LINUX

Cours de Laurent Crosnier pour les SRL01

Table des matières

MANDRAKE 10

I) Historique et Philosophie du produit

1991 : Linus Torwald se base sur les codes sources d'UNIX.
Unix existe depuis 1971 (propriété de AT&T qui en donne les droits à l'université de Berkeley qui le fait évoluer.)
L'Université crée une société : Berkeley Software Distribution (BSD) AT&T continu le développement en parallèle

2 grandes familles :

- Communautaires : Debian, slackware, freeBSD
- Distribution (Société a but lucratif) : Redhat, Suse, Mandrake
  L'os est gratuit mais pas les services

Attention : logiciel libre, on peut le modifier mais pas le revendre (free=gratuit ou libre, dans ce cas c'est libre)

Philosophie de fonctionnement :
« TOUT EST FICHIER » (écran, disques durs, clavier, programme…)

Système d'exploitation :
Multi-tâches préemptif : l'os décide du temps processeur affecté a chaque application (contrairement au coopératif ou les applis se partagent elles même le temps cpu)
Multi-utilisateur : différenciation des utilisateurs connectés

I) La ligne de commande

Linux et Unix utilisent un interpréteur de commande (qui tourne en mémoire en permanence pour répondre aux commandes).
Emplacement : /bin/bash
BASH: Bourne Again Shell (jeu de mot BORN+BASH coup de poing qui déménage)
Sous Unix :

- Bourne shell : sh
- Korn shell : ksh
- C-shell : csh

Structure des lignes de commande:
#Nom_commande –options argument1 argument2

- L'espace est séparateur d'argument
- Sensibles a la casse : File1 différent de file1
- Les extensions ne sont pas significatives (utilisées pour nous ou par certaines application en mode graphique.
- Complétion des noms utilisant la touche TAB

Les Répertoires :
Ce sont des fichiers de type REP
. : rep courant
.. : rep parent
d : fichier de type Directory (répertoire)
- : Fichier ordinaire
l : Fichier de type lié
c : Fichier en mode caractère
b : Fichier en mode Bloc
p : pipe (tube nommé)
s : socket

LES COMMANDES

- **Structure des commandes:**
- L'interpréteur de commande l'appel le Bash (Bourn shell)
- Sensible a la casse.
- Les noms d'extension de fichiers ne sont pas significatifs.
- Nombre maximum de caractères = 255
- Caractères spéciaux a ne pas utiliser: * £ $ > < ... (espaces entre guillemet)
- complétions des noms avec la touche TAB
- \commande (lance la commande sans alias)

- **Descripteurs de la ligne de commande:**
- 0 : Entrée standard de la commande (clavier)
- 1 : Sortie standard de la commande (par défaut: écran de connexion)
  #tty indique la sortie courante. Ex: /dev/pts1
- 2 : Sortie d'erreur standard de la commande (par défaut: écran de connexion)
  #tty indique la sortie courante. Ex: /dev/pts1

Donc implicitement :
#cmd 1>écran 2>écran

Pour modifier ces éléments:
#ls -l 1>file1 => écrit dans le fichier file1
#ls -l 1>>file1 => Ajoute dans le fichier file1 le résultat de la commande
#ls 2> /dev/null => Redirige les erreurs vers le fichier /dev/null qui est un puits sans fond

- **Les caractères spéciaux**:

";" : exécute plusieurs commandes
Exemple:
cd /etc;pwd;ls -l

& :exécute une commande en arrière plan
Ex: #find /-name samba >fichier.samba &

"$" : affiche le contenu d'une variable
#echo $HOME

"\" : supprime la signification spéciale du caractère suivant le slash

"$" : joker remplace plusieurs caractères

"?" : Joker remplace un caractère.

` ` : Utilise le résultat de la commande comme argument.
#echo 'il est `date +%T` heure'
Il est 10h00 heures

- **Les pipes:**

#cmd1 | cmd2
cmd2 prend l'argument de la commande 1

Exemple:
who | grep -w poste1 => n'affiche que les données contenant le mot poste1 de la commande Who
who | wc -l => affiche le nombre de lignes dans le résultat de who

- **Les commandes**
- **su**

#su =>Permet de changer d'utilisateur par défaut il passe en root et demande le password
#su - => délogue l'utilisateur
#su – toto => délogue l'utilisateur et connecte toto en demandant le password toto

- **clear** (nettoyer l'écran)

#clear

- **head**

#head -5 file1 => affiche les 5 premières lignes du fichier nommé file1

- **echo**

#echo "texte" => affiche le texte (entre guillemet nécessaires si le texte contient des espaces)

- **uname**

#uname -a => Infos sur le système Linux
#uname -n => Donne le nom du poste
#uname -r => Affiche la version du kernel

- **whoami**

#whoami => donne le nom d'utilisateur courant.

- **who**

#who => donne le nom du compte courant et la connexion
Note: pts/7 (indique /dev/pts/7) = connexion de l'utilisateur sur la console pts/7

- **id**

#id nom_user => vérifie si le compte donné existe.

- **w**

#w => Infos sur les ressources /user

- **wc**

Donne le nombre de mots, lignes ou caractères contenu dans un fichier
#wc -l /etc/passwd => Donne le nombre de lignes dans le fichier passwd

- **cal**

#cal2004 => calendrier 2006
#cal2004 X Y => X 1 à 12 pour les mois Y Année

- **date**

#date +%T =>donne seulement l'heure (voir man date)

- **ps**

Indique les travaux en cours
#ps -ef => affiche les process avec nom users et pid

- **history**

Gère la liste des commandes tapées
#history !121 => affiche la commande 121
#history !date => lace la dernière commande commençant par 'date' tapée

- **tree**

Affiche l'arborescence d'un système de fichiers
#tree -d => affiche seulement les répertoires

- **pwd** (Print Working Directory)

Affiche le chemin absolu du répertoire de travail actuel
Exemple:
#pwd => /home/poste7

- **cd** (Change Directory)

change votre emplacement dans l'arborescence du système de fichier

Options :
#cd .. =>Retour au répertoire parent

Argument:
nom_du_répertoire_relatif =>entre dans le répertoire enfant du courant

Exemple: En étant dans /home
#cd poste7 => /home/poste7

Argument:
nom_du_répertoire_absolu => entre dans le répertoire indiqué

Exemple: En étant dans /home
#cd /var/poste7 => /var/poste7
sans_argument => retour au répertoire d'accueil

- **ls** (List Contents)

Affiche le contenu des répertoires et/ou des informations concernant la présence et les droits d'accès aux fichiers et répertoires.
Options :
-l : Liste longue
-a : Affichage des fichiers cachés (fichiers commençant par un.)
-d : Affiche les répertoires sans le contenu

Arguments:
nom_fichier ou nom_répertoire
sans arguments : affiche le répertoire courant

Exemple:
#ls -al

```
drwxr-xr-x  2 root   root     4096  jui  25 10:27  ./
drwxr-xr-x 20 thierry thierry 4096  jui  25 10:26  ../
-rw-r--r--  1 root   root        0  jui  25 10:27  file1
-rw-r—r--  1 root   root        0  jui  25 10:27  file2
-rw-r—r--  1 root   root        0  jui  25 10:27  .tit
12  3 4      5    6      7        8        9          10
```

1 : type de fichier
2 : permission utilisateur propriétaire
3 : permission groupe propriétaire
4 : permissions autres utilisateurs
5 : nombre de liens logiques
6 : utilisateur propriétaire
7 : groupe propriétaire
8 : taille du fichier en octets
9 : date de dernière modification du fichier
10 : nom du répertoire ou fichier

- **mkdir** (Make a Directory)

Permet de créer un répertoire. **Vous devez avoir le droit d'écriture sur les répertoires parent.**
#mkdir rep1 => Crée le répertoire rep1

Options: Voir les pages man
-p => crée le répertoire avec ses dossiers parents si nécessaires

Exemple:
#mkdir rep1/rep2 -p => il crée rep1 puis rep2 à l'intérieur

Arguments:
nom_répertoire
"nom du répertoire avec espace"

ATTENTION: #mkdir mes docs => crée deux répertoires mes et docs

- **touch**

Permet de créer un fichier vide ou de mettre à jour la date de dernière modification d'un fichier existant. Cette commande est utilisée pour créer des pointeurs (témoins) pendant l'accomplissement de certaines tâches.

Options: Voir les pages man

Arguments:
nom_fichier

exemple:
#touch file1 file2 file3 => crée les fichiers vides file1, file2, file3

- **cat** (Concatenate)
- **tac** (cat à l'envers:lit le fichier à l'envers)

Permet d'afficher le contenu d'un fichier. Utilisé avec un symbole de redirection, cat permet aussi de créer un fichier et d'y ajouter un contenu linéaire.

Options:
-n : numérotation des lignes

Arguments:
nom_de_fichier

exemple:
```
#cat > file1
ligne un du fichier file1 [entrée]
ligne deux du fichier file1 [entrée]
[CTRL][d]        => quitte le mode édition
#

#cat file1
ligne un du fichier file1
ligne deux du fichier file1
#
#cat >> file1     => ajoute les lignes suivantes au fichier file1
ligne trois du fichier file1
[CTRL][d]
#
#cat file1
ligne un du fichier file1
ligne deux du fichier file1
ligne trois du fichier file1
```

- **less**

#less => Permet une lecture séquentielle d'un fichier

- **more**

#more => Permet de gérer le défilement du résultat d'une commande

Options : Voir man

Arguments:
nom_du_fichier

Fonctions:

Q :	Quitte la lecture
/mot :	recherche la chaîne de caractère "mot"
b :	remonte d'une page
[espace]:	avance d'une page
[entrée]:	avance d'une ligne

exemple:
#ls -l | more

- **cp** (Copy)

#cp => Permet de copier des fichiers ou des répertoires avec leur contenu.

Options:

-i	demande confirmation avant d'écraser le fichier existant
-r	copie récursive d'un répertoire et son contenu
-f	copie forcée sans demande de confirmation

Arguments:
fichier_ou_rep_source fichier_ou_rep_destination

ATTENTION:
ls -al home/poste7
-dir1
-file1
-file2
-.fichier1
-dir2

Si dans dir1:
#cp -r dir1 dir2
création d'un répertoire enfant dir1 de dir2 et copie de tous les fichiers dans ce sous répertoire

Si copy dir1 vers dir3
#cp -r dir1 dir3
création d'un répertoire dir3 dans le répertoire courant et copie des fichiers et sous dossiers à l'intérieur
#cp -r dir1/* dir2 (* indique tous les caractères sauf le. (Donc ne copie pas les fichiers cachés)
copie les fichiers dans le répertoire dir2 directement mais pas les fichiers cachés

si le répertoire d'arrivée existe déjà
#cp -r dir1/. dir2 (ne copie pas le répertoire parent)

- **mv**

#mv => Permet de déplacer, ou renommer un fichier ou répertoire.

Options:
-i => demande de confirmation avant écrasement d'un fichier ou répertoire existant

Arguments:
nom_fichier_ou_rep nouveau_nom

Exemples:
mv file1 file2 (renomme le fichier file1 en file2)
mv file1 /home/perso (déplace le fichier file1 dans le dossier perso contenu dans /home)

- **rm** (remove)

#rm => Permet de supprimer des fichiers ou des répertoires.

Options:
-i demande confirmation de suppression
-r Suppression récursive d'un répertoire avec tout son contenu
-f Force la suppression (sans demande de confirmation)

Arguments:
nom_fichier_ou_rep

ATTENTION: Pour la suppression d'un répertoire vide taper : #rmdir

- **ln** (link)

#ln => Permet de créer des liens physiques et symboliques de fichiers

- ***Lien physique :***
  2 liens pointant sur un seul fichier (inode identique)
  ATTENTION: Les liens physiques ne concernent que les fichiers d'une même partition et ne fonctionnent pas avec les répertoires.
  Note:
  (ls -li pour afficher l'inode des fichiers)
  Faire un find sur l'inode du fichier pour retrouver les différents liens physiques d'un fichier

- ***Lien symbolique (soft) :***
  Pointeur pour les liens entre partitions ou répertoires
  Les deux fichiers ont un inode différent.
  Le fichier lien ne possède que le nom du fichier réel

Options:

-p	Lien physique
-s	Lien symbolique

Arguments:
#ln -s fichier_départ fichier_lien_symbolique

- **grep** (Grant expression)

#grep => Permet de rechercher une chaîne de caractères dans un fichier.

Arguments:
#grep -options chaîne nom_fichier

Options:

-i	ignore la casse
-w	recherche le mot exact
-r	recherche récursive à partir du répertoire courant.
-v	recherche inversée (qui ne contient pas)

Caractères spéciaux:
^chaîne => recherche la chaîne commençant par
chaîne$ => recherche la chaîne finissant par
Exemple:
#ps -ef | grep poste1 | grep -v grep

liste les travaux en cours | garde ceux concernant le poste1 | enlève du résultat le process grep

- **find**

#find => Permet de rechercher des fichiers dans une arborescence selon divers critères

Options:

-name	par nom de fichier (respecte la casse)
-iname	par nom de fichier (ne respecte pas la casse)
-type	par type de fichier (d=directory, f= fichier...)
-user	par nom de l'utilisateur propriétaire
-size	par taille (+ ou – Xy ou X=taille et y=o pour octet, k pour Ko) si pas + ou alors strictement de cette taille.
-ls	Affiche le résultat sous forme de fichiers détaillés
-not (ou !)	négation de la recherche
-path `/mnt` -prune	ne pas rechercher dans le répertoire /mnt

ATTENTION:
Par défaut, la commande est en **ET logique**. La requête retourne la réponse correspondant à toutes les options.
-o => Or remplace AND

Exemple:
#find / \(-user poste7 -o -user poste8 \) -type d

Le \sont nécessaire pour que ce qui est entre ( ) soit interprété comme caractère hors commande
Attention aux espaces après les \(et avant \)

*	joker remplace un ou plusieurs caractères
^	commence par.
$	fini par
^$	prendre les lignes vides

Exercice1 :
Trouver le nombre d'utilisateurs appelés postex (x=n'importe quoi) dans le fichier /etc/passwd
Réponse:
#grep -i poste /etc/passwd | wc -l

Exercice 2 :
Compter le nombre de répertoires dans votre répertoire d'accueil.
Réponse:
#tree -d | grep directories
(Affiche la ligne de la commande tree -d qui indique le nombre de répertoires total (X directories))

- **head**

#head -n => Affiche les n premières lignes d'un fichier (10 premières par défaut)

- **tail**

#tail => Affiche les dernières lignes désignées (les dix dernières par défaut)

Options:
-n => affiche les n dernières lignes
n => affiche les n premières lignes
-f => (follow) Permet d'afficher les rafraîchissements d'un fichier toutes les quelques secondes

- **cut**

#cut -d"x" -fn nomfichier => x est le séparateur de champs
=> n est le champ à sélectionner
Exemple:
#cut -d":" -f1-3 /etc/passwd

Autre syntaxe:
#cut -cy nom_fichier => y est le numéro de la colonne

Exemple:
#who | cut -c10- => récupère le who sans les 10 premiers caractères
#ps -e | cut -c-5 | grep -v PID => récupère les PID seulement

Syntaxe des champs:
2 => champ 2
2, 3,5 => champs 2, 3 et 5
1-3 => champs de 1 à 3
-10 => jusqu'à la colonne 10
10- => Depuis la colonne 10

- **sort**

#sort => trie les lignes d'un fichier par ordre alphanumérique

- **uniq**

uniq => supprime les lignes identiques d'un fichier

L'ÉDITEUR DE TEXTE VI

VI est un éditeur de texte qui fonctionne en mode texte, c'est à dire sans système de menus. On ne dispose donc que de raccourcis claviers pour accéder aux fonctionnalités de l'éditeur. Le grand nombre de raccourcis à gérer obligeait:

- Soit à les rendre complexes et difficiles à retenir (combinaison de plusieurs touches).
- Soit à dissocier la saisie du texte des commandes à utiliser sur ce texte, ce qui permettait d'associer une touche à une fonctionnalité. C'est cette solution qui a été retenu par les créateurs de VI

- **Le mode commande (pour agir sur le texte) :**

=> Sous les anciennes versions, il n'y avait pas de flèche alors:

h	=> Déplacement vers la gauche
l	=> Déplacement vers la droite
j	=> Déplacement vers le bas
k	=> Déplacement vers le haut
a (append)	=> Commence l'insertion de texte après le curseur
i (insert)	=> Commence l'insertion de texte avant le curseur
A	=> Commence l'insertion de texte à la fin de la ligne
	=> Commence l'insertion de texte au début de la ligne
o	=> Ajoute une ligne au dessous de la ligne courante
O	=> Ajoute une ligne au dessus de la ligne courante
r	=> Remplace le caractère courant
R	=> Remplace le texte courant (mode remplacement)
x	=> Efface le caractère courant
d	=> Efface les caractères précisés
d^	=> Supprime depuis le début de ligne
d$	=> Supprime jusqu'en fin de ligne
dG	=>Supprime jusqu'en fin de fichier
dGG	=> Supprime le début du fichier jusqu'au curseur
dd	=> Effacer la ligne courante (dans un buffer) comme COUPER
yy	=> Copier la ligne courante (dans un buffer) comme COPIER
p (paste)	=> Copie le contenu du buffer après le curseur comme COLLER
P (paste)	=> Copie le contenu du buffer avant le curseur comme COLLER
u	=> Undo efface la dernière manipulation
CTRL+r	=> Redo refaire ce qui est défait
gg	=> Reviens au début du fichier
G	=> Fin de fichier
v	=> Sélection: puis SHIFT+> ou SHIFT+< pour indenter

:/split nomfichier => Permet de splitter l'écran avec les deux fichiers

CTRL+W	=>passe dans le fichier inférieur
:q	=> Ferme les fichiers du dernier ouvert vers le 1er

- **Le mode saisie (pour taper du texte) :**

indiqué par le mot INSERTION en bas de page.

- **Le mode exe (pour effectuer des commandes complexes) :**

On accède à ce mode en tapant [ESC]:

:w	=> Enregistre les modifications
:q	=> Quitter vi
:wq ou :x	=> Enregistrer et quitter
:q!	=> Forcer la sortie de vi
:r/nom_de_fichier	=> Inclure le fichier indiqué
:r!commande_bash	=> Inclure le résultat de la commande.
:n	=> Aller au fichier suivant dans la liste ouverte
:N	=> Aller au fichier précédent dans la liste ouverte
:s/chaine1/chaine2	=> Remplace dans la ligne lchaine1 par chaine2
:s/chaine1/chaine2/g	=> Remplace dans la ligne toutes les occurrences de chaîne1 par chaîne2
:%s/chaine1/chaine2/g	=> Remplace dans le fichier toutes les occurrences de chaîne1 par chaîne2
:%s/chaine1/chaine2/g	=> Remplace dans le fichier toutes les occurrences de chaîne1 par chaîne2 avec demande de confirmation: y=yes n=no a=all q=no+exit l=oui+exit
:n°	=> Déplacer le curseur à la ligne n°x
:set_number	=> (ou se_nu) affiche la numérotation des lignes
:set_nonumber	=> Dé numérote
:syntax on/off	=> Active / désactive la coloration syntaxique
/chaîne	=> Recherche la prochaine occurrence de *chaîne*
?chaîne	=> Recherche la précédente occurrence de *chaîne*
ncommande	=> lance n fois la commande tapée

L'INSTALLATION DU SYSTÈME (MANDRAKE 10)

CTRL+ALT+F3+F4 Affiche les infos+erreurs
CTRL+ALT+F7+F1 retourne à l'install

hda : Master IDE 1 (hda1 à hda16: de 1 a 4 partitions primaires et de 5 à 16 lecteurs logiques)
hdb: Slave IDE 1
hdc: master IDE 2
hdd : Slave IDE 2
sda: SCSI ou USB

Partitionnement:
Racine
/ = système de fichiers "racine" (root): Obligatoire pour installer un os linux
Au niveau du root : 250 a 300Mo sont nécessaire

Détail du root
/ le répertoire racine

- /bin les fichiers exécutables (en binaire) (initialisation du système + commandes "essentielles")
- /boot le noyau **vmlinuz** et les fichiers de démarrage
- /dev répertoire de fichiers spéciaux, qui servent de canaux de communication avec les périphériques (disques, adaptateur réseau, cartes son etc...)
- /etc les fichiers de configuration du système et les principaux scripts de paramétrage
  - /etc/rc.d *scripts de démarrage du système*
  - /etc/X11 *scripts de configuration du serveur X*
  - /etc/sysconfig *configuration des périphériques*
  - /etc/cron *description des tâches périodiques à effectuer*
  - /etc/skel *fichiers recopiés dans le rép. personnel d'un nouvel utilisateur* (squelette du profile)
- /home la racine des répertoires personnels des utilisateurs
- /lib les bibliothèques et les modules du noyau
- /mnt la racine des points de montage des systèmes de fichiers périphériques ou extérieurs (cd, disquette, nfs).
- /opt lieu d'installation d'applications supplémentaires (comme starOffice, java ..)
- /root répertoire personnel du super-utilisateur root
- /sbin les fichiers exécutables pour l'administration du système
- /tmp stockage des fichiers temporaires
- /usr programmes accessibles à tout utilisateur; sa structure reproduit celle de la racine /
- /var données variables liées à la machine (fichiers d'impression, traces connexions http, smb dans **/var/log**)
- /proc ce pseudo-répertoire contient une "image" du système (/proc/kcore est l'image de la RAM).

SWAP:

- Partition de swap: gère la mémoire disque (échange entre la mémoire disque et la mémoire RAM)
- Taille: Entre une et deux fois la RAM
- Obligatoire sous Mandrake
- Pas de point de montage: c'est le noyau qui va gérer cette partition

/USR:

- Taille:2,5 Go

/var:

- Taille: 2Go

/home:

- Taille: 2Go

Choix des paquetages à installer:
Passer en mode console: CTRL+ALT+F1 ou F2 ou F3 …F6 (il existe 6 consoles)
Passer en mode graphique: CTRL+ALT+F7 ou F8 (pour la 2ème session)

Configuration POST-INSTALL:

Vérifier les pilotes hardware.

Démarrer l'interface graphique ou non.
#startx pour lancer le mode graphique

CTRL+ALT+ (BACKSPACE) : tuer l'interface graphique si elle a planté

Par défaut la commande #startx lance #startx –DISPLAY=IP:0.0 (le premier chiffre détermine l'écran et le second détermine la session)

Arrêt et démarrage du système:

Démarrage:

1. Bios (POST)
2. Lecture du MBR (master boot record) du disque d'amorçage
3. Lecture du premier secteur de la partition active
4. Vérification (matériel) du système de fichier racine
5. Lecture du fichier /etc/inittab (il saura quoi faire au démarrage)

Lecture du fichier INITTAB
Remarque: le # est une remarque et donc non interprété

Initdefault
0 =>halt (arrêt du système)
1 =>single user mode. Mode maintenance parfois appelé niveau S (single). Si on perd le mot de passe root, il suffit de passer en mode S et de trouver ainsi le mot de passe root perdu
2 => Multi-utilisateur sans réseau
3 => Multi-utilisateur avec réseau (le plus courant)
4 => inutilisé (utilisation future)
5 => serveur graphique (X11)
6 => Reboot de la machine

#init X => Commande pour changer le mode (X est le numéro du mode)

Démarrage des Démons:

Pour chaque niveau, un répertoire rc est crée dans /etc
/etc/rc0.d/ |
ex: K09smb |
/etc/rc1.d/ | Contiennent les liens symboliques vers init.d avec options
/etc/rc2.d/ | S en majuscule pour Start
/etc/rc3.d/ | K en majuscule pour Kill
ex: S91smb | un chiffre pour indiquer la priorité
/etc/rc4.d/ |
/etc/rc5.d/ |
/etc/rc6.d/ |
/etc/init.d/ => Contient les fichiers de gestion des services
ex: smb
/etc/rc.d/rc.3 => Indique les démons à lancer dans les différents modes /rc.0

ATTENTION: derrière le chiffre le nom doit être identique au nom du fichier dans init.d

La gestion automatique du lancement des démons (création des liens symboliques) se fait par l'utilisation du logiciel **chkconfig:**

Dans le programme /etc/init.d/APPLI, il faut 2 lignes de commentaire
#chkconfig X Y Z => Créer les liens symboliques nécessaires
X correspond aux niveaux sur lesquels on lance le service.
Y correspond au niveau de priorité de démarrage
Z correspond au niveau de priorité d'arrêt

Exemple:
#chkconfig 345 20 80 base
#Description : description quelconque (obligatoire)
ensuite la commande à taper est:
#chkconfig –level X *nomservice* reset|on|off prioritéstart prioritéstop
(Attention deux tirets level)

Exercice de simulation de lancement d'un service base de donnée (base):

ATTENTION: ICI on créer des fichiers et on n'utilise pas de liens symboliques

- Pour le démarrage de la base:

Dans /etc/rc3 (ou 5), créer le fichier S20base
S20base

#!/bin/bash
echo "Démarrage de la base" >/dev/console =>affiche démarrage de la base et redirige vers la console
sleep 3 =>temporisation de 3 secondes
touch /var/lock/subsys/base => Création d'un fichier témoin prouvant le démarrage de base. DOIT s'appeler comme le service (ici base).

Le rendre exécutable par root (chmod 700)

- Pour l'arrêt de la base :

Dans /etc/rc0 et rc6 on crée un fichier K80base
K80base

#!/bin/bash
echo "Arrêt de la base" >/dev/console =>affiche arrêt de la base et redirige vers la console
sleep 3 => Temporisation de 3 secondes
rm -f /var/lock/subsys/base => Supprime le fichier témoin prouvant le démarrage de base
---
Le rendre exécutable par root (chmod 700)

Arrêt du système:
#init n => "n" est le niveau de fonctionnement souhaité (0 arrêt ; 6 reboot) **doit être root.**
#halt
#shutdown –option paramètre_temps "message"

Les options sont:
-h => Halt (power off)
-c => Cancel (annulation du shutdown en cours)
-r => reboot au niveau par défaut
-t x => Signale d'arrêt après x secondes pour les processus (pas les utilisateurs)
-h -t n => envoi un signal de shutdown à tous les processus puis attente de n secondes avant de les arrêter.
-r -t => parfois (multi-processing) il faut mettre un –t pour éteindre le pc proprement car le –r ne regarde pas si les process ont le temps de s'arrêter.
-a =>lecture du fichier /etc/shutdown.allow et si l'utilisateur qui lance la commande est dans ce fichier et est connecté, la commande s'exécute sinon elle ne se lance pas (pratique pour donner le droit à un opérateur de sauvegarde pour stopper le poste après sauvegarde).

Paramètre de temps:
Now : maintenant
+m : en minute
HH:MM : coupure a l'heure indiquée
ATTENTION : -t est différent des paramètres de temps

Exemples:
#shutdown –r +5 "coupure dans 5 minutes"
Lancer un reboot normal différé de 5 minutes avec un message

#shutdown –c "fausse alerte"
Annuler le shutdown précédant avec message

GESTION DE L'ACTIVITÉ DU SYSTÈME (ADMINISTRATION CHAPITRE 17)

- Espace Disque:

#df : Disque free
#df –h : human, permet une lecture plus facile avec les valeurs tels que Ko, Mo, Go…

du indique la taille du répertoire.
#du -option nomrep
-h => Human readable
-s => Total occupé sans les détails

ATTENTION:
#df => compte aussi 4ko par répertoire vide.
#du =>compte uniquement la taille des fichiers et dossiers sur lesquels il a accès en lecture (R)

- *Espace mémoire*

#free : infos sur la mémoire (Ch17 page 4)

- *Contrôle des processus*

#ps –options
Options:
Pas d'options: affiche les travaux en cours de l'utilisateur
-e : affiche les travaux de tous les utilisateurs
-f : identique au e mais avec plus d'infos
-a : (all) travaux issus de la ligne de commande
-u : travaux en cours avec des infos relatives à l'occupation CPU
-x : travaux en cours avec leur état.
Souvent on utilise –ef ou –aux

La colonne status possède 4 options:
N => Priorité différente
S => Sleeping
W => Réside en RAM
R => Running

Tous les processus ont un processus père. Quand on tue le père, on tue tous ses fils
En fin d'exécution, tous les processus envoient un signale a leur père
Les commandes sont toujours lancées à un processus père.
Pour effectuer une recherche:
#ps –ef | grep startx (pour connaître tous les processus dont la commande startx est incluse)

PID : Processus d'identification
TTY: numéro de la console
PPID: processus père qui initialise le processus.

#kill –signal_interruption PID
Signale d'interruption:
-l : Liste des signaux (C'est un L minuscule)
-1 : Arrêt + redémarrage
-9 : Arrêt forcé
-15 : Arrêt "soft"(option par défaut)
-17 : Suspension (pour les processus sous terminal de connexion)
-18 : Reprise après suspension
-19 : Suspension pour les processus rattachés à un terminal
#killall -signal nomcommande
ATTENTION: Si utilisé sur BASH il arrête tous les bash y compris celle qui le lance.

Comment trouver le père du père d'un processus:
#ps –ef | grep process = 10345
Puis
#ps –ef | grep 10345

Il existe une solution plus intéressante:
#pstree –p permet de voir les PID
#pstree –u permet de voir les UID
#pstree –pu permet de voir les deux

- *Zombi: processus fils dont le père a été suspendu*

#killall : envoi un signale à des processus indiqués par leur nom

#ps –ef | grep telnet
Puis la commande #killall -9 telnet

#top est un programme (q pour sortir) qui permet d'identifier les zombies

- Le répertoire /PROC

Faire un #cd/proc puis #more cpuinfo ou #more meminfo
Il n'y a que des fichiers virtuels

- Le contrôle des logs : /VAR/LOG

Dans les fichiers
auth.log => Contient les infos sur les connections et déconnections utilisateurs
Cat auth.log|grep failure|grep root|more
Cette commande fait apparaître les utilisateurs root qui ont essayés de se logger

boot.log => contient tous les messages qui sont générés pendant le boot de la machine
messages =>fichier qui contient tous les événements dignes d'intérêt.
syslog => regroupe auth.log et messages

COMMENT CONFIGURER LES LOGS (CHAPITRE 7 PAGE 38):

Aide : man syslog.conf
man logrotate
./etc/logrotate.conf =>défini le fonctionnement de rotation des logs.
./etc/syslog.conf =>niveau des messages a enregistrer dans les logs par service (pas tous)

la structure du fichier est la suivante:
nomservice.niveaualerte nomfichierstockage
niveaux: debug
info
notice
warning
erreur
critic
alert
emergency

Exemple:
mail.notice /var/log/mail/erreurs => logue tout depuis le niveau Notice jusqu'à alert
mail.=notice => Uniquement les messages de niveau notice

On peut envoyer les infos ou l'on veut vers un fichier, une console, ou un serveur de centralisation.

Exemple:
Envoi de logs sur une machine distante:
coté client => #vi /etc/syslog.conf
ajouter la ligne : service.niveau @nom_ou_ip_srvlog
coté serveur => #service syslog stop
vi /etc/init.d/syslog (ajouter -r comme remote à syslog)

ATTENTION: les modifications ne sont prises en compte que après redémarrage
#service syslog restart

GESTION DES SYSTÈMES DE FICHIERS

Les formats supportés:
EXT2, EXT3, NTFS (en lecture), FAT, V-FAT, ISO 9660(cd-rom), smbfs (smb linux), nfs (réseau linux) autres (voir la commande "mount")

CRÉATION D'UN NOUVEAU SYSTÈME DE FICHIERS:

Sur disquette:
La création se fait en s'adressant au périphérique matériel
Pour la disquette: /dev/fd0
La commande \ls /dev/floppy permet d'afficher la liste des lecteurs de disquette reconnue par le système

La commande pour créer un nouveau système de fichiers:
#mkfs –t type_système_fichier /dev/fd0 (184inodes=184 fichiers maximum)

Sur disque dur:
#mkdfs -t type_sf /dev/hda10 ext2 -j
/de/hda10 => indique la partition
ext2 -j => indique ext2 journalisé = ext3

supermount (commande Mandrake) permet le montage automatique des périphériques de stockage.

ATTENTION: même si il n'y a pas de disquettes, il écrit tout de même dans le répertoire mnt et donc rempli le disque et ce disque ne sera, par défaut, accessible que par /mnt/floppy

Les commandes pour désactiver supermount:
#supermount disable
#umount –a
Sans supermount:
Pour Accéder a la disquette:
Créer un point de montage (répertoire) ex: /floppy
Monter le lecteur: avec la commande
#mount /dev/fd0 /floppy (floppy est le point de montage)

Pour démonter le lecteur:
- Le point de montage ne doit pas être occupé
- Un fichier accessible par le point de montage ne doit pas être ouvert

#umount nom_dossier_monté
#umount /dev/fda

Partition sur un disque IDE

1. **Crée la partition avec #cfdisk**
   cfdisk /dev/hda
- espace libre
- nouveau
- logique
- taille en Mo
- début d'espace libre
- Ecrire
- oui (en entier)
- quitter
- reboot

ATTENTION: Création de la partition mais pas formatage

2. **Formater la partition:**

En EXT2 :
#mkfs -t ext2 /dev/hdan => n est le numéro de partition
En EXT3:
#mkfs -t ext2 -j /dev/hdan => n est le numéro de partition le -j est pour journalisation

3. **Le montage manuel de partition**

#mkdir /disk
#mount *chemin_partition chemin_répertoire_de_montage*
exemple:
#mount /dev/hdan /disk

Monter une partition avec des permissions différentes
#mount -o remount, rw /fs

Forcer le démontage
#amount *point_de_montage* -l
#umount -a

voir la liste des utilisateurs et processus qui dépendent du point de montage:
#fuser -vm *point_de_montage*

Sauvegarde des partitions lorsqu'elles sont dans /etc/fstab:
#dump *niveau_sauvegarde* uf *nom_fichier_archive nom_système_fichier*

ATTENTION:
- Si plusieurs points de montage sont sur le même répertoire, c'est le dernier point monté qui gagne.
Si on démonte le dernier montage sur ce répertoire c'est le montage précédant qui gagne.
- après démontage, le répertoire reste donc les données copies dans ce répertoire sont dans une autre partition!!!

Montage automatique au démarrage:
Dans le fichier /etc/fstab

Champs1	*Champs2*	*Champs3*	*Champs4*	*Champs5*	*Champs6*
Partition à monter ex: /dev/hda10	Point de montage ex: /disk	Type de système de fichier ex: EXT2	Option de montage default	Sauvegarde par dump 0=non 1=oui (défaut)	Type de vérification 0=pas verif 1=verif obligatoire + succès obligatoire 2=verif obligatoire si 1=ok

Maintenance (réparation) d'un système de fichiers: Chapitre 5 page 37

ATTENTION: A utiliser hors montage

Simulation de crash avec récupération des données
1. Vérification du montage du système de fichiers à crasher
#mount /dev/hda10 /newdisk
2. copier des données
#cp -r /etc /newdisk
3. sauvegarder la liste des inodes du système de fichier dans un fichier.
#ls -Ri /newdisk > /var/backup.newdisk.inodes
4. Planter le système de fichier en écrasant les premiers blocs avec dd
#dd if=/dev/zero of=/dev/hda10 bs=1k count=5
5. démonter le newdisk
#amount /newdisk -l
6. tenter une réparation avec e2fsck
#e2fsck /dev/hda10
7. remonter le système de fichiers
#mount /dev/hda10 /newdisk
8. refaire le 3.
#ls -Ri /newdisk > /var/backup.newdisk.inodes2
9. comparer les deux fichiers
#dif /var/backup.newdisk.inodes /var/backup.newdisk.inodes2

La gestion de packages:

3 Grandes familles:
- Les "rpm" => Redhat Package Manager (RH, Mandriva, Sus...)
- Les "deb" => Debian, Knoppix,...
- Les "tar" => Tout le monde

1. Les paquetages RPM:
#rpm *-options nom_paquetage.rpm*

-i => Install
-e => Supprimer
-U => Update (mise à jour)
-v => Verbose
-h => Hash = Barre de progression
-qpi => Query Previous Information
-qpR => Query Previous Requierements (dépendances)
-qpl => Query Previous List (liste les fichiers qui vont être installés)
-qpd => Query Previous Documentation (liste des fichiers d'aide)

Options avancées:
--force => Ne pas tenir compte des conflits
-- nodeps => Ne pas tenir compte des dépendances (ex : deux pack sont mal faits et dépendent tous les deux l'un de l'autre)

options sur les paquetages installés
-e => Erase (supprimer)
-q => Dit si le pack est installé ou pas mais demande le nom exacte du pack
-qa => Query All = Liste tous les paquetages installés (utilise grep pour retrouver la cible)
-ql => Query List
-qd => Query documentation
-qc => Query Configuration: Liste des fichiers de configuration pour le paquetage.

#rpm -qf nom_fichier => Indique le paquetage d'origine du fichier.
ATTENTION: certains fichiers ne font partie d'aucun paquetage.
(ex: /etc/shadow)

ATTENTION: Sous MANDRAKE uniquement:
#urpmi nom_paquetage
- Consulte la liste des sources disponibles (/etc/urpmi/urpmi.cfg)
- Recherche le paquetage
- L'installer

#urpmi.update => mise à jour d'une source
#urpmi.addmedia => Création d'une source

les fichiers de configuration:
/etc/urpmi/urpmi.cfg
On peut indiquer ou modifier l'emplacement des paquetages rpm
NOM SOURCE file://répertoire contenant les paquetages
hdlist => Fichier contenant la liste du nom de tous les paquetages connus par urpmi
list => fichier crée à l'ajout de la source

Ajout d'une source pour l'utilitaire URPMI:
1. Choix de la source réseau
2. Montage de cette source réseau sur un point local
#mount -t nfs 192.168.9.200:/SOURCES /rpm
3. Configuration de cette source locale pour urpmi
#urpmi.addmedia *nom_donné_a_la_ressource* file://*chemin_de_la_source*
4. Contrôle de la source dans /etc/urpmi/urpmi.cfg
5. Installation d'un paquetage à partir de la nouvelle source
#urpmi *nom_donné_a_la_ressource*
où
#urpmi –media RPM *nom_donné_a_la_ressource*
6. Mise à jour d'une base de données urpmi
#urpmi -update *fichier_a_jour*

2. Les programmes au format TAR:
Précompilés mais pas fini.
- Extraire le fichier archive (avec ou sans décompression suivant l'extension du fichier)
- Lire les infos d'installation (INTALL, README....)
- Le plus souvent on crée un fichier de configuration avant compilation: . Configure
- Puis on compile les sources: make (il faut un compilateur C++ installé)
- puis on installe les sources compilées (make install)

Les programmes sous DEBIAN:
la commande apt:
apt-get update => met à jour par Internet une liste des paquetages à installer
apt-get upgrade => met à jour les paquets listés dans le fichier liste /etc/apt/sourceslist
apt-cache search => permet de rechercher les paquetages disponibles dans la liste par mot clé
apt-get install nom_paquetage => installe le paquetage et demande l'installation des dépendances
apt-get remove nom_paquetage => supprime le paquetage (si --purge => désinstalle tout)

Le fichier /etc/apt/sourceslist indique le type de paquetages disponibles: stable, unstable...

UTILISATEURS ET GROUPES LINUX

1. Gestion des utilisateurs et groupe

1.1 Les utilisateurs

Unix est un système multi-utilisateurs. Plusieurs personnes peuvent l'utiliser de façon simultanée (dans le cas de configurations en réseau).

Pour le système, un utilisateur n'est pas obligatoirement une personne physique. Un utilisateur peut détenir des fichiers, exécuter des programmes ou encore déclencher automatiquement des fonctions systèmes.
Par exemple, un utilisateur peut être créé dans le seul but de détenir des fichiers publics. On parle alors de pseudo utilisateur.

Un utilisateur possède un nom d'utilisateur appelé aussi login lui permettant d'établir une connexion. Ce login est associé à un mot de passe personnel. Pour accéder aux ressources du système, l'utilisateur doit entrer la bonne paire login/mot de passe : c'est l'authentification (le login).

Les utilisateurs sont identifiés par le système grâce à un UID (identifiant d'utilisateur) unique. Cet identifiant est une valeur numérique.

1.1.1 Le fichier /etc/passwd

Le fichier /etc/passwd contient les informations relatives à tous les utilisateurs du système. On y trouve leur :
login
mot de passe (chiffré)
UID
GID principal
nom complet et autres informations
répertoire principal
leur shell

La syntaxe de "/etc/passwd" est très simple, chaque ligne concerne un utilisateur. les différents champs sont séparés par des ":" :
login:mot-de-passe:UID:GID:info-utilisateur:répertoire-principal:shell

Détaillons les champs :
Login : c'est l'identifiant que doit entrer l'utilisateur pour s'authentifier. La convention veut qu'un utilisateur John Smith possède jsmith ou smith_j comme login.

Mot de passe : il est évident que le mot de passe n'apparaît pas en clair dans le fichier, il est chiffré en md5. C'est la commande passwd qui s'occupe de chiffrer le mot de passe. Ce champ peut prendre plusieurs significations :
"*" : il est impossible de s'authentifier sur le système avec ce compte
"!!" : Le compte est désactivé
"x" ou "!" : le mot de passe est dans un fichier shadow (voir ci-après)
champ vide : Il n'y a pas de mot de passe pour ce compte.

UID : il s'agit de l'identifiant unique de l'utilisateur.
L'utilisateur root possède l'UID 0
Par convention, les UID inférieurs à 100 sont réservés aux comptes système.

GID : l'identifiant du groupe principal de l'utilisateur

Info utilisateur : des informations sur l'utilisateur. chaque information est séparée par une virgule (le nom complet, numéro de poste ...).

répertoire personnel : Il s'agit du répertoire dans lequel sont stockés les fichiers appartenant à l'utilisateur. En général de la forme /home/*login*

shell : c'est l'interpréteur de commandes qui sera lancé après l'authentification.

1.1.2 Le fichier /etc/shadow

Le fichier /etc/passwd est accessible à tout le monde. en effet, certaines commandes ont besoin de connaître la liste des utilisateurs ou la correspondance login/UID.
La présence du mot de passe dans /etc/passwd, même crypté, pose un problème de sécurité. La solution a ce problème est de stocker les mots de passe dans un fichier différent : /etc/shadow.
Pour garantir la sécurité, seul l'administrateur peut le lire.

Note:
sous Linux, si le fichier /etc/shadow n'est pas utilisé, l'utilitaire **pwconv** permet, à partir d'un fichier /etc/passwd unique, de créer le fichier /etc/shadow qui lui correspond.

Stratégies de mot de passe:
Ce fichier /etc/shadow permet d'imposer des règles de gestion de mots de passe
ch1 ch2 ch3 ch4 ch5 ch6 ch7 ch8 ch9
ch1 => Login
Ch2 => Mot de passe crypté
- * : compte système
- !! : compte non autorisé à se connecter, sans mot de passe ou désactivé

ch3 => Nombre de jours écoulés entre le 1er jan 70 et la dernière modification du mot de passe
ch4 => Nombre minimum de jours entre deux modifications de mot de passe
ch5 => Nombre de jours de validité de mot de passe
ch6 => Nombre de jours avant expiration du mot de passe à prévenir l'utilisateur
ch7 => Nombre de jours avant que le compte soit verrouillé, quand le mot de passe est expiré
ch8 => Date d'expiration de mot de passe (en nombre de jours depuis 1 jan 1970)
ch9 => Champs réservé (depuis longtemps)

Changement des infos d'expiration du mot de passe d'un utilisateur:
#chage -l nom_user => infos sur le password

[thierry@localhost thierry]$ chage -l thierry
Minimum : 0 => en jours
Maximum : 99999 => Validité en jours
Avertissement : 7
Désactivé : -1
Dernier changement : sep 09, 2005
Expiration du mot de passe : Jamais
Mot de passe désactivé: Jamais
Expiration du mot de passe: Jamais

Lors du login, contrôle de passwd (compte, appartenance au groupe.....), shadow (droits ok) et pam.d (authentification)

1.2 Les Groupes
Le fichier /etc/group contient les informations relatives aux groupes présents sur le système.
Voici sa syntaxe :
groupe:*:GID:utilisateurs

groupe : le nom du groupe

- : la présence de ce champ est lié aux anciennes versions d'Unix et n'est plus utilisé. Il peut rester vide ou contenir le caractère "*" ou "x".

GID : c'est l'identifiant unique du groupe sous la forme d'une valeur numérique.

utilisateur : liste des utilisateurs appartenant au groupe. Ils sont séparés par des virgules :
compta:x:230:pierre, romain, jerome

1.3 Les commandes de gestion des utilisateurs
Afin de manipuler les fichiers passwd, shadow et group facilement, des commandes ont été crées. Elles automatisent les vérifications (l'utilisateur à créer existe-t-il déjà ? le mot de passe est-il assez compliqué ?...) évitant ainsi toute erreur de saisie.

1.3.1 Ajouter un utilisateur
La commande **useradd** permet de créer un utilisateur :
useradd [options] login

Principales options de useradd

options	Explications
-c commentaires	Informations concernant l'utilisateur (nom, poste)
-d répertoire	Chemin du répertoire personnel de l'utilisateur
-e date	Date d'expiration du compte. Le format est AAAA-MM-JJ
-f nbre de jours	C'est le nombre de jours suivant l'expiration du mot de passe après lequel le compte est désactivé. La valeur 0 permet de désactiver le compte dès que le mot de passe expire. La valeur -1 permet de désactiver cette caractéristique. La valeur par défaut est -1
-g groupe principal	Le nom du groupe ou le numéro du groupe de connexion initial de l'utilisateur. Le nom ou le numéro du groupe doivent exister. Le numéro de groupe par défaut est 1
-G groupes supplémentaire	Les autres groupes auxquels appartient l'utilisateur (séparés par des virgules)
-m	Le répertoire de l'utilisateur sera créé (par défaut, cela n'est pas fait)
-k [répertoire]	A utiliser si et seulement si l'option -m est présente. Permet de copier les fichiers et répertoires contenus dans le répertoire (/etc/skel si non spécifié) dans le répertoire de l'utilisateur
-p mot de passe chiffré	Vous pouvez saisir le mot de passe en option. Il doit être chiffré (pour récupérer la version cryptée d'un mot de passe il faut utiliser la bibliothèque crypt). Le comportement par défaut est de désactiver le compte
-s chemin vers un exécutable	Shell lancé à la connexion de l'utilisateur
-u uid	L'identifiant unique de l'utilisateur

Exemple : nous voulons créer l'utilisateur lambda :
commentaire : "utilisateur lambda"
son shell : /bin/zsh
son répertoire personnel : /home/lambda
nous recopions le contenu de /etc/skel dans son répertoire
il appartient aux groupes "dev" et "final"
le compte sera désactivé immédiatement après expiration du mot de passe
root@localhost # useradd -c "utilisateur lambda" -f 0 -G final, dev -m -k /etc/skel -s /bin/zsh lambda

La commande useradd -D montre les options par défaut :
GROUP=100
HOME=/home
INACTIVE=-1
EXPIRE=
SHELL=
SKEL=/etc/skel

Il est possible de changer les valeurs par défaut grâce aux options suivantes :
Options de useradd -D

options	explication
-h répertoire	Répertoire dans lequel créer les répertoires utilisateurs
-e date	La date d'expiration du compte
-f nombre de jours	Délai de désactivation après expiration du mot de passe
-g GID	Le groupe par défaut
-s chemin vers un exécutable	Le shell par défaut

1.3.2 Modification d'un utilisateur
#usermod -options
-u,-g,-c,-d,-s => identique à useradd
-l nouveau_nom nom_actuel => renomme le compte
-G => Applique les groupes auxiliaires **Attention:** Tous à chaque fois

1.3.3 Suppression d'un utilisateur

La commande userdel permet de supprimer un utilisateur :
userdel options login
REMARQUE: L'option -r efface le répertoire personnel de l'utilisateur.

1.3.4 Changer le mot de passe d'un utilisateur

La commande passwd permet de changer le mot de passe d'un utilisateur. L'administrateur peut changer n'importe quel mot de passe. Un utilisateur normal ne peut changer que son propre mot de passe.

passwd [options] [login]

Si l'argument login n'est pas spécifié, le changement de mot de passe s'applique sur l'utilisateur courant.
Options passwd

Option	Explication
-k	Indique que seul le mot de passe doit être mis à jour sans toucher aux propriétés d'expiration
-l	Permet de verrouiller le compte spécifié en préfixant le mot de passe crypté par le caractère "!". Seul l'utilisateur root peut utiliser cette option
--stdin	Le mot de passe doit être lu à partir de l'entrée standard qui peut alors être un tube (pipe)
-u	Déverrouille le mot de passe du compte. Seul l'utilisateur root peut utiliser cette option
-d	Supprime le mot de passe d'un compte. Le champ réservé au mot de passe crypté sera supprimé dans le fichier de configuration. Seul l'utilisateur root peut utiliser cette option
-S	Affiche des informations sur le statut du mot de passe pour un compte donné. Seul l'utilisateur root peut utiliser cette option

1.3.5 Afficher des informations sur un utilisateur

Pour connaître l'identité de l'utilisateur courant (bien que cela soit affiché dans la majorité des prompts par défaut) on utilise la commande whoami.
Elle affiche le login de l'utilisateur courant. Les commandes who, users et w permettent de connaître les utilisateurs actuellement connectés sur la machine.

1.4 Les commandes de gestion des groupes

1.4.1 Créer un groupe

La commande groupadd permet de créer un nouveau groupe :

root@localhost # groupadd option groupe

Options de groupadd

Option	Explication
-g	Permet de choisir la valeur numérique du GID du nouveau groupe. Cet identifiant doit être unique
-r	Cette option permet d'ajouter un groupe système (dont le GID est inférieur à 500)
-f	Permet de stopper la commande lorsque le groupe ou le GID du nouveau groupe existe déjà

1.4.2 Suppression d'un groupe

Pour supprimer un groupe, on utilise la commande groupdel :
#groupdel GID

REMARQUE:
on ne peut pas supprimer un groupe si c'est le groupe principal d'un utilisateur.

1.4.3 Modifier les groupes secondaires d'un compte

Pour modifier les groupes secondaire d'un utilisateur, on utilise la commande usermod qui est similaire à useradd et supporte les mêmes options :

root@localhost # usermod -G toto, users, fileshare, dev toto
Ceci permet d'ajouter l'utilisateur "toto" dans les groupes "toto", "users", "fileshare" et "dev".

Attention : lors de l'utilisation de la commande usermod -G il est nécessaire de rappeler l'ensemble de groupes secondaires auxquels appartient l'utilisateur.

On peut aussi ajouter et enlever des utilisateurs d'un groupe grâce à la commande gpasswd
Options de gpasswd

Option	Explication
-a	Ajout d'un utilisateur
-d	Suppression d'un utilisateur

1.4.4 Afficher des informations sur les groupes
Pour connaître les groupes auxquels appartient un utilisateur, on utilise la commande groups. Sans argument, elle affiche les groupes de l'utilisateur courant. Pour connaître less groupes d'un utilisateur particulier, il suffit de passer son login en argument de la commande :

root@localhost # groups
root wheel disk adm sys daemon bin
luser@localhost $ groups toto
toto users fileshare

Sur de très anciens SystemV, il n'était pas possible d'activer plusieurs groupes simultanément pour le même utilisateur. La commande id permet de connaître les groupes actifs :

root@localhost # id
uid=0(root) gid=0(root) groupes=0(root), 10(wheel),6(disk),4(adm),3(sys),2(daemon),1(bin)

Pour changer le groupe actif sur un tel système, on utilise la commande newgrp. Lorsqu'elle est utilisée sans argument, elle active le groupe principal de l'utilisateur (le group e qui figure dans "/etc/passwd").

1.4.5 Modifier un groupe:
identique à usermod pour les groupes
#groupmod

1.5 Changer d'identité (SU)
Il se peut qu'un utilisateur soit obligé d'effectuer des tâches avec une identité différente. La situation la plus courante étant un utilisateur normal voulant effectuer quelques tâches en tant que administrateur.

La commande su (switch user), permet de changer d'identité:

Commande	Explication
su	Sans option, su permet de se connecter en tant que root
su nom_user	Agir en tant qu'utilisateur nom_user
su - nom_user	Se connecter en tant que nom_user avec "-" on récupère tout l'environnement de l'utilisateur lambda
su -c "mount /dev/cdrom /mnt/cdrom"	l'option -c permet simplement d'exécuter la commande entre guillemet en tant que root (pas d'option). Une fois la commande terminée, on reprend son identité

Droits d'accès :

Tout fichier du système appartient à la fois à un utilisateur (son "propriétaire") et à un groupe. Ainsi, pour ***chaque fichier le monde de ses utilisateurs potentiels est scindé en 3 catégories, nommées*** :

1. **u**, l'**u**tilisateur normal, son propriétaire, bien souvent son créateur, qui n'a pas pour autant tous les droits sur lui !
2. **g**, son **g**roupe, ensemble d'utilisateurs ayant parfois des "permissions" particulières.

o, tous les (**o**thers) autres.
Attention: l'utilisateur propriétaire et le groupe propriétaire du fichier peuvent être indépendants :

- le groupe propriétaire n'est pas forcément le groupe primaire de l'utilisateur propriétaire,
- et même, le propriétaire n'est pas forcément membre du groupe !

Mais (heureusement) une règle générale simple s'applique à la création de tout nouveau fichier (ou rép.)

- son propriétaire est l'utilisateur (humain ou système) qui l'a créé
- son groupe est le groupe *primaire* de ce même utilisateur

Droits d'accès des utilisateurs aux fichiers

Généralités

Linux permet de spécifier les droits d'action sur un fichier, que peuvent exercer les utilisateurs des 3 catégories précédentes, ou plutôt les ***permissions*** que leurs accordent les fichiers et les répertoires.

Linux a repris les 3 protections d'UNIX sur les fichiers et les répertoires. Leur notation symbolique est :

1. **r,** lecture
2. **w**, écriture
3. **x**, exécution

De façon générale, ces permissions sont consultable complètement par la commande :

ls -l

Rappel : ll est un alias plus court, pour la commande *ls -l*

Par exemple :

[stagex@p0x stagex] ll *.html
-rw-r--r-- 1 stagex stagex 1200 oct 19 12 : 39 amoi.html

Description globale

On trouve de gauche à droite

- le 1er caractère indique la nature du fichier
  "**-**" fichier normal, "**d**" un fichier répertoire, "**l**" un lien.
- le système de droits est spécifié symboliquement par les 9 attributs suivants, correspondants aux 3 catégories d'utilisateurs du fichier.
  ...|...|...
  u g o

La section u fixe les droits accordés au propriétaire du fichier.

La section g fixe les droits accordés aux utilisateurs faisant partie du groupe auquel appartient le fichier.

La section o fixe les droits des autres utilisateurs.

- nombre de liens sur le fichier 1 signifie que le fichier n'a aucun lien qui pointe vers lui, 2 (ou plus) signifiant qu'il existe un lien (ou plus) vers lui.
- le nom du propriétaire du fichier
- le nom du groupe propriétaire
- la date de dernière modification
- le nom complet du fichier

Changement de droits

De façon générale, l'utilisateur qui crée un fichier en devient le propriétaire, et le groupe auquel l'utilisateur appartient (au moment de la création) devient le groupe du fichier.

Remarques préalables

- Mais les droits accordés au propriétaire, au groupe et aux autres dépendent du processus qui a créé le fichier et du masque des droits.
- D'autre part l'administrateur peut être amené à effectuer des changements de propriété (par exemple pour permettre un travail en groupe) et des changements de droits sur des ensembles de fichiers et de répertoires, les étendre ou les restreindre.
- Et root n'est pas soumis à ces restrictions, il a le pouvoir absolu sur ... le système de fichiers. En contrepartie il peut être considéré comme responsable de tout dysfonctionnement !

- **Changer le propriétaire**
  chown [-R] nv-user fichiers
  Commande réservée au propriétaire actuel des fichiers ou des répertoires (et à root)
  L'option **-R** (récursif) permet d'agir sur l'ensemble des sous répertoires.
  Exemple : chown -R stage4 /home/stage1

- **Changer le groupe propriétaire**
  chgrp [-R] nv-groupe fichiers
  Ceci doit être effectué par root ou le propriétaire, à condition que celui-ci soit membre du nouveau groupe.
  Exemple : chgrp -R stage4 /home/stage1

- **Changer les 2 en même temps**
  chown nv-user.nv-groupe fichiers
  chown new-user.fichiers
  Dans ce cas, en plus, le groupe propriétaire des fichiers est changé pour le groupe primaire du nouveau propriétaire.

- **Changer les permissions sur les fichiers**

- Les droits d'accès peuvent être modifiés par le propriétaire des fichiers ou par root (ou équivalent, d'uid 0).
- La commande **chmod** (*change mode*, change le "mode" des fichiers) peut s'écrire de plusieurs façons équivalentes, sur le modèle :
  chmod *droits* fichiers
  Le paramètre *droits* permet de calculer les nouveaux droits d'accès.
- Ceux-ci peuvent s'obtenir de façon *relative*, par ajout (symbole +) ou retrait (-) par rapport aux droits existants, ou bien de façon *absolue*, en fixant les nouveaux droits qui remplacent les anciens (symbole =).

Ajout, retrait ou fixation des permissions

Pour chaque fichier, on désigne par :

- **u, g et o** les 3 catégories d'utilisateurs (user, group, other) et de plus par **a** (=all) tous les utilisateurs.
- **r,w,x** les 3 attributs de chaque fichier, pour chaque catégorie d'utilisateur.
- **+ - =** l'action d'ajouter, de retirer ou de fixer un droit, qui s'applique à chaque catégorie séparément.
- les changements, sur le modèle "*à quelle(s) catégorie(s), quelle action, quel(s) droit(s)*" sont alors notés symboliquement :
  [u g o a] [+ - =] [r w x]
- par exemple **chmod u+x fichier** signifie "ajouter le droit d'exécution au propriétaire du fichier"
- on peut regrouper les catégories si on veut exercer la même action :
  chmod ug+w fichier "ajouter le droit d'exécution au propriétaire et au groupe"
  chmod go-rwx fichier "enlever tous droits d'accès à tous les utilisateurs, sauf au propriétaire"

Compléments indispensables
Notation octale des permissions
Il existe une autre façon d'indiquer les permissions de chaque catégorie, plus simple en utilisant la numération octale
Voici la table de correspondance entre les 8 chiffres en numérotation octale (base 8) et les 8 valeurs de droits fichiers.
Par convention la présence d'un droit est notée 1, l'absence 0.

Binaire	Droit	Octal
000	(---)	0
001	(--x)	1
010	(-w-)	2
011	(-wx)	3
100	(r--)	4
101	(r-x)	5
110	(rw-)	6
111	(rwx)	7

Synthèse : notation globale pour les 3 catégories

propriétaire			groupe			autre		
lecture	écriture	exécution	lecture	écriture	exécution	lecture	écriture	exécution
400	200	100	40	20	10	4	2	1

Pour obtenir les permissions exprimées en octal, il suffit d'ajouter en octal les nombres de la table de correspondance ci-dessus, pour lesquels les droits sont positionnés.

Exemples
chmod 700 /home/rep-a-moi *droits par défaut pour un rép. personnel*.
ls -l /home/rep-a-moi
--> drwx------

Les 2 commandes suivantes sont équivalentes :
chmod 764 test
chmod u=rwx,g=rw,o=r test
ls -l test
-rwxrw-r--

Le masque de protection umask

- Rappelons les règles simples de propriété qui s'appliquent à la création d'un fichier ou d'un répertoire :
  - son propriétaire est l'utilisateur qui l'a créé
  - son groupe est le groupe *primaire* de ce même utilisateur
- Mais quelles sont les permissions attribuées *par défaut* à l'utilisateur propriétaire, au groupe propriétaire et à tous les autres ?
  Les permissions maximales accordées par un fichier et un répertoire sont 666 (-rw-rw-rw-) et 777 (-rwxrwxrwx).
  On peut restreindre ces permissions lors de sa création. C'est le rôle de la commande **umask** de fixer les permissions *masquées*, autrement dit **les droits non accordés aux fichiers et répertoires lors de leur création**.

Exemple de calcul de permissions effectives, affectées lors de la création d'un répertoire, par un utilisateur dont le masque de protection est **027**

777 = 111 111 111 permissions maxi = rwx rwx rwx
\- 027 = 000 010 111 masque de protection
= 750 = 111 101 000 permissions effectives = rwx r-x ---

- **La commande umask**
  - **umask** affiche le masque de l'utilisateur actif
    Quelles sont les valeurs des masques par défaut de root et des autres utilisateurs ?
  - **umask -S** affiche les permissions correspondantes au masque, sous forme symbolique.
  - **umask *masque*** fixe les permissions ultérieures de création des fichiers de l'utilisateur actif, conformément à *masque*, en notation octale.
    Attention ! le changement ne s'applique qu'à la présente session.
  - Pour la rendre permanente, on peut intervenir sur un fichier *profile* :
    - Dans le fichier profil général **/etc/profile**, on peut modifier la règle habituelle :
      if [$UID == 0] ; then umask 022 ; else umask 077 ; fi
    - Pour agir au niveau des utilisateurs, ajouter la ligne umask *masque* dans le fichier de profil personnel $HOME/.bash_profile

La commande umask est défini par défaut est 022 :

On part des droits maxi :	777
umask :	-022
Droits sur les REP	=755
Retrait du droit X pour les fichiers :	-111
Droits sur les fichiers :	=644

On ne s'occupe que des trois derniers chiffres, le premier concernant les droits spéciaux.

Exemple : umask 077 défini des fichiers crées par l'utilisateur en cours avec les permissions 777-077 = 700-111=600 (ils seront cachés aux autres utilisateurs).

LES TÂCHES PLANIFIÉES ET SAUVEGARDES

ATTENTION: Les Droits et l'environnement de l'utilisateur sont chargés avant l'exécution de la tâche.

1. les tâches ponctuelles

#at paramètre_de_temps -option [ENTER]
#commande1 [ENTER]
#commande2 [ENTER]
#[CTRL]+[d]

Temps de manière absolue: La tâche aura lieu à l'heure indiquée:
HH:MM
MM/DD/YY
midnight (00h00)
noon (12h00)
teatime (16h00)
tomorrow (même heure le lendemain)

Le temps de manière relative: Now+ temps T
now+n min
now+n hours
now+n days
now+n weeks
now+n months
Remarque: pas de temps inférieur à la minute, les minutes sont résolues
ex: now +1min => à 10h03 et 15 secondes, il agit à 10h04 pile

dæmon du planificateur de tache **.atd** qui vérifie la liste des tâches toutes les minutes

exemple:
#at now+1min
at > echo coucou > /dev/pts/0 => (le chemin de console actuelle est obtenu avec " tty")
[CTRL]+[D]

#at -l => ou #atq affiche la liste des travaux dans la file at

Pour supprimer une tâche dans la liste (root):
#at -dn | Permet de supprimer la
#atrm n | tâche n

pour lancer un scripte à un moment donné:
#at -f nom_fichier paramètre_de_temps

Visualiser le contenu d'une tâche planifiée:
#ls /var/spool/at => Affiche la liste des tâches
#cat numtâche => affiche une tache
Attention : ne pas les modifier avec vi => bloque la tâche

/etc/at.allow	*/etc/at.deny*	***Permissions utilisateurs***
existe		Permissions de at.allow
existe	existe	Permissions de at.allow (prioritaire)
n'existe pas	n'existe pas	Seulement root
n'existe pas	existe	Utilisateur qui ne sont pas dans le at.deny
n'existe pas	Existe et vide	Tout le monde

2. les tâches périodiques avec CRONTAB

Le spooler crontab est dans /var/spool.cron
#crontab -e => lance vi qui écrit dans un fichier (tmp/crontab.n à ne pas éditer manuellement)
#crontab -l -u *nomuser* => liste les tâches d'un utilisateur donné
#crontab -e -u *nomuser* => Editition crontab d'un utilisateur donné
#crontab -r => Arrête le crontab actif

Champ 1	***Champ 2***	***Champ 3***	***Champ 4***	***Champ 5***	***Champ 6***
00-59 minutes	00-23 heures	01-31 jour du mois	01-12 mois	0-7 jour de la semaine dimanche à dimanche (prog)	Commande script

* toutes les valeurs
*/n toutes les valeurs en partant de la première possible
m, n pour la valeur m et la valeur n
m-n pour les valeurs de m à n
m-n/p pour les valeurs de m à n une fois sur p
ex: 0-30/3 toutes les 3 minutes de 0 à la demi

Exercice : programmer les tâches suivantes:

A 16h30 et 17h30 la veille de Noël et du jour de l'an
=>30 16,17 24,31 12 * cmd

de 22h à 4h30 toutes les 10min dans la nuit du samedi au dimanche
=> */10 22,23 * * 6 cmd
=> 0/10 0-3 * * 7 cmd

Le 1er jour de chaque trimestre à 1h15
=> 15 1 1 1,4,7,10 * cmd
=> 15 1 1 1-12/3 cmd
=> 15 1 1 */3 * cmd

Le dernier jour du mois à 23h45
=> 45 23 31 1,3,5,7,810,12 * cmd
=> 45 23 30 4,5,9,11 * cmd
+programme pour le mois de février

le 15 du mois, à 2h30, si c'est le week-end
=>30 2 15 * 6,7 cmd
PB: Dû à l'interdépendance des champs 3 et 5 : Crontab le fait le 15 et tous les samedi et dimanche, donc test par programme (si le 15 est en week-end?)

Les autorisations clients de crontab:

/etc/cron.allow	***/etc/cron.deny***	***Permissions utilisateurs***
existe	n'existe pas	Utilisateurs indiqués dans cron.allow
n'existe pas	existe	Utilisateurs qui ne sont pas dans cron.deny
n'existe pas	n'existe pas	Root au delà du niveau de sécurité standard tout le monde jusqu'à standard

Exercice:
Créer un programme qui enregistre toutes les 3 minutes de 10h15 à 10h30, les 5 derniers messages du système dans un fichier /tmp/lastmsg, avec l'heure d'enregistrement de ces 5 messages, sans écraser le fichier.
A 10h30, prévenir root que la surveillance est fini.
Rem:La liste de tous les messages est dans /var/log/messages
surveillance

```
#!/bin/bash
echo "liste des 5 derniers messages à `date +%T `:" /tmp/lastmsg
tail -5 /var/log/messages >> /tmp/lastmsg
echo 0                    =>permet un retour chariot
```

Dans le fichier crontab:
21-36 /3 10 * * /root/surveillance
36 10 * * * echo "surveillance terminée" > /dev/tty

3. les tâches en arrière plan

Commandes de gestion des tâches en arrière plan:
#cmd & => place la commande en arrière plan (on garde la main sur le bash pendant l'exécution de l'application)
#jobs =>liste des tâches actuellement en arrière plan
fg *numjob* => replace le job portant le numéro numjob en avant plan
bg *numjob* => replace le job portant le numéro numjob en arrière plan
#CTRL + Z => suspend la tâche en avant plan

4. les tâches par priorité

Commandes de gestion des priorités:
#nice =>Affiche la priorité du travail en cours
Valeurs possibles de -20(plus prioritaire) à +19(moins prioritaire) 0 par défaut

Remarque: les valeurs négatives sont réservées à root
#nice -n priorité commande => Affecte la priorité indiquée à la commande donnée.
#renice new_priorité PID => Modifie la priorité du processus indiqué
#renice new_priorité -u nom_user =>Modifie la priorité de tous les travaux de l'utilisateur
#ps -elf => affiche les processus avec une colonne NI (nice)

SAUVEGARDE ET RESTAURATION

dd: Disk Dump Permet de copier des disques et convertir avec choix de la taille des blocs. Surtout utilisé pour copier des partitions

#dd if=*nom_fichier_source* of=*nom_fichier_destination* [options_supplémentaires]

options:

bs= => (bloc size) taille des blocs 512,1024,2048...

count= => nb de blocs

#dd if=/etc/passwd of=/dev/fd0 (if= input file of= output file)

- Il ne s'occupe pas du point de montage et effacera la disquette si il la trouve (l'avantage c'est que les données sont écrites de façon contigu
- la restauration restaure le fichier et les données présentes sur la disquette
- Il enregistre les données sous forme de blocs de 512o par défaut
- on ne peut pas lire la disquette avec un ls car il écrit en brut les données

Pour restaurer le fichier:

#dd if=/dev/fd0 of=restaure_passwd

Pour créer un fichier de taille prédéterminé:

#dd if=/dev/zero of=file.10Mo bs=1k count=10000

tar: (Tape Archival)

Sauvegarde:

la sauvegarde sauve les fichiers sous le / de début du nom. Il faut donc se placer dans le répertoire racine /avant de restaurer:

#cd /

#tar -xvf /dev/fd0

#tar -C / -xvf dev/fd0 => ***ATTENTION:*** pas de / devant dev

ATTENTION: la destination est devant la source.

#tar -options *fichier_de_sauvegarde sources_à_sauvegarder*

options:

-c => Create : créer une sauvegarde.

-x => Extract: restaurer une sauvegarde.

-t => Lister le contenu d'une sauvegarde.

-v => Verbose

-f => file : Nom du fichier archive (si – alors le f doit être la dernière option juste avant le nom du fichier

-C => restaure dans le répertoire nommé

-M => Multi Volume

-l 1440 => Longueur du support

-k => n'écrase pas les fichiers communs

--exclude => Exclus in fichier à restaurer

Options de compression

-z => Compression ou décompression au format gzip

-j => Compression ou décompression au format bzip2 (plus performant)

ATTENTION:

Les options de compression DOIVENT être fourni lors de la restauration de la sauvegarde sinon erreur. On essai donc de leur donner une extension gzip ou bz3 pour les repérer.

Restauration:

#tar -xvf /dev/fd0 => la destination est le chemin relatif

Pour restaurer un seul fichier:

#tar -cf /backup /home/poste23

#tar -xf /backup home/poste23/fichier1 fichier2

cpio (copy input output)

Remarque: Contrairement à tar, il sauvegarde le chemin absolu donc pas de problèmes lors de la restauration. Elle ne gère pas la compression.

#cpio options >nom_archives => Ensuite il demande les fichiers
-o => output : sauvegarde
-i => input
-v => Verbose
-t => Liste : Visulisation

Remarque: On utilise souvent find | cpio pour sauvegarder la réponse de find

Exemples:
sauvegarde des fichiers appartenant à poste1
#find /home -user poste1 | cpio -ov > /dev/fd0

sauvegarde du fichier /etc/passwd en mode interactif:
#cpio -ov > /dev/fd0
/etc/passwd [ENTER]
/etc/pass [ENTER]
[CTRL][D]

en mode non interactif
#cpio -o > save.passwd < /etc/passwd /etc/pass

Visualisation d'une archive sur disquette:
#cpio -ivt < /dev/fd0

Restauration d'une archive sur disquette
#cpio -iv < /dev/fd0

ATTENTION: La commande de restauration doit être la même que celle de sauvegarde (référencez-la).

ADMINISTRATION RÉSEAU

Configuration du réseau:

Les commandes de configuration du réseau:
#ifconfig =>gestion des interfaces
#ifconfig nom_interface up options
ATTENTION config dynamique down jusqu'au redémarrage car pas de modification des fichiers de configuration

#hostname => Change et affiche le nom d'hôte de la machine (dynamique)

route: gestion de table de routage (dynamique)
#route -n => Affichage numérique

ping : test ip
#ping -c 4 => Nombre de ping ip (ici 4)
#ping -b => test de la plage par broadcast

traceroute permet de retracer les routeurs par lesquels nous passons
#traceroute ip_destination

netstat : liste les infos réseau et services
#netstat => liste les ports ouverts LISTEN=service actif
ESTABLISH=en cours d'utilisation
#netstat -na => écoute tous les ports de la machine

Les Fichiers:
Mandrake et Redhat:
/etc/init.d/network => scripte de démarrage du réseau
/etc/sysconfig/network-scripts/ifcfg-eth0 => Configuration de la carte 0

Exemple:

DEVICE=eth0
BOOTPROTO=static => IP manuelle
IPADDR=192.168.107.7 => Adresse ip
NETMASK=255.255.255.0 => masque de sous réseau
NETWORK=192.168.107.0 => réseau associé
BROADCAST=192.168.107.255 => Adresse de broadcast
ONBOOT=yes => lancer au démarrage
MII_NOT_SUPPORTED=no => détection automatique de présence du câble réseau

---

/etc/sysconfig/network => Nom de la machine, passerelle par défaut, routage de paquet

NETWORKING=yes
GATEWAY= 192.168.10.254 => Passerelle
/etc/sysconfig/network/device/eth0.route => Routes statiques liées à l'interface nommée ou /route-eth0
/etc/sysconfig/static-routes => Routes statiques
/etc/services => liste des ports TCP et UDP et services associés
/etc/nsswitch.conf => ordre des services à consulter pour la résolution de nom
/etc/resolv.conf => Liste des adresses ip des serveurs DNS nameserver 194.2.0.20
/etc/hosts => liste des correspondances nom hôte <=> ip

exemple
127.0.0.1 localhost
192.168.10.7 thierry tit

Mise en place d'un réseau routé:

1. **Configurer une carte réseau**
   si il y a deux cartes, commencer par paramétrer la deuxième carte
   /etc/sysconfig/network-scripts/ifcfg-eth0
   /etc/sysconfig/network-scripts/ifcfg-eth1
   Si il y a une seule carte, copie répertoires fichier eth0 en eth0:0 pour créer une autre carte virtuelle avec adresse ip différente
   ATTENTION: renommer la carte eth0 en eth0:0 par exemple

2. **Activer le routage**
   /etc/sysconfig/network
   HOSTNAME= nom_hôte
   FORWARD_IPV4=true => active le routage ipv4 pour toutes les interfaces
   ATTENTION: pas la même option pour toutes les versions Mandrake et autres!!!

3. **Redémarrer le service network**
   #service network restart

4. **Créer une route:**
   - temporaire
     #route add -net 192.168.107.0 netmask 255.255.255.0 gw 192.168.10.254
     réseau dest masque du réseau passerelle à joindre
   - permanente (lancée au démarrage du système)
     dans /etc/sysconfig/networking/devices/eth0.route
     route add -net 192.168.107.0 netmask 255.255.255.0 gw 192.168.10.254

5. **Déclaration de la passerelle par défaut:**
   - temporaire:
     #route add default gw ip_passerelle

   - permanent:
     /etc/sysconfig/network
     GATEWAY= ip_passerelle

LE SERVICE XINETD (MODULE 7 PAGE 12)

But:

- Superviser des services réseaux à activité faible (ou moyenne) de manière centralisée, homogène et flexible (protocoles utilisant TCP).
- Optimiser des ressources CPU en évitant d'avoir des services inactifs, mais utilisant de la CPU
- **Désavantages:** les services dépendent de xinetd pour fonctionner (un seul point faible réseau pour tous ces services.

Exemple de services pris en charge par xinetd:

```
                                    Ports TCP
pop             |                   | 110   |
swat            |                   | 901   |          Client attaque le port telnet
telnet          |---------xinetd------ | 23    | => du serveur xinetd avec un port client aléatoire Ex:3596
imap            |                   | 143   | => xinetd lance un processus fils telnet pour la connexion
```

Configuration:

```
/etc/xinetd.d/telnet
                /swat
                /imap
```

Exemple:

```
-------------------------------------------------------
service nom_du_service
{
disable             = yes
# désactiver le service au démarrage (yes ou no)
server   = /usr/sbin/in.ftpd
# Commande de lancement du service
only_from           = nom_host_ou_ip_host_ou_ip_reseau autre_nom
# liste des machines autorisées
no_access           = nom_host_ou_ip_host_ou_ip_reseau autre_nom
# liste des machines interdites
access_times    = HH:MM-HH:MM  HH:MM-HH:MM
# restreindre l'accès au service
instances           = 10
#limiter le nombre de connections simultanées autorisés (ici 10)
redirect = 192.168.10.254 23
#redirige les requêtes vers ce serveur sur le port indiqué

#utilisation en tant que détecteur/bloqueur d'intrusions
flags               = SENSOR
# sensor bloque le service xinetd entier pour cet utilisateur
deny_time           = forever
# combien de temps on le bloque (n= en minute, forever=jusqu'au redémarrage du service, never=jamais)
}
```

ATTENTION:

en cas de conflit entre only from et no access c'est la commande la plus précise qui gagne. Si égalité c'est no access suite à toute modification, on redémarre et on regarde les logs (alias dans le .bashrc permet de simplifier ça).

Dans /root/.bashrc
ajouter
alias msg='tail -30 /var/log/messages'
et relancer le .bashrc
#. .bashrc

Exercice:

Rediriger toutes les connections telnet vers l'IP 192.168.10.254
vi /etc/xinetd.d/telnet

```
service telnet
{
disable = no
server   = /usr/sbin/telnetd
flags    = REUSE
redirect = 192.168.10.254 23
}
```

Exercice:
Pour toutes les ip du réseau 10.0, stocké dans un fichier, celles qui répondes au ping

```
#!/bin/bash
touch /var/scan

for i in `seq 1 254`
do
        ping -c 1 192.168.10.$i > /dev/null && echo 192.168.10.$i >> scan
done

clear
cat /var/scan
rm -f scan
```

DOMAINE NAME SERVICE (DNS BIND)

Ports associés:
TCP 53, 953 : ports standard et sécurisé de transfert de zone
UDP 53, 953 : ports standard et sécurisé de requête DNS

Packages : Bind, bind-utils, caching-nameserver

Les fichiers pour Mandrake:

/etc/named.conf	=> Liste des zones administrées
/var/named/	=> Répertoire contenant un fichier de description par zone administrée.
/var/named/named.local	=> zone locale pour le cache
/var/named/named.ca	=> liste des serveurs racines
/etc/nsswitch.conf	=> ordre de résolution des noms d'hôte
/etc/resolv.conf	=> ip des serveurs DNS pour le client DNS

mode sécurisé
/etc/rndc.conf
/etc/rndc.key

Commandes côté serveur:

#service named stop|start|restart|reload	=> Mandrake
#/etc/init.d/named stop|start|restart|reload	=> toutes distributions
#rndc flush	=> Vidage du cache client et serveur
#named-checkconf	=> Vérifie la syntaxe de /etc/named.conf
#named-checkzone	=> Vérifier la syntaxe d'une zone

Requêtes côté client:
#nslookup

#host: (résout le nom spécifié)
host nom_dns
host -l nom.zone => Récupère le contenu d'une zone entière à partir du 1er serveur dans /etc/resolv.conf
host -t type nom_zone => type: SOA, PTR, A, MX...

#dig

DOMAINES: ILS SONT COMME DES BOITES DANS LESQUELLES ON RETROUVE DES ENREGISTREMENTS
Types d'enregistrements DNS:

SOA => Start of autorité (défini les données sur le transfert de zone) Un SOA par zone (obligatoire)

A => Hôte (donne FQDN reçoit une ip associée)
www.free.fr. IN A 192.168.1.24

PTR => Pointeur équivalent à A dans les zones inversées (donne ip et reçoit le FQDN)
253 IN PTR www.free.fr.

NS => name server indique les serveurs (primaire ou secondaires qui hébergent cette zone)

FQDN_server	IN	NS	FQDN_domaine
dns1.free.fr.	IN	NS	free.fr.
ou			
nom-hôte_server	IN	NS	FQDN_domaine
dns1.	IN	NS	free.fr.

CNAME => Canonical name (alias)

nom_alias	IN	CNAME	nom_FQDN_existant
www.free.fr.	IN	A	192.168.2.1
ftp.forteam.fr.	IN	CNAME	www.free.fr.

MX => Messagerie SMTP de la zone

nom.zone.dns.	IN	MX	x	FQDN_serveur_messagerie *(x = priorité de 1 à 99(le plus faible est le plus prioritaire) pas de force du poids (1 et 5 = 10 et 70)*
free.fr	IN	MX	15	nom-FQDN_serveur_messagerie.

SRV **=>** enregistrements de service permettant de retrouver les serveurs de ce service par le biais de la recherche centralisée DNS

```
_tcp.nom_service.    IN    SRV    x    y    FQDN_serveur_hébergeant_ce_service
ou
_tcp.numero_port.    IN    SRV    x    y    FQDN_serveur_hébergeant_ce_service
```

x=priorité de ce serveur pour ce service
y=poids de ce serveur en % par rapport aux autres serveurs de même priorité pour ce service

Exemple:

```
_tcp.23.    IN    SRV    1    75    telnet1.free.fr
_tcp.23.    IN    SRV    1    25    telnet2.free.fr
_tcp.23.    IN    SRV    2    100   telnet3.free.fr
```

les serveurs 1 et 2 seront prioritaires sur le dernier et seront distribués 75 fois sur 100 pour telnet1 et 25 fois sur 100 pour telnet2
telnet3 ne sera utilisé par le client DNS que si les deux premiers sont HS

REMARQUE:
TCP peut être remplacé par UDP

MISE EN PLACE DE L'ADMINISRTATION D'UNE ZONE DNS

1. Déclarer la zone dans /etc/named.conf

```
zone "nom_zone_dns" {
type master;                    => Type de gestion : master, slave, forward.
file "nom_fichier_zone";        => relatif au répertoire /var/named par défaut
        };
```

2. Créer le fichier décrivant la zone à administrer

ATTENTION: les lignes ne commencent **pas** par ESPACE
/var/named/nom_fichier

```
$TTL 1d
@       IN      SOA     FQDN.serveur.dns. Mail.admin_du_serveur. (
                valeur1         => numéro de série de la zone (maximum 10 chiffres)
                valeur2         => refresh : temps en seconde entre deux synchro (28800)
                valeur3         => retry : temps supplèmentaire en cas d'échec de connections
                valeur4         => expire: durée de vie des enregistrements après désynchronisé
                valeur5)        => minimum: ttl dans le cache avant de reconfirmer
                                IN      NS      FQDN.serveur.DNS.     =>serveurs de zone
FQDN.serveur.DNS.               IN      A       ip_du_serveur.dns
nom_host.zone.                  IN      A       ip_host
nom                             IN      A       ip_host => pas de . après host car nom relatif à la zone
```

ATTENTION: les parenthèses du SOA doivent être à la suite du texte et non décollées ou en retour chariot

3. Déclaration du ou des serveurs à contacter

=>/etc/resolv.conf

```
nameserver ip_serveur_DNS          3 maximums
nameserver ip_serveur_DNS
```

ATTENTION: le client ne passe au serveur suivant que si le premier serveur n'est pas TROUVE

4. Tester

#host nomhost.nom.zone

REDIRECTIONS

Comment diriger la requête vers un serveur quand notre serveur ne connaît pas le domaine demandé:
Dans /etc/named.conf
dans la section OPTIONS {
Ajouter:
forwarders {ip_serveur_dns;ip_autre_serveur_dns};

Pour une redirection conditionnelle, il faut créer une zone de type forward dans named.conf puis y mettre les serveurs de redirection pour cette zone
Ex: une redirection vers le serveur dns 192.168.10.56 pour toute demande vers le domaine poulet.malade.net:

```
zone "poulet.malade.net" {
        type forward;
        forward only     => seul le redirecteur indiqué doit être contacté
                first    =>  il est contacté en premier mais si échec on contacte le redirecteur général
        forwarders {192.168.10.56;};     => liste des redirecteurs
};
```

DÉLÉGATION DE ZONE DNS

On ne connaît que ce qui se trouve en dessous de soi. Ce qui est un bon principe car le serveur racine n'a pas à connaître toutes les adresses mais seulement les serveurs qui les possèdent.
On fait donc une délégation pour une zone pour que la zone présente sache quel(s) serveur(s) gère (nt) la zone enfant (forcément inférieure)

dans un fichier de zone on indique quel(s) serveur(s) gère (nt) quel(s) domaine(s) enfant(s).

pour test.fr on ajoute:

ifc.test.fr.	IN	NS	toto.ifc.test.fr	=> nom du domaine enfant
toto.ifc/test.fr	IN	A	192.168.10.59	=> ip du serveur de ce domaine enfant

MISE EN PLACE D'UNE ZONE DNS ESCLAVE (SECONDAIRE)
Les zones secondaires sont des zones DNS de sauvegarde identiques à leur zone maître en lecture seule. Elle peut répondre aux requêtes DNS de cette zone et se synchronise à intervalle régulier défini dans le SOA de la zone principale.

La synchronisation IXFR (Incremental eXchange.) se fait en tirer-pousser entre le serveur principal et le secondaire.

Côté zone principale:
on ajoute dans le /etc/named.conf, dans la section de la zone concernée:

```
allow-transfert {
ip_autorisé_à_obtenir_la_zone;
autre_ip_autorisé_à_obtenir_la_zone;
};
notify yes;                     =>indique que le master envoi des notifications de modifications
also-notify {ip.dns.slave;};    => Indique la liste des serveurs slaves à notifier.
```

Côté zone secondaire:
dans /etc/named.conf on crée une nouvelle section de zone:

```
zone "nom_zone_principale" {
        type slave;
        file "nomfichier.slave";
        masters {               => ATTENTION:au pluriel
        ipdnsmaster;
        };
        };
```

Options de synchronisation du SOA:

v	=> serial	: Numéro de série pour déterminer la synchronisation
w	=> refresh	: interval entre chaque synchronisation
x	=> retry	: interval avant nouvel essai après échec de synchronisation
y	=> expire	: interval depuis panne de synchronisation avant arrêt de la zone
z	=> minimum	: durée de vie minimum du cache DNS

Répartition de charge: avec une zone secondaire:
-sur 1000 clients, je dois mettre en configuration clients sur :
500 d'entre eux, serveur1 puis serveur2 (en backup)
les 500 autres, serveur2 puis serveur1 (en backup)

Tolérance de panne:
automatique car les clients qui ne trouvent pas leur serveur primaire, iront voir le serveur de sauvegarde

REMARQUE: Une zone slave peut être slave d'une autre zone slave

Autres options intéressantes:
au niveau général ou au niveau de zone (prend le pas sur général si présent) :

allow-query	=> spécifie qui a le droit de faire des requêtes.
Allow-recursion	=> autorise les requêtes récursives
allow-transfert	=> autorise les transferts de zone
blackholl	=> désigne une liste de machines non autorisées à faire une requête
listen-on port 1234 { !1.2; 1.2/16; };	=> indique sur quelles interfaces et ports, le serveur DNS réponds (port 53 si non spécifié)

Création d'une zone de recherche inversée
Les zones inversées sont utilisées par des services (messagerie, serveur web...) pour obtenir le nom du domaine d'un client par rapport à sont adresse ip pour vérifier sa validité.

Dans .etc/named.conf
```
zone "10.168.192.in-addr.arpa." {
type master;
file "domtom.fr.rev";
};
```

dans le fichier /var/named/domtom.fr.rev
```
---------------------------------------------------------------------------------------------------------
@       IN      SOA     10.168.192.in-addr.arpa.        Titi.free.Fr (
                        1
                        3600
                        3600
                        3600
                        3600)
        IN      NS      thierry.domtom.fr.
254     IN      PTR     thierry.domtom.fr.
10      IN      PTR     www.domtom.fr.
---------------------------------------------------------------------------------------------------------
```

Exercice:
Chercher le nombre de transferts de zone:
- que vous avez effectués depuis ce matin en tant que slave.
- que vous avez autorisé depuis ce matin en tant que master.

Indiquez ensuite si votre serveur est plutôt master ou slave

```
-------------------------------------------------------
#!/bin/bash
$transfert = `cat /var/log/messages | grep "Aug 25" | grep -w transfer | wc -l `
$client = `cat /var/log/messages | grep Aug 25 | grep -v client | wc- l `
if [ $transfert -gt $client ]
then
        echo "DNS Serveur plus actif"
else
        echo "DNS slave plus actif"
fi
-------------------------------------------------------
```

MISE À JOUR DYNAMIQUE D'UNE ZONE DNS

Coté master
dans le fichier /etc/named.conf on modifie la section du domaine concerné:

```
zone "nom_zone." {
        type master;
        file "fichier.dns";
        allow-update {
```

Cas 1:
Laisser les postes clients s'enregistrer eux-mêmes dans la zone DNS.
Dans /etc/named.conf

```
zone "corse.fr" {
        type master;
        file "corse.fr";
        allow-update {                                  => seulement sur zone master
                ip_autorisées_pour_maj_dynamique;
                ip_séseau_cidr;
                0.0.0.0;                        => Tout le monde
                };
};
```

Les mises à jour dynamiques sont inscrites dans un fichier /var/named/corse.fr.jnl (journal non texte) crée après activation des MAJ dynamiques qui enregistre les clients dynamiques avant de les transférer dans le vrai fichier de zone à intervalles régulier.
Un TTL est présent dans le fichier de zone pour tout enregistrement dynamique.

ATTENTION:
Pour voir les enregistrements dynamiques avant leur copie dans le fichier de zone, on tape:
#host -l corse.fr => Affiche tous les enregistrements statiques et dynamiques mais ne différencie pas les deux types d'enregistrement.

ATTENTION:
tout client DNS peut s'enregistrer et donc saturer le serveur avec des enregistrements. C'est une faille de sécurité.

PRINCIPE DU ROUND-ROBIN:
si le serveur DNS possède plusieurs réponses pour une requête, il donne la liste des réponses de manière cyclique (et linéaire)
Exemple:

```
        ftp.free.fr     IN      A       192.168.2.1
        ftp.free.fr     IN      A       192.168.2.2
```

le premier client aura l'ip 2.1 le suivant 2.2 le suivant 2.1

Remarque:
Chez Linux, il est possible de rendre aléatoire la distribution des ip par le round robin. Il est activé par défaut (mots clés : round robin first (uniquement le premier trouvé) ou random).

DYNAMIC HOST CONFIGURATION PROTOCOL

(DHCP)

Module 10 (page 6)

Packages:
dhcp-common
dhcp-server
dhcp-client
dhcp-relay

Ports:
TCP/UDP 67:Serveur
68: Client

Fichiers
/etc/dhcpd.conf => n'existe pas par défaut
/etc/dhcpd.conf.sample => Exemple de fichier de configuration
/var/lib/dhcpd/dhcpd.leases => Liste des baux ip en cours attribuées aux clients
/var/lib/dhcpd/dhclient.leases => Liste des baux ip obtenu par le client.
/etc/dhcpclient.conf => Configuration du client

Commandes:
#service dhcpd start| stop | restart..
dhcpd -t => vérifier la syntaxe du fichier de configuration
dhcpd -T => Vérifier le fichier des baux actifs
#dhclient => Commande client pour obtenir un bail
#dhcrelay => Commande de démarrage des relais DHCP
#dhcpreport.pl => Visualisation "conviviale" des baux
#dhcping => test le serveur DHCP par l'envoi d'un DHCPREQUEST suivi d'un DHCPRELEASE

Aide:
#man dhcpd.conf
#man dhcp-options
#man dhclient.conf
/usr/share/doc/dhcp-common-30../doc/dhcp-dynamic-dns-exemples

Fonctionnement:

DHCP DISCOVER (cherche un DHCP)
---------->
DHCP OFFER (Offre des serveurs DHCP) ip, mask, TTL minimum (wins, nom domaine...)
<----------
CLIENT DHCP REQUEST (choisi un bail) Linux peut choisir la qualité du bail (options) **SERVEUR**
---------->
DHCP ACK (donne le contrat)
<----------

Tout en broadcast car pas d'IP
DHCP REQUEST et DHCP ACK sont en unicast ethernet mais broadcast IP

IMPORTANT: Le contrat accepté ne peut être annulé par le serveur.

Mise en place d'une configuration minimum:
Dans le fichier /etc/dhcpd.conf
ddns-update-style none; => méthode de MAJ dynamique du DNS par le DHCP
subnet adresse_reseau netmask mask_reseau {
rang ip_debut_plage ip_fin_plage; => **ATTENTION**:le serveur doit posséder une route vers ce réseau
default-lease-time *valeur_en_seconde;* => Durée du bail avant reconduction tacite
max-lease-time *valeur_en_seconde;* =>Durée maxi du bail avant expiration forcée
} **=> pas de ; comme DNS**

Options de configuration fournies aux clients:

option router *IP_passerelle;*
option *domain_name_server IP_server_dns,ip_server_dns*;
option domain-name "*nom_domaine_dns*";
option static-routes *adresse_reseau_a_joindre ip_passerelle*;

Remarque:
on peut utiliser les numéros d'option plutôt que les noms
exemple:
option option-03 *IP_passerelle;*

Filtrage des clients inconnus:
Proposer le service DHCP seulement pour les clients identifiés (par leur adresse MAC)
=> /etc/dhcpd.conf
deny unknown-clients;
ATTENTION: Obligation de répertorier tous les clients

Le nom d'hôte à fournir sert au serveur pour identifier le client dans les logs et la liste des contrats
host *nom_d_hôte* {
hardware-ethernet *adresse_mac* => (format d'adresse MAC XX:YY: ...)
}
ATTENTION: Si cette section est dans la section subnet, les options du subnet lui sont ajoutées. Si elle est en dehors d'une section subnet, il faut lui rajouter les options pour lui et surtout, lui ajouter une ip fixe.
=> options d'étendu ou options clientes

Exemple:
host *nom_d_hôte* {
hardware-ethernet *adresse_mac* (format d'adresse MAC XX:YY: ...)
fixed-adress *ip_fixe*;
}

MISE À JOUR DYNAMIQUE DES ENREGISTREMENTS CLIENTS DANS LES SERVEURS DNS PAR LE DHCP:

Cas n°1:
Après obtention d'un bail, le client s'enregistre lui même. Le bail doit avoir au mois le nom du domaine dns, l'adresse d'au moins un des serveurs DNS qui gère ce domaine

Coté DHCP:
Le bail doit avoir au mois le nom du domaine dns, l'adresse d'au moins un des serveurs DNS qui gère ce domaine.

Côté DNS:
dans .etc/named.conf
zone " " {
allow-update { ips des clients autorisés a mettre a jour leurs enregistrements };
};

Remarque : aucune communications entre le serveur DNS et le serveur DHCP

Cas n°2:
Après obtention d'un bail, le serveur DHCP 'enregistre le nom et adresse ip du client dans le DNS. Le bail doit avoir au mois le nom du domaine DNS, l'adresse d'au moins un des serveurs DNS qui gère ce domaine

Coté DHCP:
Le serveur demande au DNS l'enregistrement du client.
Le bail doit avoir au mois le nom du domaine DNS, l'adresse d'au moins un des serveurs DNS qui gère ce domaine.

Dans le dhcpd.conf

 # Mise à du serveur DNS par le DHCP activée (**attention au UPDATES au pluriel**):
ddns-update-style interim;
ddns-updates on;

 # Nom du domaine à mettre à jour:
ddns-domainname "morbihan.com";

 # Mise à jour pour les clients dont l'IP est fixe:
update-static-leases on;

 # Refus des clients dont on ne connait pas l'adresse MAC:

```
deny unknown-clients;

subnet 192.168.7.0 netmask 255.255.255.0 {
        range 192.168.7.100 192.168.7.103;

# Liste des paramètres passés au client:
default-lease-time 300;
max-lease-time 450;
option domain-name "morbihan.com";
option domain-name-servers 192.168.7.200;
option routers 192.168.7.254;

# Déclaration pour chaque client en fonction de son adresse MAC:
host 2000Pro {
        hardware ethernet 00:0c:29:e1:73:d9;
        #ddns-hostname "2000Pro";
}
}

# Indication de la zone à mettre à jour et de son serveur DNS Master:
zone morbihan.com {
        primary 192.168.7.200;
}
```

Côté DNS:

le serveur DNS accepte ou refuse l'enregistrement du client par le serveur DNS
dans .etc/named.conf
zone "*nom_zone*" {
allow-update { ips des clients autorisés à mettre à jour leurs enregistrements };
};

LE RELAIS DHCP

Il faut créer un scripte de démarrage pour le démarrer automatiquement!!!

Joue le rôle d'un serveur DHCP pour les clients DHCP dans un réseau sans serveur DHCP séparé par un routeur des autres sous réseaux avec serveur DHCP.
Il reçoit et retransmet en unicast les requêtes des clients vers le serveur DHCP indiqué derrière le routeur.

Fail over DHCP

Permet à deux serveurs DHCP de partager une même plage ip et de fournir une tolérance de panne complète.

Serveur Primaire de DHCP

dhcpd.conf:
Ce sont les déclarations particulières pour permettre le failover. Quand je dis "déclarations particulière" cela signifie les détails pour le serveur primaire dhcp comme l'Ip, la mise en communication, et quelques autres options. Dans ce cas-ci, ces options sont seulement pour le serveur primaire dhcp. Mettez le texte suivant dans votre dhcpd.conf sur le serveur primaire dhcp:

```
ddns-update-style none;
one-lease-per-client true;
option domain-name "mydomain.com";
option domain-name-servers 10.254.0.3, 10.254.0.4;
option subnet-mask 255.255.0.0;
default-lease-time 300;
max-lease-time 300;
authoritative;
failover peer "dhcp" {
primary;
 address 10.254.0.9;
port 519;
peer address 10.254.0.8;
peer port 520;
```

```
max-response-delay 60;
max-unacked-updates 10;
mclt 600;
split 128;
load balance max seconds 3;
}
 include "/etc/dhcpd.master";
```

Serveur DHCP Secondaire:

dhcpd.conf

Vous noterez qu'il y a quelques options manquantes à la déclaration de failover. C'est parce que le serveur secondaire dhcp n'a pas besoin de ces options. Mettez maintenant ce texte dans votre dhcpd.conf sur votre serveur dhcp secondaire

```
ddns-update-style none;
one-lease-per-client true;
option domain-name "mydomain.com";
option domain-name-servers 10.254.0.3, 10.254.0.4;
option subnet-mask 255.255.0.0;
default-lease-time 300; max-lease-time 300;
authoritative;
failover peer "dhcp" {
secondary;
address 10.254.0.8;
port 520;
peer address 10.254.0.9;
peer port 519;
max-response-delay 60;
max-unacked-updates 10;
}
include "/etc/dhcpd.master";
```

Maîtrisez le dossier de configuration de DHCP

dhcpd.master

Ceci devrait être sur les deux serveurs. Ce dossier contient toutes les déclarations de sous-réseaux que vous définissez. Mettez celui-ci dans /etc/dhcpd.master:

```
subnet 10.254.0.0 netmask 255.255.0.0 {
pool {
failover peer "dhcp";
range 10.254.0.10 10.254.255.254;
deny dynamic bootp clients;
}
option routers 10.254.0.1; }
```

Script de démarrage:

Mettez ceci dans /etc/init.d/dhcp:

```
#!/bin/bash
DAEMON=/usr/sbin/dhcpd
CONF=/etc/dhcpd.conf
NAME=DHCP
PIDFILE=/var/run/dhcpd.pid
IFDEV=eth0
DHCPOPTS="-q $IFDEV"

[ -x $DAEMON ] || exit 0
[ -f $CONF ] || exit 0

 # Safety check
if [ ! -f /var/state/dhcp/dhcpd.leases ]; then
touch /var/state/dhcp/dhcpd.leases fi
fi
case "$1" in
start)
echo -n "Starting $NAME Server: "
start-stop-daemon --start --quiet --pidfile $PIDFILE --exec $DAEMON -- $DHCPOPTS
echo "done"
;;
stop)
echo -n "Stopping $NAME Server: "
start-stop-daemon --stop --quiet --pidfile $PIDFILE --exec $DAEMON -- -$DHCPOPTS
echo "done"
```

```
;;
restart)
$0 stop
sleep 3
$0 start
;;
*)
echo "usage: $0 start|stop|restart"
exit 1
;;
esac
```

Création des clés de mise à jour sécurisé entre dns et dhcp

Dans le cas d'une protection de zone, les deux clés sont identiques (clés symétriques)

1. Création de la clé

#dnssec-keygen -a *hmac*-md5 -b *512* -n *toto.fr*

-a => algorithme choisi : ici md5
-b => taille de la clé en bit (1 à 512)
-n => nom du domaine

Génère 2 fichiers:
Knom_clé+X.private => ou X est un nombre
Knom_clé+X.key => identique et aléatoire

2. Déclaration de la clé dans /etc/named.conf

```
Key "nom_de_clé" {
        algorythm hmac-md5;
        secret"copie_de_la_clé_crée";
        };
```

3. Protection de la zone choisie avec cette clé

```
zone "nom_zone_dns" {
        allow-update {
        key nom_clé;
        };
        }
```

4. Déclaration de cette clé dans le /etc/dhcpd.conf

```
Key nom_clé {
        algorythm hmac-md5;
        secret "copie_de_la_clé_crée";
        }
```

LE SERVICE NFS (NETWORK FILE SYSTEM)

But:
Partager des répertoires en réseau entre les postes Unix/Linux. Il prend tout sont intérêt en cas d'utilisation avec un domaine NIS (Network Information Service)

Ports:
TCP/UDP 111, 2049

Fichiers de configuration:
Côté serveur:
/etc/exports => liste de configuration des partages

Côté client (facultatif):
/etc/fstab => Montages permanents et automatiques au démarrage

Commandes:

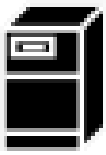
Côté serveur:

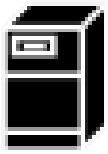
#service nfs status | start..
#exportfs => Active/ Désactive les partages indiqués dans exports
#showmount => équivalent à mount pour les montages nfs seulement
#exportfs -r => Relit le contenu de /etc/exports

Côté client (facultatif):

#showmount => équivalent à mount pour les montages nfs seulement
#mount => monter un point de partage.

SERVEUR NFS **CLIENT NFS**





/var/data/file1 <-----mount -t nfs options ---------- /mnt/import/file1
/file2 (indique le répertoire import) file2
/etc/exports /var/data options

Création d'un partage dans /etc/exports

Côté serveur:

1. **Choisir une ressource à partager.**

2. **Renseigner ce partage dans le fichier /etc/exports**

champs1 champs2 (champs3)
champs1 => Répertoire à partager
champs2 => IP.autorisée, ip.réseau, nom_hôte, nom_domaine, *(tt monde)
champs3 => liste des options (option1, option2, option3,)...
le champs3 indique les permissions de partage (Si pas options Lecture seule en nfs)

3. **Activer le partage**

#exportfs -r => Relit le contenu de /etc/exports
ATTENTION:
on ne redémarre pas le serveur (sources de problèmes)

4. **Vérifier le montage**

#showmount -e

Côté client:

1. **Créer/choisir un point de montage**

2. **Choisir un partage disponible**
#showmount -e nom_ou_ip_serveur

3. **Effectuer le montage**
#mount -t nfs nom_ou_ip_server:nom_rep_partagé point_de_montage_local
ex:
#mount -t nfs thierry:/var/partage /mnt/data

4. **Vérifier**
#s point_de_montage

Structure des enregistrements dans /etc/exports:

champs1 champs2 (champs3)
champs1 => Répertoire à partager
champs2 => IP.autorisée, ip.réseau, nom_hôte, nom_domaine, *(tt monde)
champs3 => liste des options (option1, option2, option3,)...
le champs3 indique les permissions de partage (Si pas options Lecture seule en nfs)

options de partage du champs3:
ro ou rw => read only ou read write
sync | no sync => Synchronisation (par défaut) ou non : des opération de lecture écriture sont écrites en temps réel ou après transfert complet.
anonuid=uid => ou udi est l'uid de l'utilisateur anonyme (par défaut 65534 nobody)
anongid=gid => ou gdi est l'gid du groupe anonyme (par défaut 65534 nogroup)
root_squash => Le root client perd ses privilèges (vu en tant l'anonyme) sur le répertoire nfs (root a rien par défaut)
no_root_squash => le client root garde ses privilèges
all_squash => tous les utilisateurs sont transformés en anonymes
squash_uid=uid => ou uid-uid (ensemble d'uid) qui seront squashés
squash_gid=gid => ou gid-gid (ensemble de gid) qui seront squashés

ATTENTION:
il n'est pas possible de préciser plusieurs listes séparées par des ","

Exercice:
Partager un répertoire en lecture-écriture de telle sorte que toutes les opérations effectuées le soient en tant que compta1 du groupe compta (uid=601 gid=600)

```
#groupadd  -g 600 compta
#useradd -u 601 -g 600 compta1
#mkdir -m 775 /var/test
#chown compta1:compta /var/test
#ls -ld
#vi /etc/test
        /var/test *(rw,anonuid=601,anongid=600,all_squash)
#exportfs -r
```

Options avancées de montage coté client:
#mount -t nfs -o option1, option2
ro => Montage en lecture seule (réduit les permissions si RW sur le serveur)
rw => Montage en lecture écriture (par défaut) => plus restrictif gagne entre client et serveur
hard => Tentative de montage illimité
intr => (interruption) Possibilité d'interrompre les tentatives de montage (indispensable avec "hard" si connections impossible)
soft X => tentatives de montage jusqu'à expiration d'un certain nombre de connections (avec timout)
timeo=x => délais x en seconde avant de retenter un montage
retrans=x => Nombre de réessaie avant de déclencher un message d'erreur.
Rsize=1024 => taille des packets en lecture en octets (1024 2048 4096 8192) wsize=1024=> taille des packets en lecture en octets (1024 2048 4096 8192)

Tests de performance:

Côté serveur:
1. Partage d'une ressource
2. création de deux fichiers de 20Mo dans /var/perf

#dd if=/dev/zero of=/var/perf/20Mo.1k bs=1k count=20000
#dd if=/dev/zero of=/var/perf/20Mo.4k bs=4k count=5000

Côté client pour 1024:
1. Montage de /var/perf avec rsize et wsize à 1024
2. Lecture des fichiers de 20Mo

#time cat /mnt/import/20Mo.1k
#time cat /mnt/import/20Mo.4k
3. Création d'un fichier de 20Mo

#time dd if=/dev/zero of=/mnt/impot/20Mo.1024.client bs=1k count=20000
#time dd if=/dev/zero of=/mnt/impot/20Mo.1024.client bs=4k count=5000

Dépanner un service nfs:
Purger des fichiers de cache qui gardent en mémoire des infos sur le passé de l'utilisation de NFS:
supprimer le contenu de ces fichiers
/var/lib/nfs/etab => Liste des partages déjà effectués
/rmtab => liste des montages déjà effectués
/xtab => Liste des partages actifs

ATTENTION:
ne pas les supprimer.

Montages NFS permanents
dans le fichier /etc/fstab
nom_server:rep_partagé point_de_montage nfs liste_de_options 0 0

ATTENTION:
pas de contrôle pour voir si le réseau est disponible

SAMBA

Historique :
1980 : IBM =>Netbios (Network Basic Input Output Système) implémenté plus tard dans les protocoles autres de netbeui tel que TCP/IP (netbios over TCP ou NBT), DECNET IPX/SPX
1985 : IBM =>Netbeui (Netbios Enhanced User Interface) Protocole non routable.

NBT: Permet:
- Voisinage Réseau (Browsing)
- Résolution de nom Netbios (NBNS Netbios name système =>WINS chez Microsoft)
- Authentification centralisée (Domaine)

Depuis Windows 2000 le protocole netbios utilise directement TCP/IP sur le port 445

1993:nbserv 1.5
1994: SAMBA
07/94:1ere version assez stable
95: Apparition dans les premier serveurs Linux ou free BSD
99 : Samba 2.0

Fonctionnalités samba:
- Partage de fichiers (arborescence)
- Partage d'imprimantes Connectées sur le serveur ou les clients)
- Parcours du voisinage réseau
- Services d'authentification des clients Windows et la gestion des autorisations et droits
- Joue le rôle d'un PDC NT pas CSD si le PDC est un Windows.
- Serveur DFS
- Traduction entre les ACL (Access Control List) et les permissions UNIX
- Résolution de nom Netbios (serveur NBNS)
- Le Démon *winbindd* permet l'authentification sur un système Unix des utilisateurs dont les informations sont stockées dans une base SAM (Domaine NT4)
- Extensions CIFS UNIX (HP) =>partage entre système UNIX
- Outils clients UNIX

Commandes SAMBA:

#nmblookup	=> interrogation d'un serveur de nom NETBIOS
#smbclient	=> Programme interactif multifonction
#smbcontrol	=> interrogation simple auprès des démons
#smbmount	=> Montage d'une ressource SMB (uniquement UNIX)
#smbpasswd	=> Gestion des mots de passe des utilisateurs samba.
#testparm	=> test le format du fichier de configuration smb.conf
#testprns	=> Vérification des informations des imprimantes
#smbstatus	=> état des connexions
#smbspool	=> Gestion des imprimantes
#smbtar	=> utilitaire de sauvegarde
#wbinfo	=> interrogation du démon winbindd

obsolète avec la version 3:

make_smbcodepage	=> gestion de l'internationalisation
make_unicodemap	=> gestion de l'internationalisation

Service SWAT: Samba Web Administration Tool
HTTP://localhost:901

Le Service SAMBA est composé de trois sous-services:
- smbd : Gère l'accès aux ressources
- nmbd : Gère les noms sur le réseau (Voisinage réseau) Génère un processus fils par connection active Une seule occurrence du démon nmbd (sauf en cas de serveur WINS)
- winbindd : permet l'authentification sur un système Unix des utilisateurs dont les informations sont stockées dans une base SAM (Domaine NT4)

Les fichiers:
/etc/samba/smb.conf => Fichier de configuration
/etc/samba/smbpasswd => Fichier de configuration pour les clients

ATTENTION:
Un utilisateur Samba est un compte autorisé à accéder aux ressources partagées par samba.
Tous les utilisateurs samba doivent avoir aussi un compte linux (/etc/passwd)
Si
valid user => n'est pas indiqué, tous les utilisateurs samba ont accès.
Public = yes|no => autorise ou non tout le monde a utiliser le partage (compte ou pas) comme tout le monde de Microsoft

Création d'utilisateur samba:
#groupadd -g 2000 windows
#useradd -u 2001 -g 2000 samba1
#smbpasswd -a samba1 =>crée un compte samba1 à partir du compte linux

ATTENTION: le password ne sera pas enregistré si le compte linux associé n'est pas trouvé (-a)

le fichier smbpasswd contient :
user:501:password NTLM:password NTLM:flag:last change
user => nom du compte
501 => UID
LM: => Lan Manager pour 95 98 ME
NTLM : => NT Lan Manager pour nt/2000/XP
Flag: => U:utilisateur,
D:utilisateur désactivé
N:utilisateur sans password
w:compte d'ordinateur
Last change: => Date de dernière modification en seconde depuis le 1er janvier 1970

Création automatique des users SAMBA:
pour faire automatiquement en sorte que les utilisateurs linux (passwd) soient crées automatiquement en tant que user samba (smbpasswd)

#cat /etc/passwd | mksmbpasswd.sh > /usr/local/samba/private/smbpasswd

```
#!/bin/sh
awk 'BEGIN {FS=":"
        printf("#\n# SMB password file.\n#\n")
        }
{ printf( "%s:%s:XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX:XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX:[UD
]:LCT-00000000:%s\n", $1, $3, $5) }
```

ATTENTION : ce scripte ne trie rien. Il récupère tous les comptes (même système).
Il permet de ne pas taper le -a mais ne remplace pas le smbpasswd pour mettre un password et activer le compte (flag UD)

Suppression de samba 3.01a puis installation de samba 3.014a
#rpm –qa | grep samba
#rpm -e PACK
puis décompression du package tar
#tar -xvzf samba-3.0.14a.tar.gz => Décompresse le pack
Dans le répertoire /samba-3.0.14a/sources
#./configure => Compile les sources (Crée un fichier makefile correspondant au matériel de la machine)

Les options suivantes peuvent être utilisées:

```
--with-acl-support
--with-automount
--with-configdir=dir      =>emplacement du smb.conf
--with-logfilebasedir
--with-msdfs              => Support du FS DFS
--with-winbind            => Support du demon winbind (yes par défaut)
--with-smbmount
--with-privatedir=dir     => Emplacement des fichiers d'authentification

#make                     => Compile les sources dans le répertoire de travail
#make install             => Installe le produit
```

le fichier /source/config.status permet de voir les options choisies avant compilation
avant d'effectuer une recompilation il est préférable de nettoyer les sources:

```
#make clean
#rm config.cache
```

On doit créer un fichier /usr/local/samba/lib/smb.conf

```
---------------------------------------
#fichier de configuration samba
[global]
        workgroup = mdkgroup
[test]
        path = /mnt/test
        comment = Répertoire de test
----------------------------------------
```

ensuite on lance les démons:

```
# /usr/local/samba/sbin/smbd -D
# /usr/local/samba/sbin/nmbd
# /usr/local/samba/bin/smbstatus
```

Pour lancer le client de n'importe ou, on modifie le .bashrc et on ajoute /usr/local/samba/bin dans le path.

Pour lancer le démon automatiquement au démarrage:
copier le fichier /samba-3.0.14a/packaging/Mandrake/smb.init vers /etc/init.d
dans ce fichier, en ligne 23, remplacer le chemin du fichier smb.conf par celui défini plus haut:
usr/local/samba/lib/smb.conf

on utilise ensuite chkconfig pour le lancer automatiquement au démarrage en niveau 345 et l'arrêter automatiquement au niveau 16

```
#chkconfig 16 smb.init off
#chkconfig 345 smb.init on
```

Ces commandes crées les liens de fichier dans les bonnes sections avec par défaut un ordre de priorité de S91 et K09 (Voir chkconfig.)

Mise en place de SWAT:

- ll /etc/xinetd.d | netbios-ns => vérifier si ce fichier existe.

Si il est absent, le créer:

- #vi /etc/xinetd.d/netbios-ns

```
-----------------------------------------------------------------
service swat
{
        port            = 901
        socket_type     = stream
        wait            = no
        user            = root
        server          = /usr/local/samba/sbin/swat
        log_on_failure  += USERID
        only_from       = localhost
        disable         = no
}
#
-----------------------------------------------------------------
```

- **redémarrer xinetd.d**

```
#/etc/init.d/xinetd restart
```

- **accéder à swat:**
  http://localhost:901

Syntaxe du fichier smb.conf:

Les modifications du fichier sont détectées automatiquement après maximum 1 minute
les valeurs sont des valeurs de chaîne de caractère, décimales, ou booléennes.

```
[global]
global
option=valeur                   => Avec ou sans espaces
option='valeur1 valeur2'        => Espace entre liste de valeur, peuvent être entre ' (pas obligatoire)
# commentaire                   => Commentaire
; option = valeur3               => Commentaire des valeurs (convention)
option= ValEUr4                 => Casse y est préservée
```

Les variables SAMBA disponibles:

```
%a          => Architecture du client samba (win95 pour 95 98 me, win2k pour 2000 et XP ou unknown)
%I          => Adresse IP du client
%M          => Nom DNS client
%m          => Nom netbios du client
%u          => Identité de l'utilisateur pour le partage concerné
%g          => Groupe principale de l'utilisateur %u
%U          => Identité souhaitée par l'utilisateur
%G          => Groupe principale de l'utilisateur
%H          => Répertoire de connections de l'utilisateur %U
%S          => Nom du partage
%d          => PID du processus courant
%h          => Nom du serveur DNS SAMBA
%L          => Nom netbios du serveur SAMBA
%v          => Version SAMBA
%T          => Date et Heure système
%$var       => Valeur de la variable var
@group      => nom du groupe
```

root preexec et root postexec:

Permet d'exécuter des commendes seulement lors de l'accès ou de la déconnection à une ressource

Exemple:

Mettre à disposition de chaque poste windows du réseau, un partage contenant un répertoire au nom du groupe de l'utilisateur, et dans ce répertoire, un fichier au nom de l'utilisateur

Etape 1:

Ajouter le partage suivant dans smb.conf

```
[%m]
path = /var/%m
root preexec = [ -d /var/%m ] || mkdir /var/%m;[ -d /var/%m/%G ] || mkdir /var/%m/%G;chgrp %G /var/%m/%G;echo "une connection supplémentaire... à `date +%T` >> /var/%m/%U
```

```
        => -d : vérifie si le répertoire existe déjà
||      => OU logique si cmd1 ok alors cmd2
```

Dans [GLOBAL]

Option	***description***	***default***
Config file	Utiliser plusieurs fichiers smb.conf (ex: un par groupe) config file = /usr/local/samba/lib/smb.conf.%g cherche un fichier smb.conf.*groupe_principale* Si le fichier n'est pas trouvé la ligne est ignorée Si il le trouve, il ouvre un processus fils et charge uniquement le .conf.group (pas le smb.conf)	
include	identique à config file mais les options sont dans le fichier smb.conf	
copy	permet de recopier une section dans une autre **Attention:** la section à copier **DOIT** être écrite en amont ex: [proto1] guest ok = yes guest only = yes [toto] path = /var/pub copy = proto => Copie de la section proto	
	indique si les imprimantes partagées sont disponibles	yes
default service = pub	redirige vers la section indiquée si la connections échoue	
name resolve order	Définit l'ordre de recherche de résolution de nom défaut: lmhosts host wins bcast	
wins support	joue le rôle d'un serveur wins	no
wins server	désigne un serveur wins	no
wins proxy		no
default service ou default	nom d'un partage par défaut	

Dans une section de partage:
Options liées au Browsing (voisinage réseau)
Samba prend par défaut une valeur d'élection de 20 (pdc nt 2000 => 32, nt/2000 no pdc=>16, 9x=>1)

Exemple de fichier smb.conf:

```
---------------------------------------------------------------------------------------------------------------
# Samba config file created using SWAT
# from 127.0.0.1 (127.0.0.1)
# Date: 2005/08/16 16:31:08

# Global parameters
[global]
        workgroup = MDKGROUP
        # Nom de workgroup
        netbios name = thierry
        # Nom netbios du serveur SAMBA
        server string = Serveur samba de Thierry sur %L
        # Commentaire sur le serveur SAMBA
        local master = yes
        # Participe en tant que maître explorateur
        domain master = no
        # Maître explorateur de son domaine (si contrôleur de domaine)
        preferred master = no
        # Provoque une élection pour devenir maître d'opération
        os level = 50
        # Niveau pour l'élection (suppérieur à 35 pour être prioritaire sur les windows Microsoft
        domain logons = no
        # Mettre à yes si le serveur est PDC
        security = user
        # Mode d'authentification (share, user, domain, server, ads)
        load printers = yes
        # Indique si les imprimantes partagées sont disponibles
        printcap name = /etc/printcap
        # Fichier de configuration des imprimantes
        printing = cups
```

```
        # Système d'impression utilisé
        username map = /etc/samba/usermap.txt
        # Fichier de correspondance de nom (attention à la casse et la longueur des logins
        unix password sync = yes
        # Synchronise les password samba vers unix (pas inverse)
        passwd program = /usr/bin/passwd %u
        # Donne le chemin de passwd
        passwd chat = *nouveau*password* %n\n *retapez*le*password* %n\n *Réentrez*le*nouveau*password* %n\n
*Password*change*
        # Indique les données à fournir pou changer le password
        socket option = TCP_NODELAY           SO_KEEPALIVE
        # rendre illimité la connection TCP
        keepalive = 300
        #intervalle en seconde entre chaque vérification de la présence du client
        deadtime = 10
        # nombre de minutes d'inactivités avant déconnections (0 par défaut)
        interfaces = all
        #indique les interfaces utilisés par samba (la première seulement est utilisée par défaut)
        bind interfaces only = yes
        # Utilise uniquement les interfaces dans l'option interface
        log file = /usr/local/samba/var/log.smbd.%m%u
        # Indique le nom et emplacement du journal SAMBA
        log level =
        # synonyme «debug»  niveau de détail (0 par défaut) de 0 à 10(normal:0-1)
        max log size = 2000
        # Taille en kilo octet du fichier de log. (0 = illimité)
        # Renommé en .old si arrive à la taille maximum. Est écrasé si existe déjà


[HOMES]
      comment = partage du répertoire de base de %u

[profiles]
        path = /home/profile/%u
        browsable = no
        writable = yes
        create mode = 0600
        # Affecte les permissions à 600 pour les fichiers
        directory mode = 0700
        # Affecte les permissions à 700 pour les Dossiers

[test]
        comment = Répertoire de test
        # commentaire visible dans le voisinage réseau
        browsable = yes
        # Indique si le partage est visible sous le voisinage réseau (yes par défaut)=browseable
        path = /mnt/test
        # Path est le chemin absolu du répertoire partagé et peut utiliser la variable %u
        guest ok = no
        # synonyme de 'public': Accès anonyme au partage (yes/no)
        guest account = nobody
        # nécessite 'guest ok = yes' Nom d'un compte invité
        guest only = no
        # nécessite 'guest ok = yes': uniquement les anonymes (synonyme: only guest)
        writable = yes
        # écriture possible (synonyme: writeable ou write ok)
        read only = no
        # lecture seule
        encrypt passwords = yes
        # Permet la gestion des passwords cryptés Microsoft (ntlm et Kerberos)
        veto files =  /*Security*/*.tmp/*root*/;/*.mp3;/*.MP3
        # n'affiche ni n'autorise l'accès aux fichiers définis (Attention à la casse)
        delete veto files = no
        # Supprime les fichiers mentionnés (pratique pour interdire en écriture les .mp3 par ex)
        hide files = /*.bak/*.old/
        # cache les fichiers mais ils sont accessibles
        hosts allow = 192.168.10. EXCEPT 192.168.10.19
```

```
        # Machines  autorisés avec exceptions
        valid users = bruno,thierry,@compta
        # Indique les utilisateurs et groupes (@) autorisés
        invalid users = toto14;@aftec
        # comptes et groupes non autorisés
        admin users = thierry
        # peuvent tout faire sur le partage comme root
        locking = yes
        # permet de verrouiller un fichier ouvert en écriture mais n'empêche pas la lecture
        hide dot files = yes
        # cache les fichiers commencant par un point (yes par défaut)
        dont descend = /etc/tot/go
        # Bloque l'accès aux sous répertoires définis
        follow symlinks = yes
        # Autorise les liens symboliques (yes par défaut)
        wide links = yes
        # Autorise ou pas les liens en dehors de l'arborescence (yes par défaut) avec répertoires symlinks
        force user = root
        # l'utilisateur se connecte avec son compte mais les fichiers qu'il créé  appartiennent à root
        force group = admin
        # idem que force user mais pour le groupe propriétaire
        read list = titi
        #  acces que en lecture pour titi (ATTENTION: prend le pas sur les permissions samba et UNIX
        write list = tutu
        # prend le pas sur les permissions SAMBA mais pas les permissions UNIX
```

Modes de Sécurité:

share	=> Password sur les partages. Lui seul est nécessaire pour accéder au partage
user (défaut)	=> géré par ce serveur et dépend du compte utilisateur et groupe
domain	=> Le serveur SAMBA est dans un domaine NT, l'authentification est gérée par un pdc.
server	=> essai d'authentifier l'utilisateur auprès d'un NT si pas possible il bascule en mode user
ads	=> (>samba3) Doit être membre de Active Directory et avoir Kerberos

Comment créer un serveur virtuel (un serveur => plusieurs noms netbios

```
netbios name = ifc1
netbios aliases = compta,prod
server string = samba sur %L
include = /usr/local/samba/lib/smb.conf.%L    => Ce %L exploite le nom netbios appelé par le client
```

DFS:

Fournis avec la version 3 et à lancer avec –with_msdfs sur la version 2.2 de SAMBA

```
[global]
        host msdfs = yes

[data]
        path = /usr/local/samba/dfs
        # le répertoire doit appartenir à root (user) et avoir les droits 755
        msdfs root = yes
```

Dans le répertoire racine DFS on crée des liens DFS (= liens symboliques)

```
#ln -s `msdfs:nom_serveur\partage` nom_lien_dfs
```

ATTENTION: le lien symbolique ainsi crée sera perçu comme défectueux par unix mais fonctionnera pour dfs

Samba comme PDC d'un domaine NT4

- **Rôles à assumer**
  - gérer les postes clients (SID ordinateur)
    => Jonction des machines au domaine
  - gérer l'exploration réseau (domain master, serveur wins, ...)
  - gérer la base de comptes utilisateurs
    => Profils, scripts de logon, stratégies systèmes ...

```
workgroup = nom_workgroup
domain logons = yes
security = user
local master = yes              |
domain master = yes             |
preferred master = yes          | Dépend de la topologie du réseau
os level = 33                   |
wins support = yes              |
```

1. **Il faut valider la ligne 245 permettant de faire un compte linux pour la machine :**

```
add machine script = /usr/sbin/useradd -d /dev/null -g machines -c 'Ma      chine Account' -s /bin/false -M '%u'
-d      /dev/null       => pas de répertoire d'accueil
-g      Machine         => groupe machine
-s      /bin/false      => pas de shell
-M                      => pas de répertoire par défaut
%u                      => nom netbios de la machine client
```

2. **il faut aussi activer un partage spécial appelé [netlogon] (ligne 364) emplacement des scripts et des fichiers config.pol et ntconfig.pol**

```
path = /var/lib/samba/netlogon
guest ok = yes
writable = no
```

3. **Il faut aussi ajouter un compte samba pour 'root'**

```
#smbpasswd -a root
```

Les profils errants:

```
logon path = \\%L\Profiles\%u   => indique le répertoire de stockage des profils utilisateurs
browsable = no
writable = yes
create mode = 0600              => Affecte les permissions à 600 pour les fichiers
directory mode = 0700           => Affecte les permissions à 700 pour les Dossiers
```

REMARQUE:dans ce répertoire : renommer ntuser.dat en ntuser.man pour le rendre obligatoire

la gestion des scriptes de logon:

```
logon script = %m.bat           => indique le nom du scripte dans le partage netlogon
```

Jonction d'un serveur SAMBA dans un domaine Microsoft

1. **modifier le smb.conf**

```
workgroup = domcrosoft                          => nom netbios du domaine
domain logons = no                              => pas DC
security = domain                               => Sécurité assumée par un DC
password server = 192.168.10.254                => Gestionnaire de compte (mettre le nom dns)
add user script = /usr/sbin/useradd -s /bin/false %u    => Ajouter un utilisateur linux si autorisé par le DC
```

2. **lancer une commande pour joindre le serveur SAMBA au domaine:**

```
#net join -S nom_netbios_dc -w nom_netbios_domaine -U compte_admin_jonction_domaine
```

Gestion Cups :

URI parallel :/dev/lp1
liste des périphériques pris en charge:
lpinfo -v
/etc/init.d/cups
chkconfig --list
--level cups
--add

PPD (postscript printer definition)

lien de drivers imprimantes :
http://www.linuxprinting.org/

/usr/bin/lpadmin -p HP(-E) -V parallel:/dev/lp1
-m laserjet.ppd

options:
lpadmin -p imprimante -c classe=> ajoute à la classe d'imprimante (si une occupé la 2eme)
-m => le fichier ppd
-r classe => retirer de la classe
-E => active l'imprimante

http://localhost:631/admin
/etc/init.d/cups restart

supprimer une imprimante lpadmin -x printer
imprimante par défaut lpadmin -d printer
/usr/bin/disable imprimante
/usr/bin/enable imprimante
/usr/sbin/accept imprimante
/usr/sbin/reject imprimante
lpadmin -p imprimante -c classe ajoute à la classe

exemple:
connection de l'imprimante du poste de SEB
lpadmin -p laserjet -E -v ipp://ifc6/printers/HPlaserjet
lpadmin -p laserjet -E -v ipp://192.168.10.6/printers/HPlaserjet

Imprimante partagée sur le poste seb
ajout à cups imprimante-partagée-seb, donc on repartage celle de seb
mettre ; load printers dans global smb.conf
load printers = yes

IMPRIMER SOUS LINUX

Paquetage:
cupsys => serveur
printconf => Fichiers admin et client

Configurer l'imprimante avec l'une des méthodes suivantes:
1. par http://localhost:631 site web embarqué (pas apache)
2. par /etc/cups/cupsd.conf
3. par /etc/cups/printers.conf
4. par lpadmin
5. par webmin: http://localhost:100000

Common Unix Printing System (CUPS)
- les + :
- Fichier de configuration similaire à apache
- Fichier de configuration largement commenté
- Docs sur le site www.easysw.com
- interface d'administration graphique
- émulation des commandes des anciens systèmes d'impression (lpstat, lpadmin, lpq,...)

- les - :
- Rôle client/serveur mail défini par défaut (difficile de déterminer si l'imprimante est locale ou distante)
- Par défaut tout le monde voit tout le monde (directive browsing à enabled)

cupsd.conf
Ligne 143: Fileterie Yves : indique pour une imprimante, un fichier de sortie (périphérique)

printers.conf contient les imprimantes ajoutées par interface web

La ligne de commande

1. Rechercher le pilote
lpinfo -m | grep -i nom_constructeur

2. ajouter l'imprimante
#lpadmin -p nom_file_attente -v nom_interface
nom_interface => file:/dev/lp0 (nécessite FileDevice Yes dans /etc/cups/cupsd.conf
ipp://nom_serveur/printers/nom_file_attente_sur_le _serveur

3. Démarrer l'imprimante (activer la file d'attente)
#/usr/bin/enable nom_file_attente (car enable est une commande interne et celle ci est différente)

4. Accepter les travaux d'impression
#accept nom_file_attente

APACHE

Manuel d'aide: file:/usr/share/doc/apache2-manual-2.0.48/index.html

Viens de « A Patchy » fait de patches

Ports TCP	80	HTTP
	443	HTTPS
	8080	Proxy

Fichiers:
2 arborescences distinctes (dans Mandrake, Redhat):
/etc/httpd => pour les fichiers de configuration
- dans /etc/httpd/conf/commonhttpd.conf => Fichier de configuration
- dans /etc/httpd/conf/httpd2.conf => Gestion des modules externes

/var/www/html => Emplacement des données à partager

Les Logs:
- /var/log/httpd/access_log
- /var/log/httpd/sslaccess_log
- /var/log/httpd/error_log

ATTENTION: Les Options respectent la casse.

Options de configuration d'une balise <Directory> dans le fichier commonhttpd.conf

C'est une balise englobante à fermer quand terminée:
<Directory nom_répertoire>
Options Indexes FollowSymLinks SymLinksIfOwner MultiViews Includes All

Indexes	=> Affiche le contenu du répertoire si la demande du client ne trouve pas le fichier de démarrage
FollowSymLinks	=> Permet ou non l'affichage des fichiers pointant sur des liens en dehors de la racine Internet (DocumentRoot = /var/www/html)
SymLinksIfOwner	=> presque comme FollowSymLinks mais ok si le fichier lui appartient
MultiView	=> Permet ou non la négociation du contenu affiché en fonction du langage du navigateur client
Includes	=> Permet ou non l'utilisation des langages de programmation pour interpréter le contenu des documents.
ExecCGI	=> (Common Gateway Interface) Permet ou non l'exécution immédiate des fichiers ayant une extension identifiée par une directive Addtype (association de fichiers)
All	=> Active ou non toutes les options

IMPORTANT:
- Pour chaque options, un + activera celle ci alors qu'un – la désactivera (si rien devant = +)
- Si l'option n'est pas défini, elle est héritée du dossier parent jusqu'à la racine (ou il n(y a aucune option)

Options:
AllowOverride All | None | AuthConfig | Options
Permet de définir pour un sous répertoire, comment sera traité le fichier .htaccess

All	=> toutes les options peuvent être modifiées par un fichier .htaccess pour l'accès au sous répertoire.
None	=> l'accès aux sous répertoires du répertoire courant ne pourra pas être modifié par un fichier tel qu'indiqué dans la directive Access File Name (.htaccess)
AuthConfig	=> Seules les options d'authentification sont modifiables
Options	=> Seuls les paramètres de la directive Options peuvent être modifiés
</Directory>	=> on ferme la balise Directory

Exemple:
- Créer un lien symbolique pointant sur .home et en autorisant la visualisation

#ln -s /home /var/www/html/home
- dans le fichier commonhttpd.conf à ajouter en fin

<Directory /var/www/html/home>
- Options +Indexes +FollowSymLinks

</Directory>
- redémarrer le service httpd

#service httpd restart

Filtrage des accès au niveau Machine:
<Directory>
Order Allow, Deny => indique l'ordre de lecture des options Allow From et Order Deny Allow ou Deny From

Allow From 192.168.2.0/24 => autorise la plage 192.168.2.0
Deny From 192.168.2.50 => interdit la plage défini (ou ici l'hôte 50)

Remarque: pour ces deux options on peut utiliser une IP, une plage IP/CIDR, un nom d'hôte ou un nom de domaine (nécessite un DNS pour les noms d'hôte et domaines)

Filtrage des accès au niveau Utilisateur:
IMPORTANT:
- Les comptes Apache sont totalement indépendants des comptes linux
- La navigation se fait en tant que utilisateurs "nobody" (Seuls les fichiers en lecture pour others sont visibles

Création d'un compte apache:

1. **Créer un fichier de compte:**
#touch /etc/httpd/conf/*nom_fichier* => souvent users

2. **Ajout d'un compte**
#htpasswd /etc/httpd/conf/*nom_fichier*
Remarque:
htpasswd -c /etc/httpd/conf/*users user1* => -c crée le fichier avant d'y mettre l'utilisateur

Création d'un groupe :
#vi /etc/httpd/conf/*nom_fichier_groupes* => souvent groups
dans le fichier
nom_groupe: utilisateur1 utilsateur2
ATTENTION:
pas d'espace avant les deux points mais un espace après les deux points.

Dans le fichier commonhttpd.conf
<Directory>
Order Allow, Deny
Allow From 192.168.2.0/24
Deny From 192.168.2.15
AuthName "texte_de_l'invite_de_login"
AuthType Basic => ou Digest mais nécessite un paquetage
AuthUserFile *nom_fichier_utilisateurs* => liste des utilisateurs authentifiés (chemin complet)
AuthGroupFile *nom_fichier_groupes* => seulement si filtrage par groupe
Requirevalid user => autorise tout le monde
user nom_user1 nom_user2 => Seuls les users mentionnés sont autorisés
group nom_groupe1 => seuls les groupes mentionnés seront autorisés
Satisfy All => autorisé si authentification machine ET user
Any => autorisé si authentification machine OU user

Exemple:
pour le répertoire /home autoriser les liens externes et le parcours du répertoire.
Autoriser tout le réseau 10.0 mais pas la machine 10.117
Créer 2 comptes Apache web1 et web2 et filtrer l'accès à /var/www/html/home en autorisant web1 et web2 après authentification
n'autoriser que les utilisateur ET ordinateurs authentifiés

<Directory /var/www/html/home>
Options +Indexes +FollowSymLinks
Order Allow,Deny
Allow From 192.168.10.0/24
Deny From 192.168.10.117
AuthName "Bienvenue sur mon site web"
AuthType Basic
AuthUserFile /etc/httpd/conf/users
AuthGroupFile /etc/httpd/conf/groups
Require valid-user
Satisfy All
</Directory>

Exercice:
Créer un compte Apache pour chaque utilisateur de votre système ayant son répertoire d'accueil dans /home

```
#!/bin/bash
rm -f /etc/httpd/conf/users
touch /etc/httpd/conf/users
cd /home
for i in `ls -d *`                    #attention a ne pas avoir de fichiers dans /home
do
      echo "création du compte apache" $i
      htpasswd -b /etc/httpd/conf/users $i $i
done
```

Utilisation des fichiers .htaccess :
=> Nom précisé dans la directive Access File Name (ligne74)
=> Pris en compte dynamique (sans redémarrer le serveur)
=> Récursivité (les .htaccess sont exécutés dans l'ordre de la structure des répertoires)

```
<Directory /var/www/html/home>
Options +Indexes +FollowSymLinks
AllowOverride All
Order Allow,Deny
Allow From 192.168.10.0/24
Deny From 192.168.10.117
AuthName "Bienvenue sur mon site web"
AuthType Basic
AuthUserFile /etc/httpd/conf/users
AuthGroupFile /etc/httpd/conf/groups
Require valid-user
Satisfy All
</Directory>
```

Pour chaque dossier dans home, je vais créer un fichier .htaccess
Require user compta1 => pour /home/compta1
Quand le client tape: http://localhost/home/compta1 il vérifie si il existe un fichier .htaccess et si oui remplace les valeurs du commonhttpd.conf par ceux du .htaccess mais tient compte des autres balises

Filtrage des accès à l'aide du compte anonyme:
Toujours dans la balise <Directory>

```
Anonymou_Authoritative On | Off          => On : seulement les comptes anonymes
                                            Off : comptes anonymes et comptes apache
Anonymous nom1 nom2                      => nom des comptes autorisés en anonyme
Anonymous_MustGiveEmail On | Off         => le password doit être une adresse mail ou pas.
Anonymous_VerifyEmail On | Off
AuthName "Bienvenue sur mon site web"
AuthType Basic
AuthUserFile /etc/httpd/conf/users       => si Anonymou_Authoritative=Off
Require valid-user
```

Création des sites virtuels:
But : faire cohabiter des sites web différents sur une même machine avec 3 possibilités:
- Socket différent par ip différent
- Socket différent par port différent
- Socket identique et utilisation d'entête d'hôte. (host hearder)
- un mélange des options précédentes

ATTENTION:
Il est conseillé de considérer le site par défaut comme un site virtuel et de déplacer sa configuration.
=> créer une balise serveur virtuel pour /var/www/html

dans le Fichier /etc/httpd/conf/httpd2.conf:
Server Name *nom_du_serveur* => Doit exister dans hosts ou DNS
A fin de test:
en ligne 65 on désactive la ligne qui redirige toute requête dans le vent vers le site web sécurisé par défaut.

Cas n° 1 :

- **2 sites avec socket identique et entêtes d'hôte.**

– Mettre en commentaire la ligne 65 du httpd2.conf (évite les Pb avec le site web par défaut)
– en fin de fichier httpd2.conf

NameVirtualHost * => * indique que le serveur aura une entête d'hôte
<VirtualHost *>
ServerName *nom_site* => nom à donner dans l'URL
DocumentRoot *rép_racine_site* => Indique le répertoire racine du site
<Directory>
...
</Directory>
</VirtualHost>

<VirtualHost *>
ServerName *nom_site* => nom à donner dans l'URL pour le site 2
DocumentRoot *rép_racine_site* => Indique le répertoire racine du site 2
<Directory>
...
</Directory>
</VirtualHost>

REMARQUE:
vider le cache du client.

Cas n°2 :

- **Deux sockets différents par leur adresse IP:**

– configurer une seconde adresse IP
copier /etc/sysconfig/network-scripts/ifcfg-eth0 en /etc/sysconfig/network-scripts/ifcfg-eth0:0
modifier eth0:0 avec la seconde adresse IP

– configurer httpd2.conf
<VirtualHost *IPsite1*>
ServerName *nomsite1*
DocumentRoot *racine_site1*
</virtualHost>

<VirtualHost *IPsite2*>
ServerName *nomsite2*
DocumentRoot *racine_site2*
</virtualHost>

#service httpd restart

– Affecter un nom dans hosts ou DNS (facultatif)
#vi /etc/hosts
192.168.10.7 *nomsite1* => *Ip site1*
192.168.10.8 *nomsite2* => *Ip site2*

Cas n°3 :

- **Deux sockets différents par leur Port d'écoute:**

– configurer httpd2.conf
Listen *ip_site:n°_port_site1* => *Indique le port d'écoute pour le site1*
Listen *ip_site:n°_port_site2* => *Indique le port d'écoute pour le site2*

<VirtualHost *ip_site:n°_port_site1*>
ServerName *nomsite1*
DocumentRoot *racine_site1*
</virtualHost>

<VirtualHost *ip_site:n°_port_site2*>
ServerName *nomsite2*
DocumentRoot *racine_site2*
</virtualHost>

#service httpd restart

Cas n°4 :

- ***Combinaison des différents cas.***

Deux entêtes d'hôte pour site1 et site2, un site sur une ip différente et un site sur la même ip que site1 et site2 mais sur le port 8000.

NameVirtualHost *
<VirtualHost *>
ServerName www.site1.fr =>*Pointe sur 192.168.10.7*
DocumentRoot /var/www/html/site1
<Directory>
...
</Directory>
</VirtualHost>

<VirtualHost *>
ServerName www.site2.fr =>*Pointe sur 192.168.10.7*
DocumentRoot */var/www/html/site2*
<Directory>
...
</Directory>
</VirtualHost>

<VirtualHost *192.168.10.17*>
ServerName www.site3.fr => *Pointe sur 192.168.10.17*
DocumentRoot */var/www/html/site3*
</virtualHost>
Listen 192.168.10.7:8000
<VirtualHost *192.168.10.7:8000*>
ServerName www.site4.fr => *Pointe sur 192.168.10.17*
DocumentRoot */var/www/html/site3*
</virtualHost>

LE SERVICE DE MESSAGERIE POSTFIX

Paquetages :
- postefix
- imap-2002d

Ports: TCP 25 (SMTP: Simple Mail Transfert Protocol)
110 (POP3)
143 (IMAP4)

Fichiers:
/etc/postfix/main.cf => Configuration du service
/etc/postfix/master.cf => Configuration des composants du service
/etc/postfix/aliases => Configuration des alias de noms
/var/spool/mail => Répertoire contenant les boites aux lettres
/var/log/mail/ Infos =>trace des envois de messages
/var/log/mail/Warnings
/var/log/mail/errors
/var/spool/postfix/etc/resolv.conf => Liste des serveurs DNS du serveur de messagerie
Attention: pas de synchronisation entre resolv.conf client et serveur
/var/spool/postfix/etc/hosts => fichier hosts pour postfix

Commandes:
#service postfix start|stop|restart => gestion du service postfix
#postalias => Misa à jour du fichier des alias
#postmap => Création des tables de la base de données interne à postfix
#mail => envoi, consultation des messages

Mise en place d'une configuration de base:

1. **sauvegarder le fichier /etc/postfix/main.cf**

#cd /etc/postfix
#cp main.cf main.cf.backup

2. **utiliser à sa place le fichier main.cf.dist**

#mv main.cf.dist main.cf
#vi main.cf
quelques variables sont à renseigner:
sendmail_path = /usr/sbin/sendmail.postfix => (ligne 621) chemin de l"exécutable postfix
newaliases_path = /usr/bin/newaliases.postfix => (ligne 626) chemin d'emplacement des alias
mailq_path = /usr/bin/mailq.postfix => (ligne 631)
setgid_group = postdrop => (ligne 637)
manpage_directory = /usr/share/man => (ligne 641) emplacement des fichiers man

On peut mettre ces lignes en commentaire:
sample_directory = /usr/share/doc/postfix-2.1.0/samples => (l 645) emplacement des exemples
readme_directory = /usr/share/doc/postfix-2.1.0/readme => (l 649) répertoire du readme

myhostname = nom_host => (ligne 67) nom d'hôte du serveur
ATTENTION: nom_host doit exister dans /etc/host et /var/spool/postfix/etc/hosts
mydomain = nom_domain_dns => (ligne 75) domaine smtp géré par le serveur
ATTENTION: le nom_hote.nom_DNS doit être résolu
myorigin = $myhostname => (ligne 91) ce qui sera affiché derrière le @
Remarque : une autre variable indique que le nom derrière @ sera nomhote+domaine

mydestination = $myhostname,$myhostname.$mydomain,$mydomain => (ligne 159) indique comment essayer de résoudre le nom du compte

Modifier l'emplacement du fichier contenant les alias:
alias_maps = hash:/etc/postfix/alias => (ligne 385)
créer une table des alias pour la base de donnée interne à postfix
#postalias /etc/postfix/aliases

dans la partie: receving mail
#inet_interfaces = localhosts |
inet_interfaces = all | =>choisir la configuration réseau d'écoute

#inet_interfaces = $myhostname | lignes 153à 156
#inet_interfaces = $myhostname, localhost |

Vérification après redémarrage du service:
netstat -na | grep -w 25 => le port 25 est à LISTEN

Test d'un envoi de message:
#mail nom_user
subject: sujet [ENTER]
texte_du_message [ENTER]
[CTRL][D]
Cc: *autre_nom_user* [ENTER]

Options de consultation des mails en consultation:
#mail => ne permet que de consulter la BAL (pas les fichiers lu)
U1
U2 U (unread) indique un message déjà listé mais pas lu
N3 N (new) indique un nouveau message
N4
1[ENTER] => Lecture du message
d3 [ENTER] => Suppression du message 3
r2 [ENTER] => Répondre au 2
q [ENTER] => Quitte

Après mise en place d'un MX dans la zone DNS, l'adresse mail peut être @nom_domaine
ATTENTION: Tous les messages envoyés à root sont redirigés sur la BAL POSTFIX

Modification d'origine des messages dans le champ From:
Par défaut le champ from des mails contient le nom de machine de l'expéditeur
dans main.cf
myorigin = $mydomain

Exercice:
Vous savez que votre utilisateur compta1 doit recevoir un message très important en provenance des dom-tom.
Créez un programme qui vous avertira dès que compta1 aura reçu ce message, puis dès qu'il l'aura lu.
Remarques:
les messages reçus sont dans un seul fichier : /var/spool/mail/nom_user
les messages lus sont dans un fichier: /home/nom_user/mbox

maildomtom:

```
#!/bin/bash
clear
echo -e "entrez le nom de domaine de l'émetteur:" \\c
read domaine
echo -e "entrez le nom du compte récepteur du message:" \\c
read user
nbmaildomtom=`cat /var/spool/mail/$user | grep -i $domaine | wc -l`
nbmaildomtomlu=`cat /home/$user/mbox | grep -i $domaine | wc -l`
nbmailnow=$nbmaildomtom
lu=$nbmaildomtomlu

while [ $nbmaildomtom == $nbmailnow ]
do
      nbmailnow=`cat /var/spool/mail/$user | grep -i $domaine | wc -l`
      sleep 5
done
clear
echo "Compta1 a reçu un message des DOM-TOM à `date +%T` heures"

while [ $lu == $nbmaildomtomlu ]
do
      lu=`cat /home/$user/mbox | grep -i $domaine | wc -l`
      sleep 5
done
echo "il viens de le lire à `date +%T` heures"
```

Utilisation de mail en mode batch:
#mail adresse_user -s "sujet_du_message" < fichier_contenant_le_message
ou
#mail adresse_user -s "sujet_du_message" < --MOT texte_jusqu_au_prochain_MOT MOT

le premier nécessite la création d'un fichier contenant les données mais la seconde ne permet pas de retour chariot

Commandes du protocole SMTP:
#telnet nom_serveur 25 Connected to site1 (192.168.10.7).
Escape character is '^]'.
220 thierry ESMTP Postfix (2.1.0-pre-20040209) (Mandrake Linux) **remarque**: configurées dans la directive: smtpd_banner
HELO toto => connection
250 THIERRY
MAIL_FROM: nom_user_nom_domaine => émetteur
250 OK
RCPT TO: toto@free.fr => destinataire
250 OK
DATA
354 End data with <CR><LF>.<CR><LF> => fin de saisie [ENTER]. [ENTER]
donnes [ENTER]. [ENTER]
250 Ok: queued as BA33EFA44 => sauve avec un numéro aléatoire
QUIT
221 Bye

Installation du pop3:
Les messages sont stockées dans des BAL et redistribuées par le serveur POP3 aux clients.

Configuration du service pop sur le MTA
/etc/xinetd.d/ipop3
service pop {
        disable = no
        }
1. vérifier que le service est bien configuré
#urpmi imap-2002d
2. vérifier que le service est actif
#netstat -na | grep -w 110

Commandes POP (Accès en telnet):
telnet nom_serveur_pop3 110
Trying 192.168.10.7...
Connected to thierry (192.168.10.7).
Escape character is '^]'.
+OK POP3 site1 v2003.83mdk server ready
user thierry
+OK User name accepted, password please
PASS pass_thierry
+OK Mailbox open, 0 messages
RETR n°_message => Retrieve pour récupérer le mail
+OK 397 octets
Return-Path: <root@thierry.corse.fr>
X-Original-To: thierry
Delivered-To: thierry@thierry.corse.fr
Received: by thierry (Postfix, from userid 0)
        id 7B220FC22; Thu, 1 Sep 2005 16:07:51 +0200 (CEST)
To: thierry@thierry.corse.fr
Subject: hello boys
Message-Id: <20050901140751.7B220FC22@thierry>
Date: Thu, 1 Sep 2005 16:07:51 +0200 (CEST)
From: root@thierry.corse.fr (root)
Status:
coucou
DELE 1 => supprimer le message 1
+OK Message deleted
LIST
+OK Mailbox scan listing follows
1 538

2 704
TOP n°_message nb_lignes => Affiche le nombre de lignes indiquées du message indiqué
+OK Top of message follows
kl
jhkj
quit
+OK Sayonara

Gestion d'un compte imap4:
L'intérêt principal de IMAP par rapport à pop est qu'il permet de travailler sur les fichiers mails centralisés sur un serveur alors que pop (par défaut) récupère les mails en local avant de les lire.

1. **Vérifier que le service imap4 est disponible**

dans xinetd
/etc/xinetd.d/imap
service imap {
disable=no => yes par défaut
}

2. **Vérifier qu'il est actif**

#netstat -na | grep -w 143

3. **Côté serveur:**

Créer/choisir un compte pour son voisin

4. **Côté client:**

Configurer une BAL de type imap pour ce compte

En telnet:
[root@thierry thierry]# telnet 192.168.10.7 143
Trying 192.168.10.7...
Connected to site1 (192.168.10.7).
Escape character is '^]'.
* OK [CAPABILITY IMAP4REV1 LOGIN-REFERRALS AUTH=LOGIN] site1 IMAP4rev1 2003.338m dk at Fri, 2 Sep 2005 09:07:08 +0200 (CEST)
a1 LOGIN compta1 compta1
a1 OK [CAPABILITY IMAP4REV1 IDLE NAMESPACE MAILBOX-REFERRALS BINARY UNSELECT SCA N SORT THREAD=REFERENCES THREAD=ORDEREDSUBJECT MULTIAPPEND] User compta1 authent icated
a2 select INBOX
* 14 EXISTS
* 1 RECENT
* OK [UIDVALIDITY 1125583891] UID validity status
* OK [UIDNEXT 15] Predicted next UID
* FLAGS (\Answered \Flagged \Deleted \Draft \Seen)
* OK [PERMANENTFLAGS (* \Answered \Flagged \Deleted \Draft \Seen)] Permanent flags
* OK [UNSEEN 2] first unseen message in /var/spool/mail/compta1
a2 OK [READ-WRITE] SELECT completed
a3 FETCH 12 ALL => lit tout le message 12
* 12 FETCH (FLAGS () INTERNALDATE " 1-Sep-2005 15:29:06 +0200" RFC822.SIZE 532 ENVELOPE ("Thu, 1 Sep 2005 15:30:14 +0200 (CEST)" "POIPOIPOI" (("root" NIL "root" "bretagne.fr")) (("root" NIL "root" "bretagne.fr")) (("root" NIL "root" "bretagne.fr")) ((NIL NIL "compta1" "corse.fr")) NIL NIL NIL "<20050901133014.1496BAFB2@IFC4>"))
a3 OK FETCH completed

Gestion des alias :
permet d'envoyer des mails à un alias plutôt que taper son nom complet de messagerie
permet aussi de simuler une liste de distribution.

Le fichier alias :
/etc/postfix/aliases
---
root: postfix
toto: toto.monbateau
ouest: toto@free.fr,titi@wanadoo.fr => liste de distribution

Pour mettre à jour le fichier alias :
#postalias /etc/postfix/aliases
REMARQUE : pas instantané (environ 30 secondes)

Sécurité du serveur de messagerie:

Ordre des restrictions à mettre en oeuvre :

1. Restriction sur l'IP ou le nom d'hôte => smtp_client_restrictions
2. filtrage sur le protocole SMTP => smtpd_ehlo_restrictions
3. Filtrage de la commande MAIL FROM => smtpd_sender_restrictions
4. Filtrage sur le destinataire RCPT TO => smtpd-recipient_restrictions (commande RCPT To)
5. Filtrage du contenu => Contenu de l'entête et du corps du message

dans **/etc/postfix/main.cf**
message_size_limit = 400000 => taille des messages courants en octets
mailbox_size_limit = 1000000 => taille maximum des BAL en octets
header_size_limit = 10000 => taille limite des entêtes de mail (failles) en o
smtpd_helo_required = yes | no => authentification obligatoire (yes par défaut)
smtpd_helo_restrictions = permit_mynetworks
smtp_sender_restrictions = nom_dom_exped_rejeté
smtp_client_restrictions = checkclientaccess hash:/etc/postfix/access
=> Restrictions dans le fichier indiqué

exemple de fichiers de configuration:/usr/share/doc/postfix-2.1.0/samples

Mise en place de restrictions sur le contenu (regexp =voir expressions régulières):
body_checks = regexp:nom_du_fichier_contenant_les_filtrages_au_format_regexp

Mise en place de restrictions sur l'entête du message
header_checks = regexp:nom_du_fichier_contenant_les_filtrages_au_format_regexp

Syntaxe des fichiers "regexp"
/etc/postfix/bodycheck
/chaîne_à_filtrer/*flag* ACTION
flag => répertoires=ignorer la casse

ACTION (attention à la casse):
- DUNNO =>Do Nothing (ne fait rien, on suppose que le message sera traité ultérieurement)
- OK => Le message est accepté (à ce niveau de filtrage)
- DISCARD => mail détruit sans prévenir l'expéditeur
- REJECT "message de rejet" => Le message est renvoyé à l'expéditeur, le mail est rejeté
- WARN "text" => message accepté mais avec trace dans les logs du message et du contenu
- REDIRECT adresse_mail => Le message est redirigé vers l'adresse indiquée

Remarque: le ! Permet d'inverser la balise.

Créer la table liée à ce fichier pour la base de données interne à POSTFIX
#postmap /etc/postfix/bodycheck
#postfix reload

RELAIS SMTP : Filtrage des clients SMTP :
(ligne 231 ...) mynetworks_style permet de définir qui peut envoyer des mails externes au domaine SMTP du serveur
mynetworks_style = host => seulement l'hôte
subnet => Autorise seulement les postes du même subnet que le serveur
class => autorise tout ordinateur dans la même classe ip que le serveur

mynetworks = ip_réseau/, ip_autre_réseau => indique les ips autorisées à relayer (ex:10.0.0.0/8)
$mydomain => Seulement des ordinateurs de mon domaine (nécessite la recherche DNS inverse)
hash:/etc/postfix/network_table => fichier contenant la liste des réseau autorisés

ex:
/^From:*@domtom.fr/
le ^ indique le début de ligne
le . Indique n'importe quel caractère
le * Indique n'importe quel nombre de caractères

Le Filtrage des entêtes (man header_checks):

1. Dans Main.cf

header_checks = regexp:*nom_fichier_contenant_les filtres*

2. Création de la table correspondant à ce fichier

#postmap *nom_fichier_contenant_les filtres*
Ex: /etc/postfix/entête

/^content-(type|disposition).*name.*\.(exe| vbs|bat|cmd)/ REJECT

^ => commence par la chaîne suivante
(type|disposition) => Liste de mots possibles 1er argument ($1)
. => Indique n'importe quel caractère
* => N'importe quel nombre de fois le caractère précédent
(exe| vbs|bat|cmd) => Défini les extensions choisis (séparés par des |) 2eme argument ($2)

/^From:.*@domtom\.fr/ WARN encore un mail des domtom
/^Subject:.* mot_non_souhaité.*/DISCARD

ex:
/^Subject:.*sex.*/ DISCARD => rejette tout mail dont le sujet contient sex (sexiste...)
/^Subject:.* sex .*/ DISCARD => rejette tout mail dont le sujet contient le mot sex grâce aux espaces devant et derrière

Filtrage du volume de messages:

dans main.cf:
#pour le même client:
smtpd_client_connection_count = n => n est le nombre de connexions maximum du même client dans l'intervalle de temps défini par la directive suivante.
smtpd_client_connection_rate_limit = n => n= intervalle de temps en seconde.
smtpd_client_connection_limit_exceptions= $mynetwork => exceptions sur les options précédentes

Répartition de charge et tolérance de panne sur les serveur de messagerie:

On place deux serveurs de messagerie avec 2 Mx dans le DNS pour une tolérance de panne et répartition de charge mais les clients doivent pouvoir avoir leurs données quel que soit le serveur qui répond.
On ajoute un serveur de centralisation de BAL (serveur NFS ou disque SCSI partagé) et on indique à chaque serveur de messagerie le point de montage de copie des BAL

Sur le DNS:

- Pour la zone de domaine SMTP => créer deux Mx pour les serveurs de messagerie

Sur les serveurs POSTFIX:

- mettre le même nom de domaine SMTP pour les deux serveurs
- faire un point de montage vers le serveur NFS pour le point de stockage des mails (/var/spool/mail)
- Créer des comptes utilisateurs utilisant le même uid sur les deux serveurs

Sur le serveur NFS

- Créer une ressource partagée NFS en lecture écriture pour les utilisateurs de postfix
- Attention aux droits et aux balises anonuid et anongiu.

Pour la capture de trames:
For i IN `seq 1 500`; do mail postman@irlande.net -s "mail$i" <pouipoui;usleep 100000; done

ADMINISTRATION AVANCÉE DE LA LIGNE DE COMMANDES

Développement des noms de fichiers:

Pour une variable $i existante:
${i%% partie_à_ignorer_à_droite}
${i##partie_de_$i_à_ignorer_à_gauche}

Les expressions régulières avancées (Disponibles avec grep -E):

expr*	=> Jocker de 0 à x caractères
?	=> Jocker de 1 caractère
^expr	=> Commence par
expr$	=> exp en fin de ligne
.	=> n'importe quel caractère
\.	=> Le caractère "." (\ indique que le prochain caractère est lu tel quel)
expr+	=> Au moins une fois l'expr
expr\{n\}	=> exactement n fois l'expr
expr\{n,m\}	=> Entre n et m fois l'expr
expr\{n,\}	=> Au moins n fois l'expr
[char1char2char3]	=> char1 ou char2 ou char3 à cette position
[0-9] ou [a-zA-Z]	=> caractères compris entre 0 et 9, a et z ... à cette position
[^0-9]	=> Dans les crochets le ^ = exclusion de l'intervalle de valeurs
{val1,val2,val3}	=> développement pour chaque caractère de la lister entre accolades

Exemples:
pour les fichiers /file1, /File1, File2, /Formulaire1, /Formulaire2, /pile1, /Pile1, filet, /Fileur
- Comment afficher les fichiers dont le nom commence par un f ou un F suivi de ile et suivi d'un chiffre?
#ls [Ff]ile[0-9]
- Afficher les fichiers dont le nom commence par un f ou un F suivi de ile et suivi d'un caractère non numérique
#ls [Ff]ile[^0-9]
#touch {a,p}ile{3,4} => création de aile3,aile4,pile3,pile4

Comment renommer les fichiers: fichier1.txt, fichier2.txt, fichier3.txt, fichier4.txt en .doc?
Boucle for avec les expressions régulières:
```
for i in `ls fichier?.txt`
do
mv $i ${i%%.txt}.doc     =>%% extraction de la partie de $i se trouvant à gauche de .txt
done
```

Trouver les lignes qui commencent et ne contiennent que des 1:
#cat fichier-sed | grep ^11*$ => si on met ^1*$ il met aussi les lignes vides car * =0 ou plus
#cat fichier-sed | grep -E ^1+$ => identique à celle du dessus

Trouver des lignes qui ne contiennent aucun caractères numériques:
#cat fichier-sed | grep ^[^0-9]*$

Chercher les lignes qui contiennent 3 chiffres seulement et contigus:
#cat fichier-sed | grep ^[0-9][0-9][0-9]$

chercher les lignes qui contiennent 3 chiffres quelle que soit leur position
#cat fichier-sed | grep ^[^0-9]*[0-9][^0-9]*[0-9][^0-9]*[0-9][^0-9]*$

Une lettre minuscule en 3eme position
#cat fichier-sed | grep ^..[a-z]

Chercher le nom d'une personne avec l'initiale seulement en majuscule sans monsieur
#cat fichier-sed | grep [Mn][Oo][Nn][Ss][Ii][Ee][Uu][Rr] [A-Z][a-z]+

Afficher dans le fichier liste.machines pour les quelles, le dernier octet de l'ip est supérieur ou égale à 100
#cat liste.machines | grep -E ^[0-9]\{1,3\}\.[0-9]\{1,3\}\.[0-9]\{1,3\}\.[1-2][0-9][0-9]

L'OUTIL SED (Stream Editor):

Utilisation du fichier :
fichier-sed

Edite un fichier et le traite ligne par ligne.
#sed -option 'action/expression/autre_action' nom_fichier

Afficher les lignes d'un fichier contenant une expression
#sed -n '/*expression*/p' *nom_fichier* => Equivaut à #grep -E expr nom_fichier

Remplacer expr1 par expr2
#sed 's/*expr1*/*expr2*/'*nom_fichier*

Remplacer expr1 par expr2 en mentionnant expr1 dans expr2
#sed 's/*expr1*/&*expr2*/'*nom_fichier* => le & est remplacé par expr1

Supprimer les lignes contenant l'expression donnée
#sed '/expr/d' nom_fichier

remplacer caractère par caractère:
#sed 'y/1234/abcd/' nom_fichier => 1 est remplacé par a, 2 par b ....

Remplacer expr1 par expr2 seulement pour les lignes qui contiennent expr3
#sed '/exp3/s/expr1/expr2/' nom_fichier

Faire une substitution pour toutes les occurrences de expr1
#sed 's/expr1/expr2/g' nom_fichier

Inverser la recherche
exemple: rechercher les lignes ne contenant pas expr1
#sed -n '!/expr1/p' nom_fichier

Utilisation d'un script sed
#sed -f nom_script_sed nom_fichier
pour ce fichier:
ligne1
ligne2
ligne2
ligne5
ligne6
ligne6
ligne8
ligne9
ligne10

Dans le script sed:

2a\	=> Append (ajouter après la ligne 2)
ligne3	
6c\	=> Chang (modifie la ligne 6 en l'appelant ligne7)
ligne7	
9s/LIGNE/ligne	=> Substitute
7i\	=> insert:avant la ligne indiquée
texte	
y/char1/char2/	=>remplacer le char1 par char2
3,8d	=> Suppression de la ligne 3 à 8

ATTENTION: les lignes sont numérotées au départ par sed et il garde cette numérotation jusqu'à la fin du traitement.

Exemple:
sur un numéro de sécurité sociale 1-75-5-689-456-78 je veux changer le 5 (mois de mai) en 05
je découpe en trois parties le numéro (avant valeur, valeur a changer , après valeur) avec des ()
ensuite les parenthèses sont appelées 1, 2 et 3
#sed 's/^\([0-9]-[0-9][0-9]-\)\([0-9]-\)\(.*\)/\10\2\3/' *nom_fichier*
---------- ---------- ---
Ch1 Ch2 Ch3

L'OUTIL AWK (Aho, Weinberger, Kernighan)

Mini langage qui traite des fichiers ligne par ligne comme une boucle FOR, (caractère par caractère) avec notion de champ.
Une ligne est appelée enregistrement. Chaque enregistrement est composé de champs.

#awk -F séparateur_de_champ '{commande interne}' nom_fichier

Séparateur de champ

$0	=> Ligne complète
$n	=> Champ n de la ligne
NF	=> Nombre de champs dans la ligne
NR	=> numéro de la ligne
FNR	=> nombre de lignes dans le fichier

Syntaxe plus complète

#awk -F "car" '/option_de_tri/{ }'

Options de tri:

/expressions_régulières/ => Seulement pour les lignes contenants l'expression

opérateurs de test:

$n> val	|
$n>=val	| => exécution seulement pour les lignes ou la comparaison arithmétique est VRAIE
$n<val	|
$n<=val	|
$n==val	|
!=	=> différent
$n~/expr/	=>$n contient expr
!(test)	=> inversion du test
test && test2	=> ET logique
test || test2	=> OU logique

opérateurs arithmétiques:

+
*
-
/
var=$n
var2=$1+$2
var3=($1+$2)/NF
var4+=$2 => Somme de tous les champs n°2
var5++ => incrémente la valeur val5

La commande printf:

#awk '{printf("moyenne_de_%s:%2.2f\n,"$1,($2,$3,$4,$5)/4}'
%s => Chaîne de caractère (%4.2f nombre de 4 chiffres et 2 chiffres âpres la virgule)
%d => nombre décimal
%f => nombre réel
\n => retour à la ligne

Exemple:

pour sortir du passwd les users qui ont un uid supérieur à 500:
en bash:

```
for i in `cat /etc/passwd`
do
        if [ `echo $i | cut -d":" -f3\ -gt 500 ]
        then
        echo $i
done
```

en awk:

#awk -F: '$3>500{print $0}' /etc/passwd

LES QUOTAS SUR SYSTÈME DE FICHIERS

Paquetage:
quota-3.09

Principes:
- les quotas s'appliquent sur la totalité du système de fichiers (crée par mkfs)
- Les quotas s'appliquent soit par utilisateur, soit par groupe, soit les 2
- Il y a deux limites mises en place:
  - Limite soft : préviens l'utilisateur mais pas bloqué (limite de temps pour redescendre)
  - Limite hard: L'utilisateur est bloqué

Deux types de contrôle:
- sur la taille totale des fichiers
- Sur le nombre total de fichiers

(ou les deux)

Mise en place des quotas (exemple: sur /home)

1. **Indiquez l'utilisation des quotas dans /etc/fstab**

/dev/hda11 /home ext3 default,usrquota 1 2
monter /home avec l'utilisation des quotas

2. **Remonter le système de fichier**

#mount -o remount,usrquota /home
cette commande peut aussi fonctionner en dynamique (quota jusqu'au redémarrage du système)

3. **Créer le fichier indexé listant les quotas pour /home (liste l'état actuel d'utilisation)**

#quotacheck -m /home => -m permet de faire le fichier même si certains fichiers sont utilisés

4. **Visualiser l'état actuel d'utilisation**

#repquota /home | more
*** Report for user quotas on device /dev/ide/host0/bus0/target0/lun0/part8
Block grace time: 7days; Inode grace time: 7days

	Block limits				File limits			
User	used	soft	hard	grace	used	soft	hard	grace
root	40356	0	0		712	0	0	
thierry	220744	0	0		1632	0	0	
tech	1072	0	0		274	0	0	
tech1	2144	0	0		548	0	0	
totobash	36	0	0		9	0	0	

5. **Editer les quotas pour un utilisateur (éditer avec VI: changer les limites)**

#edquota -u nom_user /home

Disk quotas for user thierry (uid 501):

Filesystem	blocks	soft	hard	inodes	soft	hard
/dev/ide/host0/bus0/target0/lun0/part8	220744	0	0	1632	0	0

6. **Activer la surveillance des quotas:**

#quotaon

7. **Indication d'un délai de grâce:**

#edquota -t => Mettre une valeur de grâce sur les utilisateurs qui nous intéresse
REMARQUE: cette valeur est la même pour tous les utilisateurs sous quota

Arrêt de la gestion des quotas:
#quotaoff /home

limite soft:
- Déclenche un avertissement lorsqu'elle est dépassée mais ne bloque pas l'écriture en cours
- Déclenche un compte à rebours indiqué dans le délai de grâce

Limite hard:

- empêche toute écriture lorsqu'on tente de la dépasser

Délais de grâce :

- Transforme la limite soft en limite hard s'il est écoulé sans retour du quota utilisateur sous la limite soft

Exercice:

Choisir un utilisateur et lui appliquer des quotas en nombre de blocs (soft=10 hard=20)
Créer un programme qui crée un fichier dont la taille va lui faire dépasser la limite soft mais pas la limite hard
Créer un autre programme qui crée un fichier dont la taille va lui faire dépasser la limite hard.
Vérifier si le fichier est tronqué pendant l'écriture ou s'il n'est pas crée du tout

Le fichier copié qui dépasse la limite hard est interrompu en cours de copie. Les données restantes sont perdues

Mise en place des quotas par groupe:

1. **Indiquez l'utilisation des quotas dans /etc/fstab**

/dev/hda11 /home ext3 default,usrquota,grpquota 1 2
monter /home avec l'utilisation des quotas

2. **remonter le système de fichier**

#mount -o remount,grpquota /home
cette commande peut aussi fonctionner en dynamique (quota jusqu'au redémarrage du système)

3. **Créer le fichier indexé listant les quotas pour /home (liste l'état actuel d'utilisation)**

#quotacheck -m -g /home => le -m permet de faire de fichier même si certains fichiers sont déjà utilisés

4. **Visualiser l'état actuel d'utilisation**

#repquota /home | more

5. **Editer les quotas pour un utilisateur (éditer avec VI: changer les limites)**

#edquota -g nom_group /home

6. **Activer la surveillance des quotas:**

#quotaon

7. **Indication d'un délai de grâce:**

#edquota -t => Mettre une valeur de grâce sur les utilisateurs qui nous intéresse
REMARQUE: cette valeur est la même pour tous les utilisateurs sous quota

Arrêt de la gestion des quotas:

#quotaoff /home

LES PERMISSIONS SPÉCIALES SUR LES FICHIERS

Ces permissions sont gérées sous forme de bits qu'on applique aux droits des fichiers

1. Le doit d'endossement: Bit SUID:
REMARQUE: uniquement sur les fichiers exécutables
#chmod u+s *nom_fichier*
#ls -l *nom_fichier* => le x est remplacé par un s dans les permissions utilisateurs

L'utilisateur qui exécute un tel fichier récupère les privilèges du propriétaire de ce fichier durant toute son exécution.

Notation arithmétique:
4000 en octal de permission (4XYZ)

Exemple:
#chmod u+s /bin/cat => cat peut être lancé avec les prérogatives de root (cat /etc/shadow)
ou
#chmod 4775 /bin/cat

ATTENTION: DANGER

2. Le bit SGID
Identique au bit UID mais pour le groupe
#chmod g+s *nom_fichier*
#ls -l *nom_fichier* => le x est remplacé par un s dans les permissions de groupe

Notation arithmétique:
2000 (2XYZ)

ATTENTION:
Le bit sgid appliqué à un répertoire change la propriété de groupe de tous les fichiers le groupe propriétaire du répertoire

3. Le sticky bit appliqué à un répertoire:
Seul le propriétaire d'un fichier peut le supprimer lorsque celui-ci est placé dans un tel répertoire.

#chmod o+s nom-répertoire
#ls -ld nom_répertoire
résultat: drwxrwxrwt

Notation arithmétique:
1000 (1XYZ)

REMARQUE: le sticky bit appliqué à un exécutable n'a aucun effet sur Linux

LES MODULES D'AUTHENTIFICATION (PAM)

pluggable authentication module

But :
Vérifier, pour la liste des services présents dans /etc/pam.d , les permissions et contraintes éventuellement mises en place.
Les PAM sont un moyen modulaire d'authentifier des utilisateurs. On peut ainsi définir des stratégies sans avoir à recompiler les programmes qui supportent PAM. En fait PAM, évite que chaque application ne vous redemande une authentification avec des règles différentes à chaque fois.
Avez vous besoin de toucher à cela?
La plupart du temps non, mais si vous souhaitez renforcer la sécurité (obliger des mots de passe qui ne sont pas dans un dictionnaire par exemple), l'uniformiser ou adopter une autre méthode d'authentification (ldap par exemple) alors la réponse est oui.

Il est installé sur tous les unix - linux par défaut. Toutefois il existe des petites différences.
Dans certains cas on trouve l'ensemble de la configuration dans /etc/pam.conf (BSD), sous Redhat, Mandrake on a un fichier par service dans /etc/pam.d/. Si le fichier /etc/pam.d correspondant existe le fichier /etc/pam.conf est ignoré.
Le fichier principal dans le cas de l'existence de /etc/pam.d est system-auth.
Les modules appelés dans les différents fichiers de configuration se trouvent dans /lib/security (ou /usr/lib/) cela dépend de la distribution. Vérifiez donc que le module que vous souhaitez utiliser existe.
Enfin on trouve dans /etc/security des fichiers complémentaires (access.conf, group.conf, limits.conf, time.conf) utilisés par certaines applications.

ATTENTION: Toujours garder ouvert une console root, car si la modification ne marche pas elle peut bloquer la connexion de tout le monde

Pour chaque fichier, le contrôle se fait ligne par ligne.

Structure des fichiers de "service" :
ch1 ch2 ch3 ch4

ch1:type de gestion:	- account	=> Vérifier la validité du compte
	- auth	=> Vérification de l'identité de l'utilisateur
	- password	=> paramètres de modification de l'identité (mdp...)
	- session	=> paramètres liés à l'ouverture et la fermeture de session
ch2:type de contrôle:	- required	=> Succès obligatoire
	- sufficient	=> Succès suffisant pour valider tout le monde pour ce module
	- requisite	=> Succès de la vérification obligatoire: l'échec entraîne l'arrêt immédiat des autres vérifications.
	- optional	=> l'échec n'entraîne pas l'arrêt du module.
ch3:Action:		=> Le plus souvent, appel une bibliothèque dynamique ou redirection vers un autre service
ch4:Paramètres:		=> Liés à l'appel de bibliothèque

Exemple. Fichier pam login

```
#%PAM-1.0
auth     required     pam_securetty.so
auth            required    pam_stack.so service=system-auth
auth            required    pam_nologin.so
account         required    pam_stack.so service=system-auth
password        required    pam_stack.so service=system-auth
session         required    pam_stack.so service=system-auth
session         optional    pam_console.so
```

Exemple de modification
service : login
Edition du fichier /etc/securetty qui liste les terminaux à partir des quels root peut se loguer
supprimer la ligne correspondant au TTY à refuser à root
désactiver la première ligne du fichier pam login : (auth required pam_securetty.so) qui contrôle le fichier securetty

Modules à connaître

- pam_cracklib :
  Permet d'accepter ou de rejeter un mot de passe, si celui-ci se trouve dans un dictionnaire. Il permet aussi de vérifier que vous ne réutilisez pas le même mot de passe. Vous pouvez le faire suivre de retry=n (le nombre de tentatives) minlen=n (la longueur imposée) difok=n (nombre de caractères qui sont dans le vieux mot de passe et que l'on ne peut pas retrouver dans le nouveau).

- pam_env :
  Permet de spécifier des variables d'environnements spécifiées dans /etc/security/pam_env.conf à tout utilisateur qui se connecte sur la machine.

- pam_unix :
  Module de base. Gère à la mode unix la politique d'authentification. Il peut être avec les quatre types de modules : account (établi la validité utilisateur/mot de passe et peut forcer la modification de celui là), auth (compare avec la base le mot de passe), password (la politique de changement du mot de passe), session (pour loguer les connexions).
  Vous pouvez associer quelques options dont : nullock pour autoriser un mot de passe vide, md5 pour le type de cryptage, debug pour loguer les informations à syslog, remember=n pour ce souvenir des n derniers mots de passe utilisés.

- pam_pwdb :
  module de base, qui a les mêmes options que pam_unix.

- pam_time :
  autorise un accès par heure. La configuration se faisant dans le fichier /etc/security/time.conf

- pam_wheel :
  permet de limiter l'accès à root via la commande su qu'aux seuls membres du groupe wheel. On peut changer le nom du groupe par défaut avec l'option group=mon_group.

- pam_limits :
  Permet de limiter les ressources mises à la disposition d'un utilisateur. Il faut alors configurer le fichier /etc/security/limits.conf

- pam_nologin :
  permet de désactiver les comptes. Il faut alors créer le fichier /etc/nologin et alors il n'y a plus que root qui puisse se connecter.

- pam_access :
  Ce module permet de contrôler les utilisateurs par nom, machine, domaine, adresse IP, terminal. Vous devez alors configurer le fichier /etc/security/access.conf

- pam_deny :
  comme sont nom l'indique, vous pouvez (devez !) l'utiliser dans /etc/security/other pour auth, account, password et session avec required. Si dans le répertoire /etc/security vous avez des noms d'applications que vous n'utilisez pas vous pouvez renommer ces fichiers avec un autre nom au cas..! Si quelqu'un cherche à utiliser l'application le other sera alors utilisé par défaut.

- pam_securrety :
  Vérifie que le compte root a la possibilité de se connecter sur cette console. Pour cela il faut qu'elle soit indiquée dans le fichier /etc/securrety.

- pam_warm :
  log les informations à syslog

- pam_console :
  permet de spécifier les autorisations d'accès à la console. Il faut alors configurer /etc/security/console.perms.

- pam_stack :
  généralement suivi de service=system-auth, permet de renvoyer sur system-auth.

- pam_ldap :
  permet d'effectuer l'authentification sur une base ldap. Ce module demande une documentation à lui tout seul.

Exemple
Voici le fichier /etc/pam.d/login.

```
#%PAM-1.0
auth required /lib/security/pam_securetty.so
auth required /lib/security/pam_unix.so shadow nullok
auth required /lib/security/pam_nologin.so
account required /lib/security/pam_unix.so
password required /lib/security/pam_cracklib.so
password required /lib/security/pam_unix.so shadow nullok use_authtok
session required /lib/security/pam_unix.so
```

Ligne 1 => est un commentaire.
Ligne 2 => interdit à root de se connecter sur la console (enfin directement on peut utiliser su), si la console n'est pas autorisée dans /etc/securrety.
Ligne 3 =>Vérifie le mot de passe, on peut aussi utiliser pam_pwdb
Ligne 4 => Vérifie l'existence du fichier /etc/nologin. Si celui-ci existe, il n'est plus possible de se loguer sauf pour root. Affiche son contenu.
Ligne 5 => Comptabilise la "vie" du mot de passe.
Ligne 6 => Teste la validité du mot de passe.
Ligne 7 => Impose les règles de modification du mot de passe.
Ligne 8 => Ne fait rien, mais indique qu'il doit être utilisé pour gérer la session.

Un lien :
Http://www.kernel.org/pub/linux/libs/pam/Linux-PAM-html/pam.html

LE PROXY (SQUID)

Définition d'un proxy: service qui fait une action à la place de...
Dans le cas du proxy Internet, il fait des requêtes à la place du client. Il permet de "partager" une adresse Publique pour X clients

Paquetages :
- squid (proxy web ftp gopher)
- squidGuard (additif de filtrage)

ports TCP :
3128 (modifiable)

Fichier squid:
/etc/squid/squid.conf
/var/log/squid/access.log => Fichier log des translations

Commandes:
#service squid start | restart | ...

Configuration minimum:
(1763) http_access allow all => all = nom de l'acl correspondant à n'importe quelle IP, n'importe quel masque
(1977) visible_hostname nom_hote_du_serveur

Gestion des ACL:

Déclaration d'une ACL:
acl nom_acl type_d_acl paramètres

Types d'acl
src => Source : IP, IP/CIDR, nom_hôte, nom_domaine (nécessite une recherche dns inversée pour le nom d'hôte et de domaine)
dst => Destination : IP, IP/CIDR, nom_hôte, nom_domaine
time => MTWHFAS : lundi mardi ...(A pour Saturday mais S déja pris pour Sunday) HH:MM-HH:MM

Remarque: on utilise les heures, jours ou une combinaison des deux.
url_regex => expression_regulière_definissant_l_url_a_filtrer
urlpath_regex => expression_regulière_definissant_un_bout_du_chemin_de_l'URL

Exemple

acl	reseau100	src	192.168.100.0/24
acl	reseau10	src	192.168.10.0/24
acl	TRAVAIL	time	MTWHF 08:00-18:00

Filtrage de l'accès au proxy à partir des ACL:
IMPORTANT: L'ordre des filtrages "http_access" est TRES important.
Dès qu'une requête correspond à un filtrage, elle est acceptée et ne passe pas dans les autres filtrages
http_access allow *nomacl nomacl* => un ET logique est sous entendu si plusieurs ACL
deny

Un "!" inverse le sens de l'ACL:
exemple: http_access allow !reseau10 =>concerne tout ce qui n'est pas dans le reseau 10

Deux acl peuvent avoir le même nom si leur type est identique:

acl TRAVAIL	time	MTWHF 09:00-12:00
acl TRAVAIL	time	MTWHF 13:30-18:00

Exercice:
Pour les clients du réseau 100.0
- Accès illimité au web en dehors des plages horaires 9h-12h et 13h30-18h du lundi au vendredi
- Accès interdit à tous les sites google, altavista, wanadoo pendant les horaires ci dessus, mais ok pour les autres sites
- accès autorisé en ftp seulement le week-end

Pour le client 192.168.10.254
-accès limité

```
acl illimite254    src 192.168.10.254/32
acl MOTEURS ulr_regex -i ^http://.*(wanadoo|google|altavista).*
acl RESEAU100                  src 192.168.100.0/24
acl TRAVAIL    time     MTWHF 09:00-12:00
acl TRAVAIL    time     MTWHF 13:30-18:00
acl FTP        proto    ftp
acl WEEKEND    time     AS

http_access deny RESEAU100 TRAVAIL MOTEURS
http_access allow illimite254
http_access allow RESEAU100 WEEKEND FTP
http_access allow RESEAU100
http_access deny all
http_access deny all
```

Authentification des acces au proxy (authentification pour tout le monde)

1. Créer un service "squid" pour PAM ou vérifier que le fichier existe
#vi /etc/pam.d/squid

2. Configurer squid.conf
ATTENTION: Lignes à placer avant les déclarations d'ACL
auth_param basic program /usr/lib/squid/pam-auth => Appel au programme d'authentification
auth_param basic realm "texte de l'invité de login => texte fourni au client
auth_param basic credentialsttl 1 hours => limite l'authentification à 1 heure
error_directory /usr/répertoires/squid/errors/French => donne le chemin des erreurs en fr

REMARQUE: après ces deux lignes:
acl *LOGIN* proxy_auth REQUIRED
.
.
http_access allow .....LOGIN => A mettre devant les http_access en authentification

REMARQUE:
Pour une authentification par user: utiliser openldap

Exercice:
trouver les sites visités et les compter:
#cat /var/log/squid/access.log | awk '{print $7}' | awk -F"/" '{print $3}' > liste.site.proxy
#cat liste.sites.proxy | sort | uniq | wc -l => uniq doit être placé après un tri (sort)

Filtrage du débit des accès:
Quand le débit dépasse la taille du buffer défini, il découpe les données en petite partie et les renvois au client en définissant le débit par seconde de ces paquets

Cas n°1
Limite inconditionnelle du débit.

Dans squid.conf
delay_pools 1 => Déclaration d'une entité de filtrage
delay_class 1 1 => Le premier chiffre correspond à la pool, le 2eme la classe du pool
1=Pour tout le monde (totalité des accès au proxy est limité par un seul filtrage général pour les utilisateurs défini par les ACL)

delay_parameters n°_pool val1/val2
val1=débit en octet/s
val2=taille du buffer(réservoir)
delay_access n°_pool allow nom_ACL1 nom_ACL2....

Exemple:
limiter le réseau 100 à un débit maximum (partagé entre tous les postes du réseau 100)
delay_pools 1
delay_class 1 1
delay_parameters 50/50
delay_access 1 allow RESEAU100

Cas n°2
Limitation du débit général ET du débit particulier

delay_pools 2
delay_class 2 2 => Le premier chiffre correspond à la pool, le 2eme la classe du pool

delay_parameters 2 val1/val2 val3/val4 => 2=limitation générale plus une limitation par poste (pour les utilisateurs défini par les ACL)
val1=débit en octet/s | Pour tout le
val2=taille du buffer(réservoir) | monde

val3=débit en octet/s | Pour chaque
val4=taille du buffer(réservoir) | poste
delay_access 2 allow nom_ACL1 nom_ACL2....

Exemple:
Créer un pool qui concernera le réseau 10.0 (réseau comprenant 7 Postes) et:
- Leur attribuer un débit général de 1Mo/s
- Leur attribuer un débit particulier de 150k/s

delay_pools 2
delay_class 2 2
delay_parameters 1000000/1000000 150000/150000
delay_access 1 allow RESEAU10

L'OUTIL SQUIDGUARD

Programme appelé par squid pour filtrer les requêtes http

Fichier à créer à partir d'un exemple:
/etc/squid/squidGuard.conf.sample

dans
dbhome /usr/share/squidGuard-1.2.0/db => répertoire contenant les bases de données de filtrage
logdir /var/log/squidGuard =W Répertoire de log

dans squidGuard.conf

```
time NOMFILTRAGE {
        weekly smtwhfa HH:MM-HH:MM HH:MM-HH:MM
        date AAAA/MM/JJ ....
                        }
src NOMFILTRAGE {
        iplist nom_repertoire_defini_dans_dbhome/nom_fichier_contenant_ip_à_filtrer
ou
        ip      iphost_ou_ipcidr
                        }
dest NOMFILTRAGE {
        domainlist nom_fichier_contenant_dom_à_filtrer      | nécessite une reconstruction
        urllist nom_fichier_contenant_url_à_filtrer         | de la base
        expressionlist nom_fichier_mots_à_filtrer           |

ATTENTION: Obligation de mettre au moins domainlist ou urllist
ou
        domain nom_dom1 nom_dom2...                         |
        url url1 url2...                                    | pas de reconstruction
        expression mot1 mot2...                             |
                }

acl {
        nom_filtrage_source {
                pass nom_filtrage_dest !AUTREFILTRAGE all
                redirect url_en_cas_de_refus
                                }
autre_filtrage_source within FILTRAGE_HORAIRE {         => ou outside
                pass FILTRAGE ... none
                redirect ULR_en_cas_de_refus
                                }
        }
```

exemple de fichier squidGuard.conf

```
dbhome /usr/share/squidGuard-1.2.0/db
logdir /var/log/squidGuard

time TRAVAIL {
      weekly mtwhf 10:00-15:00
           }
src RESEAU100 {
      ip 192.168.10.254
           }
dest GOOGLE {
      url google.fr
           }
acl {
      RESEAU100 within TRAVAIL {
      pass !google all
      redirect http://192.168.10.254
      }
     SERVER {
      pass all
      redirect http://192.168.2.254
           }
      }
```

Indiquer à squid l'utilisation de squidGuard:
dans /etc/squid/squid.conf
redirect_program /usr/bin/squidGuard -c /etc/squid/squidGuard.conf

#mkdir -m 775 /usr/share/squidGuard-1.2.0/db/GOOGLE
#chown squid:squid /usr/share/squidGuard-1.2.0/db/GOOGLE
#vi /usr/share/squidGuard-1.2.0/db/GOOGLE/domains
www.google.fr

Test des filtrages

1. créer un fichier contenant des requêtes
#vi *filtrage*
ch1 ch2/ch3 ch4 ch5

ch1 => URL demandée
ch2 => IP du client
ch3 => domaine du client « - » si pas de domaine
ch4 => nom de l'utilisateur « - » si pas d'utilisateur
ch5 => méthode d'accès à l'URL GET, POST

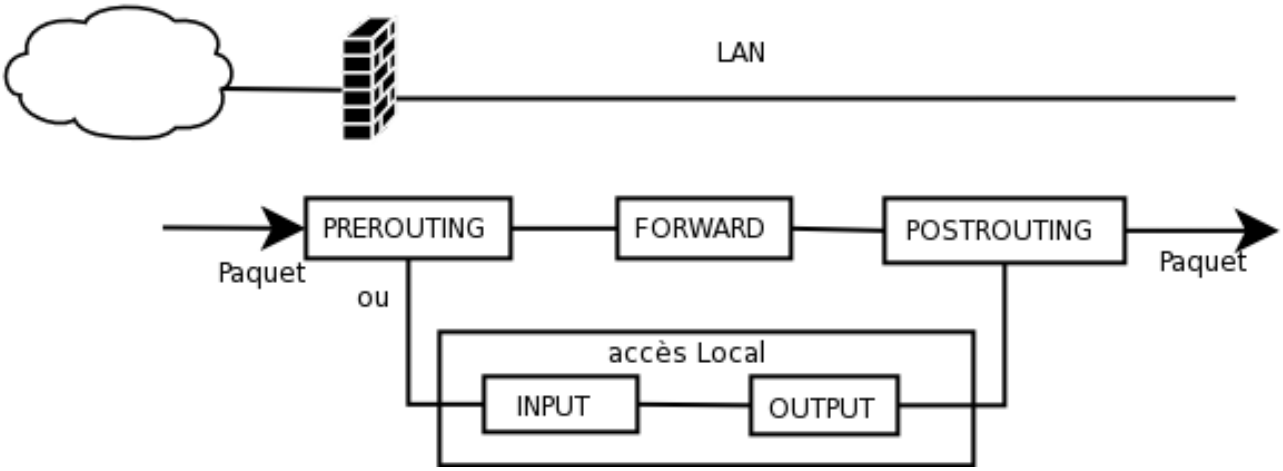
Exemple:
http://www.google.fr 192.168.100.1/- - GET

2. Lancer le test
#squidGuard -c /etc/squid/squidGuard.conf < *filtrage*

Résultat:
- ligne blanche = Requête ok
- URL indiqué en redirection = Requête refusée

LE PARE-FEU IPTABLE

Permet de filtrer les paquets entre deux (ou plus) réseaux



Méthodes de traitement des règles de firewall:

Si le traitement concerne:
- un service sur le serveur : utilise les règles PREROUTING, INPUT, OUTPUT, POSTROUTING
- un service en Forward : utilise les règles PREROUTING, FORWARD, POSTROUTING

Fonctionnement:
Si la première règle correspond => Application de celle ci
si non => Consultation de la suivante
Si aucune règle ne correspond => application de la politique par défaut

ATTENTION: Après avoir appliqué la première règle qui convient, le client s'arrête.

- **Commandes:**

#iptables
#service iptables start|stop|save|panic
save: => Sauvegarde les règles en cours dans /etc/sysconfig/iptables
panic: => Bloque immédiatement tout trafic

- **3 Tables:**
- filter (par défaut) => INPUT, OUTPUT, FORWARD
- nat (network address translation) => PREROUTING, POSTROUTING, OUTPUT
- mangle (travail sur les paquets) => Les 5 chaînes

- **5 chaînes:**
- INPUT => Filtre les paquets à destination du pare-feu lui-même(entrée).
- OUTPUT => Filtre les paquets à en provenance du pare-feu(sortie).
- FORWARD => Filtre les paquets transitant par le pare-feu .
- PREROUTING => Filtre les paquets dont l'adresse de destination sera modifiée.
- POSTROUTING => Filtre les paquets dont l'adresse source sera modifiée.

- **4 politiques (cibles) de base:**
- DROP => paquet rejeté
- ACCEPT => paquet accepté
- REJECT => paquet rejeté (retourné à l'envoyeur avec indication de refus)
- QUEUE => paquet stocké (différé, dépend des modules

- **Autres politiques:**
- RETURN => application de la politique par défaut
- LOG => paquets logués
- DNAT => (Destination) rebalance les paquets vers un proxy
- SNAT => équivalent du secure NAT Microsoft (NAT transparent)
- MASQUERADE => LE NAT

- **Connaître l'état actuel du pare-feu**

#iptables -L -n -t *nom_table*

t => si pas -t la table par défaut est affichée:filter
L => Liste des règles
n => ne traduit pas les ports en nom de protocole et IP en HOST

- **Définition de la politique par défaut associée à chaque chaîne, dans chaque table**

#iptables -t table -P CHAINE POLITIQUE
ex:
#iptables -P INPUT DROP => **ATTENTION**: Bloque tout en entrée et sortie (qd un, l'autre aussi)

- **Construction basique d'une règle de filtrage**

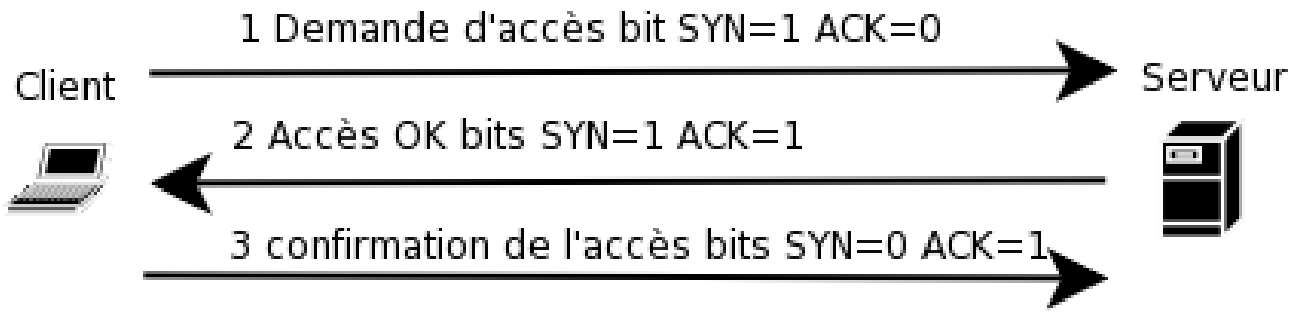
#iptables -t *table* -A *CHAINE* -p *protocole* -s *IP_Source* --sport *port_source* -d *ip_destination* --dport *port_destination* -j *POLITIQUE*

-t => Table concernée
-p => Protocole: TCP UDP ou ICMP
-s => Source: Machine, réseau(CIDR), 0/0=toutes les ips
-d => Destination: Machine, réseau(CIDR), 0/0=toutes les ips
-i => Input interface: eth0, ppp0
-o => Output interface: eth0, ppp0
--dport => port de destination
--sport => Port source
-j => Politique
! => tout sauf : exemple: ! icmp => TCP et UDP

Modules:
-m multiport 53,80 => indique un ensemble de ports
-m mac => permet d'indiquer une adresse MAC avec l'option --mac-source FF:FF:...
-m state => Voir plus bas

Supprimer toutes les règles d'une table
#iptables -t *table* -F
ATTENTION: Cette commande n'affecte pas les politiques par défaut:

Filtrage plus précis du trafic WEB en fonction des bits de status du protocole TCP:



Utilisation du module "state" d'iptables.
Définition des états possibles d'une communication
NEW: => nouvelle connexion (demande de connexion) bit SYN=1 ACK=0
RELATED => Paquet en relation avec une connexion connue bit SYN=1 ACK=1
ESTABLISHED => Paquet en relation avec une connexion établie
INVALID => Paquet dont les bits de status sont incohérents

Mise en place du masquerading (Masquage de réseau ou encore serveur NAT):
#iptables -t nat POSTROUTING -s 192.168.100.0/24 -i eth1 -o eth0 -j MASQUERADE

L'utilisation de la chaîne PREROUTING pour la gestion d'une DMZ (DeMilitarized Zone) :
Correspond à du NAT inverse:
#iptables -t nat -A PREROUTING -p tcp -d 194.1.2.3 --dport 80 -j DNAT --to-destination 192.168.10.2:80

ou 194.1.2.3 est l'ip publique du firewall

Il ne reste plus qu'à l'accepter dans la chaîne FORWARD
#iptables -A FORWARD -p tcp -d 192.168.10.2 --dport 80 -j ACCEPT
#iptables -A FORWARD -p tcp -s 192.168.10.2 --sport 80 -j ACCEPT

ou 192.168.10.2 est le serveur WEB en interne à attendre

Remise à zéro des compteurs de trafic:
Visualisation du trafic iptables:
#iptables -L -t table -v

Remise à zéro des compteurs de trafic:
#iptables -L -t table -Z

Tracer des paquets spécifiques:
#iptables -A FORWARD -s ip_réseau_privé --dport 80 -j LOG
#iptables -A FORWARD -s ip_réseau_privé --dport 80 -j ACCEPT

Règle de sécurité à indiquer dans iptables:
Refuser les paquets arrivant sur notre carte réseau côté publique Dont l'ip source prétend être dans notre réseau privé
#iptables -A FORWARD -s ip_reseau_privé -i carte_reseau_publique -j DROP

Exemples

Refuser l'accès à notre serveur WEB pour tous les clients ne provenant pas du réseau 10.0

Solution 1:
#iptables -A INPUT -p tcp -s 192.168.10.0/24 -d192.168.10.254 --dport 80 -j ACCEPT
#iptables -A INPUT -p tcp -d192.168.10.254 --dport 80 -j DROP

Solution 2:
#iptables -A INPUT -p tcp -s ! 192.168.200.0/24 -d192.168.10.254 --dport 80 -j DROP
ATTENTION: Mettre un espace entre ! Et l'adresse IP

interdire l'accès depuis le réseau 200.0 à mes serveurs telnet, smtp, pop, imap, DNS, DHCP
#iptables -A INPUT -p tcp -m multiport -s 192.168.200.0/24 -d 192.168.10.254 --dport 23,25,53,67,110,143 -j DROP

Filtrer avec l'état de connexion :
règle en sortie:
#iptables ..... -m state --state NEW,RELATED,ESTABLISHED
règle en sortie:
#iptables ..... -m state --state RELATED,ESTABLISHED

Scénario:
- Politique par défaut à DROP
- Autoriser le trafic web par IP à destination du réseau 200.0

iptables -A FORWARD -p tcp -m multiport -d 192.168.200.0/24 --dport 80,443 -s 192.168.10.0/24 -j ACCEPT
iptables -A FORWARD -p tcp -m multiport -s 192.168.200.0/24 --sport 80,443 -d 192.168.10.0/24 -j ACCEPT

Créer un fichier de configuration de iptables:

vi firewall

```
#!/bin/bash
#Bloque tout dans la table filter (politique par défaut)
iptables -t filter -P INPUT DROP #ne permet plus d'entrer dans le serveur
iptables -t filter -P OUTPUT DROP # ne permet plus de sortir du serveur
iptables -t filter -P FORWARD DROP #ne permet plus le routage de paquets

#bloque tout dans la table nat
iptables -t nat -P PREROUTING ACCEPT
iptables -t nat -P POSTROUTING ACCEPT
iptables -t nat -P OUTPUT ACCEPT

#suppression des règles actuelles
iptables -t filter -F
iptables -t nat -F

#Permet d'indiquer que les services locaux puissent fonctionner
iptables -t filter -A INPUT -d 127.0.0.1 -j ACCEPT
iptables -t filter -A OUTPUT -d 127.0.0.1 -j ACCEPT

#autoriser le poste en tant que serveur telnet:
iptables -A INPUT -p tcp -d 192.168.10.7 --dport 23 -j ACCEPT
iptables -A OUTPUT -p tcp -s 192.168.10.7 --sport 23 -j ACCEPT
```

LA CRYPTOGRAPHIE

- **Paquetages:**
  - openssl
  - libopenssl

- **But :**
  - assurer la confidentialité des données
  - assurer l'authentification d'utilisateurs, de sites, de machines

- **Méthodes de cryptage:**

  - Algorithmes à clé symétrique (la même clé pour crypter et décrypter)
  - Algorithme à clé asymétrique : clé publique /clé privé (cryptage avec l'une => selle l'autre peut décrypter.)
  - Générateurs n'emprunte (Hachage de mots de passe)=> processus non réversible

  Il faut la chaîne non cryptée pour essayer de décrypter les données

Exemple de chiffrage à clé symétrique: Méthode de César:

L'algorithme décale les lettres de l'alphabet. La clé correspond au nombre de lettres de décalage
EX: Bonjour => Clé de 2 lettres => Dqplqwt

Faiblesse: on vois les lettres françaises les plus utilisées et on essai de trouver la correspondance

Exemple soviétique : donne lettre page dans un livre donné et on reconstruit le message :

27.105 21.109.... 27eme caractère page 105 21eme car page 109 ...
La clé est le livre utilisé

- **Utilisation basique d'une clé symétrique:**

1. Choisir/créer un document à crypter
2. Le crypter en choisissant un algorithme et une clé

#openssl algorithme -in doc_à_crypter -out fichier.crypté -k chaîne_servant_de_clé

Exemple:

#openssl rc4 -in text.txt -out text.rc4 -k gateau
décryptage (-d):
#openssl rc4 -d -in text.rc4 -out text.decrypté -k gateau

- **Utilisation des clés asymétriques:**

Signature Numérique = authentification

Compta 1				Compta2
Clé privée	---------->	message crypté avec la clé privée de Compta1	---------->	Décrypte le message avec la clé publique de compta1
Clé publique				

Les données ne sont pas sécurisées car tout le monde peut récupérer la clé publique de Compta1 mais le message décrypté avec sa clé publique ne peut avoir été crypté qu'avec la clé privée (donc Compta1

Enveloppe Numérique = cryptage

Compta 1				Compta2
Clé privée	<----------	message crypté avec la clé publique de Compta 1	<----------	crypte le message avec la clé publique de compta1
Clé publique				

Exemple:

1. **génération d'une paire de clé privée/publique**

#openssl genrsa -out fichier_clé_privée rsa 512 => 512 est la taille en bits de la paire de clés

2. **Extraction de la clé publique à partir du fichier de clé privée:**

#openssl rsa -in fichier_clé_privée -pubout fichier_clé_publique

3. **cryptage d'une chaîne de caractères avec la clé publique.**
#echo "chaîne" | openssl rsautl -inkey clé_publique -pubin -encrypt -out fichier_codé

4. **Décryptage avec la clé privée:**
#openssl rsautl -inkey clé_privé -decrypt -in fichier_codé

REMARQUE: marche qu'avec 4 caractères maximum???

- **Générateur d'empreinte (hachage):**

#openssl passwd -crypt linux
#openssl passwd -1 linux => méthode utilisée sur linux pour les mots de passe
résultat dans /etc/shadow:
1*graine*$*mot_de_passe_crypté* (le 1 indique crypté)

LE SERVICE SSH

Permet le login sur les postes distants en liaison cryptée (asymétrique) et permet l'exécution de commandes sur des postes distants

- **Port:** TCP 22

- **Fichiers:**

/ect/ssh/sshd_config => serveur
/ect/ssh/ssh_config => client

3paires de clés pub/priv (pour RSA1,RSA2,DSA)identifiant la machine:
/etc/ssh/ssh_host_key | Clés RSA1
/etc/ssh/ssh_host_key.pub |

/etc/ssh/ssh_host_rsa_key | Clés RSA2
/etc/ssh/ssh_host_rsa_key.pub |

/etc/ssh/ssh_host_dsa_key | Clés DSA2
/etc/ssh/ssh_host_dsa_key.pub |

- **Commandes:**

#service sshd start | status | ...
#ssh => Login et ou exécution à distance
-l nom_user ip_ou_host_srv => indique le nom du compte à utiliser en connexion ssh
-v => verbose (peut mettre deux ou trois v pour plus d'infos)
#ssh -keygen => Génération des clés
#scp => copie d'un fichier

Etablissement d'un canal SSH:

Client SSH ----------1. demande de connexion (négociation des protocoles)--------> **Serveur SSH**
<---------2. OK mais en crypté (envoi clé publique SRV)---------- ------ **SRVpub/priv**
3. Stockage de la clé publique dans /etc/ssh/known_hosts
----------4. Entente sur une clé symétrique générée à ---------->
<----------partir d'un fragment de la clé publique ----------

Dans le répertoire .ssh en 700
/home/compta1/.ssh/clépub
/home/compta1/.ssh/clépriv

/home/tech1

----------5. signature numérique pour l'utilisateur compta1------> /.ssh

/autoriezd_keys2
/clépub compta1

<-----6. envoloppe numérique crypté avec clé pub de Compta1-------
----------7. renvoi le message au serveur---------->

<---------- 8. Si pas de clé : authentification par login -------->

Session ssh avec authentification par signature numérique du client:

Côté client:
1. Générer un couple de clé pub/priv avec l'utilisateur
#ssh-keygen -t rsa => RSA2
2. Envoyer la clé publique au serveur

Coté serveur
1. créer un répertoire .SS dans le répertoire d'accueil de l'utilisateur
#mkdir -m 700 /home/tech1/.ssh
2. créer un fichier authorized_keys2 dans le répertoire précédent
#touch /home/tech1/.ssh/authorized_keys2
3. Copier la clé publique de l'utilisateur dans ce fichier

Les fichiers de configuration du serveur:
/etc/ssh/sshd_conf

ATTENTION: Les commandes en # indiquent la configuration par défaut prévu à la compilation du programme par ssh

PermitRootLogin yes|no
AuthorizedKeysFile .ssh/authirized_keys => suivant distribution peut être keys2
PamAuthenticationViaKbdInt yes | no => UsePAM sous Debian permet ou non l'utilisation
PasswordAuthentication yes | no => d'un mot de passe à la place des clés

Le fichier de configuration du client
/etc/ssh/ssh_config
IdentifyFile nom_fichier_clé_privée => plusieurs lignes possibles

Exécuter des commandes à distance sans ouverture de session:
#ssh -l nom_user nom_serveur_ssh commande_a_executer_sur_serveur

Copier un fichier entre le client et le serveur :
#scp nom_fichier_local user@machine => copié par défaut dans son répertoire d'accueil
#scp user@machine:nom_fichier_distant nom_fichier_local
#scp nom_fichier_local user@machine:nom_fichier_distant nom_fichier_local

Utilisation des tunnels SSH
un tunnel crypté pour un service donné défini par son port (redirection de ports)
#ssh -L n°_port_local:nom_serveur:n°_port_distant

On souhaite avoir sur le port 11000 le port 80 du serveur distant, ceci afin de sécuriser la connection entre les 2 postes.
#ssh -l user nom_serveur –L 11000:nom_pc:80

le trafic entre le port 80 du serveur distant et le port 11000 local sera crypté de manière à ce qu'aucune personne ne puisse y accéder.

pour accéder à cette ressource, il faut utiliser : http://localhost:11000

exemple pour accéder au serveur de messagerie 192.168.200.2

sur le client :
il faut un client ssh ainsi qu'un client pop
Dans Kmail :
il faut définir le serveur localhost ainsi que le port 11000
dans la ligne précommande qui sera exécutée avant la connection, il faut y inscrire :
#ssh -l nom_user -C -f nom_serveur -L 11000:nom_serveur:port_serveur sleep 20
-C ==> compression lors du transfert
-f ==> fork, permet le dédoublement, il va exécuter 2 commandes en même temps (la commande ssh et les commandes liées au protocole pop)
sleep 20 ==> commande laissant le tunnel ouvert pendant 20s

sur le serveur :
il doit y avoir un serveur ssh ainsi qu'un serveur smtp et pop

LES CERTIFICATS X509

protocoles :
SSL (avant) => développé par Netscape
TLS (maintenant) => a racheté les droits de conceptions de SSL

exemple à travers la sécurisation d'un site web
Protagonistes :
- la société qui souhaite faire authentifier son site sécurisé
- l'autorité de certification (CA), Verisign, Thawte, ... s'engage sur l'exactitude des coordonnées fournies par la société gérant le site sécurisé.
- le client, qui peut vérifier auprès du CA la validé du site auquel il accède.

Coté société

1/ Créer le site web à sécuriser
commonhttpd.conf
<Directory /var/www/html/banque>
config avec accès par mdp
</Directory>

2/ Donner un nom à ce site et le renseigner au niveau DNS

3/ Déclarer ce site comme étant sécurisé
/etc/httpd/conf.d/41_mod_ssl.default-vhost.conf
<VirtualHost _default_:443>
DocumentRoot /var/www/banque
</VirtualHost>

4/ Dé configurer les sites virtuels existants

5/ Revalider l'appel au site virtuel sécurisé
Pour configurer l'accès sous Debian, voir http://www.destination-linux.org/article30.html

6/ Générer la demande de certificat
la commande suivante va nous permettre de créer un nouveau certificat :
cd /etc/ssl
/usr/lib/ssl/misc/CA.pl -newreq
il suffit ensuite de répondre à toutes les questions posées.
/!\ Common name : Nom FQDN complet du site sécurisé
Créer le fichier newreq.pem qui contient .la demande de certificat
.la clé privée

7/ faire parvenir cette demande (ce fichier) au CA

8/ récupérer le certificat signé par le CA (fichier newcert.pem)

9/ il faut ensuite dire à apache de prendre en compte le nouveau certificat et plus celui créé lors de l'installation du serveur.

10/ supprimer la clé privée et le certificat actuels
#rm -f /etc/ssl/apache/server.key /etc/ssl/apache/server.crt

11/ extraire la partie "clé privée" du fichier newreq.pem et la copier dans /etc/ssl/apache/server.key

12/ extraire la partie "certificat" du fichier newcert.pem et la copier dans /etc/ssl/apache/server.key

13/ suppression de la pass phrase protégeant la clé privée (car si elle existe, elle est demandée au démarrage du service apache)
#cp server.key server.key.nopass
#openssl rsa -in server.key.nopass -out server.key

Coté client

1/ accéder au site sécurisé et constater qu'il n'est pas connu des C.A du navigateur.

solution 1:
accepter quand même le certificat du site sécurisé
/!\ Seul le site sécurisé sera connu, pas le CA (cf config navigateur / certificats)

solution 2:
refuser de faire confiance au site et au CA

solution 3:
déclarer le CA dans la liste des CA connus, afin de faire confiance à l'avenir à touts les sites authentifiés par ce CA.

Devenir CA(Autorité de certification)
cd /etc/ssl
/usr/lib/ssl/misc/CA.pl -newca

signer une demande de certificat
cd /etc/ssl
/usr/lib/ssl/misc/CA.pl -signreq
/!\ nécessite que la demande de certificat soit dans /etc/ssl sous le nom newreq.pem
Cela crée le fichier newcert.pem à renvoyer à la société demanderesse.
ajoute les coordonnées de la société au fichier demoCA/index.txt

LE SERVICE LDAP

Lightweight Directory Access Protocol

Paquetages à installer:

#rpm -qa | grep ldap
perl-ldap
openldap-2.

Openldap-clients..

nss_ldap

openldap-migration

libldap2-devel...
libldap2-2....

openldap-servers....

pam_ldap-.....
openldap-guide-2....

Ports :

TCP 389 (LDAP)
TCP 636 (LDAP sécurisé)

Fichiers

/etc/openldap/slapd.conf	=> Configuration du serveur
/etc/openldap/ldap.conf	=> Configuration du client (déclaration de l'accès au serveur)
/etc/ldap.conf	=> Configuration pour l'authentification du client
/etc/nsswitch.conf	=> Résolution des noms LDAP
/var/lib/ldap	=> Répertoire qui contient les composants de l'annuaire

Commandes :

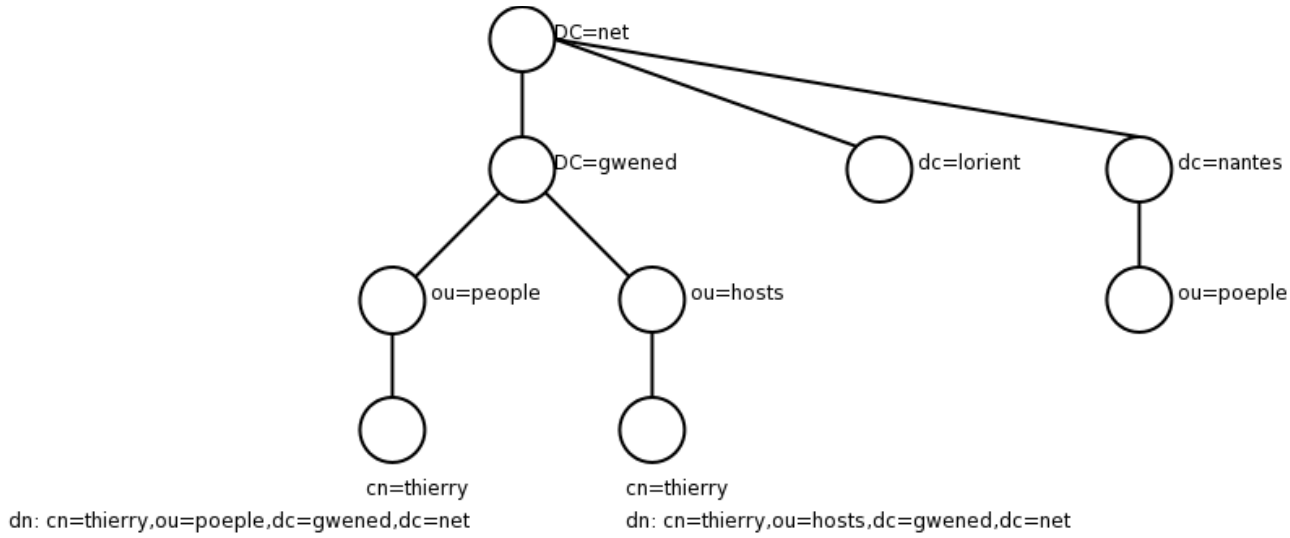
#service ldap start | stop | restart....

#ldapadd	=> ajouter un enregistrement
#ldapsearch	=> rechercher un enregistrement
#ldapmodify	=> Modifier un enregistrement
#slapcat	=> Visualisation
#slapadd	=> Modification

Composants d'un annuaire :

DIT:	Directory Information Tree (Base organisée en arbre)
Naming Context :	Suffixe de la base (ex: DC=linux,DC=net)
DSE :	Directory Service Entry : Enregistrement dans l'annuaire
DN :	Distinguished Name Identifie de manière unique un enregistrement ldap
RDN :	Relative Distinguished Name : Nom relatif de l'entrée par rapport au suffixe
Attributs :	composent l'enregistrement
OU:	Organisational Unit : Conteneur d'enregistrement
Schema:	Entité comprenant des attributs et conteneurs spécifiques permettant de composer des enregistrements en fonction de leur rôle dans l'annuaire

Structure de l'arborescence LDAP:



Mise en place d'un annuaire LDAP:

/etc/openldap/slapd.conf

(74) suffix 'dc=tit,dc=fr" => déclaration du nom de la base
(76) rootdn "cn=manager,dc=tit,dc=fr" => Déclaration de l'administrateur
(82) rootpw *mot_de_passe* => Donne password root (en clair pour l'instant)
(88) allow bind_v2 => Autorise les requêtes client en version 2 du protocole LDAP

ACL de base à partir de la ligne 98:

Modifier la ligne 102 pour indiquer la base

```
98      #Basic ACL
99      access to attr=userPassword
100     by self write
101     by anonymous auth
102     by dn="cn=manager,dc=tit,dc=fr" write
103     by * none
104
105     access to *
106     by dn="cn=manager,dc=tit,dc=fr" write
107     by * read
```

Créer un fichier au format LDAP décrivant les deux enregistrements mentionnés:

- Un enregistrement pour le nom de la base
- un enregistrement pour le nom de l'administrateur

Créer un fichier ldabe base.ldap contenant les données suivantes dans /var/lib/ldap

dn: dc=tit,dc=fr

objectClass: top

objectClass: domain

objectClass: domainRelatedObject

associatedDomain: dc=tit,dc=fr

dc: tit

dn: cn=Manager,dc=tit,dc=fr

objectClass: top

objectClass: organizationalRole

cn: Manager

Lancer la commande suivante:

#ldapadd -x -W -D "cn=manager,dc=tit,dc=fr" -f base.ldif

-x => authentification simple (non-ssl)
-W => Invite pour le mot de passe

Rechercher un enregistrement dans la base:

#ldapsearch -x -W -D "cn=managmen,dc=tit,dc=fr" -b "dc=tit,dc=fr"

Ajouter une OU:

dans /var/lib/ldap/ou.ldif

dn: ou=hosts,dc=tit,dc=fr

ou: hosts

objectClass: top

objectClass: organizationalUnit

objectClass: domainRelatedObject

associatedDomain: dc=tit,dc=fr

#cd /usr/share/openldap/migration
#./migrate_base.pl /var/lib/ldap/newbase.ldif

#cd /var/lib/ldap

/!\ supprimer le 1er enregistrement du fichier newbase.ldif (déjà crée précédemment)

#ldapadd ..... -f newbase.ldif

Configurer l'outil de migration:
/usr/share/openldap/migration/migrate-common.ph

71 $DEFAULT_MAIL_DOMAIN = "tit.fr";
74 $DEFAULT_BASE = "dc=tit,dc=fr";
86 $DEFAULT_MAIL_HOST = "mail.tit.fr";
90 $EXTENDED_SCHEMA = 1;

Le répertoire /usr/share/openldap/migration contient des exemples de scripts de migration:

fstab, /etc/hosts.....

Pour migrer les enregistrements de /etc/hosts dans la base on utilise le script migrate_hosts:

#cd /usr/share/openldap/migration => doit être le répertoire courant pour include dans le migrate_hosts.pl

#/usr/share/openldap/migration/migrate_hosts.pl /etc/hosts > hosts.ldif

ensuite on enregistre le fichier dans la base
#ldapadd -x -W -D "cn=manager,dc=tit,dc=fr" -f hosts.ldif

Migration des comptes utilisateurs :
ETC_SHADOW=/etc/shadow

./migrate_passwd.pl /etc/passwd > /var/lib/ldap/users.ldif

ldapadd ...

Migrer les groupes, services, montages avec le script associé

/etc/group | /etc/services | /etc/fstab
migrate_group.pl | migrate_services.pl | migrate_fstab.pl

Remplacer l'administrateur actuel (cn=Manager) par root
#vi /etc/openldap/slapd.conf
:%s/cn=Manager/uid=root,ou=People/

Installer gq pour accéder au serveur LDAP par un GUI

pour le configurer ==>File-> Preferences-> Server ->General->Details

Configuration du client LDAP

1- Indiquer le nom du serveur et le nom de la base

/etc/openldap/ldap.conf
BASE dc=tit,dc=fr

host @IpserveurLDAP

2- Indiquer l'appel à LDAP pour la résolution des noms

/etc/nsswitch.conf
passwd: files ldap

shadow: files ldap

group: files ldap

hosts: files dns ldap

3- faire appel aux bibliothèques LDAP dans les modules PAM

nécessite pam_ldap_167
SAUVEGARDER LE REPERTOIRE /etc/pam.d ACTUEL

copier le répertoire pam.d fourni par le paquetage pam_ldap_167 dans /etc
#cp -r /usr/share/doc/pam_ldap_167/pam.d /etc

4- tester en supprimant l'utilisateur (lui laisser son home) compta1

on supprime les lignes des fichiers, ne pas passer par userdel car il va aussi passer par LDAP.

5- ouvrir une session avec compta1

ATTENTION: Différences entre les modules PAM-LDAP et PAM-Linux:

/etc/pam.d/login (de LINUX)	/etc/pam.d/login (de LDAP)
session optional /lib/security/pam-console.so	la ligne a un # a supprimer pour autoriser startx

Cohabitation entre les annuaires openldap et Active Directory

CAS 1: Fonctionnement entre deux annuaires indépendants
Les deux annuaires openldap et AD sont utilisées en redirection sur l'autre si la première ne connais pas l'information demandée (pas de synchronisation entre les deux)

1. **Configuration des outils smbldap :**

/etc/samba/smbldap_conf.pm

Ligne 67 $UIDstart=10000;

Ligne 68 $GIDstart=10000;

Ligne 72 $SID=numero_SID_recuperé_avec_net_getlocalsid;

Ligne 94 $masterLDAP=ip_serveur_ldap;
Ligne $masterPort=389;
Ligne 104 $suffix="dc=tit,dc=fr";

Ligne 109 $userou=q(people);

$computersou=q(computers);

$groupsou=q(group)

$binddn="cn=manager,$suffix";

à modifier:$binddn="uid=root,ou=people,$suffix);

$bindpasswd="password => mot de passe root

2. Copier la valeur (ligne 72) dans /etc/samba/smbldap_conf.pm

3. Pour récupérer le SID samba
#net getlocalsid

4. Vérifier la bonne configuration des outils
#smbldap -usershow
#smbldap -useradd nom_user

5. Créer le groupe ad avec le gid=10000

6. Créer un compte user pour samba
#smbldap-usermod -a nom_user_existant
ou
#smbldap-useradd -a nouveau_nom_user

7. Créer un password pour ce compte
#smbldap-passwd

8. Configuer smb.conf

(20) workgroup=nom_domaine_netbios_AD
(84) security=domain
(88) password server=ip_ou_host_serveur_AD
realm=domaine_kerberos
(197) domain logons=no
(231) add user script=/usr/bin/smbldap-useradd -a %u
(280) ldap admin dn=uid=root,ou=people,dc=tit,dc=fr
(281) ldap ssl=off
(283) ldap port=389
(284) ldap suffix=dc=tit,dc=fr

9. Installer le client Kerberos

```
#urpmi krb5-workstation
#vi /etc/krb5.conf
[libdefaults]
      default_realm = SAMPLEDOMAIN.INVALID
[realms]
      SAMPLEDOMAIN.INVALID = {
            kdc = controller.sampledomain.invalid:88
               admin_server = 192.168.200.1:749
      }
[domain_realms]
      .controller.sampledomain.invalid = SAMPLEDOMAIN.INVALID
```

Tester Kerberos
#kinit Administrator@SAMPLEDOMAIN.

10. Créer la structure générale de la base (les ou=conteneurs)

#cd /usr/share/openldap/migration
#./migrate_base.pl /var/lib/ldap/newbase.ldif
#cd /var/lib/ldap

/!\ supprimer le 1er enregistrement du fichier newbase.ldif (déjà crée précédemment)

#ldapadd ..... -f newbase.ldif

Migration des comptes utilisateurs
ETC_SHADOW=/etc/shadow
./migrate_passwd.pl /etc/passwd > /var/lib/ldap/users.ldif
ldapadd ...

Migrer les groupes, services, montages avec les scripts associés
/etc/group | /etc/services | /etc/fstab
migrate_group.pl | migrate_services.pl | migrate_fstab.pl

Chercher des informations dans l'annuaire
#ldapsearch -x -W -D “cn=Manager,dc=jojo-ifc8,dc=net” -b “dc=jojo-ifc8,dc=net”

Remplacer l'administrateur actuel (cn=Manager) par root
#vi /etc/openldap/slapd.conf
:%s/cn=Manager/uid=root,ou=People/

Installer gq pour accéder au serveur LDAP par un GUI
pour le configurer ==>
File -> Preferences -> Server -> General -> Details

Configuration du client LDAP

1- Indiquer le nom du serveur et le nom de la base
/etc/openldap/ldap.conf
BASE dc=jojo,dc=net
host @IpserveurLDAP

2- Indiquer l'appel à LDAP pour la résolution des noms
/etc/nsswitch.conf
passwd: files ldap
shadow: files ldap
group: files ldap
hosts: files dns ldap

3- faire appel aux bibliothèques LDAP dans les modules PAM
nécessite pam_ldap_167
SAUVEGARDER LE REPERTOIRE /etc/pam.d ACTUEL
copier le répertoire pam.d fourni par le paquetage pam_ldap_167 dans /etc
#cp -r /usr/share/doc/pam_ldap_167/pam.d /etc

4- tester en supprimant l'utilisateur (lui laisser son home) compta1
on supprime les lignes des fichiers, ne pas passer par userdel car il va aussi passer par LDAP.

5- ouvrir une session avec compta1

LE SERVICE MYSQL

Système de gestion de base de données = SGBD

Port TCP:
3306

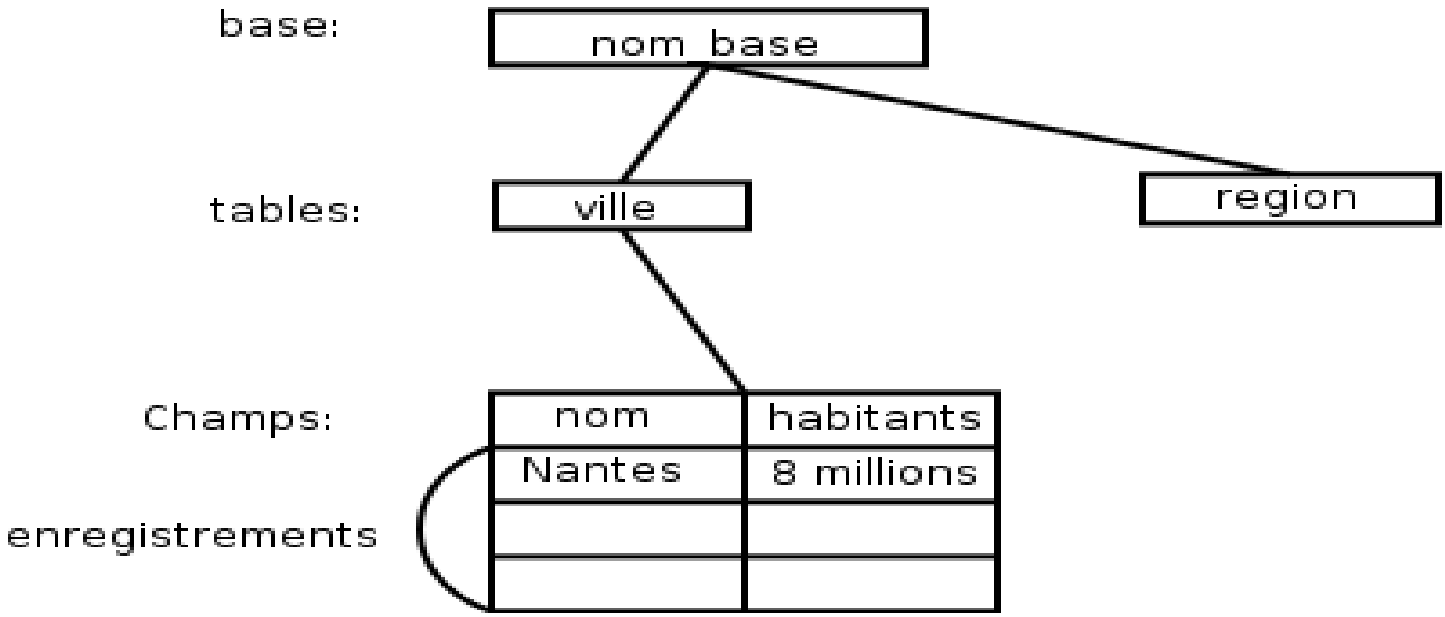
Paquetages:
MySQL
MySQL-common
MySQL-client
libmysql12

Commandes:
#mysql => Connexion au service
#mysql admin => Administration des comptes
#mysqlrepair => Outil de réparation
#mysqldump => Sauvegarde
#mysqlimport => Restauration/importation des fichiers
#myisamchk => Outil de réparation/test

Fichiers:
/var/lib/mysql => Répertoire des bases de données
/var/lib/mysql/*nom_base/table.MYD* => Données de la table
/var/lib/mysql/*nom_base/table.MID* => Index et clé de la table
/var/lib/mysql/*nom_base/table.frm* => Structure de la table

ATTENTION: Le répertoire et son contenu doivent être accessibles en lecture et écriture à l'utilisateur mysql

Organisation d'une base:



CRÉER UNE BASE DE DONNÉES

1. **Connexion au service**
#mysql

2. **Création**
>CREATE DATABASE *nom-base*; => ; indique la fin de la commande

3. **connexion à la base**
>CONNECT *nom_base*;

Commandes de connexion:
#mysql -u *nom_user* -h *nom_ou_ip_mysql* -p
-u => donne un utilisateur
-h => donne un serveur mysql
-p => demande de password au client

Créer une table:
Pour chaque champ décrivant la table, il faut préciser:
- le type de données enregistrées dans le champ
- La longueur du champ

Types:

INT	=> Nombre entier écrit sur 4 octets = 0 à 2 puissance 32
SMALLINT	=> Nombre entier écrit sur 2 octets = 0 à 65535 (+127 à -127)
TINYINT	=> Nombre entier écrit sur 1 octets = 0 à 255
BIGINT	=> Nombre entier écrit sur 8 octets = 0 à 2 puissance 64
INT(n)	=> Nombre entier écrit sur n caractères
FLOAT	=> Nombre réel décimal sur 4 octets
CHAR(n)	=> Chaîne de n caractères (bloque les n emplacements)
VARCHAR(n)	=> Champ de n caractères (ne prend que les données fournies)
TEXT	=> Champ de 255 octets
ENUM(val1,val2)	=> Champ ou la donnée écrite sera forcément val1 et val2

4.création de la table
> CREATE TABLE *nom_table (champ1 type(longueur), champ2 type(longueur)..)*;

5. Lister les tables d'une base:
>SHOW TABLES;

exemple:
>create table linux (nom CHAR(15),prenom CHAR(15),age INT(2),cours VARCHAR(6));

6. Ajouter un enregistrement à la base:
>INSERT INTO *nom_table* VALUES ('champ1','champ2',....);
ex:
>insert into linux values ('Plantive','Thierry','31','SRL01');

7. Visualiser le contenu de la table:
>SELECT * FROM *nom_table*

8. Visualiser les champs:
>desc *nom_table* => affiche les champs de la table (= description)

9. Créer un fichier pour importer des données dans la base:
le séparateur de champ sera le ":"
Troutrou:bob:34:SRL01
Boudboul:toto:12:SRL02

L'importer dans la table:
>LOAD DATA LOCAL INFILE "*fichier*" INTO TABLE *nom_table* FIELD TERMINATED BY ':';

Indique par 'FIELD TERMINATED BY', le séparateur de champs

Rechercher des infos dans la table:
>SELECT *champ1,champ2,...* FROM *nom_table* WHERE c*lause_de_recherche*;

clauses de recherche:

champ="valeur"	=> champ contient la valeur
champ>"valeur"	=> champ Supérieur à la valeur
champ<"valeur"	=> champ inférieur à la valeur
champ>="valeur"	=> champ Supérieur ou égal à la valeur
champ<="valeur"	=> champ inférieur ou égale à la valeur
champ!="valeur"	=> Champ différent de la valeur
champ LIKE "chaîne%"	=> Champ commence par chaîne
champ LIKE "%chaîne"	=> Champ fini par chaîne
champ LIKE "%chaîne%"	=> Champ contient chaîne

champ REGEXP "expr_reg" => Utilise les expressions régulières
>SELECT COUNT(*) => Donne le nombre d'enregistrements trouvés
>SELECT MAX(champ)

exemple:
>SELECT nom_table,MAX(champ) FROM nom_table
>SELECT MIN()
>SELECT SUM() => Donne la somme
>SELECT AVG() => Moyenne
>SELECT *champ1,champ2* FROM *nom_table* ORDER BY champ2 DESC LIMIT 1;
ORDER BY =>permet de trier
- ASC => Trier par ordre ascendant
- DESC => Trier par ordre descendant
LIMIT 1 => indique le nombre d'enregistrements à retourner

10. Suppression d'un enregistrement:

>DELETE FROM *nom_table* WHERE *nom_champ="valeur"*;

11. Modification d'un enregistrement

>UPDATE *nom_table SET* champ="*nouvelle_valeur*" WHERE *clause_recherche*

12. Suppression d'une table

>DROP TABLE nom_table;

13. Modification d'une table

- Ajout d'un champ:

>ALTER TABLE *nom_table* ADD n*ouveau_champ*;

- Mise en place d'une clé dans un champ
  la clé permet d'identifier de manière sûre, l'unicité de l'enregistrement
  Souvent, la clé primaire est une valeur auto incrémentée dans un champ dédié id

>CREATE INDEX *nom_index* ON *nom_table(champ)*;

- Suppression d'un index

>DROP INDEX *nom_index* ON *nom_table*;

- Ajouter un champ auto incrémenté , indexé servant de Clé primaire:

>ALTER TABLE *nom_table* ADD *champ_clé* INT NOT NULL AUTO_INCREMENT FIRST, ADD INDEX(*champ_clé*)

14. Sauvegarde de tout ou partie d'un table dans un fichier

>SELECT * FROM *nom_table* INTO OUTFILE "*nom_fichier*" FIELDS TERMINATED BY ':';

15. Sauvegarde de la base entière:

Méthode 1:
Utiliser mysqldump pour sauvegarder la base sous forme fichier.
#mysqldump nom_base -u nom_user -p password > /tmp/sauvegarde.sql
pour restaurer :
#mysqlimport < fichier.sql

ATTENTION : prend du temps CPU et disque car entre les commandes l'une après l'autre.

Méthode 2:
Sauvegarde du répertoire correspondant à la base: /var/lib/mysql
on décompresse la base avec la commande TAR au bon endroit pour restaurer

Méthode 3:
sauvegarder les données table par table (bof bof)
pour restaurer: >LOAD DATA..

Administration des utilisateurs:

1. Création d'un utilisateur :
Donner la permission "usage" à un utilisateur:
>GRANT USAGE ON *nom_base.nom_table* TO *nom_user* IDENTIFIED BY '*motpass*';

nom_base.nom_table	=> Pour la table donnée de la base donnée
base.*	=> Pour toutes les tables de la base
.	=> Pour toutes les tables de toutes les bases
user	=> se connecte depuis partout sauf localhost
user@nom_ou_ip	=> Se connecte depuis la machine indiquée
user@'%'	=> depuis partout y compris localhost

ATTENTION: Après cette commande il n'a que le droit de se connecter à la base.

2. Affectation de droits:
Pour voir les droits d'un utilisateur:
>SHOW GRANTS FOR *nom_user*;
ou modification de la table user dans la base MYSQL

Les Droits:
>GRANT *droits* ON *nom_base.nom_table*
- USAGE
- SELECT
- SELECT,UPDATE
- SELECT,UPDATE,INSERT
- SELECT,UPDATE,INSERT,DELETE
- SELECT,UPDATE,INSERT,DELETE,ALTER
- SELECT,UPDATE,INSERT,DELETE,ALTER,DROP
- ALL PRIVILEGES (tous les droits sauf GRANT)
- GRANT (Permet de donner des droits à des droits utilisateurs)
- SHUTDOWN

ATTENTION: si on donne plusieurs droits, c'est le plus permissif qui gagne!!!

on peut gérer champ par champ mais dans ce cas on n'utilise pas les options ci-dessus car c'est le plus permissif qui remporte.
>GRANT *droit(champ1,champ2)*,*autre_droit(champ4,champ5*) ON *nom_base.nom_table;*

3. Retrait des droits:
>REVOKE *droits* ON *base.table* FROM *nom_user*
ATTENTION: la suppression des droits d'un utilisateur ne supprime pas son compte

4. Modifier le mot de passe d'un utilisateur:
#mysqladmin -u *nom_user* -p

5. Suppression d'un compte utilisateur

6. Forcer la relecture de la table des privilèges (user dans mysql):
>FLUSH PRIVILEGES;

pour supprimer le champ 1 d'un fichier et le remplacer par un champ vide:
#awk -F: 'OFS=":"{print $2,$3,$4,$5,$6,$7,$8}' /tmp/table.villes.93 | sort | uniq | sed 's/3/:&/' > /tmp/93

Exercice :
dans une table régions et villes, je doit trouver les 10 départements les – peuplés en utilisant une variable dynamique minipop:
>SELECT dp,SUM(popu) AS minipop FROM villes GROUP BY dp ORDER BY minipop LIMIT 10;

LE SERVEUR X

Développé par le consortium X-Windows

Définitions:

- Serveur X: Ensemble matériel carte+écran+clavier+souris+applications serveur X
- Terminal X=Serveur X dédié: Terminal passif graphique
- Client X:
  - Application graphique (Konqueror, xclock,...)
  - Gestionnaire des fenêtres (disposition des fenêtres, Déplacement de la souris,....)
    Ex: fvwm,kde,gdm....
  - Bureau=environnement : ensemble cohérent d'applications graphiques
    Ex: KDE,Gnome,icewm

Configuration du serveur X:

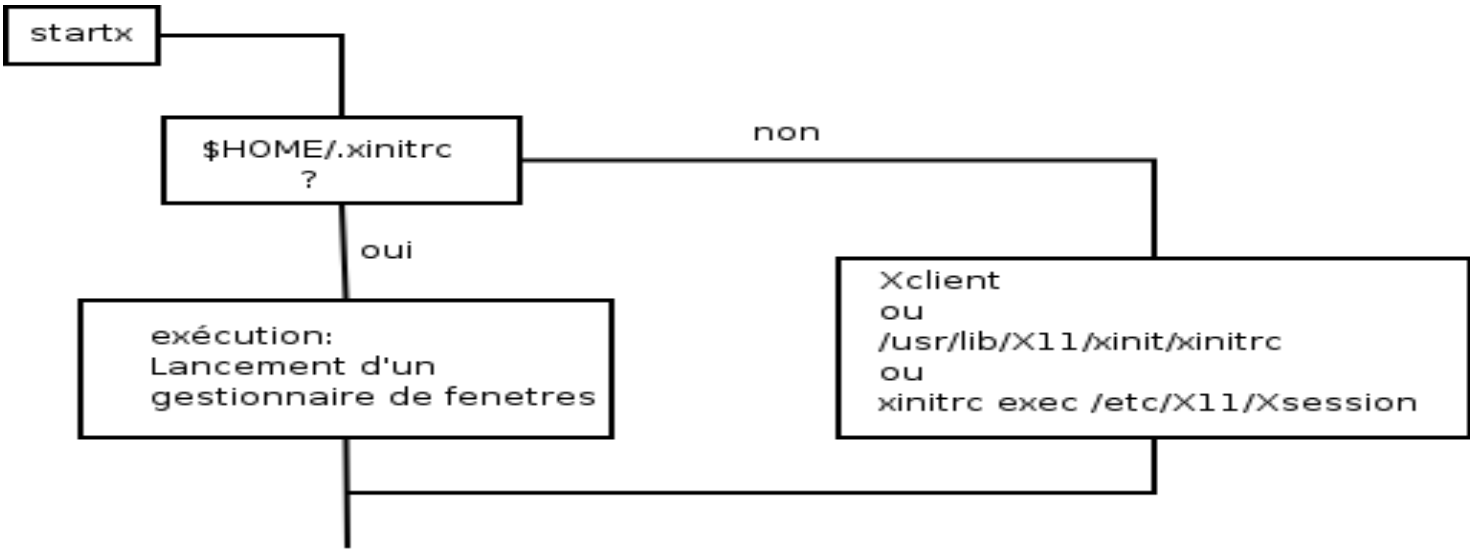
Méthode 1:utiliser un outil lié à la distribution
XFdrake => sous Mandrake
yast => fait toute la configuration sous Suse
Xconfigurator => Redhat

Méthode 2:avec la commande xf86config, xorgcfg, xorgconfig

Méthode 3: éditer le fichier
/etc/X11/XF86config-4 ou
/etc/X11/XF86config ou
/etc/X11/xorg86.conf

Démarrage d'un session X
#startx
#kde
#gnome-session
#xinit

ou
lancer à partir du service graphique :
/etc/init.d/dm
/gdm
/xdm
/kdur



Démarrage d'un gestionnaire de fenêtre.

- Démarrage d'un bureau
  ou
- démarrage d'une console et d'une horloge

Connexion au serveur X via sdm:
relation Client-Serveur gère les ressources du serveur :
Répertoire de configuration /etc/X11/xdm:
xdm-config => Emplacement des autres fichiers de configuration et variables
Xserveurs => Liste des serveurs
Xsession => Script de connexion (Xaccess)
Xaccess => Configuration des accès via XBMC

Gestion du display
#application_graphique -display 127.0.0.1:X.Y
le X correspond à la carte 0=carte 1 1=carte 2...
Le Y correspond à l'écran 0=ecran1 1=ecran 2...

Autoriser l'accès à notre serveur X à des applications clientes distantes
#xhost + => permet à n'importe qui de lancer à distance une interface graphique
#xhost +ip_ou_nom => uniquement l'ip ou nom donné
#xhost - => inverse de +

REMARQUE: à lancer depuis le mode graphique
Autoriser de manière permanente:
/etc/X0.host + nom_ou_ip_client

exemple
#ssh -l nom_srv appli_a_lancer –display ip_locale:0.0

Dans Debian:
/etc/X11/xinit/xserverrc
enlever le nolisten

Mise en place d'un serveur d'application basé sur xhost et ssh

1. Stocker les clés publiques pour les utilisateurs SSH autorisés.
2. Sur le client générer la paire de clés privées
3. Donner la clé publique du serveur aux utilisateurs autorisés
4. autoriser le serveur à autoriser notre bureau
5. lancer une session Xlocale
6. Via SSH, lancer l'application souhaitée sur le serveur avec affichage sur le poste client.

LE SERVEUR XDMCP

X Display Manager Choosing Protocol

Côté serveur :
sous Mandrake et Debian (installer le pack gdm):
/etc/X11/sdm/sdm-config

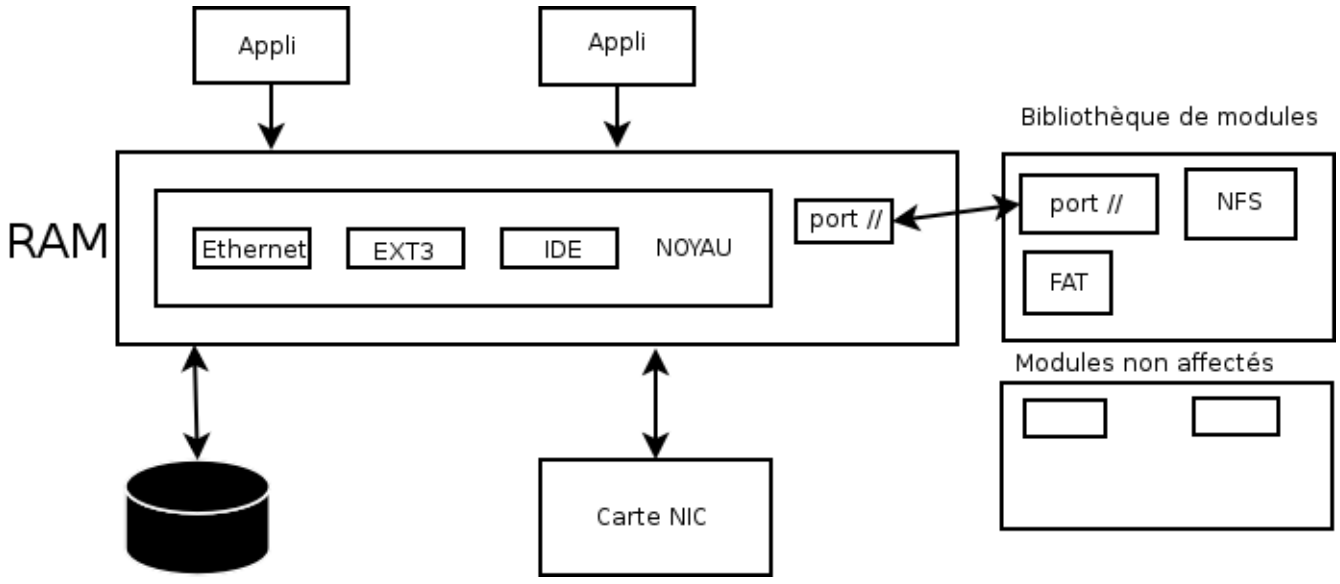
DisplayManager.requestPort 177 => Port d'écoute

Sur Ubuntu
/etc/X11/gdm/gdm.conf
[xdmcp]
enable=true
port=177

Côté client:
#X -broadcast :1 => le premier disponible
#X -indirect nom_ou_ip :1 => Celui fourni et si pas possible un autre trouvé
#X -query nom_ou_ip :1 => Celui la et pas un autre
Le :X indique la sortie écran(CTRL+ALT+F8).

LE NOYAU

La compilation:
But:
- Installer des fonctionnalités supplémentaires non prévues à l'origine
- Optimiser le fonctionnement



1. **Installer les sources du noyau:**
paquetage Kernel-source et éventuellement Kernel-headers
http://www.kernel.org

2. **Se placer dans le répertoire permettant la compilation:**
#cd /usr/src/linux-version_noyau

3. **Configurer le noyau avec les fonctionnalités souhaitées:**
#make config => (Déconseillé) montre les fonctionnalités une par une
#make menuconfig => Menu semi-graphique
#make xconfig => Mode graphique

REMARQUE: sous Debian besoin d'installer les packages kernel-source et libncurses5-dev

Commande rapide sous Debian
#make-kpkg –revision=1.0 kernel-nom_image
1.0 est une variable que l'on choisi

4. Gérer les dépendances entre fonctionnalités
#make dep

5. Nettoyer les fichiers temporaires
#make clean

6. Compiler
#make bzImage

7. En cas de changement de version, compiler les modules
#make modules
#make modules_install

8. Installer la nouvelle image dans /boot, et créer une entrée dans /etc/lilo.conf pointant sur cette image
Sous DEBIAN: /boot/grub/menu.lst

9. Vérifier dans /etc/lilo.conf

10. Tester en rebootant sur la nouvelle image

La gestion des modules

Fichiers:
/etc/modules.conf => alias et chemins de recherche pour les modules

/lib/modules/*version_noyau*/kernel/drivers
/net
/sound
Chaque module contient des sous répertoires pour chaque module

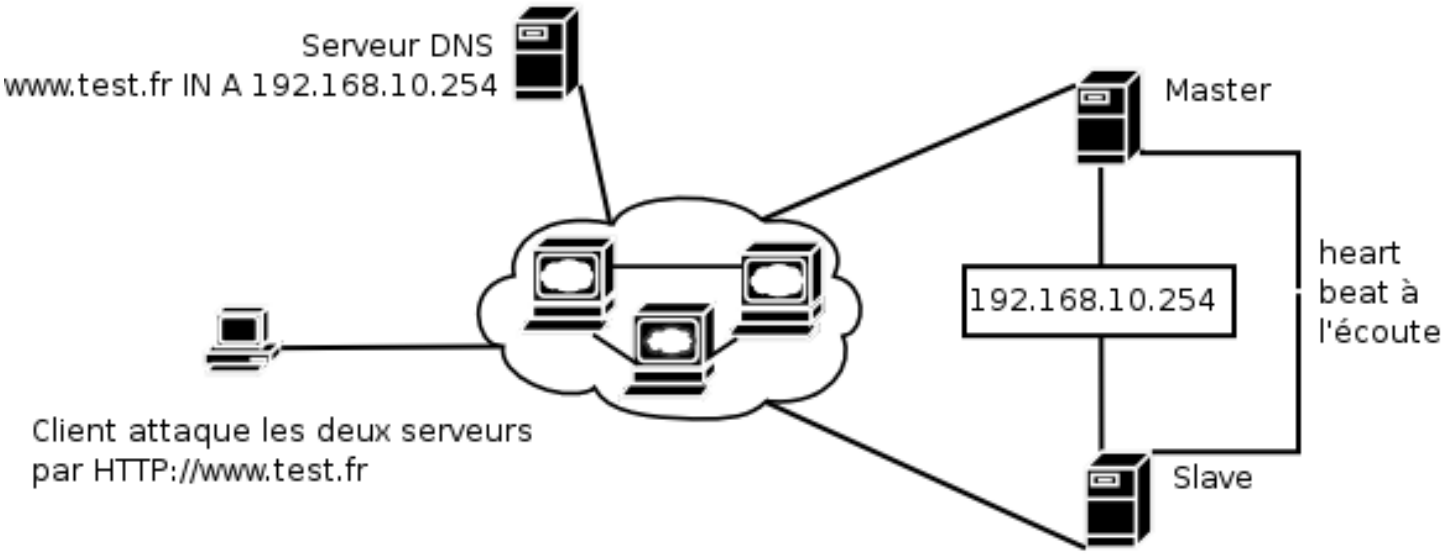
Les commandes:
#modprobe -option
-l => liste de tous les modules installables
-l -t type => liste de tous les modules installables de type indiqué (usb, vidéo...)
-c => Configuration actuelle
-r nom_mod => retrait du module du noyau
nom_mod => insérer un module dans le noyau (pas si périphérique inexistant)
#lsmod => Liste les modules présents dans le noyau

Visualisation des périphériques/cartes reconnues:
#lspci => vois les cartes pci et isa
#lshw
#lsusb

LES CLUSTERS

Les clusters de
- disque: RAID 0,1,5,10...
- processeurs: Architecture SMP
- services : intégré au protocole (DNS), géré par un super-service
- machines: parallélisation des processeurs

Le service heartbeat:



Le service heartbeat permet à la machine Master de répondre aux requêtes client et au slave de prendre le relais si le master ne répond plus aux battements de coeur (clusterisation Active passive sur adresse IP)
Le heartbeat est à l'écoute par port série ou avec une autre carte réseau entre les deux PC.

Dans le fichier :
/etc/ha.d/haresources
nom_host_master IP_virtuelle nom_service

ATTENTION: le nom du service correspond au non du fichier à lancer dans /etc/init.d

dans /etc/ha.d/ha.cf
keepalive n => fréquence en secondes des battements de coeur
deadtime p => délais en secondes avant de considérer que le master est mort (ex:3 secondes)

serial nom_peripherique_serie_heardbeat ex: /dev/ttyS0
baud vitesse_lien_serie ex: 19200
OU
udp nom_int_réseau_pour_hartbeat
udpport port_udp_utilisé ex: 1001
node nom_host_master
node nom_host_salve

/etc/ha.d/authkeys
auth 1|2|3
1 CRC
ou
2 SHA1 Chaîne_cryptée
ou
3 MD5 Chaîne_cryptée

Gros problèmes:
- la deuxième ne marche que si la première tombe (gâchis) souvent les deux machines ont deux services HTTP et Mail et ils sont master du http pour l'une et mail pour l'autre
- ne protège que des pannes hardware (si arrêt du service => le slave prends le relais)
- Pas de répartition de charge

- **Le service ipvsadm: Cluster à tolérance de pannes et répartition de charge:**

1. Etre routeur de paquet

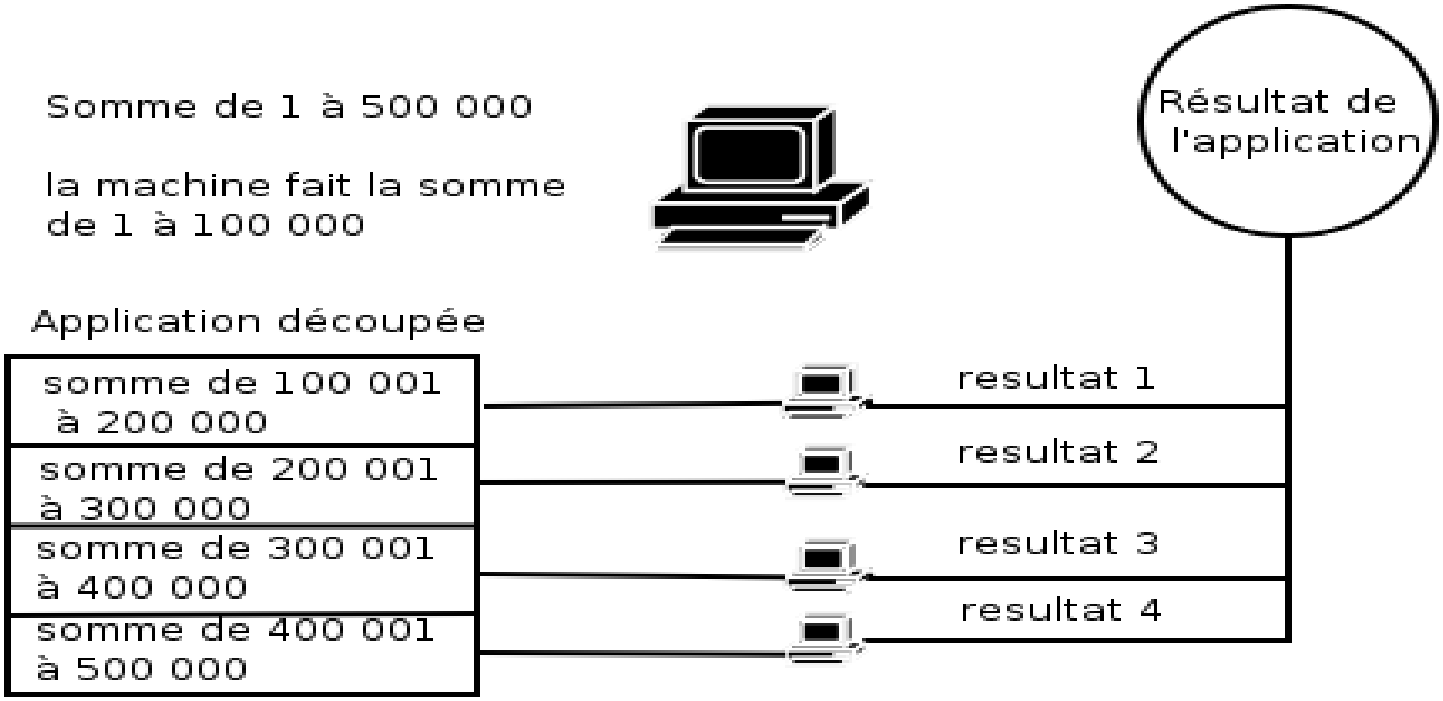
2. Masquer le réseau des serveurs virtuels
#iptables -t nat -A POSTROUTING -s réseau_serveur_réel -j MASQUERADE

3. Déclaration du service à clustériser en précisant la méthode de réparation de charge
#ipvsadm -A -t *ip_publique:port_service* -s *méthode*
-A => ajout d'un serveur virtuel
-t => Target (cible)
méthode => rr : Round Robin
wrr : Weighted Round Robin avec poids
lc : Least Connection (serveur le plus disponible)
wlc : Weighted Least Connection (LC avec poids)

4. Déclarer les serveurs réels
#ipvsadm -a -r *ip_serveur_reel:port_service -t ip_publique:port* -m -w poids
-r => real
-a => ajout d'un serveur réel
-m => masquerading
poids => 0 = quiescent (tranquille) empêche toute nouvelle connexion mais pas celles établies
1 à 65535

- **Le Cluster de CPU avec parallélisation de processus**:

On segment l'opération de l'application et chaque poste renvoi le résultat dans un FIFO (ou canal nommé) qui ne sera lu que à la fin de toutes les opérations pour récupérer le résultat globale.



Le cluster maître est désigné:

- il distribue sa clé publique à tous les clusters esclaves
- Exporte un répertoire via NFS, contenant le fichier de visualisation du déroulement de l'application.
- Lance l'application, la segmente, distribue les segments, récupère les résultats partiels, les assemble

Les clusters esclaves:

- possède la clé publique du maître
- monte via NFS le répertoire partagé du maître
- Exécute le segment de l'application envoyé par le maître, retourne le résultat partiel
- possède l'application localement
- Ecrit dans le fichier NFS de manière distinctive (couleurs différentes pour chaque noeud)

coté client

```
#useradd cluster
#mkdir -m 700 /home/cluster/.ssh
#mount -t nfs 192.168.10.254:/cluster /var/cluster
#cat  /var/cluster/id_rsa.pub > /home/cluster/.ssh/authorized_keys
#cp /var/cluster/bigcalcluster /home/cluster
```

On utilise un programme de calcul de somme nommé calcul
calcul

```
#!/bin/bash
i=$1
j=0
while [ $i -le $2 ]
do
        let j=j+i
        let i=i+1
done
echo la somme des nombres de $1 à $2 est: $j
```

et aussi le fichier de segmentation et de calcule bigcalcluster

```
#!/bin/bash

DEBUT=`cat debut`
DEBUTBIS=$DEBUT
j=0
FIN=`cat fin`

while [ $DEBUT -le $FIN ]
do
      let j=j+DEBUT
      let DEBUT=DEBUT+1
      let k=$DEBUT%200
# le 31 de la ligne dessous est la couleur a changer pour chaque noeud et le 1 est l'id du cluster à changer
# pour chaque noeud du cluster (voir /etc/DIR_COLORS)
      [ $k -eq 0 ] && echo -en "\\033[1;31m"1"\\033[0;39m"  >> /cluster/FILE2
done
echo "La somme des nombres de $DEBUTBIS à $FIN est: $j"
```