

# Cours GNU/Linux

## Les Filesystems

# présentation

## Organisation des données sur disque

Gestion des droits et des accès

Gestion de l'Arboressance

## Gestion des écriture sur disques

Géré par le Kernel (I/O = matériel)

Maîtrise de plusieurs types de FS

## Différentes sortes :

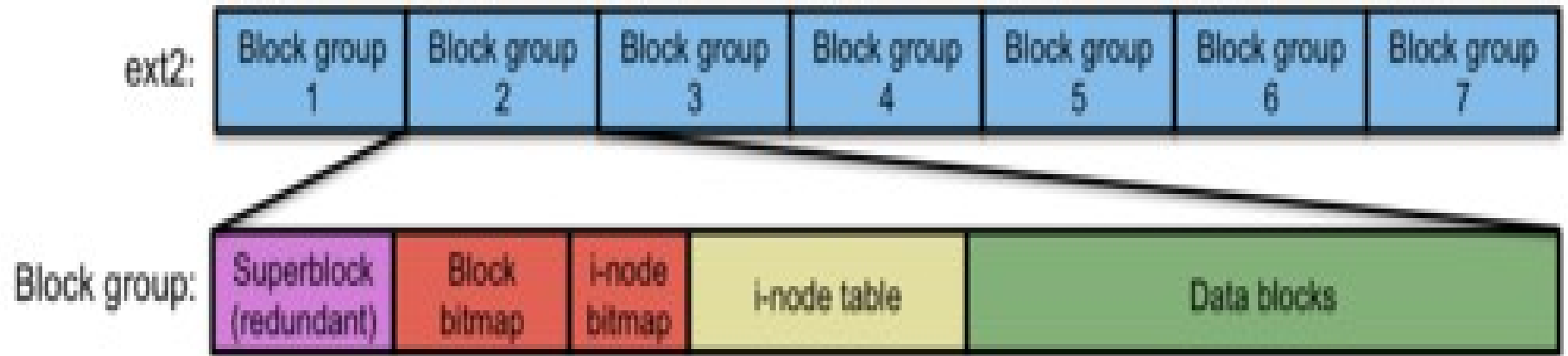
### Locale

- Sur support de stockage (disques / mémoire)

### Distant

- Système de fichier réseau (NFS CIFS)

# File Systems



Ext2 : Groupe de cylindre ou groupe de blocs contigus

Copie du Super bloc

Bitmap des inodes et de blocs

Tables d'inodes

Blocs de données

# Super bloc

## Architecture du FS

- La taille du FS

- La taille des blocs

- La taille des groupes de blocs

- La taille et la position des tables d'inodes et de blocs

## Statut du filesystem

- La liste et le nombre des blocs libre et occupé

- Statut : Ouvert / fermé synchronisé

# Un fichier

## Une inode :

Type de fichier

- - (ou f) : fichier de données
- d : directory
- l : lien symbolique

Mode (644/rwxr-xr-x),

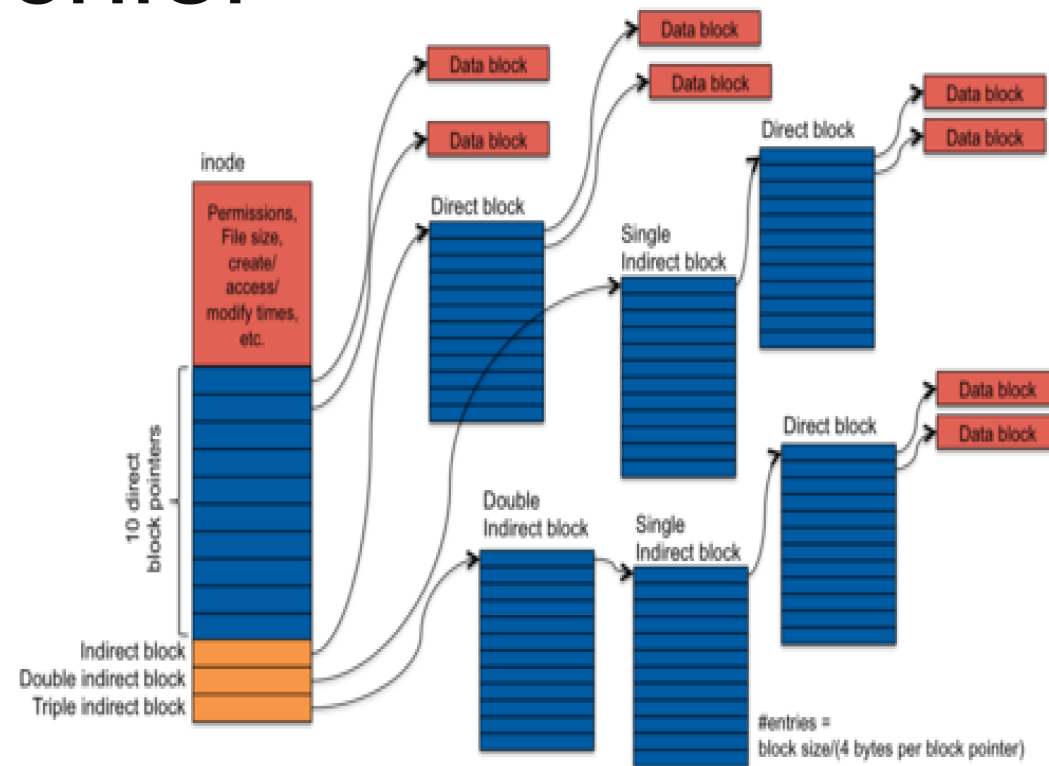
owner et group (uid gid),

Nombre de blocks et Taille en octets,

Timestamps (atime ctime mtime),

Nombre de liens (nombre de référence de ce fichier)

Pointeurs adressant les blocs de données



Un fichier = une inode sur un filesystem

Un fichier n'a donc pas de nom

et donc pas d'extention de fichier (sous windows .exe .dll)

# Les fichiers spéciaux 1/2

## Les répertoires

Les données sont spécialement formatés

- La liste de fichiers
  - inode / nom de fichier
  - Parfois des meta données de l'inode du fichier

## Les liens physiques

Une autre entrée dans un répertoire pour la même inode

## Exemples :

```
touch fic1 fic2 ; ln fic1 fic3 ; ls -ali
```

inode	T-mode	#ln	uid	gid	size	date	nom
656530	drwxr-xr-x	2	alan	alan	4096	2013-11-20 17:20	.
655362	drwxr-xr-x	56	alan	alan	12288	2013-11-20 17:10	..
<b>657769</b>	-rw-r--r--	2	alan	alan	0	2013-11-20 17:21	fic1
657676	-rw-r--r--	2	alan	alan	0	2013-11-20 17:21	fic2
<b>657769</b>	-rw-r--r--	2	alan	alan	0	2013-11-20 17:21	fic3

```
echo TOTO > fic3 ; ls -ali fic1 fic3
```

<b>657769</b>	-rw-r--r--	2	alan	alan	5	2013-11-20 17:21	fic1
<b>657769</b>	-rw-r--r--	2	alan	alan	5	2013-11-20 17:21	fic3

# Les fichiers spéciaux 2/2

## Les liens symboliques

Un lien symbolique est un fichier donc une inode

Les données

- **une chaîne de caractère** : chemin relatif ou absolu vers le fichier cible

Sous extX, Si la chaîne est inférieure à 60 caractères elle est contenue à la place des pointeurs de blocs

Exemple :

```
ln -s ./fic1 fic4 ; ln -s fic1 fic5 ; ls -ali
```

inode	T-mode	#ln	uid	gid	size	date	nom
657769	-rw-r--r--	2	alan	alan	5	2013-11-20 17:21	fic1
657676	-rw-r--r--	2	alan	alan	0	2013-11-20 17:21	fic2
657769	-rw-r--r--	2	alan	alan	5	2013-11-20 17:21	fic3
657754	lrwxrwxrwx	2	alan	alan	6	2013-11-20 17:21	fic4 -> ./fic1
657757	lrwxrwxrwx	2	alan	alan	4	2013-11-20 17:21	fic5 -> fic1

# Fonctionnement 1/2

## Création d'un fichier:

### Récupération d'une Inode libre

- Marque comme occupée dans la bitmap
- écriture de l'inode (ctime, uid gid mode etc...)

### Inscription du fichier dans le répertoire

- Modification des données du répertoire (ajout d'une entrée)
- Modification de l'inode du répertoire

### Écriture des données

- Mise à jour des la tables de blocs libres
- écriture des données sur des blocs libre
- Modification de l'inode
  - Pointeurs de blocs
  - Date de modification des données



# Fonctionnement 2/2

## Supression d'un fichier

Supression de l'entrée dans le répertoire

- Modification des données du fichier répertoire
- Modification de l'inode du répertoire (mtime)

Libération de l'Inode

- Modification de la table des inodes (inode marquée comme libre)

Libération des données

- Mise à jour des la tables de blocs libres

Cet ordonancement assure que :

Les répertoires ne contiennent que des inodes valides

- Inode réservé avant d'être référencé dans le répertoire

Les inodes ne référence que des bloc effectivement occupé

- Écriture des données avant de les pointer

Un bloc de données n'est pointé que par une inode

- Marqué comme occupé avant d'être pointé par une inode

# A maitriser

## Arborescence Unix

### Commandes :

df

mount / unmount

fdisk

mkfs / mke2fs

fsck

### Fichier : fstab

# Commande df

df : permet de montrer l'usage des filesystems actuellement gérés

df -h : h pour human readable

Format plus lisible

df -l : l pour local

Uniquement les filesystems locaux

df -i : i pour inode présente l'usage des inodes

% d'occupation des inodes

Exemple :

- df -h

Filesystem	Size	Used	Avail	Use%	Mounted on
/dev/sda1	38G	3.5G	32G	10%	/
none	498M	212K	497M	1%	/dev
none	502M	124K	502M	1%	/dev/shm

- df -i

Filesystem	Inodes	IUsed	IFree	IUse%	Mounted on
/dev/sda1	2482176	201534	2280642	9%	/
none	127248	677	126571	1%	/dev
none	128308	6	128302	1%	/dev/shm

# Montage / démontage

Ouvrir ou fermer un filesystem

Le rendre disponible dans l'arborescence

Commande mount / umount

Syntaxe :

- Mount -t \$Type -o \$Options \$Partition \$Point-de-montage

\$Type : type de filesystem / format du système de fichier

\$Options : options de montage (exemple ro pour read only, voir le man)

\$Partition : fichier device/materiel (sous /dev) formaté au format \$Type

\$Point-de-montage : répertoire de l'arborescence sur lequel on “monte” le filesystem

# mount / umount / sync

Outils système donc utilisé avec l'identité root

La commande mount sans argument retourne l'état des montages actuels

Exemple :

```
mount
/dev/sda1 on / type ext4 (rw,errors=remount-ro)
proc on /proc type proc (rw,noexec,nosuid,nodev)
none on /sys type sysfs (rw,noexec,nosuid,nodev)
```

La commande umount permet de démonter un système de fichier :

Il disparaît de l'arborescence et est fermé correctement

La commande sync permet de forcer l'écriture des données sur le support physique (vidage des caches)

# Fichier /etc/fstab

Configuration statique des systèmes de fichiers sur le système (norme POSIX)

Exemple :

/dev/sda1	/	ext4	defaults	1	1
/dev/sda2	/boot	ext4	defaults	1	2
/dev/sda3	swap	swap	defaults	0	0
tmpfs	/dev/shm	tmpfs	defaults	0	0
sysfs	/sys	sysfs	defaults	0	0
proc	/proc	proc	defaults	0	0

Format :

Device / partition

Point de montage

Type

Option

Fréquence de dump (sauvegarde au sens POSIX)

fs\_passno : ordre de vérification de l'état du filesystem au boot

0: pas de vérification 1: en premier (pour /) 2: tout les suivants

# Fsck

## Check des filesystems

Travail directement sur la partition

FS Démonté ou monté en ReadOnly

Vérifie la cohérence du filesystem

- Validation du Super bloc
- Vérification des inodes
  - Contenu de l'inode
  - Block associés
  - Tailles
- Structure des répertoires
- Structure de l'arbo (connexion inter répertoires)
- Nombre de références (nombre d'entrée de répertoire vers inode)
- Mappage de blocs (bloc libre)
- Correction du mappage de blocs

## Corrige les erreurs suivantes :

- Inodes non référencé (dans un répertoire)
- Le nombre de liens sur les inodes
- Blocs non libéré (non utilisé mais absent de la liste des blocs disponible)
- Blocs utilisé mais marqué comme libre
- Inconsistance du super bloc

# TD Gestion FS

TD créez un FS ext3 sur un volume supplémentaire de 200MB sur une de vos VM

Créez une partition : “fdisk” “dmesg”

Créez un filesystem sur la partition créée : “mkfs” / “mke2fs”

Montez le FS sur /mnt

Démontez le

**Cassez le !** : “dd if=/dev/zero of=/dev/sdb1 bs=1024 count=2”

**Reparez le !**



Pour approfondir

[alan.simon@free.fr](mailto:alan.simon@free.fr)

# Interface système : VFS

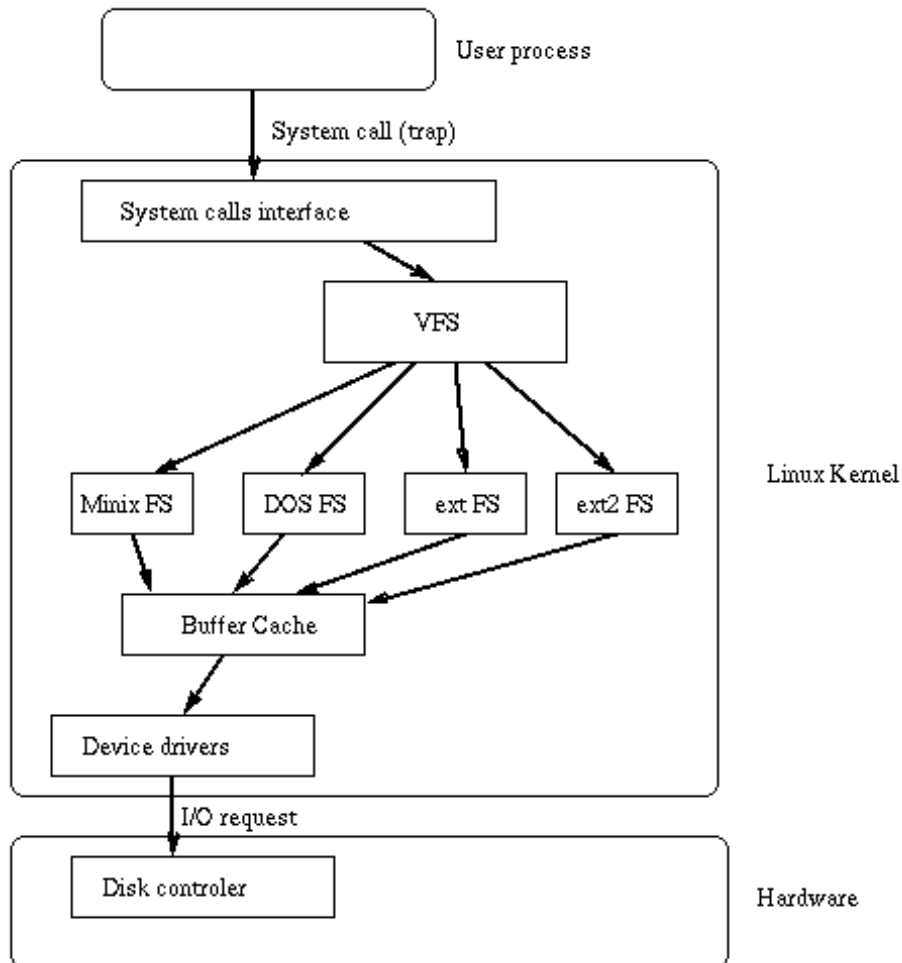
## VFS

Ensemble de fonction  
systeme

Couche d'abstraction

Utilisation indépendante de  
type de filesystem

Intégration simplifié des  
filesystems



# VFS

Filesystems descriptor : Initialisé au montage via l'utilisation de la fonction de montage associée au filesystem.

- Lecture du super block

- Initialisation des variables internes VFS (device, fs type, etc...)

- Retourne le descripteur de filesystem ouvert à VFS

- Pointeur vers les fonction du kernel associée au filesystem
  - Donnée Interne du filesystem (superblock)

Inode : Pointeur vers les fonctions de gestion des inode

- Create

- Unlink

Open Files : Pointeur vers les fonctions de gestion de fichier

- Read

- Write

# Isof / fuser

Fuser : Permet d'identifier un processus “bloquant un fichier” (descripteur de fichier ouvert par le processus)

fuser \$file

Test :

- vi ./fic1 & fuser ./fic1 : retourne le pid du processus qui a le fichier .fic1 ouvert.
- ps -ef | grep \$Process-ID

Isof : pour “list open file”

retourne tout les descripteur de fichiers actuellement ouvert sur le système

test : Isof | grep fic1

# Les devices

`ls -ald /dev/s*`

```
brw-rw---- 1 root disk 8,    0 Nov 21 18:36 /dev/sda
brw-rw---- 1 root disk 8,    1 Nov 21 18:36 /dev/sda1
brw-rw---- 1 root disk 8,    2 Nov 21 18:36 /dev/sda2
brw-rw---- 1 root disk 8,    5 Nov 21 18:36 /dev/sda5
crw-rw---- 1 root disk 21,   0 Nov 21 18:36 /dev/sg0
```

Major et Minor number

Identification du driver

Identification du device

`ls -l | grep lsf`

Type : DIR, REG, FIFO, CHR

# TP

Migrez /var/log dans un filesystem dédié

mkfs ...

mount ... /mnt/tmp

rsync ... /var/ /mnt/tmp/

umount ... / mount ...

ls -l ...

vi /etc/fstab

reboot

df