

Administration Unix

<u>Introduction.</u>	2	
<u>Usagers.</u>	3	
<u>Installation d'un serveur linux.</u>	4	
<u>Les packages RPM et la création des User</u>	6	
<u>Gestion des Users</u>	7	
<u>le File System.</u>	8	

Introduction.

Par Pascal LECUYER

- 1 installation / introduction
- 2 démarrage / arrêt – noyau – rpm
- 2 usagers
- 3 système de fichier
- 4 processus / impression
- 5 disques / périphériques / sauvegarde
- 6 réseau / sécurité
- 7 shell bash

un système informatique englobe un système d'exploitation et a pour objectif de remplir des services, combler un besoin pour des usagers => relation avec un besoin.

Système d'exploitation : consiste à réaliser une machine logique qui a pour but de gérer un certain matériel pour le rendre plus accessible pour les logiciels et les utilisateurs. Maintenant c'est plus un ensemble de logiciel dont le but est de fournir des services aux applications. Cela permet aux applications de s'exécuter.

Linux est un système d'exploitation et les distributions de linux sont le système d'exploitation plus des applications.

Le système d'exploitation est libre mais pas forcément les applications.

GNU → licence GPL

Sous licence un GPL le logiciel peut être distribué avec les sources gratuitement ou pas, le logiciel peut être approprié mais si on le distribue c'est avec les sources.

Caractéristiques de Linux :

- 1 multi plate-forme : PC, mac, Sun ...
- 2 multi tâches / multi processeur
- 2 multi utilisateurs

comment est utilisé linux :

- 3 serveur WEB (apache)
- 4 possibilités de clustering
 - continuité de service (avec un serveur actif et un service passif)
 - répartition de charge (load balancing) sur plusieurs autres serveurs
- 5 système embarqué => solutions micro-linux
 - avec une mission précise au sein du réseau (solutions clé en main)
 - temps réel
- 6 laboratoire / plate-forme d'expérimentation / test car possibilité de modifier linux
- 7 gestion avec un serveur avec sortie en texte qui va vers linux qui va sortir un flux hml vers les terminaux avec navigateurs internet.

Usagers.

Un usager est identifié et peut utiliser une application.
Utilisateur a une autre signification.

Identification d'usage : UID

Identifiant de groupe : GID

Mot de passe

Compte (home directory)

Shell (bash)

/etc/passwd (local que l'on peut centraliser grâce à NIS ou NIS+) en lecture pour tout le monde

annuaire LDAP

/etc/group

/etc/shadow avec mot de passe sécurisé + information sur la validité du mot de passe
... : lisible que pas l'administrateur

root a l'UID 0

usager système : pas de compte

lp : service d'impression

usager normal

usager toto

/home/toto

ou : /home/user/toto

/etc/skel : c'est le squelette d'un compte qui définit les fichiers et les répertoires pour les fichiers d'un compte.

Installation d'un serveur linux.

Partitionnement avec Disk Druid.

Partitions :

/boot = 32Mo

swap = Espace swap de 2 à 4 fois la taille de la RAM.

/ = reste du disque

système de fichier ext3 : système de fichier journalisé. Avantage en auto-réparation qui permet de comparer les actions qui auraient dû être faites et celles qui ont été faites. Si il y a une différence, il essaie de faire les actions manquantes et si c'est pas possible, elles ne sont pas effectuées mais il n'y a pas de vérification entière du disque.

Généralement c'est plus rapide et il y a moins de cas d'erreurs fatales.

Avec ext2 le système vérifie tout le disque ce qui peut être très long. Il y a plus de cas ou il n'est pas possible de récupérer le système de fichiers.

Pour charger le système : se trouve sur le DD pour charger le noyau en mémoire et lancer son exécution.

LILO : peut avoir des problèmes si le DD est > à 8Go

GRUB

BIOS → LILO → noyau (kernel) → init

On peut modifier /etc/lilo.conf :

8 pour changer le disque de boot (/dev/hda)

9 prompt pour choisir un autre système que celui par défaut (default et time out)

10 image = /boot/vmlinuz-2.4.2-2

label = "linux"

root = /dev/had7 (partition 7 du hda)

other=/dev/hda1

label = "windows95"

on précise donc le noyau que l'on veut charger. On peut donc en avoir plusieurs sur le DD pour tester une version.

En lançant la commande lilo

→ init lance /etc/inittab

5 :2345:respawn:/sbin/mingetty /dev/ttyS0

5 : étiquette

2345 : niveau

respawn : action

les reste : commande

il existe **7 niveaux** :

0 : arrêt du système

1 : mono-utilisateur (single user) en cas de problème avec que le root

2 : multi-utilisateur = réseau sans NFS (service de partage de fichier)

- 3 : niveau 2 + NFS
- 4 : pas défini par la norme : niveau spécifique
- 5 : niveau 3 + le graphisme
- 6 : redémarrage

Donc cette commande exécutée la commande si on démarre en niveau 2,3,4 ou 5.
respaw permet de ne pas attendre la fin de la tâche avant d'en lancer une autre. Par contre si elle se termine elle se relance automatiquement.
wait permet d'attendre la fin de la commande pour continuer le chargement.

La première ligne définit le chargement du système par défaut :
 id:5 :initdefaults \Rightarrow niveau graphique

le répertoire **/etc/rc.d/init.d** contient l'ensemble des scripts qui permettent de configurer et de lancer les services (réseau, impression ...)

exemple : **/etc/rc.d/init.d/network** pour le réseau
 Pour démarrer des services
/etc/rc.d/rc0.d
 :
 :
/etc/rc.d/rc2.d/S10network \Rightarrow S pour Start puis un nombre : ordre de démarrage
 :
 :
/etc/rc.d/rc4.d/K10network \Rightarrow K pour Kill : stoppe un service
 :
 :
/etc/rc.d/rc6.d

Chacun de ces fichiers est un lien symbolique qui lance un script. On peut lancer manuellement ces scripts. On se met dans **/etc/rc.d/init.d** et on lance la commande :
 network stop
 network start
 les arguments sont standards.

la commande **chkconfig --add nom_du_service** permet d'ajouter un service au démarrage.
 On a aussi les options **--del** et **--list**.

Les commandes **init0**, **halt**, **shutdown**, **reboot** et **init 6** redémarrent le système.

init x permet de passer en level x
runlevel donne le level où on se trouve.

Les packages RPM et la création des User

Les RPM ont été créés par Red-hat.

Chaque package a un nom

une version

parfois une release

un sommaire (phrase)

un descriptif (un texte explicatif)

la liste des dépendances (fichier : nom du package + le chemin ou le nom d'un rpm)

La taille

L'auteur

Packager

Vendeur

Licence

On a donc une partie paramètres et une partie data.

Dans les paramètres on a en plus des informations, la liste des fichiers et des scripts pour avant et après : installation / mise à jour / désinstallation.

Dans la partie data on a les fichiers.

rpm -q <fichier> ⇒ interroge la bibliothèque des packages installés pour avoir des informations sur toto

-a ⇒ tous les rpm installés sur le système.

-i ⇒ informations sur le package installé

-p ⇒ informations sur les packages non installés

-l ⇒ lister les fichiers

-r ⇒ liste des packages dépendants de lui

-f ⇒ liste des packages contenant le fichier dans les packages installés

rpm -i <fichier> ⇒ installation

-u ⇒ update

-h ⇒ hash (#) : barre de progression

-v ⇒ verbose : donne plus d'informations pendant l'installation

--force ⇒ permet d'écraser les fichiers

--nodeps ⇒ sans dépendances

rpm -e <fichier> ⇒ désinstall un package si il n'y a pas de dépendance sur lui

--test ⇒ mode de test de désinstallation

--nodeps ⇒ sans dépendance

--noscript ⇒ pas d'exécution du script d'installation : on peut essayer de démarrer un service qui est en fait désinstallé.

kpackage est un outil graphique pour la gestion des packages.

Gestion des Users

/etc/passwd	informations sur les usagers
/etc/group	liste des groupes
/etc/shadow	sécurité sur les usagers
/etc/skel	squelette / home directory utilisé lors de la création d'un usager

who	usagers connectés
who -m	usager courant
users	usagers connectés
id	affiche l'id d'un user
finger	informations user
logname	nom de l'usager qui a ouvert la connexion courante
su toto	ouvre une session avec toto
echo \$USER	affiche l'utilisateur courant
logname	nom de la première session ouverte

/etc/default/useradd	
UID_MIN	
UID_MAXIMUM	intervalle pour définir les UID
HOME	répertoire de la home directory
SHELL	shell par défaut /etc/bash
SKEL	répertoire skel /etc/skel

useradd	ajout d'un usager
usermod	modification d'un usager
-c <commentaire>	
-d <home directory>	
-g <group principal>	
-G <group supplémentaire>	
-s <shell>	
-u <UID>	
-p <password>	

quand on crée un utilisateur avec useradd chacun a un groupe spécifique, ce qui n'est pas pratique pour les droits d'accès d'ou l'utilité de

-n	pas de groupe spécifique
-e	date d'expiration du user
-f	nombre de jour de validité du password

userdel toto

-r	destruction de la home directory
----	----------------------------------

groupadd / groupmod

groupadd toto
groupadd -g 1010 titi

groupdel

le File System.

le Virtual File system permet d'accéder à plusieurs types de système de fichiers (NTFS, MSDOS, FAT16, FAT32, ext2, ext3, NFS, Samba)

mount permet de faire un point de montage.

umount détruit le point de montage.

df donne les points de montage.

more /etc/fstab \Rightarrow on a 6 champs

- 1 périphérique (device)
- 2 champs de montage
- 3 type de partition
- 4 options de montage (default, auto, noauto)
- 5 dump : 0 / 1 pour si non-dumpable ou dumpable
- 6 vérification : si >0 : ordre de vérification ; si 0 pas de vérification

sur notre installation on a :

/dev/hda3	/	ext3	default	1	1
/dev/hda2	swap	swap	defaults	0	0
/dev/hda1	/boot	ext3	defaults	1	2
/dev/fd0	/mnt/floppy	auto	noauto,owner	0	0
/dev/cdrom	/mnt/cdrom	ISO9660	noauto,owner,ro	0	0
none	/proc	proc	defaults	0	0
none	/dev/pts	devpts	defaults	0	0

mount affiche les points de montage courant.

df : espace occupé / espace libre en bloc

df -k : en Ko

df -h : en k/M/G

/bin : exe

/boot : kernel

/dev : périphérique

/etc : fichiers conf

/home et /home/users \Rightarrow home directory

/mnt : point de montage périph à monter

/proc : ressources kernel

/var : fichiers travail : lock / log / spool

/tmp : fichiers temporaires

/sbin et /usr/sbin : exe root

/usr/bin : exe utilisateur

/usr/include : headers C

/usr/lib : binaires

/usr/local : spécifique (logiciels)

Fichier :

owner

group

type ⇒ - : fichier ordinaire

d : directory

l : lien symbolique

c : périphérique caractère

b : périphérique bloc

s : socket

droits d'accès :

owner / group / other avec chacun xrw

pour changer les droits :

1 → exécution

2 → écriture

4 → lecture

-R le fait en récursif sur un répertoire.

Commandes standard :

cat : concatène des fichiers

chown : change le propriétaire

chgrp : change le groupe

sum checksum : calcule un résultat sur un fichier qui permet de le comparer à un autre

cmp : compare fichiers binaires

diff / sdiff / diff2 : comparaison de fichier texte

ed / sed : éditeurs ligne

find : recherche le fichier

grep / egrep / fgrep : recherche de chaîne dans un fichier texte

head : affiche le début d'un fichier

tail : affiche la fin du fichier

ln : crée un lien

od : visualiser un fichier binaire

sort : trier les lignes d'un fichier

du : occupation d'un fichier ou arborescence

uniq : supprime toutes les lignes identiques dans un fichier texte

wc : nombre de caractères / mots / lignes d'un fichier texte

more / pg : permet de consulter un fichier texte

mv : déplace / renomme le fichier

uname :

Les tâches.

Les périphériques

Les drivers permettent d'accéder au matériel via un fichier device dans /dev
Chaque fichier a un numéro majeur.

Exemple :

ls -l /dev/ttyS0 → redirige le ls sur le port COM1

```
crw-rw-rw-rw- 1 root uucp 4 64 ..... /dev/ttyS0
```

En rouge c le numéro majeur.

Il y a aussi un numéro mineur relatif au driver.

Par exemple pour une carte ayant 10 ports série on va avoir un numéro majeur et 10 numéros mineur.

De plus on peut spécifier le mode du fichier :

c : caractère

b : block (512 octets)

Procédure d'installation matérielle

1 Installer le driver. Deux manières :

a. L'inclure dans le noyau en le recompilant : on ajoute un fichier objet par une édition de liens du noyau.

b. Charger le driver en tant que module : pas besoin de reboot, plug 'n play, le driver n'est pas chargé avec le noyau.

insmod : installer le module

lsmod : lister les modules chargés

Si le driver est permanent, il faut ajouter la commande dans les fichiers de démarrage de la machine

2 créer les fichiers périphériques :

```
mknod /dev/periph_0 c 14 0
```

c = mode / 14=numéro majeur / 0=numéro mineur

3

/dev/mem	mémoire physique
kmem	mémoire virtuelle du noyau
console	console principale (clavier + écran)
tty0 à 5	consoles caractères de la console principale (comme telnet) Ctrl + Alt F1 à F6 pour l'afficher Ctrl + Alt F7 : session graphique
/null	la poubelle
/zero	c'est comme /dev/null mais retourne '\0' en lecture
/dev/fd0	disquette formatée
/dev/fd0H1440	disquette non formatée
/ttyS[0-3]	ports série COM
/lp0	port parallèle (imprimante)

Les disques

IDE

SCSI → 7 périphériques : @SCSI de 0 à 7 mais le 7 est réservé au contrôleur

Un disque possède des secteurs de 512 octets. Il faut obligatoirement lire au minimum 512 octets. Le secteur 0 contient le MBR qui contient la table des partitions. Cette table possède 4 entrées. Pour chaque entrée on a un octet qui permet de définir le type de partition (255 types)

Pour microsoft on a les partitions principales et la partition étendue. La partition étendue permet de définir des sous partitions.

Pour un disque IDE, Linux définit /dev/hda pour le premier disque.

/dev/hda1 définit la première partition du premier disque dur.

fdisk

fdformat → formate une disquette

mkfs → crée un système de fichiers type ext2 ou ext3

fsck → permet de vérifier le système de fichier

df → espace libre

du → récupération d'une arborescence de fichier

Les réseaux.

Applications

Socket
IP UDP
IP ARP / ICRP
Ethernet / token ring / SLIP PPP

/etc/host.conf → résolution lors de l'accès à une machine

order hosts, bind : dans un premier temps on va regarder dans le fichier hosts et si on le trouve pas on cherche sur le DNS. C'est une priorité.

/etc/hosts

```
127.0.0.1    localhost.localdomain    localhost
@machine    nom_machine            alias
```

/etc/nsswitch.conf → comment on accède à certaines informations système

les types de données : password, group, shadow, hosts, services

les méthodes d'accès : file, nis, nis+, dns (hosts), db (fichier sous forme de base de données)

exemple : on a un ordre de priorité

```
password    files    nis+
group       files    nis+
shadow      files    nis+
```

/etc/protocols : pas besoin de toucher à ce fichier. Chaque protocole y est défini avec un numéro

```
IP      0
ICMP    1
TCP     6
```

/etc/resolv.conf → adresses des serveurs DNS, réseaux par défaut. Ca donne un ordre de recherche.

```
search      .europe.com  com
nomServeur  47.115.48.10
nomSrv2     47.115.48.11
nomSrv3     47.115.48.12
```

Si on ne trouve pas sur le DNS 1 on va chercher sur le DNS 2.

/etc/services → définit les ports TCP / UDP. Ca permet de changer les ports pour protéger.

C'est surtout utile pour des services particuliers qui utilisent un même port

```
echo  7/TCP
echo  7/UDP
ftp   21/TCP
telnet 23/TCP
http  80/TCP
pop3  110/TCP
```

/etc/sysconfig/network : on peut avoir jusque 4 cartes ethernet

@dynamiques

NETWORKING = yes

HOSTNAME = sun10

DOMAINNAME = europe.com

@fixe

NETWORKING = yes

HOSTNAME = sun10

DOMAINNAME = europe.com

GATEWAY = 47.162.45.1

GATEWAYDEV = eth0

/etc/sysconf/network-scripts/ifcfg-eth0 → définition d'une interface IP

DEVICE=eth0

BOOTPROTO = dhcp

OnBOOT = yes

DEVICE = eth0

BROADCAST = 47.162.45.255

IPADDR = 47.162.45.43

NETWORK = 47.162.45.0

NETMASK = 255.255.255.0

ONBOOT = yes

BOOTAUTO = none

/etc/sysconfig/hwconf → définition du matériel

class : NETWORK

device : eth0

driver : xircom.cb

desc : "Xircom Ethernet 10/100"

bus : PCI

/etc/module.conf → interface entre le système et le matériel

alias eth0 xircom.cb

Commandes :

ifup eth0

ifdown eth0

ipconfig -a

route → gestion de la table de routage

route -n → affichage

route add 47.118.23.30 dev eth0

route del 47.118.23.30 dev eth0

hostname → pour connaître le hostname de la machine

arp → gestion de la table ARP (IP / MAN ethernet)

netstat → statistiques sur le réseau.

1) installer le package NFS (nfs-utils-0.3.1.5.rpm)

```
rpm -ivh nfs-utils-0.3.1.5.rpm  
ou  
rpm -Uvh nfs-utils-0.3.1.5.rpm
```

v : verbose → installe ou mets à jour le package

h : hach → barre de progression

2) activer le service NFS (script dans /etc/rc.d/init.d/nfs)

a. activer sans redémarrer le système

b. activer pour les prochains redémarrages

(A) /etc/rc.d/init.d/nfs start

(B) chkconfig --add nfs

ou

chkconfig --level 345 nfs on

chkconfig --level 0126 nfs off

3) ajouter un usager jacques avec :

UID : 101112

GID : 1112 (the_users)

Nom complet : Jacques Lee

```
useradd -u 101112 -g 1112 -m -n -c "Jacques Lee" -p jacques jacques
```

remarque : -m pour créer le home directory

-n pour ne pas créer un GID=UID

OU :

```
echo « jacques :x :101112 :1112 :Jacques Lee:/home/jacques:/bin/bash » >>
```

```
/etc/passwd
```

```
echo « jacques :11545 :0 :99999 :7 :: » >> /etc/shadow
```

```
passwd jacques
```

```
new password : jacques
```

```
retry new password : jacques
```

```
cp -rfp /etc/skel /home/jacques
```

```
chmod -R jacques :the_users /home/jacques
```

4) lister le répertoire de /home par occupation disque

```
du -sk /home/* | sort -rn
```

s : somme

k : en kilo-bit

r : tri renversé

n : tri numérique

5) ajouter en service (nom_service) dans /etc/inittab. La sortie sont redirigées dans un fichier log.

```
echo « mo :345 :respawn:/usr/local/bin/mon_service 1>/var/log/mon_service.log 2>
```

```
f1 » >> /etc/inittab
```

kill -1 1 → permet de relancer l'init qui va relire le fichier modifiée et lancer le nouveau service

ps -ax | grep mon_service → permet de voir si le service est bien en fonction

6) tuer toutes les tâches toto

killall toto

7) installer la carte Specialix qui gère 8 ports série.

Driver : special-rs

Mode caractère

Numéro majeur = 222

Numéro mineurs vont de 0 à 7

Device name : /dev/ttyRspX

insmod /usr/local/Modules/special-rs.o

mkmod /dev/ttyRsp0 c 222 0

c : caractère

222 : numéro majeur

0 : numéro mineur

mkmod /dev/ttyRsp1 c 222 1

mkmod /dev/ttyRsp2 c 222 2

mkmod /dev/ttyRsp3 c 222 3

mkmod /dev/ttyRsp4 c 222 4

mkmod /dev/ttyRsp5 c 222 5

mkmod /dev/ttyRsp6 c 222 6

mkmod /dev/ttyRsp7 c 222 7

8) changer le groupe des répertoires de /home/dir1 en NewGroup

find /home/dir1 -type d -exec chgrp NewGroup {} \;

/home/dir1 : départ

-type d : critère : répertoire

{ } : chacune des occurrences qui respectent le critère = occurrence

\ ; : indique la fin de la commande exec

9) changer le groupe des fichiers ordinaire de /home/dir1 en NGF

find /home/dir1 -type f -exec chgrp NGF {} \;

9bis) détruire tous les fichiers .o de l'arborescence /home/dev1

find /home/dir1 -name *.o -exec rm {} \;

10) création de lien symbolique de /home/NewGroup1 sur /home/groupe1

ln -s /home/groupe1 /home/NewGroup1

ls -l /home

d groupe1

l NewGroup1 ---> group1

11) détruire une arborescence

rm -r repl

12) création d'une arborescence

/usr existe

créer : /usr/rep1/rep2/rep3
mkdir -p /usr/rep1/rep2/rep3

13) visualiser en continu un log
tail -f /var/log/message
Ctrl+C pour l'arrêter

14) que fait : ps -ax | tee res1 | grp crond | tee res2
affiche la liste de toutes les tâches avec res1
puis affiche les processus « crond » avec res2

15) telnet 47.162.180.18 | tee res
tout ce qui apparaît sur la console pendant le telnet est stocké dans le fichier « res »

16) routage IP : routage d'un subnet
On a deux réseaux : 47.168.48.0 et 47.162.43.0
Un routeur a deux interface est au centre avec 47.168.48.1 et 47.162.43.1
La station linux a l'adresse 47.162.48.10
Comment établir la route statique sur la machine linux pour accéder au 47.162.43.0 ?

Route add -net 47.162.43.0 gw 47.162.48.1
→ on ajoute le subnet 47.162.43.0 par la gateway 47.162.48.1

routage d'une adresse IP :
route add -host 47.168.18.18 gw 47.162.48.1

routage vers une interface IP :
On a un routeur pour aller sur le net et que l'on a aussi un modem. On veut forcer le passage par le modem qui a l'interface IP ppp0.

Route add -host 47.180.15.15 dev ppp0

17) sauvegarder un fichier sur une disquette vierge
formater la disquette :

fdformat /dev/fd0H1440

créer le file system :

mkfs -t ext2 -c /dev/fd0

-t pour préciser le type du file system

-c pour les cluster défectueux

vérifier

fsck /dev/fd0

on monte la disquette

mount -t ext2 -o owner /dev/fd0 /mnt/floppy

copie du fichier

cp fichier /mnt/floppy

démonter la disquette

umount /mnt/floppy

df -s /mnt/floppy → pour connaître l'espace libre
du -s /mnt/floppy → pour connaître l'espace utilisé

e

Compilation du noyau

uname -a → informations sur le noyau

Linux plecuyer-2.europe 2.4.2-2 #1 Sun Apr 8 20 :41 :30 EDT 2001 i686 unknown

Linux = type de système

plecuyer-2.europe = hostname

2.4.2-2 = version du noyau = version . release majeure . release mineure – numéro d'édition

le majeur est une évolution du noyau (évolution fonctionnelle)

le mineur est mis pour plusieurs mise à jour ou mise à jour importante

numéro d'édition = correction mineures

#1 Sun Apr 8 20 :41 :30 EDT 2001 = date de compilation du noyau

i686 = type de machine

unknown = ?

/boot/vmlinuz → c'est le noyau

/boot/system.map → c'est un mapping de tous les modules avec leur adresse

non compressé = 2.2 à 2.5 Mo

compressé = 800Ko

1) Configuration du noyau

/usr/src/linux/.config

make -f <make file>

make xconfig → méthode graphique

make menuconfig → méthode par menu mode caractère

make config → commande en ligne

on peut configurer :

- 2 processeur
- 3 port série et //
- 4 USB
- 5 Multi processeur
- 6 SCSI / IDE / Raid
- 7 Système de fichiers

2) compilation du noyau

make dep → crée les dépendances

make clean → supprime les production intermédiaires (provenant des make intermédiaires) pour forcer la re-compilation complète du noyau

make bzImage → compilation

le noyau produit est mis dans /usr/src/linux/arch/i386/bzImage

Il faut lui ajouter les modules.

3) Ajout des modules (compilation et installation)

/usr/lib/module

make modules → compilation des modules
make modules_install → pour les ajouter dans le noyau

4) Installation du noyau

/!\ Il faut faire attention à ne pas écraser le noyau existant /!
cp /usr/src/linux/arch/i386/bzImage /boot/vmlinuz.new
cp /usr/src/linux/System.map /boot/System.map.new

5) mettre à jour LILO

```
        /etc/lilo.conf  
image=vmlinuz.new  
label=linux-essai  
:  
:  
:
```

on recharge LILO → commande « lilo »