

07/01/15

Administration Unix

• création d'un compte

useradd nom_compte

useradd -m nom_compte (forcer la créat° du répertoire personnel)

/home/nom_login : ~ nom_log
/root

• Les informat° sur les comptes sont enregistrées dans :

- etc/passwd \Rightarrow syntaxe d'une ligne (644) (ls -l etc/passwd pr voir les permissions d'accès)
nom_login : password : uid : gid : commentaire : dossier du rep personnel : shell

- etc/shadow : informat° sur les mots de passe

grep ^root \rightarrow /etc/passwd pour chercher la ligne qui commence par root.

id -u root

id -u \rightarrow user

-g \rightarrow groupe principal

-G \rightarrow groupe secondaire

(-gn : son nom)
au lieu du nom du groupe

delimiteur

champ (field)

cat -d: -f1 /etc/passwd

\Rightarrow affiche le 1^{er} champ

cat -d: -f1,3 /etc/passwd

comptes admin

uid = 0

" systemes

$0 < uid < uid_min$

" users

$uid \geq uid_min$

(/etc/login.defs)

grep -iw uid_min /etc/login.defs

useradd test001

grep ^test001: /etc/passwd /etc/shadow

su x → on change l'identité x
 su - x → on change aussi les variables d'env.
 dont Path

which ls ⇒ chemin qui sera exécuté.

est-ce que le rép personnel est créé? ⇒ ls -l /home
 on a effectivement trouvé test001

nom_login : mdp : date_dernière_modif (en jrs) : durée de vie max du mdp (en jrs) : durée de vie min (en jrs) : warning : date désactivation du compte
 (on ne peut pas changer le mdp avant cette durée)

chage	-M	durée de vie max	nom_compte
chage	-m	durée de vie min	"
	-w	warning day	

si champ mdp commence par ! → compte désactivé

awk -F: ' \$3 >= 500 {print \$1}' /etc/passwd

grep -i pass /etc/login.defs

Ex1 commande 'x' est remplacée par son résultat

Ex1 chage -M 30 test001

for x in \$(awk -F: ' \$3 > 500 {print \$1}' /etc/passwd)
 do chage -M 30 \$x

pour modifier le mdp :

passwd nom_compte

(demande de taper le mdp 2 fois pour confirmat°)

pour désactiver un compte

passwd -l nom_compte

passwd sans argument
→ change

-d → on autorise l'user à se connecter sans mdp

-u → réactive le compte (unlock)

ls -la /etc/skel ~ test001

↑
fichier cachés (dont le nom commence par .)

qd on crée un user, on l'affecte à un groupe principal du même nom,

le gid le plus petit id non encore utilisé

- bashrc
↳ script exécuté automatiquement lorsqu'on lance un nouveau shell
- bash_profile
↳ lorsqu'on ouvre une session
- bash_logout
↳ lorsqu'on se déconnecte

* options useradd

- u uid
- g gid
- G liste de groupes secondaires
- d rep personnel
- k rep skel
- o forcer
- m forcer la création du ~

* usermod options compte

* useradd -D (affiche le contenu de /etc/default/useradd)

* Pour supprimer un compte

userdel nom_compte

userdel -r nom_compte (son rep personnel aussi)

* find / -nouser

★ Gestion des groupes

fichier d'informat° : /etc/group

Syntaxe d'une ligne:

nom_groupe : passwd : GID : liste des membres

groupadd nom_groupe (4/11) gpasswd -a user groupe
groupdel -r (ajouter un user au groupe)
(enlever du groupe)

gpasswd -A admin_groupe groupe.

Ex

gpasswd -A test001 groupe 1

groupadd groupe 1

gpasswd -a test002 groupe 1

Afficher la liste des groupes d'un user ⇒ groups nomuser
par ex: groups test002 va afficher groupe 1

> f1
ls -l f1

-rw-rw-r-- 1

user 01 groupe 1

si je suis membre de ce groupe, these are the permissions I have.

• Fichiers appartenant à groupe 1 ?

find / -group groupe 1 -type f
ds Ht l'arborescence pour avoir uniquement les fichiers (sans le répertoire)

• fichiers de taille nulle ?

find / -size 0 (M G K)

+ appartenant à root
find / -size 0 -user root

• fichier auxquels on n'a pas accédé depuis 90 jrs ?

find / -atime ≥ 90

• fichier dont la taille est > 1M

find / -size +1M

; → sépare deux commandes

★ Gestion des packages → (app développée pour Linux)

rpm
pour installer un package, rpm -ivh package.rpm
yum -y install package

Ex yum -y install bind
(ping www.google.com)

→ si ça ne marche pas
vi /etc/resolv.conf
nameserver 8.8.8.8
vi /etc/yum.conf
proxy = http://196.200.

rpm -e package

rpm -qa : affiche la liste des paquets installés

rpm -qf /etc/inittab
↑

le paquet qui
a installé ce fichier

rpm -ql package → les fichiers appartenant à ce paquet
+qd → les fichiers doc
c → configurat°
i → informat°

rpm -V nom_package

→ pour vérifier l'état actuel
des fichiers du paquet
% à l'installat°

rpm -Vf /etc/inittab

résultats
(lorsqu'il
y a un
changement)

M : permission " " "

S : contenu n'est pas le même

S : taille

u : propriétaire

g : groupe

L : nombre de liens

D : type / majeur / mineur d'un fichier spécial

T : date d'accès

08/01/15

* système de fichiers

fdisk - l

au démarrage
⇒ affiche les partit° reconnues par la machine

fdisk un disque

⇒ pour gérer le disque

→ carte qu'on insère

Rappel

/dev/hda ⇒ disque dur maître du 1^{er} contrôleur ide.

/dev/hdb ⇒ " " esclave " " " "

/dev/hdc ⇒ " " maître " 2^{ème} " "

/dev/hdd ⇒ " " esclave " " " "

/dev/sda ⇒ 1^{er} disque dur SCSI

/dev/sdb ⇒ et ainsi de suite

Dans un disque

/dev/sdx1 : 1^{ère} partit° primaire

/dev/sdx2 : 2^{ème} partit° " "

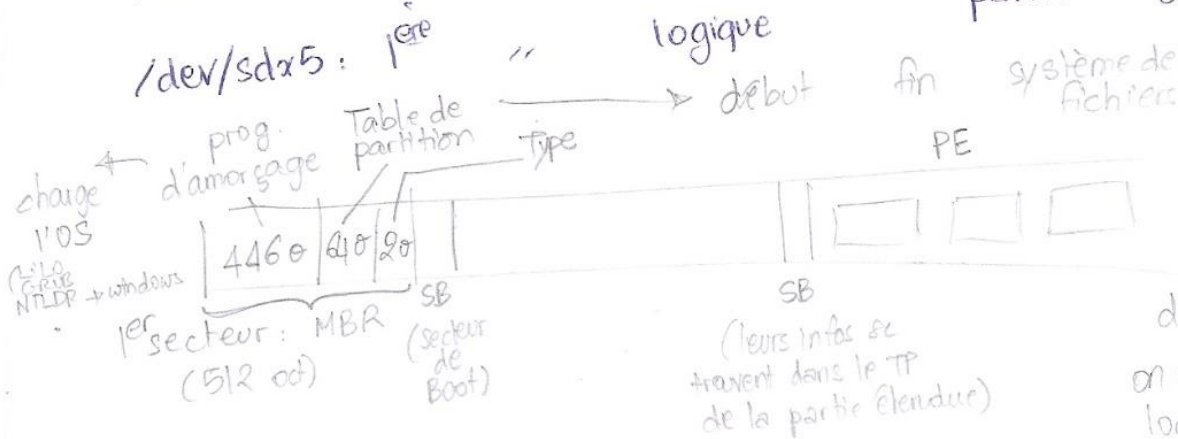
/dev/sdx3 : 3^{ème} " " "

/dev/sdx5 : 1^{ère} " " logique

on ne peut avoir que

4 partit° prim.

À partir de 5, on a les partit° Logiques.



dans des part. étendues, on peut créer des part. logiques.

appelées logiques car elles sont créées dans des partitions

fdisk /dev/sda

: attend vos commandes

: m → manuel (help)

: n → nulle partit° (ns demande secteur début et fin)

: d → supprimer une partit°

: q → quitter sans sauvegarde

: w → avec "

: P → affiche notre table de partit°

parted \Rightarrow peut faire des déplacements pr libérer de l'espace

on peut créer une partit° si l'espace n'est pas partitionné

Lvm \Rightarrow crée des partit° logiques

\hookrightarrow je peux créer des partit° \in à +^{rs} disques

MAIS normalement, une partition appartient à un seul disque.

o on a une part. et on veut créer un sys de fichiers.

`mkfs -t type partit°`

\rightarrow permet de créer un sys de f d'un type donné ds une partit° de notre choix.

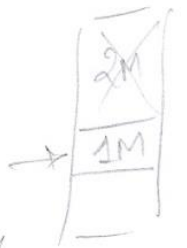
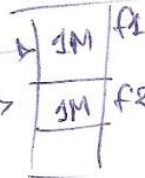
ex (formatage)

`mkfs -t ext2 /dev/sda2`

`dd` \leftarrow copie
`if = /dev/zero` \leftarrow input file
`-c` \Rightarrow permet de vérifier la partit°
`of = f` \leftarrow output file
`bs = 1M`
`count = 1`
crée un fichier f de taille 1M.
periph supposé de taille ∞
 \uparrow cmb de fois ferons-nous la copie

Créer une partit° ds un fichier

si on ne veut pas écrire ici
mais plutôt ici



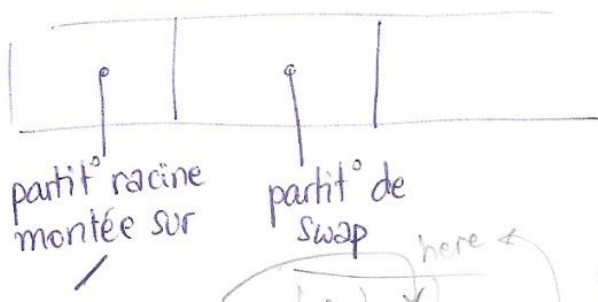
il suffit de faire

`dd if = f1 of = f2 bs = 1M count = 1`

`skip = 1` \times `bs` \leftarrow je saute
`seek = 2`
 \hookrightarrow c'est pour l'écriture. je saute $2 \times bs$

Pour utiliser une partit°, il faut la monter sur l'arborescence du fichier

À l'installat° de Linux, on crée obligatoirement 2 partitions



il est conseillé de créer une partit° pour sauvegarder les données de l'utilisateur

on la monte sur /home et dakchi sera sauvegardé sur la partit°

umount partition / ou rép de montage et avec quelles options

Pour afficher les montages actuels, `mount`

`mount -a`
fait le montage automatique

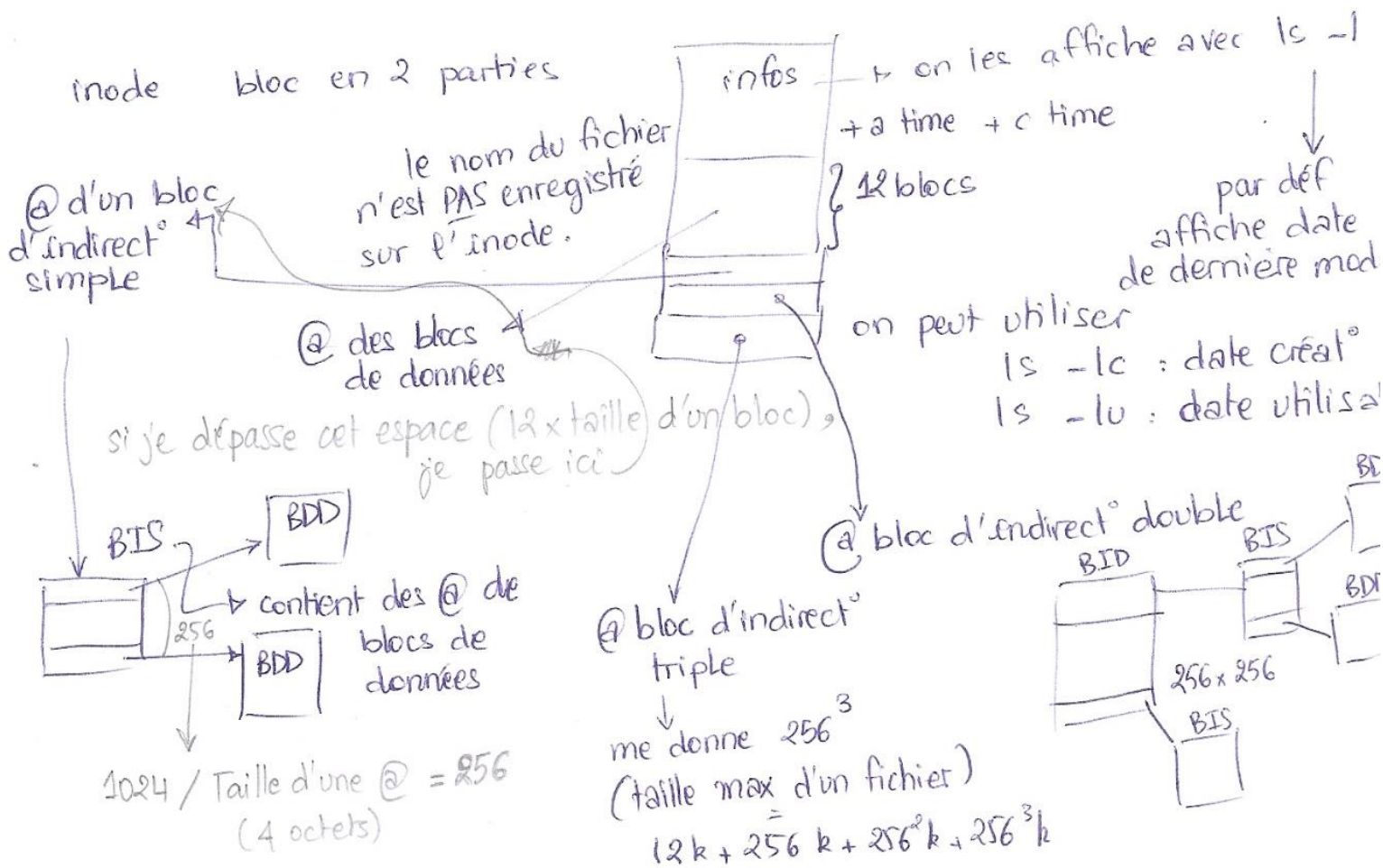
`mount -t ext2 -a` → monter uniquement les types de fichiers ext2

`df` : pour chaque part. montée cmb on a consommé et cb il reste.
(en terme de blocs)

`df -i` : mêm chose mais en terme d'inode → struct où on a des infos sur le fichier.

4 types de bloc

- superbloc, contient des infos sur le sys de f.
- inode : les infos sur 1 fichier
- bloc de données : données du fichier
- bloc d'indirect° : adresse du bloc



Les données d'un rép = correspondance entre fichier et inode.

on sup que la taille d'un bloc est 2Kb
 $12 \times 2 + 512 \times 2 + 512 \times 2 +$

Permissions

un fichier $f \in$ $\begin{matrix} \nearrow \text{user} \\ \rightarrow \text{group} \end{matrix}$

on a des perm. pour le prop, le groupe et les autres

supp. $ls -l f$

$-rw-r--r--$ 1 user g1 . . .

x est un user. Quelles st les perm. qu'il a sur f ?
 est-ce qu'il est le prop.? Je compare les uid.
 sinon, est-ce qu'il fait partie de $g1$?

$chmod \rightarrow$ pour modifier les perm.

Perm. étendues

$suid: ls -l$

'which passwd' /etc/passwd

qui peut modifier /etc/passwd?

on a comme résultat

$-rw-r--r--$ 1 root root

/etc/passwd

\downarrow
only him!

Si $suid$ est positionné sur un exécutable binaire, lorsqu'on l'exécute on est considéré comme étant le propriétaire, et donc on calcule les perm. du proprio.

rws

$suid$ est affiché à la place du x du proprio. S'il est minuscule, il y a le x également. Sinon, on a uniquement le $suid$.

$Suid: 4000$

$chmod u+s un_file$

$chmod 4744 f$

empêche de supprimer les fichiers des autres users



Sticky bit: 1000

$SGID: 2000$

$chmod g+s un_file$

$chmod o+t un_rep$

sur un rep \rightarrow ajouter et sup des fichiers du rep.

pour utiliser le rep sur un chemin \rightarrow il faut avoir un x .

find / -perm +4000

⇒ pour afficher les fichiers qui ont un suid positionné.

```
vi prg.c
void main() {
    seteuid(0);
    setuid(0);
    execp("bash", "bash", NULL, NULL);
}
```

je change l'user effectif à celui qui a un uid=0

```
# gcc -o prg prg.c
# chmod u+x prg
```

les attributs

on les affiche avec ls attr file
si le résultat est
a → on ne peut ajouter des données qu'à la fin.
i → immutable → je ne peux rien faire.

chattr ± attribut file

les acls

Il faut monter la partition avec l'option acl.

changer les opt° d'une part. déjà montée remount.

? mount -o remount, acl / ? ← Not sure!

setfacl -m u: user: permissions file
-m g: groupe: " file

(pour afficher) getfacl file
ls -l file

ex: setfacl -m u: user01:rw f1
setfacl -m u: user02: r f1