répertoire /home pour 4 heure de matin le jour suivant**

6. **Trouver la commande qui permet d'enregistrer des messages par syslog. Utiliser cette commande pour enregistrer chaque jour dans les logs systèmes le nombre de tâches exécutées par le démon atd.**

Chapitre 7. Les réseaux TCP/IP

Objectifs	⇒ Connaître les principes généraux des réseaux TCP/IP (V4) et des mécanismes d'adressage classiques.
Mots clés	adressage, classe, ICMP, IP, masque, port, sous-réseau, TCP, UDP

A. Adressage IP

a) Les adresses IP

Chaque machine connectée au réseau Internet ou sur un réseau local possède au moins une adresse IP. En fait l'adresse IP est assignée à une interface réseau de la machine.

Dans la version du protocole IP actuellement la plus utilisée, la version 4 (IPv4), l'adresse IP est composée de 4 octets (32 bits). La notation la plus connue de l'adresse IP et la notation décimale pointée : les quatre octets sont notés sous forme décimale et séparés par des points. Exemple :

```
Adresse IP : 212.50.14.82
```

En binaire cette adresse IP sera de la forme :

```
Adresse IP : 11010100.00110010.00001110.01010010
```

Ou, sans les points :

```
Adresse IP : 11010100001100100000111001010010
```

b) Les réseaux IP et les masques réseau ; les adresses de réseau et de diffusion

Un paquet IP est un morceau d'information transporté par un réseau TCP/IP. Il peut traverser plusieurs réseaux (des réseaux IP) pour aller d'une machine à une autre. Les adresses IP sont regroupées en ensembles nommés des réseaux. L'adresse IP contient l'adresse du réseau et l'adresse de la machine dans ce réseau (adresse de l'hôte).

```
Adresse IP = Adresse du réseau + Adresse de l'hôte
```

Le masque réseau spécifie la partie de l'adresse IP qui représente l'adresse du réseau. Le reste désigne l'hôte dans ce réseau. Le masque réseau est exprimé également sur 4 octets. Ce qui est spécifique est que les premiers bits, représentant la partie de l'adresse utilisée pour le réseau, sont toujours à « 1 » et que les bits restant sont à « 0 ». Exemple :

```
Masque réseau : 11111111.11111111.11111111.11000000
(255.255.255.192)
```

Cet exemple montre un masque réseau de 26 bits (le nombre de « 1 » au début). Cela veut dire que les 26 premiers bits de l'adresse IP sont réservés pour décrire l'adresse du réseau. Pour désigner les machines dans ce réseau il ne reste que 6 bits. Cela fait 2^6 = 64 adresses de machines.

En réalité, le dernier calcul n'est pas tout à fait correct car il ne tient pas compte du fait que, dans chaque réseau, il y a deux adresses réservées : la première adresse de la plage d'adresses est l'adresse du réseau et la dernière est l'adresse de diffusion. En tenant compte de cette correction nous disposons de 62 adresses pour nos machines (2^6 - 2). Actuellement, il est possible dans certains cas d'utiliser ces adresses.

Résumons cet exemple dans le *tableau 2*. Nous pouvons utiliser pour l'adresse IP la notation CIDR (*Classless Inter Domain Routing* ou routage inter domaine sans classe) : 212.50.14.82/26. Le /26 indique que l'adresse 212.50.14.82 se trouve dans un réseau de taille 26 (les premiers 26 bits déterminent l'adresse du réseau).

Tableau 2. Notation décimale/binaire des éléments d'un réseau en IPv4

	Décimal	**Binaire**
Adresse IP	212.50.14.82/26	11010100.00110010.00001110.01010010
Masque réseau	255.255.255.192	**11111111.11111111.11111111.11**000000
Adresse du réseau	212.50.14.64	11010100.00110010.00001110.01**000000**
Adresse de diffusion	212.50.14.127	11010100.00110010.00001110.0**1111111**

L'adresse du réseau est constituée en remplaçant la partie de l'hôte dans l'adresse IP par des « 0 » (zéros). En ce qui concerne l'adresse de diffusion, on l'obtient en remplaçant la partie de l'hôte par des « 1 ».

Un autre méthode, plus mathématique, permet d'obtenir ces deux adresses :

```
<Adresse réseau> = <Adresse IP> AND <Masque réseau>
<Adresse de diffusion> = <Adresse IP> OR NOT(<Masque réseau>)
```

où AND, OR et NOT sont les opérateurs logiques classiques.

c) Les classes IP

Les réseaux d'adresses IP ont été regroupés en classes. Les 2 premiers bits de l'adresse IP déterminent la classe IP à laquelle cette adresse IP appartient. De ce fait, la classe IP détermine la taille du réseau (*tableau 3*).

Tableau 3. Les classes en IPv4

Premiers 2 bit	Classe IP	Taille du réseau	Nombre d'adresses IP du réseau	Exemple
00 ou 01	A	1 octet	224 = 16777216	115.0.0.1 **01**110011.00000000.00000000.00000001
10	B	2 octets	216 = 65536	130.1.1.1 **10**000010.00000001.00000001.00000001
11	C	3 octets	28 = 65536	212.50.14.82 **11**010100.00110010.00001110.01010010

Dans chaque classe IP il existe un ou plusieurs réseaux réservés pour des besoins privés. Ce sont des adresses qui ne sont pas prises en compte par les routeurs de l'Internet et peuvent être utilisées uniquement pour des réseaux internes. Tandis que les adresses « normales » sont, en principe, uniques dans tout l'Internet (leur distribution est contrôlée), la même adresse privée peut être présente dans plusieurs réseaux privés.

Tableau 4. Réseaux privés par classe en IPv4

Classe IP	Adresses privées
A	10.x.x.x (un réseau de classe A)
B	de 172.16.x.x à 172.31.x.x (16 réseaux de classe B)

C	192.168.x.x (255 réseaux de classe C)

d) Les sous-réseaux

La division en classes IP standard provoque un gaspillage important d'adresses IP (très peu d'organisations ont besoin de 16 milliards d'adresses IP).

En utilisant des masques réseaux différents, il est possible de subdiviser un réseau en sous-réseaux en fonction des besoins. Un fournisseur d'accès le fait fréquemment pour distribuer des adresses à ses clients, mais un administrateur peut être amené à le faire pour gérer plusieurs sites.

Par exemple un réseau avec un masque /24 (normalement la taille d'une classe C) peut être divisé en 2 sous-réseaux de taille /25 ou en 4 sous-réseaux de taille /26.

B. La suite TCP/IP

TCP/IP est le nom commun d'un ensemble de protocoles. Ce sont les protocoles sur lesquels sont basées toutes les communications Internet – IP, ICMP, TCP, UDP.

Les communications par TCP/IP sont organisées en couches, comme dans le modèle OSI. La différence est qu'ici le nombre de couches est réduit à 4, sinon l'idée est la même : chaque couche réalise des fonctionnalités spécifiques et communique avec les couches voisines.

Les quatre couches du modèle TCP/IP sont décrites dans le *tableau 5*.

Tableau 5. Pile TCP/IP

Couche	Description	Protocoles
Application	La couche supérieure – comme son nom l'indique. On y trouve les applications réseau	FTP, HTTP, SMPT, POP, IMAP, ….
Transport	Transmission des données	TCP, UDP
Internet	Datagrammes et routage – connexion des réseaux	IP, ICMP, ARP(?), RIP, BGP, IGMP
Network Access	Communication au niveau physique	Ethernet, Token Ring, etc.

Les protocoles principaux assurant le fonctionnement de l'Internet sont présentés dans le *tableau 6*.

Tableau 6. Protocoles associés à la pile TCP/IP

Protocole	Description
IP	C'est le protocole utilisé pour le découpage de l'information (les segments TCP ou UDP) en paquets (datagrammes) et pour le routage de ces paquets de l'émetteur au destinataire. C'est aussi le protocole qui définit l'adressage des différentes machines connectées à l'Internet et le regroupement de ces machines en réseaux.
TCP	*Transmission Control Protocol* – c'est le protocole fiable de transmission des données. Il travaille en mode connecté pour transmettre les données d'une application à une autre tout en assurant le contrôle de l'intégrité des données.
UDP	*User Datagram Protocol* – ce protocole assure aussi la transmission des données entre applications mais en mode « non fiable » : l'intégrité des données n'est pas contrôlée. Les applications utilisant ce protocole doivent elles-mêmes vérifier cette intégrité. Ce protocole est plus performant que TCP.
ICMP	*Internet Control Message Protocol* – ce protocole est destiné à gérer les informations relatives aux erreurs pouvant survenir sur le réseau.

C. Les ports

Les protocoles TCP et UDP assurent la communication entre deux applications (*tableau 6*).

Nous savons déjà que pour distinguer les machines nous utilisons les adresses IP (chaque machine dispose de sa propre adresse IP unique). Il faut aussi pouvoir distinguer les connexions entre applications.

La solution est l'utilisation des ports. Une connexion TCP entre deux applications est identifiée par 4 informations : l'adresse IP de la première machine, le port TCP de la première application, l'adresse IP de la deuxième machine et le port TCP de la deuxième application.

Le même principe est valable pour les connexions UDP. Elles sont identifiées par des ports UDP.

Il faut remarquer qu'une connexion peut être réalisée entre deux applications qui se trouvent sur la même machine. De même une application donnée (donc utilisant toujours le même port) peut participer à plusieurs connexions (ce sont les applications serveurs). Dans tous les cas la combinaison des quatre informations est unique.

Pour initier une connexion TCP (ou respectivement UDP) il faut au préalable que l'une des deux applications commence à « écouter » sur un certain port TCP. Cette application est le serveur. La deuxième application, que l'on appelle le client, va initier la connexion vers ce port TCP (ou port UDP dans le cas d'une communication par UDP).

Les ports de 1 à 1023 sont réservés aux serveurs, les ports de 1024 à 65 535 sont utilisés dynamiquement par les clients (et par les serveurs parfois).

Une liste des services réseaux classiques peut être trouvée dans le fichier /etc/services. Voilà un extrait de ce fichier :

```
ftp-data   20/tcp
ftp        21/tcp
fsp        21/udp  fspd
ssh        22/tcp              # SSH Remote Login Protocol
ssh        22/udp              # SSH Remote Login Protocol
telnet     23/tcp
# 24 - private
smtp       25/tcp  mail
# 26 - unassigned
time       37/tcp  timserver
time       37/udp  timserver
rlp        39/udp  resource    # resource location
nameserver 42/tcp  name        # IEN 116
whois      43/tcp  nicname
re-mail-ck 50/tcp              # Remote Mail Checking Protocol
re-mail-ck  50/udp             # Remote Mail Checking Protocol
domain      53/tcp  nameserver # name-domain server
domain      53/udp  nameserver
mtp         57/tcp             # deprecated
bootps      67/tcp             # BOOTP server
bootps      67/udp
bootpc      68/tcp             # BOOTP client
bootpc      68/udp
tftp        69/udp
gopher      70/tcp             # Internet Gopher
gopher      70/udp
rje         77/tcp  netrjs
finger      79/tcp
www         80/tcp  http       # WorldWideWeb HTTP
www         80/udp             # HyperText Transfer Protocol
```

On peut remarquer par exemple que les connexions Web sont normalement réalisées sur le port TCP 80.

D. Exercices

1. Lequel des ports suivants ne devrait pas bloquer sur un système qui fonctionne comme un serveur Web public ?

- ❑ 53
- ❑ 21
- ❑ 80
- ❑ 23

2. Quand devez-vous utiliser le protocole RARP dans votre réseau local ?

- ❑ Vous en avez besoin pour activer la pile TCP/IP.
- ❑ Vous en avez besoin pour améliorer la sécurité du réseau.
- ❑ Vous devez désactiver PAM
- ❑ Vous en avez besoin pour servir les clients sans disque.

3. Votre adresse IP est 170.35.13.28 et votre masque réseau est 255.255.255.192. Quelle adresse IP N'APPARTIENT PAS à votre réseau ?

- ❑ 170.35.13.33
- ❑ 170.35.13.88
- ❑ 170.35.13.62
- ❑ 170.35.13.55

Chapitre 8. Configuration du réseau

Objectifs	⇒ Comprendre et être capable de configurer les interfaces réseau. ⇒ Comprendre et être capable de configurer les informations de la machine. ⇒ Être capable de démarrer et d'arrêter le réseau. ⇒ Connaître les outils réseau.
Points importants	Sous Linux, comme sous Unix, tout est géré par un fichier, même les interfaces réseau. On présente d'abord les commandes de configuration du réseau puis les fichiers utilisés pour sauvegarder la configuration TCP/IP de manière permanente.
Mots clés	/etc/hosts, /etc/host.conf, /etc/HOSTNAME, /etc/resolv.conf, /etc/sysconfig/networks-scripts/ifcfg-eth0, ifconfig, ifup, route, arp, dig, host, hostname, interface, netstat, ping, réseau, routage, tcpdump, traceroute

A. Les fichiers de configuration

Le nom de la machine est indiqué dans le fichier /etc/HOSTNAME.

Les informations permettant de résoudre les noms de machines sans le mécanisme du DNS (*cf.* ci-après) se trouvent dans le fichier /etc/hosts. On le renseigne généralement avec les informations concernant la machine locale, ce qui permet aux applications de fonctionner correctement même sans connexion réseau.

```
#/etc/hosts
193.54.85.245  serveur.transfer-tic.org serveur
127.0.0.1      localhost
```

La commande hostname permet d'afficher et de modifier le nom de la machine.

Le fichier /etc/resolv.conf. est le fichier de configuration du client DNS (*Domain Name System*). On trouve dans ce fichier l'adresse IP des serveurs DNS ainsi que le domaine de recherche par défaut.

```
#/etc/resolv.conf
nameserver 134.157.9.1
nameserver 134.157.0.129
search transfer-tic.org
```

Pour vérifier le bon fonctionnement du client DNS, on utilise les commandes host ou dig.

La commande host est utilisée pour la résolution des noms de machines.

```
host www.auf.org
```

La commande dig, beaucoup plus élaborée, est utilisée en général pour obtenir de l'information sur une zone plutôt que sur une machine particulière.

```
dig lpi.org
```

Le fichier /etc/nsswitch.conf permet de spécifier quelles sources d'information consulter pour résoudre les noms de domaine, et dans quel ordre.

Par exemple on peut prévoir de chercher d'abord dans le fichier /etc/hosts puis, s'il n'y a pas de résultat, d'utiliser un service DNS :

```
hosts :   files dns
```

Les informations générales sur la configuration réseau d'une machine se trouvent dans le fichier /etc/sysconfig/network. On spécifie s'il faut démarrer automatiquement le réseau, le nom « réseau » de la machine (en principe équivalent à celui de /etc/HOSTNAME qu'il remplace en général), la passerelle, et l'interface utilisée pour joindre la passerelle.

```
#/etc/sysconfig/network/
NETWORKING=yes
HOSTNAME= serveur.transfer-tic.org
GATEWAY= 134.157.9.126
GATEWAYDEV=eth0
```

Les informations spécifiques à chaque interface se trouvent dans le répertoire /etc/sysconfig/network-scripts.

Le fichier se nomme ifcfg-<nom de l'interface>, par exemple ifcfg-eth0.

```
#/etc/sysconfig/network-scripts/ifcfg-eth0
DEVICE=eth0
BOOTPROTO=static
IPADDR=134.157.9.52
NETMASK=255.255.255.0
NETWORK=134.157.9.0
BROADCAST=134.157.9.255
ONBOOT=yes
```

Il contient essentiellement quatre informations :

- le nom de l'interface ;
- la manière d'affecter l'adresse IP (statiquement ou dynamiquement) ;
- les informations IP (adresse, masque, réseau, diffusion) ;
- s'il faut démarrer automatiquement l'interface.

Enfin, il est possible de nommer les interfaces, mais ce n'est pas normalisé et reste très dépendant de la distribution Linux utilisée.

La distribution Debian stocke les informations de toutes les interfaces réseau dans le fichier /etc/network/interfaces. On y indique pour chaque interface son nom et les informations de réseau associées.

```
auto eth0
   iface eth0 inet static
   address 192.168.10.10
   netmask 255.255.255.0
   network 192.168.10.0
   broadcast 192.168.10.255
   gateway 192.168.10.1
```

B. Démarrage et arrêt du réseau

a) Démarrage « classique »

On utilise la commande traditionnelle de Unix `ifconfig`.

Syntaxe

```
/sbin/ifconfig interface [informations reseau] [options]
```

Trois exemples :

```
ifconfig eth0 193.54.85.245 netmask 255.255.255.224 up
```

```
ifconfig eth0 up
```

```
ifconfig eth0 down
```

b) Démarrage en utilisant les fichiers de configuration

On utilise la commande `/sbin/ifup` qui récupère la configuration à partir de `/etc/sysconfig/network` et `/etc/sysconfig/network-scripts/`, fichiers cités précédemment.

```
/sbin/ifup eth0
```

Pour fonctionner, la commande recherche les informations pour l'interface « eth0 » dans le fichier `/etc/sysconfig/network-scripts/ifcfg-eth0` qui doit exister et être renseigné.

c) Démarrage de toutes les interfaces

Comme pour la plupart des services sous Linux, on peut démarrer le service « réseau » (donc toutes les interfaces) avec le script `/etc/rc.d/init.d/network` qui utilise la commande `ifup` vue précédemment pour chaque interface définie. Sur les systèmes de la famille Red Hat, les interfaces sont définies par des fichiers de configuration dans le répertoire /etc/sysconfig/network-scripts/ (ifcfg-eth0 pour l'interface « eth0 »).

```
/etc/rc.d/init.d/network start
```

Ce script récupère des informations supplémentaires dans le fichier /etc/sysctl.conf. Par exemple, on peut trouver dans ce fichier la ligne `net.ipv4.ip_forward=1`, le système va ainsi relayer (*forward*) les paquets IP entre les interfaces pour se transformer en routeur. Ceci a pour effet de modifier le contenu du fichier `/proc/sys/net/ipv4/ip_forward` en remplaçant « 0 », qui est la valeur par défaut », par « 1 ».

d) Renouvellement de bail DHCP

Les outils permettant de récupérer une nouvelle adresse IP depuis un serveur DHCP (*Dynamic Host Configuration Protocol*) sont, selon la distribution utilisée, `pump` ou `dhcpclient`.

Attention, le serveur DHCP de Linux s'appelle **dhcpd**, alors que le client se nomme **dhcpcd**.

C. Routage

Lorsque l'on utilise la commande `ifup` ou le script de démarrage du réseau, les informations concernant la passerelle sont lues directement depuis le fichier /etc/sysconfig/network (champ GATEWAY).

Lorsque l'on utilise `ifconfig` ou si la passerelle n'est pas renseignée, il faut indiquer cette passerelle (et d'autres routes éventuellement) par la commande `/sbin/route` qui permet de configurer la table de routage.

Pour ajouter une route statique vers le réseau 192.168.100.0/24 en utilisant l'interface physique eth2 qui permet de joindre la machine 192.168.2.1 qui, elle, permet d'atteindre un réseau, le 192.168.100.0/24 :

```
/sbin/route add -net 192.168.100.0/24 gw 192.168.2.1 eth2
```

Pour plus de commodité, on peut définir des réseaux par un nom au lieu de mettre la notation CIDR qui dans l'exemple est 192.168.100.0/24. Il suffit pour cela de l'indiquer dans le fichier /etc/networks.

```
mon_reseau 192.168.100.0/24
```

Pour ajouter une route par défaut vers la machine 192.168.1.1 joignable par l'interface physique eth0 :

```
/sbin/route add default gw 192.168.1.1 eth0
```

Pour afficher la table de routage du noyau Linux, on utilise la commande `/sbin/route` sans options. L'option `-n` permet d'éviter simplement la résolution de nom DNS, ce qui est commode lorsque l'on travaille sur le réseau et qu'elle n'est pas opérationnelle.

```
/sbin/route -n
Table de routage IP du noyau
Destination     Passerelle   Genmask        ...  Iface
192.168.1.0     0.0.0.0      255.255.255.0  ...  eth0
192.168.2.0     0.0.0.0      255.255.255.0  ...  eth2
192.168.100.0   192.168.2.1  255.255.255.0  ...  eth2
127.0.0.0       0.0.0.0      255.0.0.0           lo
0.0.0.0         192.168.1.1  0.0.0.0             eth0
```

La route vers le réseau 0.0.0.0 signifie « vers n'importe quel réseau » : c'est la route par défaut ou passerelle par défaut (*route add default* ... ou celle se trouvant dans le fichier /etc/sysconfig/network). Cette ligne est évidemment la dernière consultée par le système lors d'une demande d'accès à une machine : cela signifie que l'adresse demandée ne se trouve sur aucun des sous-réseaux décrits par les routes précédentes.

L'autre route peut être ajoutée automatiquement à chaque démarrage en l'indiquant dans le fichier /etc/sysconfig/static-routes.

Les informations de routage peuvent être créées automatiquement à l'aide de démons spéciaux plutôt que de les indiquer statiquement. C'est le rôle des logiciels `routed` et `gated` qui peuvent mettre à jour dynamiquement les routes en récupérant les informations depuis le réseau. Cela permet d'utiliser automatiquement des passerelles de secours en cas de défaillance d'une des passerelles.

D. Les outils associés au réseau

Voici quelques outils permettant d'effectuer des tests de fonctionnement du réseau.

a) `ping`

Cette commande envoie un paquet ICMP (ECHO_REQUEST) à une machine et attend sa réponse (ECHO_RESPONSE). Cela permet de vérifier qu'une machine est joignable et qu'elle est capable de répondre. Si c'est bien le cas, cela signifie que sa configuration réseau et la nôtre sont correctes.

Quelques options utiles pour la commande `ping` :

- `-c <N>` : envoie N paquets et stoppe ;
- `-q` : mode « calme » (*quiet*), rien n'est affiché à part les lignes de résumé au démarrage et à la fin de l'exécution ;
- `-b` : envoie le ou les paquets à un ensemble de machines (*broadcast*).

```
ping www.transfer-tic.org
PING www.transfer-tic.org (81.80.122.16) 56(84) bytes of data.
64 bytes from transfer-tic.org (81.80.122.16): icmp_seq=1
ttl=238 time=4.42 ms
64 bytes from transfer-tic.org (81.80.122.16): icmp_seq=2
ttl=238 time=4.29 ms
64 bytes from transfer-tic.org (81.80.122.16): icmp_seq=3
ttl=238 time=11.6 ms
64 bytes from transfer-tic.org (81.80.122.16): icmp_seq=4
ttl=238 time=4.20 ms
©64 bytes from transfer-tic.org (81.80.122.16): icmp_seq=5
ttl=238 time=4.88 ms
64 bytes from transfer-tic.org (81.80.122.16): icmp_seq=6
ttl=238 time=4.35 ms
```

```
64 bytes from transfer-tic.org (81.80.122.16): icmp_seq=7
ttl=238 time=3.81 ms
--- www.transfer-tic.org ping statistics ---
7 packets transmitted, 7 received, 0% packet loss, time 6060ms
rtt min/avg/max/mdev = 3.815/5.373/11.635/2.573 ms
```

b) netstat

Donne des informations générales sur la configuration réseau, à savoir les tables de routage, les statistiques des interfaces, etc.

Quelques options de `netstat` :

- `-n` : ne pas résoudre les noms ;
- `-r` : affiche la table de routage, équivalent à la commande `route` ;
- `-v` : mode verbeux ;
- `-l` : liste des connexions/interfaces ;
- `-c` : mise à jour permanente.

```
netstat -n
Connexions Internet actives (sans serveurs)
Proto  Recv-Q  Send-Q  Adresse locale    Adresse distante    Etat
tcp    0       0       134.157.9.32:22   134.157.9.52:46903  ESTABLISHED
tcp    0       0       134.157.9.32:22   134.157.9.52:46639  ESTABLISHED
tcp    0       0       134.157.9.32:800  134.157.9.11:2049   ESTABLISHED
tcp    0       0       134.157.9.32:643  134.157.9.11:111    TIME_WAIT
```

On remarque deux connexions sur le port 22, une sur le port 800 et une sur le port 643.

c) arp

Cette commande permet d'afficher la table cache de résolution d'adresses du noyau (*i.e.* l'association avec l'adresse physique MAC d'une machine) pour les machines présentes sur le même réseau.

```
arp
Address                  HWtype HWaddress          Flags Mask Iface
cerbere.lodyc.jussieu.f ether  00:B0:D0:D1:8B:2A  C          eth0
nestor.lodyc.jussieu.fr ether  00:04:76:E4:BB:33  C          eth0
```

d) traceroute

Affiche les différents nœuds (ou passerelles) traversés pour joindre une machine.

Quelques options de `traceroute` :

- `-n` : ne pas résoudre les noms (DNS) ;
- `-v` : mode verbeux ;
- `-f <ttl>` : change le TTL (*Time To Live*) ;
- `-w <sec>` : change le *time-out* sur les paquets retournés.

```
traceroute -n www.transfer-tic.org
traceroute to www.transfer-tic.org (81.80.122.16), 30 hops max,
38 byte packets
1  134.157.9.126     0.640 ms   0.265 ms   0.242 ms
2  134.157.247.238   5.000 ms   0.655 ms   0.545 ms
3  134.157.254.126   1.010 ms   0.830 ms   0.826 ms
4  195.221.127.181   1.656 ms   1.130 ms   1.083 ms
5  193.51.181.102    1.447 ms   1.028 ms   1.097 ms
6  193.51.180.158    1.888 ms   1.826 ms   1.771 ms
7  193.51.179.1      1.030 ms   1.443 ms   1.383 ms
8  193.51.185.1      1.815 ms   1.384 ms   1.100 ms
9  193.251.241.97    1.361 ms   1.675 ms   1.438 ms
```

La commande `traceroute` force les nœuds intermédiaires traversés à renvoyer une réponse qui est un message d'erreur (ICMP TIME_EXCEEDED), en positionnant une valeur de TTL trop basse. Dès qu'un message d'erreur est reçu, la commande incrémente cette valeur et renvoie le message ce qui lui permet de passer au nœud suivant et ainsi de suite.

L'utilitaire `tcpdump` permet de visualiser tous les échanges se produisant sur le réseau auquel votre machine est connectée.

Cette commande permet le débogage de problèmes réseau, mais aussi de récupérer certaines informations qui circulent « en clair » sur le réseau.

```
# surveillance de la machine 192.168.10.1
tcpdump src 192.168.10.1
```

E. Exercices

1. Quel est le fichier utilisé pour associer les noms symboliques et les adresses IP des machines de votre réseau ?

- ❑ /etc/nsswitch.conf
- ❑ /etc/resolv.conf

- ❑ /etc/hosts
- ❑ /etc/services

2. Quelle commande va créer une route par défaut avec comme passerelle 192.168.1.1 ?

- ❑ netstat-add default gw
- ❑ route default 192.168.1.1
- ❑ ip route default 192.168.1.1
- ❑ route add default gw 192.168.1.1
- ❑ ifconfig default gw 192.168.1.1 eth0

3. Lesquelles des commandes suivantes sont utilisées pour activer une interface réseau? (deux réponses)

- ❑ ifconfig
- ❑ netstat
- ❑ ifup
- ❑ ifstart

4. Vous soupçonnez que l'une des passerelles de votre réseau ne fonctionne plus mais vous ne savez pas laquelle. Quelle commande va vous aider à résoudre le problème ?

- ❑ ps
- ❑ netstat
- ❑ nslookup
- ❑ ifconfig
- ❑ traceroute

Chapitre 9. Services systèmes de base

Objectifs	⇒ Connaître les principaux serveurs SMTP et être capable de faire suivre les courriers (*forwarding*) et de configurer les alias. ⇒ Être capable de conserver l'heure système et de synchroniser l'horloge via le protocole NTP.
Mots clés	~/.forward, /etc/ntp.conf, date, Exim, hwclock, newaliases, mail, mailq, ntpd, ntpdate, pool.ntp.org, Postfix, Qmail, Sendmail

A. Maintien de l'horloge du système

Il existe deux types d'horloges sur l'architecture x86, une horloge matérielle et une horloge logicielle.

L'horloge matérielle, conservée par le BIOS, maintient l'heure lorsque l'ordinateur est éteint.

Lorsque le système démarre, il lit l'horloge matérielle et règle l'horloge logicielle à la valeur qu'il récupère. Il utilise ensuite l'horloge logicielle pour ses besoins et ceux de ses processus.

a) Configuration manuelle des horloges matérielle et logicielle

La commande `date` permet de gérer l'horloge logicielle du système, alors que la commande `hwclock` permet de régler l'horloge matérielle à partir de l'horloge logicielle et vice-versa.

La syntaxe de la commande `date` est :

```
date [option] … [+Format]
```

```
date [-u|--utc|--universal] [MMJJhhmm[[CC]YY][.ss]]
```

La commande `date` sans arguments permet d'afficher l'horloge logicielle :

```
$ date
lun. déc. 14 11:00:25 CET 2009
```

On peut personnaliser cet affichage par l'utilisation des options de formatage de la commande `date` :

```
$ date +"%d-%m-%Y"
14-12-2009
[zied@ankara ~]$ date +"%A,%d %B %Y"
lundi,14 décembre 2009
```

Pour modifier uniquement la date du système :

```
# date -s 01/01/2010
ven. janv.  1 00:00:00 CET 2010
```

Pour modifier uniquement l'heure du système :

```
# date -s 12:12:59
ven. janv.  1 12:12:59 CET 2010
```

Pour afficher l'horloge matérielle :

```
# hwclock -r
lun. 14 déc. 2009 12:02:30 CET  -0.504123 secondes
```

Pour mettre à jour l'horloge système par rapport à l'horloge matérielle :

```
# hwclock -s (ou bien hwclock --hctosys)
# date
lun. déc. 14 12:02:42 CET 2009
```

Pour modifier l'heure du système et affecter cette modification à l'horloge matérielle :

```
# date -s 11:00:00
lun. déc. 14 11:00:00 CET 2009
# hwclock -w
# date
lun. déc. 14 11:00:25 CET 2009
# hwclock -r (ou bien hwclock --systohs)
lun. 14 déc. 2009 11:00:42 CET  -0.895758 secondes
```

b) Le protocole NTP : *Network Time Protocol*

Maintenir une horloge précise est important sous Linux. Plusieurs services et programmes ont besoin ou tirent partie de l'horloge du système. En effet, l'horodatage est utilisé dans les journaux. Les programmes `cron` et `make` ont besoin des dates précises des modifications des fichiers. L'horodatage est également inclus dans les en-têtes des courriers électroniques. Certains protocoles d'authentification, tel que Kerberos, doivent s'assurer de la synchronisation des horloges des machines.

Le protocole NTP permet de synchroniser l'horloge d'un ordinateur avec celle d'un serveur de référence. Il crée une hiérarchie à plusieurs niveaux de sources de temps. Au sommet, une ou plusieurs sources de temps très précises telles que des horloges atomiques. Ces sources sont désignées par **strate 0**, elles sont directement reliées aux serveurs NTP de **strate 1**.

Les serveurs NTP de **strate 1** offrent le temps aux serveurs de **strate 2**, qui fournissent le temps aux serveurs **strate 3**, et ainsi de suite.

Le protocole NTP prévoit jusqu'à 16 strates, mais la plupart des clients se situent dans les strates 3 et 4.

c) Configuration de base du serveur NTP

Le paquetage NTP comporte plusieurs paquetages, notamment le démon `ntpd` et un certain nombre de programmes utilisés pour configurer et interroger le serveur NTP.

`ntpd` utilise le fichier de configuration /etc/ntp.conf qui contient plusieurs options dont les plus importantes sont :

- `restrict`, pour définir des contrôles d'accès au serveur `ntpd` ;
- `server`, pour rediriger le serveur `ntpd` vers un serveur NTP.

Voici quelques options extraites du fichier de configuration /etc/ntp.conf :

```
driftfile /var/lib/ntp/drift
# Permit time synchronization with our time source, but do not
# permit the source to query or modify the service on this
system.
restrict default kod nomodify notrap nopeer noquery
restrict -6 default kod nomodify notrap nopeer noquery
server 0.pool.ntp.org
server 1.pool.ntp.org
server 2.pool.ntp.org
server 3.pool.ntp.org
```

Lorsque le démon `ntpd` démarre, il contacte tous les serveurs spécifiés dans le fichier /etc/ntp.conf par l'option `server`, il compare la précision de leurs

horloges et s'ajuste par rapport à un seul serveur qui le marque comme sa source de temps primaire.

La commande `ntpq` est utilisée pour envoyer des messages de contrôle NTP à un hôte pour vérifier l'état du démon `ntpd` ou changer sa configuration. La syntaxe est la suivante :

```
ntpq [options] [host]
```

La commande `ntpq` peut être exécutée soit en mode interactif, soit avec des arguments passés à la ligne de commande.

Si l'argument `host` n'est pas spécifié, la requête est envoyée à la machine locale.

L'exemple suivant montre quatre serveurs externes connectés au serveur NTP local.

```
# ntpq -c peers
remote           refid         st t  when …  delay    …  jitter
================================================================
196.216.249.2    146.164.48.5  2  u  243  …  486.574  …  12.956
+147.137.46.196  146.64.8.7    3  u  178  …  348.080  …  7.811
*196.7.156.83    146.64.8.7    3  u  404  …  223.383  …  5.933
ntp.dts.mg       193.50.27.66  3  u  174  …  678.084  …  42.805
```

La colonne « refid » montre le serveur sur lequel chaque système est synchronisé et la colonne « st » indique la strate des serveurs externes.

Le serveur avec lequel le serveur NTP local est synchronisé est marqué par une astérisque (*), les serveurs avec de bonnes précisions sont marqués par un signe plus (+) et les autres symboles (x ou -) désignent les serveurs rejetés pour diverses raisons.

La commande `ntpdate` est utilisée pour synchroniser l'horloge du système avec un serveur NTP.

Exemple :

```
# ntpdate -s ntp.loria.fr
```

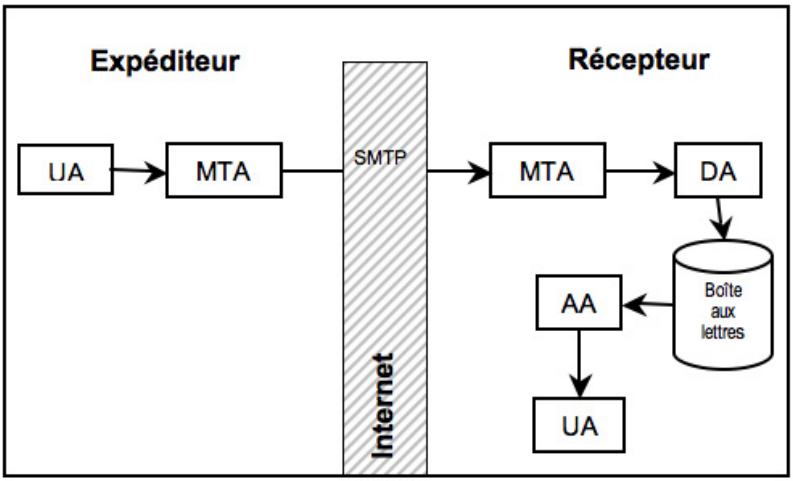
Mais la commande `ntpdate` est dépréciée, et pourrait disparaître du paquetage NTP à tout moment. À sa place on peut utiliser la commande `ntpd` avec l'option `-g`.

B. Le courrier électronique

Un système de messagerie sous Linux se compose de quatre éléments distincts (*figure 5*) :

- **MUA**, *Mail User Agent*, qui permet aux utilisateurs de lire, écrire et gérer leurs messages. Exemples de programmes MUA : Thunderbird de Mozilla, Evolution de Aka Novell, Outlook de Microsoft, la commande `/bin/mail` avec Unix et Linux ;
- **MTA**, *Mail Transport Agent* ou Agent de transport de courrier, dont le rôle est de router les messages entre les machines ;
- **MDA**, *Mail Delivery Agent*, qui permet de stocker le courrier dans la boîte aux lettres du destinataire ;
- **AA**, *Access Agent*, composant optionnel dont le rôle est de connecter le MUA à la boîte aux lettres à travers par exemple les protocoles IMAP et POP.

Figure 5. Architecture du système de messagerie



a) MTA ou Agent de transport de courrier

L'agent MTA accepte le message à partir de l'agent UA, il se charge alors de l'acheminer vers sa destination, pour cela il doit vérifier l'existence de l'expéditeur ainsi que du ou des destinataires.

Les agents MTA communiquent à travers le protocole SMTP (*Simple Mail Transport Protocol*).

Linux et Unix supportent plusieurs agents MTA, quatre agents sont les plus populaires :

- **Sendmail** : c'est le système le plus utilisé et le plus ancien. Il transmet plus de 50 % du courrier sur Internet. La configuration de Sendmail est

très complexe à cause de son fichier de configuration abscons. Sendmail est un système monolithique : il utilise un seul programme pour tous les traitements. La performance et la sécurité ont été améliorées avec les dernières versions ;

- **Postfix** : conçu comme une alternative modulaire à Sendmail, Postfix utilise plusieurs programmes qui gèrent chacun une tâche spécifique. En théorie cette conception améliore la sécurité. Postfix est rapide et plus facile à configurer que Sendmail ;
- **Exim** : les distributions Debian et Ubuntu utilisent Exim comme serveur de courrier par défaut. C'est un serveur monolithique, facile à configurer et avec un petit nombre de fichiers journaux ;
- **Qmail** : c'est un système modulaire, avec la sécurité comme objectif de conception majeur. Le code source de Qmail est disponible gratuitement mais, pour des raisons de sécurité, il est interdit de distribuer une version modifiée de Qmail sans l'accord préalable de l'auteur. Avec cette interdiction, la maintenance est devenue assez difficile. L'installation est assez compliquée, en effet il faut utiliser des *patches* pour obtenir un serveur sans bogues.

b) Gestion des courriers électroniques

La commande `mail` est un MUA permettant à l'utilisateur d'envoyer et de recevoir des courriers textuels à partir de l'invite de commandes du shell.

Par exemple l'utilisateur mejdi envoie un message à nicolas et une copie de ce message à zied :

```
[mejdi@ankara ~]$mail -s "Confirmation de réunion" -cc
zied@ankara nicolas@ankara
La réunion d'aujourd'hui est confirmée pour 15h
```

Les utilisateurs nicolas et zied peuvent ainsi consulter leurs boîtes aux lettres :

```
[zied@ankara ~]$ mail
Heirloom Mail version 12.4 7/29/08.  Type ? for help.
"/var/spool/mail/zied": 1 message 1 new
>N  1 mejdi@localhost.loca  Sun Dec 13 08:12  21/877
"Confirmation de réuni"
&
[nicolas@ankara ~]$ mail
Heirloom Mail version 12.4 7/29/08.  Type ? for help.
"/var/spool/mail/nicolas": 1 message 1 new
>N  1 mejdi@localhost.loca  Sun Dec 13 08:12  21/877
"Confirmation de réuni"
```

```
&
```

Les messages reçus sont enregistrés dans des boîtes aux lettres qui sont de simples fichiers. Ils se trouvent dans le répertoire /var/spool/mail/.

L'utilisateur peut gérer sa boîte aux lettres de façon interactive avec la commande `mail`. Il peut ainsi lire et supprimer les messages reçus ou répondre aux expéditeurs. Mais la gestion d'une boîte aux lettres avec le MUA `mail` n'est possible que sur la machine locale.

L'exemple précédent illustre la description des messages dans la boîte aux lettres de l'utilisateur : les boîtes de nicolas et de zied contiennent un message, non lu (*new*), l'expéditeur est mejdi, la date d'arrivée le 13 décembre à 08h12, le message contient 21 lignes, a une taille de 877 octets et son sujet est « Confirmation de réuni ».

Pour lire le message, on tape son numéro (ici c'est le numéro 1). On peut aussi le supprimer par la commande `d` ou bien y répondre avec la commande `r`.

Les messages en attente d'envoi sont stockés dans `/var/spool/mqueue`. Leur liste peut être affichée avec la commande `mailq`.

Le MTA Sendmail permet d'avoir plusieurs noms, ou alias (*aliases* en anglais), pour la même boîte aux lettres. On utilise pour cela le fichier /etc/aliases.

Par exemple, pour créer l'alias « nicolas.larrousse » pour un utilisateur dont le nom réel est « nicolas » :

```
nicolas.larrousse:  nicolas
```

Après avoir modifié le fichier /etc/aliases il faut lancer la commande `newaliases` pour mettre à jour la base des alias de Sendmail. Cette commande génère le fichier /etc/aliases.db qui est un fichier binaire indexé.

Un utilisateur peut rediriger les messages reçus vers la boîte aux lettres d'un autre utilisateur. Pour cela il indique l'adresse électronique de l'autre utilisateur dans le fichier `~/.forward` créé dans son répertoire personnel.

Par exemple l'utilisateur zied veut rediriger tous les messages reçus vers la boîte aux lettres niry. Il ajoute alors l'adresse électronique de la boîte niry dans le fichier ~/.forward.

```
niry@ankara
```

C. Exercices

5. Quelle commande allez-vous utiliser afin de mettre à jour l'horloge machine par rapport à l'horloge système ?

- ❑ date --sethwclock
- ❑ ntpdate
- ❑ hwclock --utc --systohc
- ❑ time --set –hw

6. Quels sont les moyens valides pour modifier l'heure de votre système ?

- ❑ ntpdate serverntp
- ❑ date -s
- ❑ date
- ❑ ntp -update

7. Salah va partir en vacances pendant deux semaines et veut que son courrier soit transmis à Ali pendant son absence. Quels changements devrait-il faire ?

- ❑ Ajouter « ali » à /etc/aliases.
- ❑ Ajouter « ali » à ~/.forward.
- ❑ Ajouter « ali » à ~/.aliases.
- ❑ Ajouter « ali » à /etc/mail.

Chapitre 10. La sécurité

Objectifs	⇒ Comprendre les risques et être capable d'appliquer les consignes de sécurité de base. ⇒ Savoir aborder la sécurité selon ses deux aspects fondamentaux : la sécurité machine et la sécurité réseau.
Points importants	Garantir la sécurité d'une machine connectée à un réseau n'est pas quelque chose d'évident. Cela dépend d'un grand nombre de paramètres qui ne sont jamais purement techniques. La sécurité du point de vue de la machine pose la question suivante : comment protéger le système contre quelqu'un qui dispose d'un accès à la machine ? En d'autres termes, le ver est déjà dans le fruit, avec soit un accès physique soit un compte qui lui permet de se connecter au système. La sécurité réseau aborde le problème d'un autre point de vue : comment nous protéger de quelqu'un qui va attaquer notre machine par le réseau (par exemple en utilisant les services que notre système offre à l'extérieur) ?
Mots clés	/etc/fstab, /etc/hosts.allow, /etc/hosts.deny, /etc/nologin, /etc/ssh_known_hosts, /etc/ssh/sshd_config, /etc/sshrc, /etc/security/access.conf, /etc/security/limits.conf, /proc/net/ip_fwchains, /proc/net/ip_fwmasquerade, /proc/net/ip_fwnames, authorized_keys, BIOS, ipchains, iptables, known_hosts, lilo, netfilter, OpenSSH, rsa, SSH, ssh-keygen, syslog, TCP wrapper

A. Les fichiers de configuration

Nous allons donner ici quelques éléments constitutifs d'une démarche de sécurisation de l'accès à notre machine.

a) Configuration du BIOS

Le cas où quelqu'un de malveillant dispose d'un accès physique à la machine est grave et quasiment insoluble. Seul le cryptage des données du système peut alors représenter une protection relativement fiable.

Il est tout de même possible de créer certaines difficultés. Le premier pas est de configurer le BIOS pour que la machine ne puisse démarrer qu'à partir du disque dur. Cela va empêcher le pirate de démarrer simplement d'une disquette ou d'un cédérom pour obtenir ensuite l'accès au système.

b) Restrictions de LILO

Le chargeur de démarrage LILO permet de passer des options au démarrage.

L'une de ces options permet de démarrer le système en « *single user mode* ». Dans certaines distributions de Linux, cette option ouvre un accès *shell* au système muni des permissions de l'utilisateur *root* sans même demander un mot de passe.

Pour éviter cela nous disposons de deux options de configuration de LILO : `restricted` et `password = '...'`. La première indique que seul l'utilisateur qui connaît le mot de passe, spécifié par la deuxième option, peut donner des options au démarrage.

c) Permissions des fichiers et répertoires

Une bonne pratique générale est de ne pas permettre à des programmes d'être exécutés à partir de /home et /tmp. Cela empêche les utilisateurs d'utiliser des logiciels qu'ils auraient installés et d'essayer d'attaquer le système par des outils téléchargés de l'Internet ou programmés par eux-mêmes.

Pour ce faire, nous pouvons utiliser les options de montage de systèmes de fichiers suivantes (/etc/fstab) :

```
/tmp /tmp ext2 nosuid 1 2
/home /home ext2 noexec 1 2
```

d) Analyser le système

Pour assurer la sécurité du système, il faut surveiller son fonctionnement. La source d'information la plus fiable pour visualiser l'activité de notre système est la consultation des fichiers de *logs*. Le démon `syslog` enregistre normalement ces informations dans /var/log/messages. Certaines distributions de Linux « loguent » l'information relative à la sécurité dans /var/log/secure : nouveaux utilisateurs ajoutés au système, échecs de connexion, etc.

La commande `w` ou `who` affiche les utilisateurs connectés au système. Elle utilise les informations du fichier de logs /var/log/utmp.

La commande `last` affiche la liste des derniers utilisateurs du système. Elle cherche ces informations dans le fichier /var/log/wtmp.

e) Des limites pour les utilisateurs

Certaines limites et restrictions peuvent être imposées aux utilisateurs.

Les restrictions d'accès pour certains utilisateurs et groupes d'utilisateurs sont configurées dans le fichier /etc/security/access.conf.

Les limitations d'utilisation sont configurées dans le fichier /etc/security/limits.conf dont chaque ligne est de la forme suivante :

```
<domaine>        <type>  <élément>  <valeur>
```

- `<domaine>` peut être un nom d'utilisateur, un nom de groupe d'utilisateurs, ou le signe « * » pour indiquer une entrée par défaut ;
- `<type>`, définit le type de limite à mettre en place : soit d'avertissement, avec le type `soft`, soit de blocage, avec le type `hard` ;
- `<élément>` définit la ressource à limiter pour le domaine en question (utilisateur ou groupe) : `core`, `data`, `fsize`, `memlock`, `nofile`, `rss`, `stack`, `cpu`, `nproc`, `as`, `maxlogins`, `priority`...

Le fichier /etc/nologin permet d'empêcher toute connexion au système autre que celle du super-utilisateur. Il contient le message que recevront les utilisateurs lors de la tentative de connexion, par exemple « système en maintenance ».

On peut également définir des quotas par utilisateurs (*cf.* support LPI 101). La commande `quota` (ou `repquota`) permet de vérifier l'utilisation des quotas par les utilisateurs.

B. Sécurité réseau

Comment contrôler l'accès à un serveur ? Nous pouvons le faire de deux manières différentes :

- restreindre l'accès en utilisant l'adresse de la machine qui se connecte au serveur ;
- restreindre l'accès en utilisant le port TCP ou UDP (service réseau) auquel le client essaie de se connecter.

a) TCP wrappers

Ce contrôle est assuré par la bibliothèque **libwrap**. Une grande partie des applications réseaux est compilée en utilisant cette bibliothèque. Cela permet de configurer l'accès à ces applications par deux fichiers, /etc/hosts.allow et /etc/hosts.deny. Chaque ligne de ces fichiers contient deux champs (ou éventuellement trois comme on verra un peu plus bas) séparés par le signe « : » (deux points) :

- le premier champ décrit le service pour lequel on ajoute des restrictions ou des permissions d'accès ;
- le deuxième décrit la liste des machines pour lesquelles cette règle s'applique.

Les mots-clés ALL et EXCEPT servent pour la spécification de ces deux champs.

Pour donner la liste des adresses des machines clientes on peut utiliser soit des noms de domaines commençant éventuellement par un point, soit des adresses IP se terminant éventuellement par un point. Par exemple la spécification .auf.org comprend tous les noms de domaines suffixés par « .auf.org ». De la même manière 10.1.1. donne la liste de toutes les adresses IP dans le réseau 10.1.1.0/24 (toute adresse IP commençant par 10.1.1.).

```
/etc/hosts.deny
ALL:       ALL    EXCEPT .auf.org

/etc/hosts.allow
ALL:       LOCAL  192.168.0.
in.ftpd:          ALL
sshd:     .auf.org
```

Par les « *tcp wrappers* » il est aussi possible de configurer l'exécution d'une commande. Pour faire cela on utilise le mot-clé spawn.

L'exemple suivant va garder une trace de chaque tentative de connexion à un des services de notre machine :

```
/etc/hosts.deny
ALL:  ALL : spawn (/bin/echo `date` client : %c service : %d >>
/var/log/tcpwrap.log)
```

La page de manuel **host_access (5)** est une bonne source d'information sur la configuration des *tcp wrappers* et en particulier sur les macros commençant par « % ».

Il faut tout de même garder en mémoire que cette technique de contrôle d'accès n'est fiable que si elle est utilisée conjointement avec d'autres démarches de sécurisation du réseau qui vont empêcher le camouflage de l'identité de la machine cliente.

b) Filtrage de paquets

Le noyau de Linux dispose d'un filtre de paquets très puissant. Ce filtre nous permet de contrôler l'accès à notre machine par le réseau en utilisant plusieurs critères comme :

- l'adresse source de l'émetteur du paquet (l'adresse IP du client de notre serveur) ;
- le port TCP ou UDP source ;
- le port TCP ou UDP de destination (le port de notre service réseau) ;
- etc.

La commande qui permet de gérer les règles de filtrage dans les noyaux de Linux de la série 2.2 est `ipchains`. La commande correspondante dans les noyaux 2.4.x et 2.6.x est `iptables`, elle configure le filtre **netfilter**.

`Ipchains` et `iptables` utilisent trois chaînes de règles : `input`, `output` et `forward` pour `ipchains` et `INPUT`, `OUTPUT` et `FORWARD` pour `iptable`. L'utilisateur peut définir des chaînes supplémentaires pour mieux structurer son filtre.

Avec `ipchains` chaque paquet qui n'est pas destiné à la machine filtre passe par la chaîne `forward`. En revanche, avec `iptables` la chaîne `FORWARD` est traversée uniquement par des paquets qui ne proviennent pas de la machine filtre et ne lui sont pas destinés.

Les chaînes `input` et `output` (ou respectivement `INPUT` et `OUTPUT` pour `iptables`) représentent les paquets qui entrent ou sortent de la machine filtre.

Les options suivantes permettent de gérer les règles de filtrage :

- `-A` pour ajouter une nouvelle règle dans une chaîne ;
- `-D` pour supprimer une règle d'une chaîne ;
- `-P` pour modifier la politique par défaut ;
- `-I` pour insérer une règle ;
- `-F` pour effacer les règles d'une chaîne ;
- `-N` pour créer une nouvelle chaîne ;
- `-X` pour supprimer une chaîne créée par l'utilisateur ;
- `-L` pour afficher la liste des règles.

Pour stopper l'accès de toutes les machines du réseau 10.1.1.0/24 au service `ssh` de la machine filtre nous allons ajouter la règle suivante :

```
iptables -A INPUT -s 10.1.1.0/24 -p tcp --dport 22 -j DROP
```

Le grand avantage de `iptables` est la possibilité de filtrer les paquets en considérant l'état de la connexion à laquelle le paquet appartient. Cette fonctionnalité est réalisée par le module *state* d'`iptables` (« `-m state` » sur la ligne de commande pour indiquer l'utilisation de ce module). À l'aide de ce module, il est par exemple possible de savoir si un paquet appartient à une connexion déjà établie ou bien s'il s'agit d'un paquet qui essaie d'ouvrir une nouvelle connexion en relation avec une autre connexion déjà établie (exemple du fonctionnement du protocole FTP qui ouvre des connexions dynamiques). Cette fonctionnalité permet de résoudre le problème du non filtrage des connexions sortantes et de celles qui leur sont associées. Cela concerne tous les services qui ouvrent des ports dynamiques.

Pour ajouter une règle qui laisse passer tous les paquets appartenant à une connexion déjà établie (ESTABLISHED) ou à une connexion en relation (RELATED) avec une autre déjà établie :

```
iptables -A INPUT -m state --state ESTABLISHED,RELATED -j ACCEPT
```

L'exemple ci-dessous montre un script bash réalisant un filtrage simple sur une machine utilisée comme passerelle pour un réseau local. Aucune connexion vers cette machine n'est autorisée, toutes les connexions qui proviennent de la machine même sont acceptées et tous les paquets du réseau local sont « masqués » (MASQUERADE), c'est-à-dire que leurs adresses IP sources sont remplacées par l'adresse IP de l'interface externe de notre passerelle :

```
#!/bin/bash

I_INT=eth0 # l'interface locale
I_EXT=eth1 # l'interface externe
IP_LOCAL=10.1.1.0/24 # le réseau IP interne

# les politiques par défaut
iptables -P INPUT DROP
iptables -P OUTPUT DROP
iptables -P FORWARD DROP


# le masquerading des adresses privées
iptables -t nat -A POSTROUTING -s $IP_LOCAL -i $I_INT -o $I_EXT
-j MASQUERADE
```

```
# accepter toute connexion initiée par le réseau local
iptables -A FORWARD -s $IP_LOCAL -i $I_INT -o $I_EXT -j ACCEPT
iptables -A FORWARD -m state -state RELATED,ESTABLISHED -j
ACCEPT


# Permettre toutes les connexions initiées par le pare-feu
iptables -A OUTPUT -j ACCEPT
iptables -A INPUT -m state -state RELATED,ESTABLISHED -j ACCEPT
```

Les informations concernant `ipfilter` sont transmises au noyau par trois fichiers situés dans l'arborescence proc :

- /proc/net/ip_fwchains qui contient les règles ;
- /proc/net/ip_fwnames qui contient le nom des chaînes ;
- /proc/net/ip_fwmasquerade qui contient les informations pour le « *masquerading* ».

c) Le shell sécurisé (SSH)

Le protocole SSH permet de se connecter en mode sécurisé à une machine à distance. Il effectue :

- le chiffrement de la connexion ;
- l'authentification de la machine serveur ;
- l'authentification des utilisateurs.

Il se décline en deux versions : la version 1 du protocole utilise l'algorithme de chiffrement **RSA1**, et la version 2 utilise soit l'algorithme **RSA** soit l'algorithme **DSA**.

L'authentification et le chiffrement sont réalisés sur la base d'algorithmes de chiffrement asymétriques. Une paire de « clé publique/clé privée » est générée pour le serveur et éventuellement une paire de clés est générée pour chaque utilisateur pour s'authentifier de cette manière.

Sous Linux on utilise en général OpenSSH, version libre du protocole SSH.

Par défaut la configuration de OpenSSH ainsi que les clés publique/privée sont enregistrés sous l'arborescence /etc/ssh.

La connexion SSH tient compte de la présence du fichier /etc/nologin décrit plus haut.

d) Authentification du serveur

Quand nous ouvrons une connexion SSH vers un serveur, nous devons nous assurer de l'identité de ce serveur. Pour cela, il envoie sa clé publique. Si c'est la première fois que nous nous connectons à ce serveur la question suivante apparait :

```
The authenticity of host machin (10.1.1.8)' can't be
established.
RSA key fingerprint is
8f:29:c2:b8:b5:b2:e3:e7:ec:89:80:b3:db:42:07:f4.
Are you sure you want to continue connecting (yes/no)?
```

Si nous acceptons, la clé publique sera enregistrée dans le fichier $HOME/.ssh/known_hosts.

Lors de la prochaine connexion à la même machine la question n'apparaîtra plus car l'identité de la machine sera connue.

Il faut admettre qu'accepter une clé publique envoyée par le réseau n'est pas parfaitement fiable. Il est préférable de récupérer cette clé directement sur une disquette ou une clé USB par exemple et de l'ajouter directement dans le fichier known_hosts.

Il est possible de configurer les clients SSH pour qu'ils n'enregistrent pas de clé dans ce fichier et de ne permettre la connexion qu'à des machines décrites dans le fichier /etc/ssh_known_hosts de façon à limiter les risques d'usurpation d'identité par une machine qui répondrait à la place de celle que l'on voulait contacter.

Sur le serveur, ses propres clés publiques/privées sont enregistrées par défaut dans /etc/ssh. Pour la version 1 le fichier contenant la clé privée est nommé ssh_host_key. Pour la version 2 il est nommé ssh_host_rsa_key pour l'algorithme RSA et ssh_host_dsa_key pour l'algorithme DSA.

Les fichiers contenant les clés publiques utilisent la même désignation avec l'extension .pub (par exemple ssh_host_dsa_key.pub).

e) Authentification de l'utilisateur

L'authentification de l'utilisateur peut se faire soit par mot de passe, soit, de façon beaucoup plus fiable, par une paire de clés publique et privée.

Pour une authentification par nom d'utilisateur et mot de passe l'utilisateur est invité à les entrer au clavier lors de la connexion. Le serveur effectue classiquement la vérification à partir des fichiers /etc/passwd et/ou /etc/shadow.

Pour réaliser une authentification par clés publique/privée il est nécessaire de générer ces clés. On utilise pour cela la commande ssh-keygen.

```
ssh-keygen -t rsa -b 1024
```

Par défaut cette commande enregistre les clés dans le répertoire $HOME/.ssh/ sous les noms **id_rsa** pour la clé privée et **id_rsa.pub** pour la clé publique.

Pour permettre au serveur d'authentifier un utilisateur à l'aide de sa clé publique, il faut enregistrer cette clé dans le fichier $HOME/.ssh/authorized_keys de l'utilisateur.

Il ne faut pas oublier de sécuriser les fichiers contenant les clés privées (dans les répertoires /etc/ssh et $HOME/.ssh) en leurs donnant les permissions d'accès 600.

f) Configuration de OpenSSH

La configuration du serveur sshd se fait par le fichier /etc/ssh/sshd_config.

Voici quelques options intéressantes extraites de ce fichier :

```
Port 22
Protocol 2,1
ListenAddress 0.0.0.0

# Clé d'authentification de l'hôte pour la version 2
HostKey /etc/ssh/ssh_host_key
# Clés d'authentification de l'hôte pour la version 2
HostKey /etc/ssh/ssh_host_rsa_key
HostKey /etc/ssh/ssh_host_dsa_key
```

La configuration du client SSH se fait dans le fichier /etc/ssh/ssh_config génériquement pour tous les utilisateurs ou dans $HOME/.ssh/config pour une configuration spécifique pour un utilisateur.

Le fichier /etc/sshrc contient les commandes à exécuter lors de la connexion d'un client SSH, donc selon le même principe qu'un fichier /etc/profile, mais adapté aux connexions SSH.

C. Exercices

1. **Avec le protocole SSH, pour réaliser une authentification par clé publique/clé privée, l'utilisateur doit enregistrer sa clé publique RSA dans le serveur SSH dans le fichier :**

   - ~/.ssh/id_rsa
   - ~/.ssh/known_hosts

- ~/.ssh/authorized_keys
- /etc/ authorized_keys

2. **Créer un script pour la mise en place des règles de filtrage simples suivantes :**

   – les politiques par défaut sont DROP ;

   – masquer les connexions initiées par le réseau local ;

   – accepter toute connexion initiée depuis le réseau local.

3. **Installer et configurer le serveur sshd.**

   Créer une paire de clés pour un utilisateur et configurer la connexion au serveur par clés publique/privées

Annexe 1 : exemple d'examen de certification 102

Voici un exemple d'énoncé d'examen 102, suivi des réponses.

Cet exemple est destiné à vous aider à évaluer vos connaissances, et également à vous préparer au style des questions posées lors de l'examen de certification. Il est en effet nécessaire de s'habituer à la formulation des questions, qui peut parfois paraître ambiguë, ainsi qu'aux questions à choix multiples qui sont courantes dans le monde anglo-saxon.

Lors de l'examen, vous disposez d'environ une minute par question.

Questions

1. **Quelle variable d'environnement allez-vous utiliser pour afficher le code de retour de la dernière commande ?**

2. **Quel ordre SQL allez-vous utiliser pour modifier des tuples dans une base de données SQL ?**

3. **Vous essayez d'exécuter la commande ls mais il existe un alias de la commande « ls ». Quelle est la manière la plus simple pour exécuter la commande originale ls et non pas son alias ?**

4. **Donnez le nom (chemin complet) du fichier qui contient le message qui est affiché à l'utilisateur au moment de l'ouverture d'une session shell.**

5. **Quel et le nom du démon NTP utilisé pour synchroniser l'horloge du système?**

6. **Quel est le nom de l'utilitaire permettant d'ajuster les différents modes graphiques ?**

7. **Quel fichier doit modifier l'utilisateur dans son répertoire personnel pour configurer la variable d'environnement PATH ? Donnez seulement le nom du fichier, sans le chemin d'accès.**

8. **Quel est le fichier qui contient la configuration de l'environnement standard pour tout le système ? Ce fichier contient normalement la variable PATH, ainsi que umask et ulimit. Donnez le chemin d'accès complet.**

9. **Quelle commande indique au serveur X d'accepter les connexions des clients X à partir de la machine tunis ?**

10. **Quel est le chemin d'accès complet au fichier qui contient la configuration du démon utilisé pour constituer les logs du système ?**

11. **Vous voulez redémarrer le service réseau d'un serveur Red Hat. Quelle commande allez-vous exécuter pour accomplir cette opération sans avoir besoin d'utiliser un chemin absolu ?**

12. **Quelle est la section du fichier xorg.conf qui contient le chemin d'accès vers les polices de caractères ?**

13. **Quel est le protocole utilisé pour offrir une interface de connexion graphique sur un réseau TCP/IP ?**

14. **Quel fichier contient les adresses IP des serveurs DNS que la machine va utiliser pour la résolution de noms ? Donnez le chemin d'accès complet.**

15. **Quel programme vous permet d'avoir un clavier d'écran ?**

16. **Quel est le fichier de configuration du fuseau horaire du système ? Donner le chemin complet.**

17. **Quelle est la variable environnement qui permet de définir le fuseau horaire du système ?**

18. **Quelle commande allez-vous utiliser pour afficher le contenu des variables de localisation LC_* ?**

19. **Quelle commande allez-vous utiliser pour changer l'encodage d'un fichier de codage UTF-8 au codage ISO-8859-1 ? Donner seulement la commande sans options et sans arguments.**

20. **Quelle est la commande utilisée pour tester le bon fonctionnement de DNS côté client ?**

21. **Donner le nom (chemin complet) du fichier de configuration de CUPS ?**

22. **Quelle commande est équivalente à route –n ?**

23. **Vous voulez régler l'horloge du système à partir de l'horloge matérielle. Quelle commande (sans options et sans arguments) allez-vous exécuter ?**

24. **Quelle commande allez-vous utiliser pour modifier votre table de cron personnelle ? Donnez la commande la plus simple.**

25. **Quelle est la commande la plus simple qui vous permet d'afficher le contenu de votre table de cron ?**

26. **L'utilisateur nicolas de votre serveur a oublié son mot de passe. Quelle commande allez-vous utiliser pour changer son mot de passe en supposant que vous avez ouvert une session en tant que root ?**

27. **L'utilisateur mejdi a été déplacé dans le département BECO. Vous voulez changer son groupe principal en beco. Quelle est la commande la plus simple pour réaliser cela ?**

28. **Quelle commande allez-vous utiliser pour verrouiller le compte de l'utilisateur nicolas ?**

29. **Quelle est la commande la plus simple pour supprimer le compte de l'utilisateur nicolas y compris son dossier personnel ?**

30. **Vous voulez ajouter l'utilisateur mejdi à votre système en créant son dossier personnel. Donnez la commande la plus simple avec ses options et arguments.**

31. Quel est le chemin d'accès complet du fichier de configuration du démon ntpd ?

32. Dans l'ordre SQL SELECT, quelle option faut-il utiliser afin de ne conserver que des lignes distinctes ?

33. En langage SQL, pour tester l'égalité de deux chaînes de caractères, quel caractère de remplacement faut-il utiliser afin de remplacer zéro à n caractères quelconques ?

34. La requête SQL suivante permet d'afficher les livres dont le prix est supérieur à 70, en ordre décroissant de prix :

- ❑ A `SELECT * FROM Livre WHERE 'Prix' >=70 ORDER BY 'Prix' DESC;`
- ❑ B `SELECT * FROM Livre WHERE Prix >=70 ORDER BY Prix DESC;`
- ❑ C `SELECT "*" FROM Livre WHERE Prix >=70 ORDER BY Prix DESC;`
- ❑ D `SELECT * FROM Livre WHERE Prix >=70 ORDER BY Prix ASC;`

35. Votre adresse IP est 170.35.13.28 et votre masque réseau est 255.255.255.192. Quelle adresse IP N'APPARTIENT PAS à votre réseau ?

- ❑ A 170.35.13.33
- ❑ B 170.35.13.88
- ❑ C 170.35.13.62
- ❑ D 170.35.13.55

36. Quelle commande vous permet de voir l'adresse MAC et la configuration IP de votre carte réseau ?

37. Laquelle des commandes suivantes permet de synchroniser l'horloge du système avec un serveur NTP ?

- ❑ A `date -q -g`
- ❑ B `hwclock -q -g`
- ❑ C `ntpdate -s ServeuNTP`
- ❑ D `ntpd Update`

38. Vous voulez que tous vos utilisateurs BASH puissent accéder aux programmes contenus dans « /opt/bin ». Vous allez ajouter PATH=$PATH:/opt/bin; export PATH dans quel fichier ?

39. La ligne correcte de la table cron qui permet d'exécuter le script /usr/local/sbin/chklog une fois par heure entre trois heures et cinq heures de l'après-midi chaque lundi et jeudi est :

- ❑ A 0 3,4,5 * * 2,5 usr/local/sbin/chklog
- ❑ B 0 3,4,5 * * 1,4 /usr/local/sbin/chklog
- ❑ C * 15,16,17 * * 1,4 /usr/local/sbin/chklog
- ❑ D 0 15,16,17 * * 1,4 /usr/local/sbin/chklog
- ❑ E 0 15,16,17 1,4 * * /usr/local/sbin/chklog

40. À quoi sert la commande export de bash ?

- ❑ A Permettre de monter les disques à distance
- ❑ B Lancer une commande dans un sous-shell
- ❑ C Mettre l'historique des commandes à la disposition des sous-shells
- ❑ D Permettre à une variable d'être accessible dans l'environnement des processus fils
- ❑ E Partager une partition NFS avec les autres ordinateurs sur le réseau

41. Quelles sont les permissions correctes pour le fichier /etc/shadow ?

- ❑ A –rw--w--w-
- ❑ B –rwxrw-rw-
- ❑ C –rw-r--r--
- ❑ D –rw-------

42. Quelle commande du bash vous empêche d'écraser un fichier avec « > » ou « >> » ?

- ❑ A set -o safe
- ❑ B set -o noglob
- ❑ C set -o noclobber

- ❑ D set -o append
- ❑ E set -o nooverwrite

43. Vous venez d'installer un système et vous voulez vous assurer que chaque utilisateur créé aura un sous-dossier bin/ dans son répertoire personnel. Dans quel répertoire allez-vous mettre le répertoire bin/ pour permettre sa création automatique au moment de l'ajout d'un nouvel utilisateur ?

44. Lesquels de ces deux fichiers dans le répertoire personnel de l'utilisateur sont utilisés pour configurer l'environnement bash ?

- ❑ A bash et .bashrc
- ❑ B bashrc et bash_conf
- ❑ C bashrc et bashprofile
- ❑ D .bashrc et .bash_profile
- ❑ E bash.conf et .bash_profile

45. Quel est le fichier dont le contenu est affiché pour des utilisateurs qui ouvrent une session localement sur la machine AVANT l'ouverture de la session ?

- ❑ A /etc/issue
- ❑ B /etc/issue.net
- ❑ C /etc/motd
- ❑ D /etc/local.banner

46. Dans la liste suivante, quels sont les « Window Manager » (plusieurs réponses) ?

- ❑ A WindowMaker.
- ❑ B KDM.
- ❑ C Xwindow.
- ❑ D twm.

47. Que va faire la commande suivante: cat hosts | lpr -#2

- ❑ A Imprimer le fichier hosts sur l'imprimante par défaut deux fois
- ❑ B Classer hosts et imprimer le classement comme tâche #2

- ❑ C Envoyer le fichier hosts à l'imprimante et le mettre dans la queue numéro 2.
- ❑ D Envoyer le fichier hosts sur la sortie standard puis envoyer la tâche en cours à l'imprimante 2.

48. Quelle ligne de la table du cron permettrait la mise à jour régulière de la date du système à partir d'un serveur de temps ?

- ❑ A 10 * * * date dt$24
- ❑ B 10 * * * settime dt$24
- ❑ C 10 * * * date<ntpl.digex.net
- ❑ D 10 * * * /usr/sbin/runcron date <ntpl.digex.net
- ❑ E 10 * * * /usr/sbin/ntpdate ntp1.digex.net \ >/dev/null 2>&1

49. Quelle commande utilisez-vous pour suspendre ou mettre en attente une queue d'impression ?

- ❑ A lpr
- ❑ B lpq
- ❑ C lpc
- ❑ D lpd
- ❑ E lprm

50. Que contient le fichier xorg.conf (plusieurs réponses) ?

- ❑ A La résolution de l'écran.
- ❑ B Le(s) chemin(s) pour trouver les polices de caractères.
- ❑ C La taille du moniteur.
- ❑ D Le nom du Window Manager à lancer

51. Vous avez décidé de basculer vos mots de passe de type standard vers des mots de passe de type MD5. Après avoir configuré les fichiers contenus dans /etc/pam.d vous devez également faire les opérations suivantes :

- ❑ A Rien, les mots de passe seront modifiés au moment de l'ouverture d'une nouvelle session par l'utilisateur
- ❑ B Rien, les utilisateurs seront avertis automatiquement de changer leur mot de passe au moment de l'ouverture d'une nouvelle session.

- ❑ C Vous devez ré-entrer un à un tous les mots de passe avec la commande passwd
- ❑ D Vous devez supprimer et recréer tous les utilisateurs
- ❑ E Vous devez revenir vers la configuration précédente de /etc/pam.d pour réinitialiser les mots de passe en MD5.

52. Vous êtes en train de vérifier la sécurité de votre système et vous vous apercevez que la plupart des enregistrements dans /etc/passwd ont des « x » dans le champ du mot de passe et que certains ont des longueurs de 13 caractères. Que faites-vous dans ce cas ?

- ❑ A Rien, les utilisateurs avec « x » comme mot de passe sont bloqués.
- ❑ B Vous utilisez la commande pwconv pour convertir les mots de passe unix standard vers des mots de passe shadow.
- ❑ C Vous utilisez la commande passwd pour créer des mots de passe shadow aux utilisateurs ayant des mots de passe unix standard.
- ❑ D Vous utilisez la commande passwd pour créer des mots de passe aux utilisateurs ayant des « x » comme mot de passe.

53. Dans quel fichier configurez vous les alias du shell pour tous les utilisateurs ?

- ❑ A /etc/bashrc
- ❑ B /etc/profile
- ❑ C ~/.bash_profile
- ❑ D /etc/skel/.bashrc
- ❑ E /etc/skel/.bash_profile

54. Avec le protocole SSH, pour réaliser une authentification par clé publique / clé privée, l'utilisateur doit enregistrer sa clé publique RSA dans le serveur SSH dans le fichier :

- ❑ A ~/.ssh/id_rsa
- ❑ B ~/.ssh/known_hosts
- ❑ C ~/.ssh/authorized_keys
- ❑ D /etc/ authorized_keys

55. Vous êtes en train de configurer un routeur mais les réseaux connectés à ce routeur ne parviennent pas à communiquer entre eux. Finalement vous en déduisez que le problème provient du fait que le « *forwarding* » n'est pas activé sur votre routeur. Pour vérifier cela vous utilisez la commande

#cat /proc/sys/net/ipv4/_____

56. La commande netstat -a stoppe un long moment sans rien afficher à l'écran. Vous soupçonnez le problème suivant :

- ❑ A problème avec NTP
- ❑ B problème avec le DNS
- ❑ C problème avec SMTP
- ❑ D problème de routage
- ❑ E le démon netstat ne fonctionne plus

57. Après avoir modifié le fichier /etc/aliases, quelle commande faut il exécuter pour mettre à jour le fichier binaire indexé /etc/aliases.db :

- ❑ A. aliases
- ❑ B. forward
- ❑ C. newaliases
- ❑ D mailq

58. Vous soupçonnez qu'une passerelle de votre réseau ne fonctionne plus mais vous ne savez pas laquelle. Quelle commande va vous aider à résoudre le problème ?

- ❑ A ps
- ❑ B netstat
- ❑ C nslookup
- ❑ D ifconfig
- ❑ E traceroute

59. Quelle commande va créer une route par défaut avec comme passerelle 192.168.1.1 ?

- ❑ A netstat-add default gw
- ❑ B route default 192.168.1.1

- ❑ C ip route default 192.168.1.1
- ❑ D route add default gw 192.168.1.1
- ❑ E ifconfig default gw 192.168.1.1 eth0

60. _____ est utilisé par la machine pour identifier quelle machine se trouve sur le même réseau et quelle machine est sur un autre réseau.

- ❑ A DNS
- ❑ B ARP
- ❑ C La passerelle
- ❑ D Le masque réseau
- ❑ E Le protocole de routage

Réponses

1. ?
2. UPDATE
3. \ls
4. /etc/motd
5. ntpd
6. xvidtune
7. .bash_profile ou .profile
8. /etc/profile
9. xhost +tunis
10. /etc/syslog.conf
11. service network restart
12. Files
13. XDMCP
14. /etc/resolv.conf
15. GOK
16. /etc/localtime
17. TZ

18. locale
19. iconv
20. dig ou host
21. /etc/cups/cupsd.conf
22. netstat -nr
23. uname –r (alternative: uname –a)
24. crontab -e
25. crontab -l
26. passwd nicolas
27. usermod -g beco mejdi
28. passwd -l nicolas (alternative: usermod -L nicolas)
29. userdel -r nicolas
30. useradd -m medji
31. /etc/ntp.conf
32. DISTINCT
33. %
34. B
35. B
36. ifconfig
37. C
38. /etc/profile
39. D
40. D
41. D
42. C
43. /etc/skel
44. D
45. A
46. A, D

47. A
48. E
49. C
50. A, B
51. C
52. B
53. A
54. C
55. ip_forward
56. B
57. C
58. E
59. D
60. D

Index des mots clés

$

/

~

A

B

C

D

E

F

G

H

I

J

K

L

M

N

O

P

Q

R

S

T

U

W

X

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 (ISET, Tunis), option réseaux informatiques. Il est ingénieur en informatique, diplomé 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.

Niry H. Andriambelo (Madagascar) : membre fondatrice de l'Association malagasy des utilisateurs de logiciels libres et formatrice de formateurs GNU/Linux, Niry Andriambelo est ingénieur informatique diplômée de l'Université d'Antananarivo et, depuis 2004, coordinatrice pour les systèmes et réseaux universitaires francophones dans l'océan Indien.

Andrei Boyanov (Bulgarie) : directeur d'Active Solutions et membre de la commission technique et développement de l'Institut professionnel Linux. Andrei Boyanov est ingénieur en informatique issu de l'Universtié technique de Sofia et formateur des personnels d'encadrement de l'enseignement supérieur dans le domaine des systèmes et des réseaux Linux.

Nicolas Larrousse (France) : concepteur de programmes Transfer dans les systèmes et réseaux sous Linux ainsi que de formations à la certification LPI. Nicolas Larrousse est ingénieur informatique. Il enseigne les systèmes d'information à l'Université de Versailles et exerce, depuis 1992, au Centre national de la recherche scientifique (CNRS), à Paris.

Véronique Pierre (France) : consultante indépendante en édition scientifique multimédia.