

Système d'Exploitation -Système de Gestion des Fichiers-

Med. AMNAI

Filière SMI-S4

Département d'Informatique

2023-2024

Plan

① Introduction

Plan

- 1 Introduction
- 2 Méthodes d'allocation des fichiers

Plan

- 1 Introduction
- 2 Méthodes d'allocation des fichiers
- 3 Description des fichiers

Plan

- 1 Introduction
- 2 Méthodes d'allocation des fichiers
- 3 Description des fichiers
- 4 Montage et maintenance des partitions

Notion de Fichier

- Toutes les informations extérieures au processus sont des fichiers;
- Un fichier peut être associé au clavier, à l'écran, à l'imprimante, ...
- Quatre catégories de fichiers :
 - **Ordinaires** (.txt, .exe,...) : Suite d'octets sans structure (ex. données ou programmes).
 - **Répertoires** : Noeud de la hiérarchie des fichiers ; ce sont des fichiers de références à d'autres fichiers.
 - **Spéciaux** (périphériques, ..) : Fichier virtuel, représentation d'un organe périphérique ; accessible par un programme de pilote.
 - **Liens symboliques** : contenant la chaîne de caractères représentant le nom d'un autre fichier ; (référence indirecte, graphe quelconque, arborescence.)

Système de Gestion de Fichiers (SGF)

Système de Gestion de Fichiers (SGF) :

- Partie du S.E. qui se charge de **gérer** le **stockage** et la **manipulation** des fichiers sur une unité de stockage (HDD, CD, USB, partition,...).
- Partie du S.E. qui maintient les données sur les périphériques mémoires (disques, disquettes, clefs USB);
- Définit la **structure** d'un disque sous forme d'une arborescence : dossiers, fichiers, etc. . . ;
- Fournit une **interface** (conviviale) à l'utilisateur.

Rôles du SGF

Rôle principal d'un SGF est de **gérer** les fichiers et d'offrir les **primitives** de manipulation de ces fichiers :

- Fournit une interface simplifiée de manipulation des fichiers (*créer, ouvrir, fermer, copier, renommer* des fichiers) ;
- Gérer l'**organisation** des fichiers sur le disque comme l'allocation et la libération d'espace ;
- Gérer les fichiers dans un milieu **Multi-Utilisateurs** ;
- Gérer l'**espace** libre sur le disque dur ;

Objectifs

Comment associer les fichiers à un espace sur le disque ?

- **Optimiser** l'utilisation des disques ;
- **Emplacement** des données ;
- **Minimiser** les temps d'accès.

Exemples

- **Windows**

- MS-DOS ou FAT (File Allocation Table), FAT32 ;
- NTFS ;
- WinFS ;
- Joliet (CD-ROM)

- **Linux :**

- ufs (unix BSD) ;
- ext2fs, ext3fs ;
- reiserfs ;
- iso9660 (CD-ROM) ;

Niveaux d'Accès

- Différents niveaux d'accès
 - Disque **Physique** (surface, cylindre, secteur) ;
 - Disque **Logique** (numéro de block) ;
 - Fichier Logique.
- Niveaux d'accès physique/Logique
 - Bas niveau (**Physique**) : **block** ;
 - Haut niveau (**Logique**) : **fichier, dossier** ;

Niveaux d'Abstractions

L'utilisateur

vision d'une arborescence

Description logique des fichiers

une suite de blocs contenant des fichiers

Organisation logique du disque

(rangement des fichiers dans les blocs)

Vue d'une suite de blocs

Structure physique du disque

Bande magnétique

Comment stocker un fichier ?

- Fichier sur Disque
 - Unité de stockage sur le disque : les **blocs**
- Stockage d'un fichier
 - **Dans un bloc**, si la taille est inférieure à celle d'un bloc ;
 - **Dans plusieurs blocs**, si la taille du fichier dépasse celle d'un bloc ;
- **Problème** : Quels blocs choisir ?

Allocation des blocs

- A la création d'un fichier, le SGF doit :
 - **Attribuer** (Allocation des blocs) de l'espace sur disque ;
 - Mémoriser son implantation et son **organisation** sur le disque.
 - Maintenir ces **informations** en cas de **modifications** de fichiers.
- Le stockage consiste à **mémoriser les adresses des blocs** de chaque fichier.
- Différentes méthodes sont utilisées : *Allocation contiguë*, *non contiguë* (liste chaînée), *liste chaînée indexée* et *Noeud d'informations (i-node)*

1-Allocation Contiguë

- **Principe**
 - Le système choisit parmi les ensembles de **blocs libres contigus** un ensemble de blocs libres contigus suffisant pour y implanter le fichier.
- **Avantages**
 - **Simplicité** : il suffit de mémoriser l'adresse du premier bloc, pour localiser un fichier ;
 - **Performances** : **Vitesse d'accès et l'accès direct** à n'importe quelle partie du fichier.
- **Inconvénients**
 - **Nécessité de connaître la taille** du fichier au moment de la création.
 - **Gaspille d'espace** en raison de fragmentation du disque.
 - **Difficilement utilisable** si le fichier augmente de taille.

2-Allocation Non contiguë (Liste chaînée)

- **Principe**
 - Le **fichier** est découpé et sauvegardé sous forme **des blocs dans une liste chaînée**. Les blocs de fichiers sont écrits sur des blocs du disque.
- **Avantages**
 - **Rapidité d'enregistrement** : Mémorisation seulement de l'adresse du **premier bloc**.
 - **Possibilité d'ajout** de blocs si la taille du fichier augmente.
- **Inconvénients**
 - Lecture d'un élément sur le bloc **n**, nécessité de parcourir les **n-1** blocs précédents.
 - Le pointeur sur le **bloc suivant** occupant quelques **octets**.

3-Liste chaînée indexée

- **Principe**
 - Pour chaque bloc, **retirer le pointeur suivant** et le placer dans une **table stocké en mémoire**.
- **Avantages**
 - Eliminer les inconvénients de la méthode précédente (Liste chaînée) ;
 - Mémorisation de l'adresse du premier bloc.
- **Inconvénients**
 - Tables énormes en mémoire d'une façon permanente.

4-Noeud d'information (i-node)

- **Principe**
 - Associer à chaque fichier une table appelé **noeud d'informations (i-node)**. Elle mémorise la cartographie des informations du fichier.
 - **L'inode** est la structure qui contient toutes les informations sur un fichier donné à l'exception de sa référence (son **nom**) dans l'arborescence.
 - **L'inode identifie de façon unique le fichier** dans le système de fichier.
 - La **table des inodes** est un **point** d'entrée **vers tous les fichiers** d'une partition.
 - **L'inode** est le passage obligatoire de tous les échanges entre le système de fichier et la mémoire.
 - Chaque inode possède un **index** dans la **table** des inodes.
 - **ls** avec l'option **-i** permet d'**afficher** l'inode d'un fichier.

4-Noeud d'information (i-node)(suite)

- Un répertoire n'est qu'un autre fichier particulier contenant une liste d'associations **<nom_de_fichier, inode>**
- Les informations (attributs) stockées dans un i-node disque sont :
 - utilisateur propriétaire,
 - groupe propriétaire,
 - type de fichier : fichier ordinaire, répertoire, ...
 - droits d'accès,
 - date de dernier accès, date de dernière modification,
 - date de dernière modification de l'inode,
 - taille du fichier,
 - adresses des blocs-disque contenant le fichier.
 - ...

Liens sous Unix

Plusieurs liens vers un même fichier peuvent existés en même temps :

- Pour faire apparaître un même fichier dans plusieurs **répertoires**, ou sous des **noms différents**.
- Ils évitent les **duplications** et assurent la **cohérence** des mises à jour.
- Types de liens :
 - Liens **durs** (hard) ou **physiques** ;
 - Liens **symboliques**.

Liens physiques

Associent deux ou plusieurs fichiers à un **même espace** sur le disque :

- Partageant le **même espace disque** et donc le **même inode** ;
- Un autre nom pour le même fichier ;
- Toutes **modifications** effectuées sur l'**un** seront **répercutées sur l'autre** ;
- Le **fichier** n'est effectivement **détruit** que lorsque son **inode** **n'est plus référencé** par aucun nom ;
- **rm** ne supprime qu'**un lien**. La suppression totale exige la suppression de tous les liens.
- **ls -l** indique le nombre de liens que comporte un fichier.
- **ln (link)** permet de créer un lien physique.

Liens physiques : Interet

- L'utilisation la plus courante concerne les fichiers **exécutables** ;
- Un **seul** et **même** code **exécutable** peut avoir **plusieurs noms** ==> **plusieurs** comportements selon le nom par lequel il a été appelé ;
- **Exp** : Commandes **gzip**, **gunzip** et **zcat** sont sur une **Mandriva** des noms désignant le même **inode**.

Liens symboliques

Fait **référence** à un **fichier** dans un **répertoire**.

- Un lien symbolique est un fichier qui contient le **chemin (raccourcis)** du fichier qu'il "**pointe**" ;
- Si vous **déplacez** ou **renommez** l'original le **lien** est **rompu** ;
- **ln -s** ou **cp -s** permet de créer un lien symbolique.

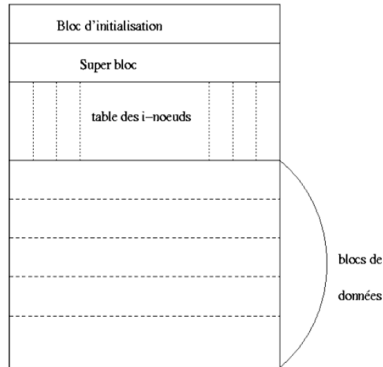
RQ

- un **lien hard** **n'a pas d'inode** propre, il a l'inode du fichier vers lequel il pointe.
- Un **lien symbolique** possède sa propre **inode**.

Organisation du Disque

- Le secteur **0** du disque est le **Master Boot Record (MBR)**
 - Utilisé pour **démarrer** l'ordinateur ;
 - La fin du **MBR** contient la table des partitions (**TP** et **Boot Loader**) ;
 - **RQ** : Une partition est marquée **active** si elle contient OS.
- Quand l'ordinateur démarre
 - Le bios lit et exécute le **MBR** ;
 - Le **MBR** trouve la **partition active** et lit les premiers blocs : **boot block** ;
 - Ce programme charge le système d'exploitation.

Description du disque (sous Unix)

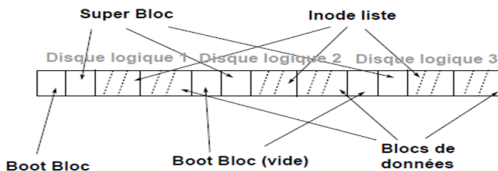


Description du disque (sous Unix)

Structure du système de fichier sur un disque logique



Plusieurs disques logiques sur un disque physique



- **Boot bloc** (secteur de boot) : utilisé au chargement du système .
- **Super Bloc** : il contient toutes les informations générales sur le disque logique.
- **Inode list** : table des inodes.
- **Blocs de données** : chaînés à la création du disque.

Table des i-nodes et bloc

Organisation des blocs et des inodes

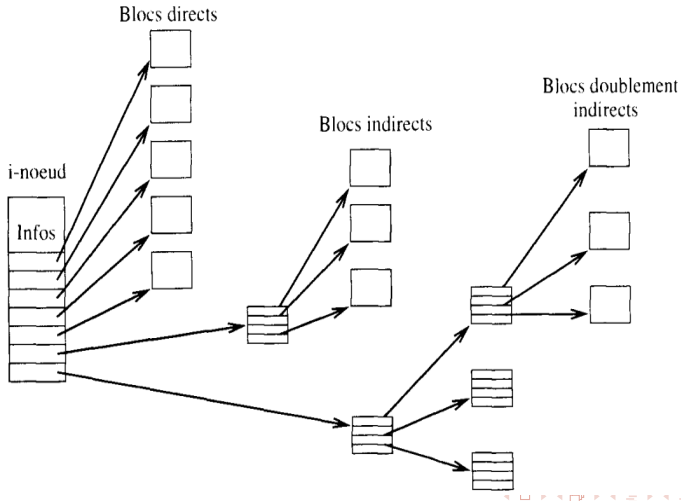


Table des i-nodes et bloc

- La table des i-nodes fournit un moyen d'accès aux fichiers pour l'utilisateur.
 - Les données sont sur le disque ;
 - Le système fournit des fonctions primitives pour y accéder.
 - **Convivialité** : comment accéder aux fichier de façon simple et transparente ;
 - **Efficacité** : minimiser le moins possible les accès aux fichiers ;
 - **Sécurité** : accès concurrent, autorisation d'accès.
- Une interface standard
 - Une vue utilisateur en **arbre** composée de **dossiers** (répertoires) et de fichiers ;
 - Chemins dans l'**arborescence** (suite de liens).

Organisation des fichiers (Linux)

Répertoire	contient
/bin	les fichiers exécutables nécessaires à l'initialisation
/boot	le noyau et les fichiers de démarrage
/dev	les fichiers spéciaux
/etc	les fichiers de configuration du système et certains scripts
/home	la base des répertoires utilisateurs
/lib	les bibliothèques système et les modules
/lost+found	le stockage des fichiers retrouvés par fsck
/mnt	les points d'ancrage des systèmes extérieurs
/proc	un système de fichiers virtuels permettant l'accès aux variables du noyau
/root	le répertoire de base du super utilisateur
/sbin	les fichiers exécutables pour l'administration du système
/tmp	les fichiers temporaires
/usr	les programmes, les bibliothèques et les fichiers accessibles pour l'utilisateur
/var	les données variables liées à la machine (spool, traces)

Types de partitions

- Deux types de partitions :
 - **Primaire** : partition capable d'accueillir un **système d'exploitation** ;
 - **Étendue** : est une partition primaire spéciale pouvant contenir jusqu'à **64 partitions logiques**.
- Un disque dur de PC peut contenir quatre (4) partitions **primaires**.

Montage de Partition

- Monter une partition est le fait de rendre accessible une partition depuis l'arborescence Linux.
- Les partitions, volumes RAID, NFS,... à monter dans l'arborescence sont référencées dans **/etc/fstab** (fichier table des partitions) .
- Etapes de montage :
 - 1 **Identifier** la partition volue dans **/etc/fstab** ;
 - 2 **Créer le point** (dossier) de montage ;
 - 3 **Monter** la partition dans le point de montage avec (**mount**) ;
 - 4 **Lister** le contenu de la partition avec **ls** ;
 - 5 **Redémarrer**.

Liste des systèmes de fichier (FS)

- Le fichier **/etc/fstab** contient la liste des systèmes de fichier (FS) connus par le système. Chaque ligne contient 6 champs :
 - C1** : Périphérique (exp : /dev/hda3)
 - C2** : Point de montage (exp : /mnt/floppy)
 - C3** : Type de FS (exp : ext3)
 - C4** : Options de montage (exp : **defaults,usrquota,grpquota**)
 - C5** : Inclusion pour les sauvegardes incrémentales par « **dump** » (0 ou 1)
 - C6** : Niveau de contrôle automatique au démarrage du système (0, 1, ou 2)
- Exemple :

LABEL=/	/	ext3	defaults	1	1
LABEL=/data	/data	ext3	defaults,usrquota	1	2
none	/dev/pts	devpts	gid=5,mode=620	0	0
none	/proc	proc	defaults	0	0
none	/dev/shm	tmpfs	defaults	0	0
none	/tmp	tmpfs	defaults,size=256M	0	0
/dev/hda3	swap	swap	defaults	0	0
/dev/cdrom	/mnt/cdrom	udf,iso9660	noauto,owner,kudzu,ro	0	0

Exemple de Montage

- **fdisk /dev/sda** (option m, p) (voir image). La partition Windows est désignée par **/dev/sda1**
- Créer le point de montage : **mkdir /mnt/windowsXP**
- Montage : **mount /dev/sda1 /mnt/windowsXP**
- Lister le contenu : **ls /mnt/windowsXP**
- **Redémarrer.**

Périphérique	Amorce	Début	Fin	Blocs	Id	Système
/dev/sda1	*	63	8191999	4095968+	7	HPFS/NTFS
/dev/sda2		8192000	9215999	512000	83	Linux
/dev/sda3		9216000	31457279	11120640	8e	Linux LVM

Outils (fdisk, gparted)

- **fdisk** est un outil de **consultation, création, suppression et modification** de partitions. Il est uniquement conçu pour modifier la table des partitions (avec **risque de perte** de données).
- **gparted** permet des opérations avancées sur les partitions. A utiliser si vous souhaitez modifier la taille d'une partition **sans supprimer les données** qu'elle contient.

Plan de Partitionnement

Les plus courants

- Il faut prévoir une partition d'échange (**swap**) au début de disque (plus rapide)
- Pour le système de fichiers **racine** : **/**
- Pour les données des utilisateurs : **/home**
- Pour d'autres systèmes d'exploitation.

Créer une partition (fdisk, gparted)

- Lancez **fdisk** avec le disque en argument ;
- Vérifiez tout d'abord l'existence de partitions avec la touche **p** (**print**) puis [Entrée].
- Pour **créer une partition**, utilisez la touche **n** (**new**). Vous devez ensuite choisir le type de partition : **primaire (p)** ou **étendue (e)**.
- Comme le **MBR** contient quatre entrées (**1-4**) vous pouvez choisir le **numéro de partition** à créer.
- Enfin choisissez la **taille de la partition**.
- Quittez **fdisk** en sauveant votre table des partitions avec **w** (**write**).

Exemple

```
# fdisk /dev/sdb
...
Commande (m pour l'aide): p

Disque /dev/sdb: 4026 Mo, 4026531840 octets
64 heads, 62 sectors/track, 1981 cylinders
Units = cylindres of 3968 * 512 = 2031616 bytes
Disk identifier: 0x0003ed63

Périphérique Amorçe   Début       Fin         Blocs      Id  Système
/dev/sdb1 *           1           1981        3930273    c   W95 FAT32 (LBA)
Commande (m pour l'aide): n
Action de commande
    e   étendue
    p   partition primaire (1-4)
p
Numéro de partition (1-4): 1
Premier cylindre (1-1981, par défaut 1):
Utilisation de la valeur par défaut 1
Dernier cylindre ou +taille or +tailleM ou +tailleK (1-1981, par
défaut 1981): +1024M
```