



Administration Unix

ENSIAS – 2015/2016

Université Mohammed V – Souissi

Pr. M. ZBAKH



Plan

Chap.1: Gestion des utilisateurs

Chap.2: Gestion des fichiers

Chap.3: Systèmes de fichiers

Chap.4: Gestion des quotas



Chapitre 1: Gestion des utilisateurs

Objectifs:

A l'issue de ce chapitre, vous serez en mesure de:

- Gérer les comptes des utilisateurs et leurs groupes.
- Connaitre également la structure des fichiers qui contiennent les informations fondamentales.



Contenu

- 1) Les fichiers */etc/passwd* et */etc/group*
- 2) Les commandes d'administration:
 - useradd,
 - passwd,
 - su,
 - id,



1. Comptes utilisateurs/groupes

1. Exemples:

```
$ more /etc/passwd
```

```
root:CuSzE74h021 CS :0:0: Super User:/:/bin/bash
```

```
ali: :200:50:Ali Ali:/home/ali:/bin/csh
```

```
brahim: OZkhUrn Yrprtpt20 1:50:Brahim Brahim :/home/brahim:/bin/sh
```

```
$ more /etc/group
```

```
root::0:root
```

```
group::50:ali,brahim
```



1. Comptes utilisateurs/groupes

2- Structure du fichier */etc/passwd*

- Le fichier */etc/passwd* est un fichier de type texte dont chaque ligne définit un compte utilisateur.
- La ligne est composée de champs. Le séparateur de champs est le symbole « : ».

Nom de connexion	Mot de Passe	UID	GID	Commentaire	Répertoire de connexion	Commande de connexion
:	:	:	:	:	:	:



1. Comptes utilisateurs/groupes

Nom de connexion: saisi lors de la demande de connexion

Mot de passe: présent dans le fichier, mais crypté. Dans les systèmes sécurisés, le champ mot de passe existe toujours mais il contient le caractère x. Le mot de passe crypté est déporté dans un fichier accessible au seul administrateur. Les caractères "!!" dans le champs mot de passe indiquent que le compte n'est pas accessible.

UID: L'administrateur attribue un numéro à chaque utilisateur. Ce numéro, l'UID (« *User Identification* »), est mémorisé dans les descripteurs de fichiers pour en identifier le propriétaire. C'est donc l'information pertinente, utilisée par le système Linux, pour identifier un utilisateur. L'UID de *root* est 0.



1. Comptes utilisateurs/groupes

GID: L'administrateur identifie le groupe de connexion d'un utilisateur grâce au champ GID (« *Group Identification* »). Le fichier */etc/group* associe un nom de groupe à ce GID et définit les groupes supplémentaires de l'utilisateur.

Commentaire: La zone est utilisée librement par l'administrateur pour commenter le compte. Il peut être structuré en suivant les recommandations de la commande *finger*. On y trouve, entre autres, le nom et le prénom.

Répertoire de connexion: Ce champ détermine le répertoire de connexion de l'utilisateur, conventionnellement */home/ali* pour l'utilisateur de nom de connexion *ali*. Ce répertoire contient les fichiers de configuration (*.bash_profile*) de l'utilisateur.

Commande de connexion: Ce champ précise le chemin d'accès absolu de la commande à exécuter lors de la connexion. C'est généralement un shell.



1. Comptes utilisateurs/groupes

Remarques:

- L'UID est une valeur comprise entre 0 et la valeur définie par la constante **UID_MAX** du fichier **/etc/login.defs**. Les valeurs inférieures à 100 sont généralement réservées pour des utilisateurs associés à des services standard du système Linux. La constante **UID_MIN** du fichier **/etc/login.defs** définit la valeur minimale des UID des utilisateurs.
- L'attribution d'un UID est de la responsabilité de l'administrateur et rien ne l'oblige à les affecter séquentiellement. Il peut définir sa propre stratégie.
- Quand un administrateur gère un parc de machines Linux sans mettre en œuvre d'administration centralisée, il est conseillé d'attribuer le même UID à un utilisateur qui possède un compte sur plusieurs machines du réseau.



1. Comptes utilisateurs/groupes

Remarques:

- Si plusieurs lignes utilisent le même UID pour plusieurs noms de connexion différents, un seul utilisateur est en fait défini. On peut ainsi définir un utilisateur stop, dont l'UID est 0 et qui exécute shutdown comme commande de connexion.
- Dans un système non sécurisé, le mot de passe peut être absent, ce qui permet de se connecter sans avoir à fournir de mot de passe.



1. Comptes utilisateurs/groupes

3- Structure du fichier */etc/group*

Le fichier */etc/group* est un fichier de type texte dont chaque ligne définit un groupe d'utilisateurs, La ligne est composée de champs. Le séparateur de champs est le symbole « : ».

Nom du				Liste des utilisateurs
groupe	:	Mot de passe	:	autorisés à se connecter au
			GID	Groupe
				Syntaxe: (util [,util ...])



1. Comptes utilisateurs/groupes

Nom du groupe: Le nom du groupe est celui utilisé dans la commande `newgrp` ou affiché par la commande `ls`.

Mot de passe: Le mot de passe est présent dans le fichier, mais crypté. Il est demandé à un utilisateur qui veut se connecter au groupe et qui ne figure pas dans la liste des utilisateurs du groupe.

Liste des utilisateurs: La liste des utilisateurs qui peuvent se connecter au groupe par la commande `newgrp` sans avoir à fournir de mot de passe.



1. Comptes utilisateurs/groupe

La commande `newgrp` permet de **changer le groupe de référence** utilisé lors de la création de nouveaux fichiers. A défaut d'avoir exécuté la commande `newgrp`, c'est le groupe de connexion qui est utilisé.

For example, if you are a member of the groups "baseball" and "racquetball", and "baseball" is your current default group, you can switch to the other group with the following command:

```
$ newgrp racquetball
```

Remarques:

- Un utilisateur n'a pas besoin d'être mentionné dans la liste des utilisateurs de son groupe de connexion.
- Le champ mot de passe est rarement utilisé dans la pratique.



2. Commandes de gestion

Les commandes de gestion des utilisateurs:

- useradd, usermod, userdel : Gèrent les comptes utilisateur
- groupadd, groupmod, groupdel : Gèrent les comptes de groupe
- pwck, grpck : Vérifient les fichiers */etc/passwd* et */etc/group*
- finger: Donne des informations sur un utilisateur
- chfn, chsh : Changent le shell ou le commentaire d'un utilisateur



2. Commandes de gestion

Les commandes de gestion des utilisateurs (suite):

- `passwd` : Permet de modifier le mot de passe d'un utilisateur
- `su` : Permet de se connecter à un compte
- `id` : Permet de connaître son identité
- `groups` : Donne la liste des groupes d'un utilisateur
- `vipw`, `vigr` : Edite les fichiers */etc/passwd* et */etc/group*,



2. Commandes de gestion

Les commandes de gestion des comptes d'utilisateurs et de groupes sont nombreuses.

Les commandes d'administration proprement dites: `useradd`, `usermod`, `userdel`, `groupadd`, `groupmod`, `groupdel`, `pwck` et `grpck` peuvent être utiles quand elles sont intégrées à un script. Elles ne présentent pas de difficultés d'emploi. Dans la réalité, il est souvent bien plus pratique de procéder aux opérations grâce à l'outil intégré d'administration `linuxconf` qui propose, pour chaque paramètre d'un compte, le choix par défaut le mieux adapté. L'outil effectue aussi une vérification des éléments fournis. Attention, quelques-unes des options de la commande `useradd`, réservée à l'administrateur, permettent de paramétrer le compte. La commande `useradd` permet en effet, en sus de la création de comptes d'utilisateurs, de définir les paramètres qui seront utilisés, par défaut, à la création d'un compte.



2. Commandes de gestion

La commande `useradd`:

Syntaxe pour la création d'un compte:

```
useradd [-c comment] [-d home_dir] [-e expire_date] [-f inactive_time] [-g  
initial group] [-G group] [, ...] [-m [-k skeleton_dir]] [-s shell] [-u uid [-o]]  
[-n] [-r] login
```



2. Commandes de gestion

Les options sont celles dont la seule lecture ne donne pas l'explication:

- c comment: Le commentaire.
- d home_dir: Le répertoire de connexion.
- e expire_date: La date d'expiration du compte.
- f inactive_time: Le nombre de jours au bout duquel un compte est inutilisable, après l'expiration d'un mot de passe.
- g initial group: Le groupe initial, par défaut, ali pour l'utilisateur ali.
- G group,.. : Les groupes supplémentaires.
- m: Il faut créer le répertoire de connexion de l'utilisateur.



2. Commandes de gestion

- k skeleton _dir: Le répertoire de peuplement du répertoire de connexion. Les fichiers qu'il contient sont copiés dans le répertoire de connexion de l'utilisateur. C'est /etc/skel par défaut.
- s shell: Le shell de l'utilisateur, par défaut bash.
- u uid: L'UID de l'utilisateur.
- n: Ne crée pas un groupe avec le même nom que celui de l'utilisateur mais ajoute l'utilisateur au groupe déterminé par l'option -g ou par la variable **GROUP** dans le fichier **/etc/default/useradd**.
- r: La commande accepte de créer un compte avec un UID inférieur à UID_MIN, défini dans /etc/login.defs.



2. Commandes de gestion

Remarques:

- Le répertoire `/etc/skel` de Linux est très important: il contient des modèles de fichiers de configuration des sessions des utilisateurs.
- L'administrateur peut ajouter des fichiers ou les modifier pour définir les paramètres communs à tous les utilisateurs du site, comme les options du shell ou la définition des touches d'édition du shell bash. Ces fichiers sont automatiquement copiés dans le répertoire de connexion des utilisateurs créés par la commande `useradd` ou l'outil d'administration `linuxconf`.



2. Commandes de gestion

Exemple:

```
# ls -aC /etc/skel  
.Xdefaults .bash profile .inputrc  
.bash_logout .bashrc  
# useradd ali  
# ls -aC ~ali  
.Xdefaults .bash -profile .inputrc  
.bash_logout .bashrc
```



2. Commandes de gestion

Syntaxe pour la définition des paramètres par défaut:

```
useradd -D [-g default group] [-b default_home] [-f default_inactive]  
          [-e default_expire_date] [-s default_shell]
```

Dans cette forme, la commande `useradd` permet de définir les valeurs utilisées par défaut quand on crée un compte utilisateur.

La commande `useradd -D` visualise les valeurs actuellement utilisées.



2. Commandes de gestion

Les autres commandes

Les commandes d'information ou de gestion courantes, utiles en mode commande: finger, users, groups, id (voir les Exemples) et su.

Les commandes générales: vipw, vigr, passwd.

La commande vipw réalise l'édition du fichier /etc/passwd. Elle en effectue d'abord le verrouillage pour en garantir un usage exclusif. L'éditeur de texte qui est exécuté est défini par la variable d'environnement EDITOR, vi à défaut. Si le verrouillage du fichier n'est pas possible, la commande vipw le signale et demande à l'administrateur d'essayer plus tard. Le fichier est automatiquement déverrouillé à la fin de l'édition. La commande vigr agit de même avec le fichier /etc/group.



2. Commandes de gestion

La commande passwd

La gestion des mots de passe et de leur pérennité (« *Aging Information* ») est réalisée par la commande `passwd`, déjà connue des utilisateurs, et par la commande `chage`. Le rôle principal de la commande `passwd` est de créer ou de modifier le mot de passe d'un utilisateur. La commande `chage` gère la pérennité des mots de passe.



2. Commandes de gestion

La commande `passwd` a plusieurs fonctions pour l'administrateur:

- Modifier le mot de passe d'un utilisateur:
`passwd Nom_utilisateur`
- Supprimer le mot de passe d'un utilisateur:
`passwd -d Nom_utilisateur`
- Verrouiller le compte d'un utilisateur, ce qui empêche sa connexion:
`passwd -l Nom_utilisateur`
- Déverrouiller le compte d'un utilisateur:
`passwd -u Nom_utilisateur` ou # `passwd -d Nom_utilisateur`
ou `passwd Nom_utilisateur`



2. Commandes de gestion

Remarques:

- Les restrictions imposées aux utilisateurs dans la définition de leur mot de passe ne s'appliquent pas à l'administrateur qui peut attribuer n'importe quel mot de passe à un utilisateur, sauf une chaîne vide .
- En l'absence de mot de passe, l'invite « password » n'est pas affichée à la connexion.



2. Commandes de gestion

Exemples:

```
# id
```

```
uid=0(root), gid=0(root)
```

```
groups=0(root), 1(bin), 2(daemon), 3(sys), 4(adm), 6(disk), 10(wheel)
```

```
# users
```

```
ali root
```



2. Commandes de gestion

```
# groups
```

```
root bin daemon sys adm disk wheel
```

```
# groups ali
```

```
ali: ali compta
```

Attribuer à "ali" le mot de passe ali:

```
# passwd ali
```

```
New UNIX password:
```

```
passwd: ail authentication tokens updated successfully
```



2. Commandes de gestion

Supprimer le mot de passe de ali

```
# passwd -d ali
```

Verrouiller le compte de ali

```
# passwd -l ali
```

ali ne peut plus se connecter

venus login : ali

Password:

Login incorrect

venus login :



2. Commandes de gestion

Déverrouiller le compte de ali

`passwd ali`

Changing password for user ali

New UNIX password:

Retype new UNIX password:

passwd: ail authentication tokens updated successfully



2. Commandes de gestion

ali peut à nouveau se connecter

venus login : ali

Password:

/home/ali \$

Visualiser les attributs par défaut

`useradd -D`

GROUP=100

HOME=/home

INACTIVE=-1

EXPIRE=

SHELL=/bin/bash

SKEL=/etc/skel



2. Commandes de gestion

Ajouter un utilisateur avec les paramètres par défaut de /etc/Login.defs

```
# useradd brahim
```

Attribuer le mot de passe unix98 à brahim

```
# passwd brahim
```

New UNIX password:

BAD PASSWORD: it is based on a dictionary word

Retype new UNIX password:

passwd: ail authentication tokens updated successfully

```
# grep brahim /etc/passwd
```

```
brahim:x:505 :505::/home/brahim:/bin/bash
```




2. Commandes de gestion

Connexion de brahim

Fedora Core Linux release 9.2

Kemel 2.6.7-10 on an i686venus

login:brahim

password:

/home/brahim \$

Créer les groupes develop, compta et achats, de GID: 1000, 1001 et 1002

groupadd -g 1000 develop

groupadd -g 1001 compta

groupadd -g 1002 achats



2. Commandes de gestion

Créer un utilisateur avec des paramètres spécifiques

```
# useradd -u 2000 -s /bin/bash -d /home/siham -c "Siham - Casa" siham
```

Visualiser le compte créé précédemment

```
# grep siham /etc/passwd
```

```
siham:!! :2000:2000:Siham - Casa:/home/siham:/bin/bash
```

```
# grep siham /etc/group
```

```
siham:x:2000:
```



2. Commandes de gestion

Ajouter siham aux groupes compta et develop

```
# usermod -G compta,develop siham
```

Lister les groupes de siham

```
# groups siham
```

```
siham: siham develop compta
```

Supprimer l'utilisateur siham (l'option -r demande la suppression de son arborescence)

```
# userdel -r siham
```



3. Modèles de fichiers de configuration

- Configuration de bash:
 - .bash_profile: script exécuté à la connexion
 - .bashrc: fichier de définition des alias
 - .bash_logout: script exécuté à la déconnexion
- Environnement graphique:
 - .Xdefaults: fichiers de définitions des ressources des applications X
- Application:
 - Modifier le profile d'un utilisateur



Exercice:

- Application:

Ecrire un shell script UNIX qui crée automatiquement les comptes d'utilisateurs dont le nom est saisi sur une ligne d'un fichier texte.



TP N° 1

Objectifs:

- Savoir créer un compte utilisateur
- Savoir gérer les utilisateurs et les groupes

Exercice 1:

Est-ce que l'utilisateur bin existe, si oui, quel est son uid ?

Exercice 2:

Comment feriez-vous pour vous connecter sous le compte de l'utilisateur « bin » ?

Exercice 3:

Existe-t-il d'autres comptes utilisateurs possédant les droits de *root* ?

Exercice 4:

A quels groupes appartient l'utilisateur bin ?



TP N° 1

Exercice 5:

Créez avec useradd, en gardant toutes les valeurs par défaut, l'utilisateur "ali". Quel est le groupe de "ali"?

Exercice 6:

Ajoutez "ali" au groupe staff. Au besoin, créez ce groupe.

Exercice 7:

Affichez les groupes de l'utilisateur "ali".

Exercice 8:

Connectez-vous au compte "ali" nouvellement créé de deux manières, à la connexion et grâce à la commande su. Expliquez les deux résultats.

Exercice 9:

Que faut-il faire pour pouvoir se connecter au compte "ali"?



TP N° 1

Exercice 10:

Changez le champ commentaire de "ali" en utilisant la commande vipw. Renseignez le champ avec le texte suivant: "Ali BenBrahim - Rabat".

Exercice 11:

Créez un compte utilisateur avec l'outil d'administration (linuxconf, ...).

Exercice 12:

Créez un compte « admin » d'UID 0. Il peut servir si, par exemple, on oublie le mot de passe de root



Objectifs

A l'issue de ce chapitre, vous connaîtrez :

- La structure des systèmes de fichiers,
- Leur gestion.



Contenu

- L'arborescence des fichiers
- Les principaux répertoires
- Les types de fichiers, les droits
- La gestion de l'arborescence
- Les attributs de fichiers



1. L'Arborescence:

1- Les principaux répertoires:

- / : Répertoire racine, là où tous les autres répertoires sont montés (accrochés)
- /bin : commandes UNIX, une partie des binaires du système et quelques commandes
exemple : ls, date, who
- /sbin : programmes exécutables indispensables à la gestion du système.
- /etc : quelques fichiers de configuration et des fichiers systèmes pour le démarrage
exemple : /etc/shutdown, /etc/init, /etc/passwd, /etc/group
- /dev : fichiers unité (périphériques, spéciaux)
exemple : /dev/lp0 imprimante 0



1. L'Arborescence:

- /home : partie où sont stockés les fichiers propres aux utilisateurs
- /var : fichiers temporaires de taille variable de quelques démons, de spools d'email et d'imprimantes, de logs, de locks ...
- /opt : lieu d'installation préféré des logiciels "modernes"
- /boot : image du noyau pour Linux
- /tmp : (temporary) fichiers temporaires, utilisés par l'éditeur de texte vi, les compilateurs...
- /usr : espace "standard"
- /usr/bin : pour les binaires
- /usr/lib : (library) fichiers d'information, pour les bibliothèques du langage C



1. L'Arborescence:

- /usr/include : fichiers d 'entête pour programmes C (.h)
- /usr/local : espace "non std", personnalisation locale du système
- /usr/local/bin : rajout de binaires en local
- /usr/local/lib : idem pour les bibliothèques
- /usr/local/include : idem pour les "includes"
- /usr/local/src : code source des différents programmes du système
- /usr/man : aide en ligne
- /mnt : (mount) montage de disquettes, donc la possibilité d'accéder aux données présentes dans la disquette à partir du répertoire /mnt, en utilisant les commandes d'UNIX
- /lost+found : (perdu et trouvé) contient les fichiers retrouvés par la commande fsck
- fsck : vérifie l'intégrité des données dans un SdF



1. L'Arborescence:

• Remarques:

- Aux yeux des utilisateurs, l'arborescence des répertoires est une entité transparente,
- Beaucoup de répertoire appartenant à l'arborescence forment différentes partitions sur un ou plusieurs disques, voire ordinateurs,
- Toute ressource est rattachée à l'arborescence du système de fichiers par un répertoire appelé point de montage.



2. Les types de fichiers

- Type de fichier: La commande « file »
- Visualiser un fichier: Les commandes « cat », « more » ou « less »
- Dump d'un fichier:
 - La commande « hexdump »: c'est un filtre qui permet d'afficher les fichiers spécifiés sous le format donné par l'utilisateur.
 - "hexdump" command is basically ASCII, decimal, hexadecimal, octal dump.
 - Hexdump provides many options to dump file contents. It can dump file contents into formats such as hexadecimal, octal, ASCII, decimal.
 - Hexdump is very helpful utility for debugging and verifying file contents written by any application program.



2. Les types de fichiers

➤ Exemples:

```
sanfoundry-> cat sample.txt
```

```
12345
```

- To display file contents in hexadecimal format(hexdump -x file_path):

```
sanfoundry-> hexdump -x sample.txt 00000000 3231 3433 000a 00000005
```

- To display file contents in decimal format(hexdump -d file_path):

```
sanfoundry-> hexdump -d sample.txt 00000000 12849 13363 00010 00000005
```

- To display file contents in octal format(hexdump -o file_path):

```
sanfoundry-> hexdump -o sample.txt 00000000 031061 032063 000012  
00000005
```




2. Les types de fichiers

➤ Exemples:

- To display file contents in character display format(hexdump -c file_path):
sanfoundry-> hexdump -c sample.txt 0000000 1 2 3 4 \n
00000005



2. Les types de fichiers

- La commande « od »:

La commande `od` ("octal dump") affiche le contenu d'un fichier en octal ou sous d'autres formats suivant les options ; l'option `-x` produit une sortie en hexadécimal, par exemple :

```
[nicolas]$ cat /etc/motd
Bienvenue sous Linux
Vous êtes connecté sur la machine doe.reso.local
[nicolas]$ od /etc/motd
0000000 064502 067145 062566 072556 020145 067563 071565 046040
0000020 067151 074165 053012 072557 020163 072352 071545 061440
0000040 067157 062556 072143 020351 072563 020162 060554 066440
0000060 061541 064550 062556 062040 062557 071056 071545 027157
0000100 067554 060543 005154
0000106
```



2. Les types de fichiers

- La commande « od »:

```
[nicolas]$ od -x /etc/motd
0000000 6942 6e65 6576 756e 2065 6f73 7375 4c20
0000020 6e69 7875 560a 756f 2073 74ea 7365 6320
0000040 6e6f 656e 7463 20e9 7573 2072 616c 6d20
0000060 6361 6968 656e 6420 656f 722e 7365 2e6f
0000100 6f6c 6163 0a6c
0000106
```



2. Les types de fichiers

Symbole	Signification	Rôle	Exemples
-	Fichier régulier	Ce sont les seuls fichiers capables de stocker de la donnée	Archives (.zip, .tar.gz, .deb...) Fichiers multimédias (.avi, .mp3, ...) Documents (.txt, .doc, .odt ...)
d	Dossier (directory)	Rassemble des fichiers	/, /home, /home/mando ...
l	Lien symbolique	Pointe sur un autre fichier	/lib/libc.so.6, /dev/cdrom
s	Socket	Communiquer par le réseau	/var/run/proftpd.sock
b	block device	Effectuer une opération sur un périphérique capable de stocker de la donnée	Disques SCSI/USB/SATA : /dev/sd* ou /dev/sr* Disques IDE : /dev/hd*
c	character device	Effectuer un périphérique incapable de stocker de la donnée (souris, webcam, ...)	/dev/psaux (souris) /dev/dsp (carte son) /dev/video* (webcam, carte tv...)



2. Les types de fichiers

- Classification des fichiers:

Fichier	Symbole (ls -l)	Création	Destruction
• Ordinaire	-	vi, ...	rm
• Répertoire	d	mkdir	rmdir, rm -r
• Périphérique caractère	c	mknod	rm
• Périphérique bloc b	mknod	rm	
• Socket locale	s	socket(2)	rm
• Tube nommé	p	mknod	rm
• Lien symbolique	l	ln -s	rm



2. Les types de fichiers

- Unités de disque:

Emplacement des unités: /dev

Fichiers son: /dev/audio (c),

Unité de CD-ROM: /dev/hdc (b)

Console: /dev/console (c)

Ports de modems: /dev/cua0 (c)

Unités de disquette: /dev/fd0 (b)

Unités à bandes: /dev/rft0, /dev/nrtf0 (b)



2. Les types de fichiers

- Exemples:

```
# file /bin/bash
```

```
/bin/bash: ELF 32-bit LSB executable, Intel 80386, version 1, dynamically  
linked, stripped
```

```
# file /etc/passwd
```

```
/etc/passwd: ASCII text
```

```
# tail -2 /etc/passwd > fichier
```

```
# cat fichier
```

```
ali:!!:500:500::/home/ali:/bin/bash
```

```
brahim:tbiYDmgxAckI2: 65536:65536::/home/brahim:/bin/bash
```



2. Les types de fichiers

- Exemples:

```
# hexdump -c /etc/passwd # -c : mode caractère
```

```
0000000a l i 1 2 3 : ! ! : 5 0 0 : 5 0
```

```
00000100 : : / h o m e / a l i 1 2 3 :
```

```
0000020 / b i n / b a s h \n a l i 1 : t
```

```
0000030 b i Y D m g x A c K i 2 : 6 5 5
```

```
0000040 3 6 : 6 5 5 3 6 : : / h o m e /
```

```
0000050 a l i 1 : / b i n / b a s h \n
```




2. Les types de fichiers

- Exemples:

```
# od -xc fichier # -x : hexadécimal -c : mode caractère
0000000  6970 7265 6572 213a 3a21 3035 3a30 3035
      a l i 1 2 3 : ! ! : 5 0 0 : 5 0
0000020  3a30 2f3a 6f68 656d 702f 6569 7272 3a65
      0 : / h o m e / a l i 1 2 3
0000040  622f 6e69 622f 7361 0a68 6170 6c75 743a
      / b i n / b a s h \ n a l i 1 : t
0000060  6962 4459 676d 4178 4b63 3269 363a 3535
      b i Y D m g x A c K i 2 : 6 5 5
0000100 3633 363a 3535 3633 3a3a 682f 6d6f 2f65
      3 6 : 6 5 5 3 6 : : / h o m e /
0000120 6170 6c75 2f3a 6962 2f6e 6162 6873 630a
      a l i 1 : / b i n / b a s h \ n
```



3. Les droits étendus

- Les neuf droits fondamentaux (valeurs octales : 400,200,100,40,20,10,4,2,1)
 - Fichier ordinaire
 - « read » : lire les octets du fichier (autorise par exemple la copie du fichier.)
 - « write » : ajouter, retirer ou modifier des octets.
 - « execute » : considérer le fichier comme une commande



3. Les droits étendus

- Répertoire

- « read » : connaître la liste des fichiers du répertoire (exécution, avec au plus l'option « -i », de la commande ls).
- « write » : créer ou de supprimer des fichiers d'un répertoire (nécessite obligatoirement le droit « x »).
- « execute » : accéder aux fichiers d'un répertoire. (clé indispensable pour que les droits d'accès d'un fichier soient contrôlés). A défaut, aucune opération n'est possible sur le fichier, quels que soient les droits de l'utilisateur. Le droit d'exécution est aussi nécessaire pour qu'un répertoire devienne le répertoire courant, grâce à la commande cd.



3. Les droits étendus

- Le « sticky bit» (valeur octale : 1000, valeur symbolique : lettre t)
 - Exécutable: il reste en mémoire, son chargement est rapide
 - Répertoire: la destruction d'un fichier est réservée au propriétaire
- Les droits d'endossement (valeurs octales : SUID=4000, SGID=2000, valeur symbolique : s)



3. Les droits étendus

- Exécutable

- **SUID** : le processus possède les droits du propriétaire du programme

- **SGID** : le processus possède les droits du groupe du programme

- exemple:

```
# ls -l /usr/bin/passwd
```

```
-r-sr-xr-x 1 root root 12345 oct 2 2001 /usr/bin/passwd
```

```
# ls -l /usr/bin/lpr
```

```
-r-sr-sr-x 1 root root 15068 oct 2 1998 /usr/bin/lpr
```



3. Les droits étendus

```
# ls -ld /var/spool/lpd/epson640
```

```
drwxr-xr-x 2 root lp      1024   avr 30 11:08  /var/spool/lpd/epson640
```



3. Les droits étendus

- Répertoire (SGID)

Les fichiers créés dans le répertoire appartiennent au groupe du répertoire et non au groupe de l'utilisateur qui les crée.

```
$ ls -ld repertoire
```

```
drwxrwsr-x 2 ali famille 1024 oct 10 10:15 repertoire
```

```
$ id
```

```
uid=500(ali) gid=500(ali) groups=500(ali)
```

```
$ cat > repertoire/f
```

Quel est le groupe

```
^D
```

```
$ ls -l repertoire/f
```

```
-rw-rw-r-- 1 ali famille 19 Oct 10 10:18 repertoire/f
```



4. Les commandes de gestion de l'arborescence

- Principales commandes:

ls -lR:	Affiche arborescence & caractéristiques
du:	Affiche une arborescence et/ou sa taille
rm -Rf:	Détruit une arborescence
cp -Rfp:	Copie une arborescence
chmod -R:	Change les droits des fichiers d'une arb.
chgrp -R:	Change le groupe des fichiers d'une arb.
chown -R:	Change le propriétaire/groupe des fichiers
find :	Effectue une recherche sur une arborescence



4- Les commandes de gestion de l'arborescence

- La commande find

Structure et technique de travail:

3 indications

A partir de quel répertoire commencer la recherche

Quels sont les critères de recherche à mettre en œuvre

Que doit-il se passer si un fichier répond à ce critère

Syntaxe:

`find répertoire [-critère [argument_critère]] ...`



4- Les commandes de gestion de l'arborescence

Critères de sélection:

- name nom du fichier
- type f, d, c, b, p, s, l
- size +-valeurcbk : taille >, < à valeur (car, blocs, ko)
sans signe, par défaut = taille
- user propriétaire
- group groupe
- perm +-droits : au plus/moins les droits (rwx)
- ctime nbjours : status fichier modifié depuis nbjrs



4- Les commandes de gestion de l'arborescence

- mtime nbjours : dernière modif. remonte à nbjrs
- atime nbjours : dernier accès remonte à nbjrs
- links

Critères d'exécution:

- print Affiche le chemin d'accès
- exec cmde {} \; Exécute cmde avec comme argument le fichier
- ok cmde {} \; Demande une confirmation pour exécuter la
cmde avec comme argument le fichier



4- Les commandes de gestion de l'arborescence

- Exemples:

```
# du -h
```

```
4.0k ./soft-jf/CVS
```

```
9.0k  ./soft-jf
```

```
10k
```

```
# df -h
```

File system	Size	Used	Avail	Use%	Mounted on
/dev/hda6	288M	134M	139M	49%	/
/dev/hda2	46M	2.6M	41M	6%	/boot
...					
usbdevfs	737M	737M	0	0%	/mnt/f



4- Les commandes de gestion de l'arborescence

- Exemples:

```
# ls -l fic*
```

```
-rw-r--r-- 1 ali ali 146 avr 30 12:29 fichier
```

```
-rw-r--r-- 1 ali ali 1044 avr 30 11:06 fic.doc
```

```
# chown brahim fichier # ne change que le propriétaire
```

```
# ls -l fichier
```

```
-rw-r--r-- 1 ali brahim 146 avr 30 12:29 fichier
```

```
# chown ali.projet fic.doc # change le propriétaire et le groupe
```

```
# ls -l fic.doc
```

```
-rw-r--r-- 1 ali projet 1044 avr 30 11:06 fic.doc
```



4- Les commandes de gestion de l'arborescence

- Exemples:

```
$ find . -name archive -print # rechercher tous les fichiers archive
```

```
./outils. copie/outils/archive
```

```
./outils/archive
```

```
$ find /home -name '*.c' -print      # afficher les fichiers dont le suffixe est .c
```

```
$ find . -type d -print      # afficher les noms des répertoires seulement
```

```
./outils
```

```
./outils/archive
```



4- Les commandes de gestion de l'arborescence

- Exemples:

```
$ ls -l f*          # le groupe de f3 n'est pas users
-rw-r--r-- 1 ali  users   24 Jun 11  16:51  f1
-rw-r--r-- 1 ali  users   21 Jun 11  16:51  f2
-rw-r--r-- 1 ali  adm    271 Oct 23  1994    f3
$                  # rechercher les fichiers dont le groupe n'est pas users
```



4- Les commandes de gestion de l'arborescence

- Exemples:

```
$ find /home/ali ! -group users -print  
/home/ali/f3
```

```
$ # rechercher les fichiers modifiés aujourd'hui
```

```
$ find /home -mtime 0 -print
```

```
$ # rechercher les fichiers qui n'appartiennent pas à ali,
```




4- Les commandes de gestion de l'arborescence

- Exemples:

dont la taille est supérieure

à 1000 octets ou le dernier accès remonte à moins de 30 jours

```
$ find /home ! -user ali \( -size +10000c -o -atime -30 \)
```

\$ # rechercher et détruire tous les fichiers réguliers dont la taille est nulle

```
$ find /home -type f -size 0 -exec rm -f {} \;
```



4- Les commandes de gestion de l'arborescence

- La commande **locate**

- # La commande **locate** a la même mission que **find** mais explore une base de données qui contient la liste des fichiers et répertoires. Cette base de données est en général automatiquement mise à jour quotidiennement par le système.
- # **locate nom_du_fichier.**
- # Bien que la commande **locate** soit très intéressante, elle ne possède pas la puissance des options de **find**. De plus, si vous créez des fichiers pendant la journée elle ne trouvera pas avant mise à jour de sa base de données (chaque redémarrage).



4- Les commandes de gestion de l'arborescence

Fonctionnement

locate utilise une base de données indexées listant tous les répertoires et fichiers du champ concerné située à `/var/lib/mlocate/mlocate.db`.

Par défaut, le fichier `/etc/cron.daily/mlocate` lance l'indexation chaque jour. Grâce à cela, **locate** va pouvoir trouver instantanément tout ce qui existait avant le dernier démarrage.



4- Les commandes de gestion de l'arborescence

Utilisation

locate trouvera instantanément le chemin de l'élément recherché , pour cela :

- Ouvrez la console. Tapez en cas de contrainte éventuelle :
- \$ sudo updatedb
- puis pour lancer la recherche, tout simplement :
- \$ locate nom_fichier
- la console affichera alors tous les chemins incluant le mot bidule.



5. Les attributs des fichiers

Les attributs d'un fichier sont des caractéristiques supplémentaires qui viennent s'ajouter, dans le système de fichiers «ext2» aux caractéristiques habituelles.

Les principaux attributs :

- a : Fichier log
- i : Fichier non modifiable
- s : Fichier physiquement détruit
- S : Fichier synchrone



5. Les attributs des fichiers

Les attributs:	Description:
A	L'heure et la date de dernier accès (« access time ») ne sont plus modifiées si cet attribut est positionné, ceci par souci de performance.
A	Un fichier qui possède cet attribut ne peut, en écriture, qu'être ouvert en ajout.
D	Le fichier ne sera pas sauvegardé par la commande dump.
i	Le fichier ne peut pas être modifié, détruit, renommé et il est impossible de créer des liens sur ce fichier. Seul l'administrateur root peut positionner cet attribut.
s	Quand le fichier est détruit, les blocs de données libérés sont remis à 0.
S	Les écritures dans le fichier sont immédiatement effectuées sur le disque. Le fichier est synchrone.



5. Les attributs des fichiers

- Les commandes

chattr : Modifier les attributs
lsattr : Afficher les attributs

La commande chattr permet de modifier les attributs d'un fichier. Sa syntaxe est la suivante:

```
chattr [-R] [-v version] [+-=Asacdisu] fichier ...
```

L'option « -R », comme dans de nombreuses commandes, permet de modifier les attributs de toute une arborescence.

L'option « -v version » permet de modifier la version d'un fichier, initialement fixée à 1.



5. Les attributs des fichiers

La commande `lsattr` visualise les attributs d'un fichier. Sa syntaxe est la suivante.

`lsattr [-Radv] [fichier. ..]`

L'option « -R » permet de visualiser les attributs de tous les fichiers d'une arborescence.



5. Les attributs des fichiers

L'option « -a », comme dans la commande `ls`, permet de visualiser les attributs des fichiers dont le nom commence par « . ».

L'option « -d », comme dans la commande `ls`, permet de visualiser les attributs des répertoires et pas leur contenu.

L'option « -v » affiche la version des fichiers.



5. Les attributs des fichiers

- Exemples

```
$ls
```

```
f1  f2  f3
```

```
$ chattr +Ss f2
```

```
$ su
```

```
password:
```

```
# chattr +i f1
```

```
# exit
```

```
$ lsattr
```

```
----i---      ./f1
```

```
s--S----- ./f2
```

```
-----      ./f3
```



5. Les attributs des fichiers

```
$ rm f1
```

rm: détruire le fichier protégé en écriture 'f1'? rm: o

Ne peut délier 'f1'.: Opération non permise

```
$ chattr -s f2
```

```
$ lsattr f2
```

```
---S--- f2
```



Enoncé du TP 2: Gestion des fichiers

Objectifs:

- Savoir changer les caractéristiques d'un fichier
- Gérer une arborescence de fichiers et de répertoires

Exercice 1:

Recherchez les répertoires de nom « cron » sur votre système.

Exercice 2:

Recherchez tous les fichiers tubes de votre système.

Exercice 3:

Recherchez tous les répertoires accessibles en écriture pour les autres.



Enoncé duTP 2: Gestion des fichiers

Exercice 4:

Connectez-vous avec le compte user01 et créez une arborescence de fichiers en utilisant les commandes suivantes :

```
$ cp /etc/passwd /etc/group -
```

```
$ mkdir ~/boot
```

```
$ cp /etc/inittab /etc/profile ~/boot
```

Listez cette arborescence en utilisant différentes commandes.

Exercice 5:

En étant connecté en tant qu'administrateur, mettez les droits 700 à l'ensemble des fichiers de l'utilisateur user01.



Enoncé du TP 2: Gestion des fichiers

Exercice 6:

En utilisant le manuel, retrouvez les différentes utilisations du droit SGID dans le système Linux.

Exercice 7:

Créez un fichier par la commande `cp` et rendez-le non modifiable. Listez ses attributs. Essayez de le modifier.



Chapitre 3: Les systèmes de fichiers

Objectifs

- ☐ A l'issue de ce chapitre, vous connaîtrez la structure des systèmes de fichiers, leur gestion.
- ☐ Vous serez capable de créer, monter, démonter un système de fichiers, et automatiser le montage.

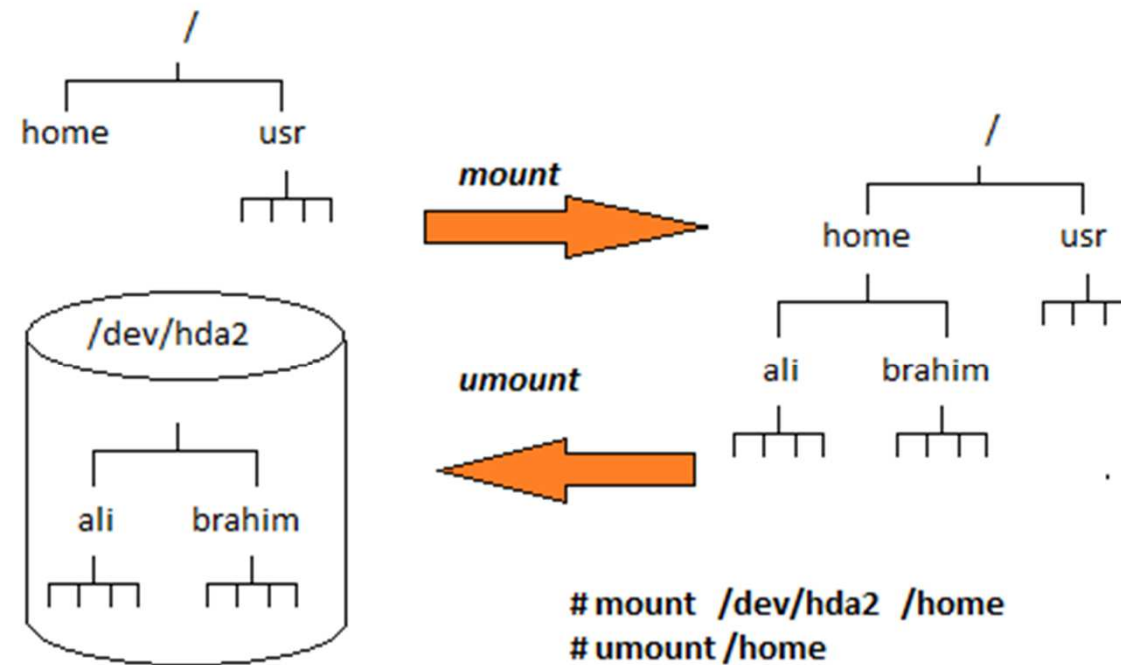
Contenu

- ☐ Montage et démontage de fichiers, Les inodes
- ☐ Les systèmes de fichiers journalisés
- ☐ La gestion de l'espace disque, la commande df
- ☐ Commandes de gestion des systèmes de fichiers
- ☐ L'automatisation du montage, La gestion des quotas



1- Rappels de cours:

1.1- Montage des systèmes de fichiers





- ☐ Le montage d'un système de fichiers consiste à attacher la racine de l'arbre du système de fichiers à un répertoire d'un système de fichiers déjà actif.
- ☐ Cette opération, qui s'appelle le montage du système de fichiers, est réalisée par la commande `mount`.
- ☐ La suppression du lien entre le répertoire de montage et le système de fichiers est effectuée par la commande `umount`.
- ☐ Les fichiers d'un système de fichiers ne sont accessibles, par les commandes usuelles (`cp`, `rm`, `mv`, `cat`, ...), que s'il est monté.
- ☐ Le système Linux mémorise les paramètres du montage des systèmes de fichiers dans le fichier `/etc/mnttab`.



1.2- Structure du système de fichiers ext2

Structure du système de fichiers

- ❑ Un système de fichiers est une structure de données.
- ❑ La commande mkfs, qui crée un système de fichiers, inscrit cette structure de données dans une partition.
- ❑ Tous les systèmes de fichiers comportent au moins trois tables système:
 - ✓ Le super bloc qui contient les informations clés concernant le système de fichiers.
 - ✓ La table des inodes, c'est-à-dire la table des descripteurs des fichiers. Chaque fichier est identifié de manière unique par le numéro de l'inode qui le décrit.
 - ✓ Les répertoires qui assurent une correspondance entre un nom de fichier et un numéro d'inode.



Structure d'un inode

- ☐ Le terme inode désigne le descripteur d'un fichier.
- ☐ Il contient les attributs du fichier, ceux affichés par la commande `ls -l`, et une table d'accès aux blocs de données.
- ☐ Il existe une table d'inodes par disque et l'espace qu'elle occupe est réservé à la création du système de fichiers sur le disque.
- ☐ La taille de la table des inodes est donc un paramètre statique important d'un système de fichiers, car elle fige le nombre de fichiers que l'on peut au plus créer sur le disque.

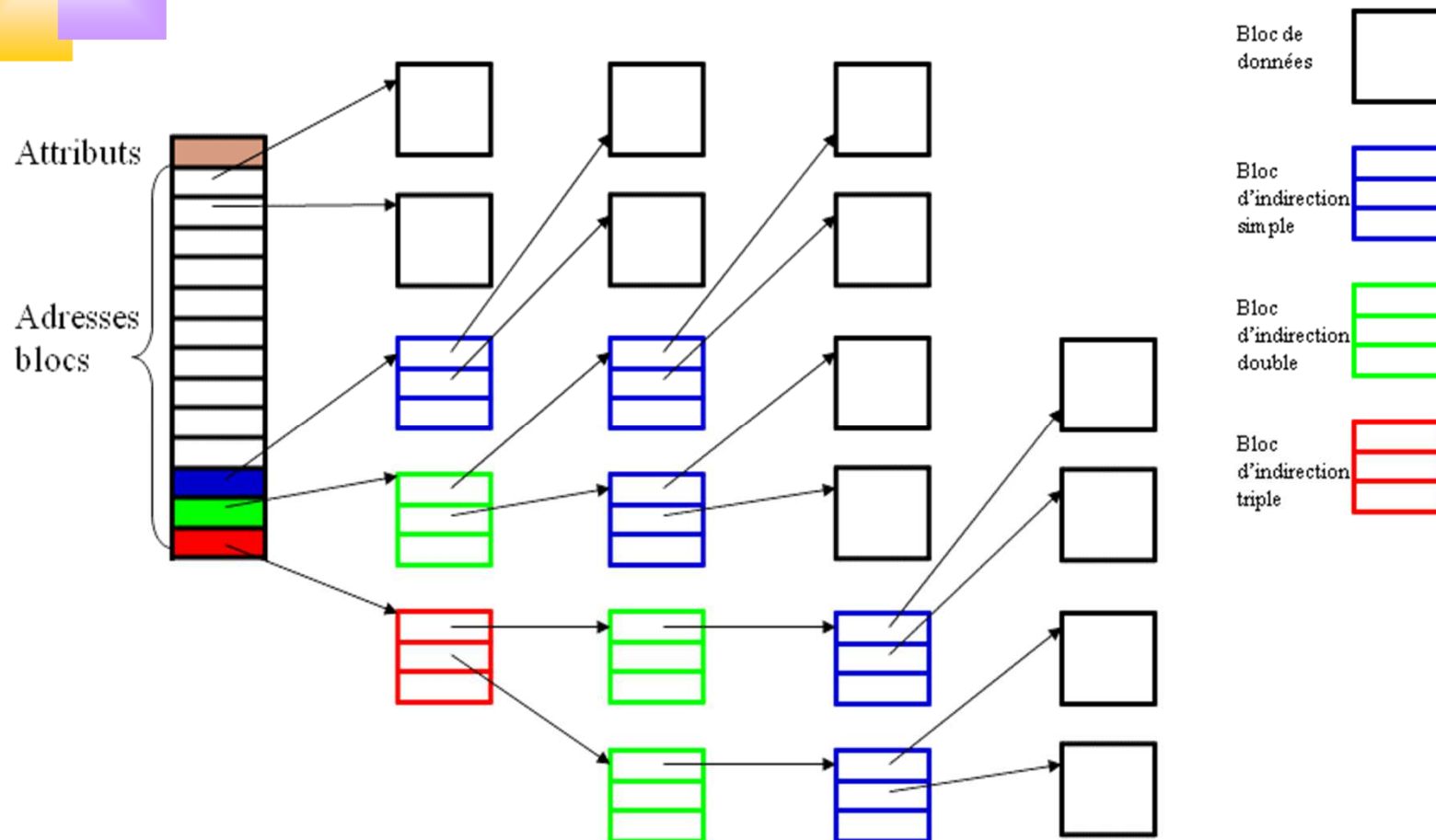


- ☐ La figure ci-dessous présente la structure de inode d'un système de fichiers de type « ext2 ».
- ☐ C'est actuellement le système de fichiers le plus performant sous Linux.
- ☐ C'est, du reste, le type de système de fichiers par défaut pour la commande mkfs qui crée les systèmes de fichiers.
- ☐ Dans le schéma, le sigle "BD" désigne un bloc de données de 1, 2 ou 4 ko et le sigle "BA" un bloc d'adresses.



Droits d'accès et type du fichier
 UID du propriétaire
 GID du groupe du fichier
 Taille en octets
 Date et heure de création
 Date et heure de dernière modification
 Date et heure de dernier accès
 Date et heure de suppression
 Nombre de liens matériels
 Nombre de blocs
 Nombre de fragments
 Drapeaux
 Réserve
 Fichier ACL
 Répertoire ACL
 Adresse de fragments
 Nombre de fragments
 Adresse du premier bloc de données

1					BD
2					BD
3					BD
4					BD
5					BD
6					BD
7					BD
8					BD
9					BD
10					BD
11					BD
12	Adresse du douzième bloc de données				BD
13	Adresse du bloc d'indirection de niveau 1	BA	BD		
14	Adresse du bloc d'indirection de niveau 2	BA	BA	BD	
15	Adresse du bloc d'indirection de niveau 3	BA	BA	BA	BD





La structure du système de fichiers ext2

- ❑ Le système de fichiers « ext2 » est le plus utilisé sous Linux.
- ❑ Il dérive du système de fichiers ffs (« Fast File System ») créé à l'université de Berkeley.
- ❑ Sa structure est la suivante:
 - Un secteur de boot de 512 octets qui ne fait pas partie du système de fichiers.
Ce secteur peut être vide.
 - Une suite d'ensemble de blocs.

Chaque ensemble de blocs contient les informations suivantes:

- Une copie du super bloc.
- Une table de descripteurs qui indique l'emplacement des tables suivantes.



- La table d'allocation des blocs de l'ensemble courant sous forme de bitmaps.
- La table d'allocation des inodes de l'ensemble courant sous forme de bitmaps.
- La table des inodes de l'ensemble courant.
- Les blocs de données des fichiers de l'ensemble courant.

Le super bloc contient notamment les informations suivantes:

- La taille des blocs.



- La taille en bloc du système de fichiers.
- Le nombre de blocs libres.
- Le nombre de blocs réservés à root.
- Le nombre de inodes
- Le nombre de inodes libres
- L'état du système de fichiers: monté ou démonté.
- Le nombre de blocs par ensemble de blocs.



- Le nombre d'inodes par ensemble de blocs.
- Le nombre maximal de montage avant de réaliser un fsck automatique.
- Le nombre de montages réalisés depuis le dernier fsck.
- Date du dernier fsck.
- Temps maximum entre deux fsck.
- La table des inodes.



Dans la table des inodes, plusieurs entrées sont réservées, notamment:

- Le premier inode qui mémorise les « bads blocs» du système de fichiers.
- Le second inode qui est celui du répertoire racine du système de fichiers.



Les différents types de SdF

minix : le premier FS utilisé par Linux

ext2 : Le FS standard du système Linux

msdos : Le FS FAT16 de MS-DOS et Windows

vfat : Le FS FAT32 de Windows

sysv : Le FS d'UNIX System V

smb : Le FS utilisant le protocole SMB de Microsoft

nfs : Le FS réseau de Sun (Network File System)

Iso 9660: Le FS utilisé par les CD-ROM



1.3- La gestion de l'espace disque, les commandes df et du:

- ☐ L'espace disque est une ressource précieuse, même si les capacités des disques ont considérablement évolué ces dernières années.
- ☐ L'administrateur doit en contrôler l'usage.
- ☐ Il dispose pour cela de commandes simples et pratiques dont nous ne faisons ici que rappeler la signification et donner quelques options significatives.



- ☐ La commande **df** indique l'espace libre des disques contenant des systèmes de fichiers montés.
- ☐ Les tailles sont affichées en kilo-octets.
- ☐ Les principales options sont:
 - i Affiche les informations sur l'utilisation des inodes et non des blocs.
 - k Affiche les tailles en kilo-octets. L'option est significative dans le cas où la variable `POSIXLY_CORRECT` est définie
 - T Affiche également le type de système de fichiers.



- ❑ La commande **du** affiche le nombre de blocs d'un kilo-octet utilisés par une arborescence qui peut coïncider avec celle d'un système de fichiers.
- ❑ Les principales options sont:
 - s Affiche le total seulement.
 - k Affiche les tailles en kilo-octets



1.4- Panorama des commandes de gestion des systèmes de fichiers

- ❑ Les commandes de gestion de systèmes de fichiers sont nombreuses.
- ❑ Dans Linux, on trouve deux familles de commandes :
 - Les commandes génériques héritées du système UNIX, comme mkfs et fsck.
 - Les commandes spécifiques à Linux et particulièrement au système de fichiers ext2.



1.4- Panorama des commandes de gestion des systèmes de fichiers

Commandes génériques:

- mkfs: Crée un FS
- mount: Monte un FS
- umount: Démonte un FS
- fsck: Vérifie un FS
- df: Espace libre
- du: Espace occupé
- lsof: Identifie les processus



1.4- Panorama des commandes de gestion des systèmes de fichiers

Commandes propres à ext2

- mke2fs: Crée un FS
- e2fsck: Vérifie un FS
- tune2fs: Paramètre un FS
- dumpe2fs: Informations sur le super bloc et les groupes de blocs
- debugfs: Débogue un FS



1.5- Montage d'un disque

Etape1: Création d'un disque virtuel (virtuel)

Etape2: Création d'une partition (fdisk)

Etape3: Formater avec mkfs

Etape4: Montage (/mnt)

Etape5: Démontage (valable pour une session)

Etape6: Montage Automatique (permanent)



Etape1: Création d'un disque virtuel (virtuel)

- 1) Démarrer une machine virtuelle (vmware)
- 2) Choisir fedora sans le lancer
- 3) Edit virtuel machine setting (création d'un disque virtuel) → Add →
Hard disque → Next → create a new virtual disk → Next → SCSI
[avec SCSI: les partitions/secteurs portent les noms: sda/sdb/...
avec IDE: les partitions/secteurs portent les noms: hda/hdb/...
disque SATA: accès rapide aux secteurs/généralement cher]
→ Next



Etape1: Création d'un disque virtuel (virtuel)

- Disk size (GB) = 2.0 (pour test)
- Next
- fc9-4.5.0.vmk
- Finish
- OK
- Power of this Virtual Machine (démarrer)
- En tant que « root »
- Pour chercher le nom du disque crée: `$more /var/log/dmesg`
[vous trouverez sda puis sdb]



Etape2: Création d'une partition sur le disque crée

→ `$fdisk /dev/sdb`

→ m (aide)

→ p (afficher)

→ n (new)

→ p

→ 1 (numéro de partition (sdb1))

→ ok

→ ok

→ p

→ w (sauvegarder les infos)



Etape3: formater avec la commande mke2fs

→\$mke2fs /dev/sdb1

Etape4: Montage (/mnt)

→\$mount

→regarder (sdb1 n'est pas encore monté)

→\$mount /dev/sdb1 /mnt

→\$mount (regarder sdb1 en rw)



Etape5: Démonter la partition

- `$umount /dev/sdb1` ou `mount /mnt`
- `$mount` (pour vérifier)

Etape6: Montage automatique (permanent)

- `$more /etc/fstab`
- `$gedit /etc/fstab`
- à la fin du fichier, rajouter la ligne:
`/dev/sdb1 /mnt ext2 defaults 0 0`
- sauvegarder et rebooter le système
- `$mount` (pour vérifier)



TP n°3: Les systèmes de fichiers

Objectifs:

- Savoir monter et démonter un FS
- Savoir surveiller l'espace disque
- Savoir gérer l'arborescence des fichiers en tenant compte des FS

Exercice 1:

Quels sont les différents FS montés, à quels fichiers spéciaux sont-ils associés, quels sont leurs répertoires de montage ?

Exercice 2:

Quelle est la place libre, en blocs de 512 octets, restant sur chaque FS ?

Exercice 3:

Quelles sont les tailles de blocs possibles dans un système de fichiers de type ext2 ?



TP n°3: Les systèmes de fichiers

Exercice 4:

Prenez un CD-ROM ayant un format compatible ISO 9660 (la plupart le sont), ou une clé USB. Montez le périphérique, listez son arborescence et démontez-le.

Exercice 5:

Les systèmes de fichiers nfs et smb sont des systèmes de fichiers distribués sur un

Exercice 6:

Résumez brièvement le rôle des principales commandes de gestion de système de fichiers de Linux.



TP n°3: Les systèmes de fichiers

Exercice 7:

On désire installer un logiciel pour jouer au tetrakis qui fait 6 Mo dans le répertoire */home/ali/games/tetrakis*, En utilisant le résultat de la commande `df`, donnée ci-après, est-ce que l'opération est réalisable? Si non, est-ce qu'il y a une autre solution?

#df

Filesystem	lk-blocks	Used	Available	Use%	Mounted on
/dev/hda1	796033	318675	436234	42%	/
/dev/hda6	101471	95777	454	100%	/home
/dev/hda5	101471	28	96203	0%	/usr



TP n°3: Les systèmes de fichiers

Exercice 8:

L'utilisateur ali visualise son fichier *.profile* (*/home/ali/.profile*) grâce à la commande *more*, L'administrateur peut-il démonter le FS */home*? Si non, pourquoi? Que doit-il faire pour y arriver?

Exercice 9:

Le FS */dev/hda8* est monté sur le répertoire */games* . Que doit-on faire pour que ce FS ne soit accessible qu'à l'utilisateur ali?

Exercice 10:

Sans utiliser la commande *debugfs*, comment l'administrateur peut-il retrouver tous les fichiers liés à un fichier dont on connaît le chemin, exemple:

/home/ali/.profile ?



TP n°3: Les systèmes de fichiers

Exercice 11:

Détruisez tous les fichiers de nom « core ».

Exercice 12:

Quels sont les gros fichiers (taille supérieure à 1 Mo) modifiés il y a peu de temps (dans la journée) ?

Exercice 13:

Quelles sont les commandes de gestion de quotas sur votre système?