

## Gestion des disques : plan et illustrations

### Plan

- Fichiers, répertoires, systèmes de fichiers
- Comment marche un disque
- La gestion d'un disque par le système
- les systèmes RAID
- l'organisation des fichiers sur disque
- les arborescences

### 1.1 Quelques chiffres

couche magnétique	10-20 nm
vitesse de rotation	5400-15000 tours/min
capacité	120 GB - 2 TB
temps d'accès	2-15 ms
vitesse de transfert	70 MB/sec (7200 t/min)

### 1.2 Résumé

La technologie

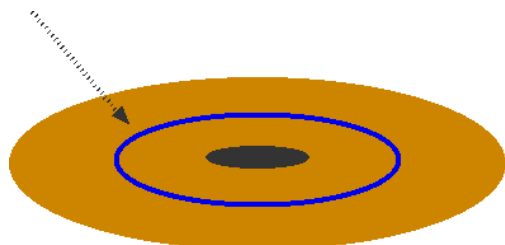
- enregistrement magnétique
- cylindres, pistes, secteur, bloc
- déplacement de pièces mécaniques

Conséquences

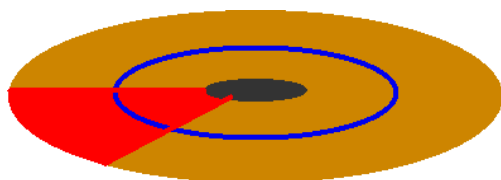
- *lenteur relative*
- *le temps d'accès aux blocs n'est pas uniforme*

## Gestion des disques

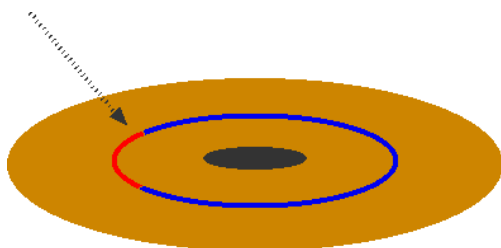
### 1 Un disque, comment ça marche ?



Une piste



Un secteur



Un bloc

Cylindre : pistes pour une position du bras.  
Temps d'accès non uniforme.

### 2 Gestion des entrées/sorties sur disque

#### 2.1 Monotâche

- pas de problème d'ordonnancement
- dégradation de performances si fragmentation

#### 2.2 Multitâches

- plusieurs processus à servir
  - *ordonnancement* des mouvements du bras
- Stratégie ?

#### 2.3 E/S : Premier arrivé, premier servi

Exemple : trois processus

- P1 : lit les blocs 100, 101, 102 ...
- P2 : lit les blocs 200, 201, 202 ...
- P3 : lit les blocs 500, 501, 502 ...

Déroulement :

**Bilan :**

- *simple* à mettre en oeuvre
- *équitable* ...
- mais *pas efficace*

- ... gagner en *performances*

⇒ 

R	edundant
A	rarray
I	nexpensive
D	isks

**2.4 E/S : plus court déplacement**

Déroulement sur exemple :

**Maintenant** : Redundant Array of *Independent* Disks

**Bilan**

- 
- 
- 

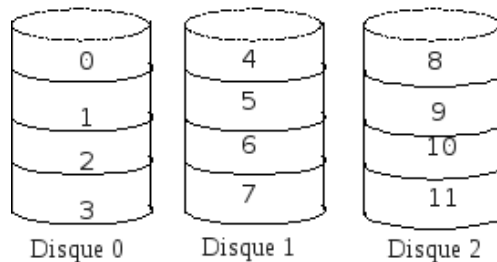
**4 les systèmes RAID usuels**

**2.5 E/S : politique de l'ascenseur**

Déroulement sur exemple :

**4.1 NRAID (JBOD)**

**Agrégation**



NRAID, JBOD : concaténation

**Bilan :**

- *assez équitable* ...
- *assez efficace*

Pas de redondance. NRAID is *not* RAID. JBOD : *just a bunch of disks*

**2.6 E/S : Deadline driven disk scheduler**

Délai fixé pour servir une requête. Requête urgente ⇔ délai dépassé

Requêtes *urgentes* traitées en priorité dans l'ordre d'apparition.

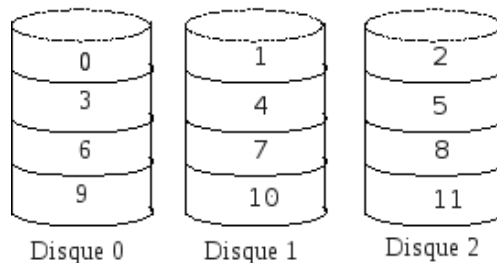
Requêtes normales : plus court déplacement.

**Bilan :**

- 
- 
- 

**4.2 RAID 0 : agrégation par bandes**

**Striping**



RAID 0 : agrégation par bandes

**RAID**

**3 Objectifs du RAID**

Segmentation du marché des disques

- 
- 
- 
- 

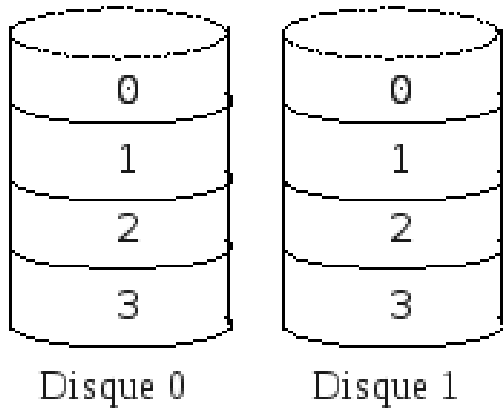
grouper des disques pour

- étendre la *capacité*
- ... améliorer la *fiabilité*

**Objectifs** : répartition de la charge + anticipation lecture suivante.

### 4.3 RAID 1 : miroir

#### Miroir

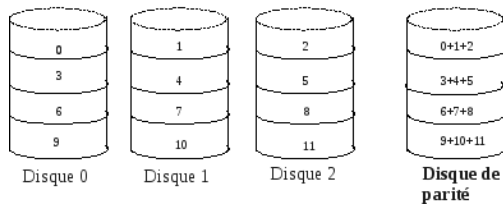


RAID 1 : miroir

Écriture en parallèle. Objectif : redondance + répartition des lectures.

### 4.4 RAID 4 : agrégation par bandes avec parité

Striping + parité

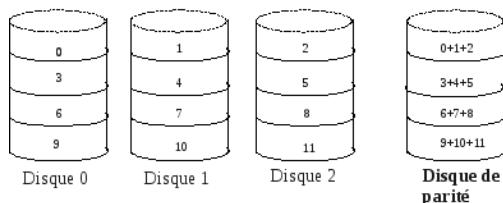


RAID 4 : agrégation par bandes avec parité

Objectif : reconstitution disque défectueux. Généralisation de l'idée de miroir.

Propriété du ou-exclusif : si  $a \oplus b \oplus c \oplus \dots = 0$  alors  $a = b \oplus c \oplus \dots$

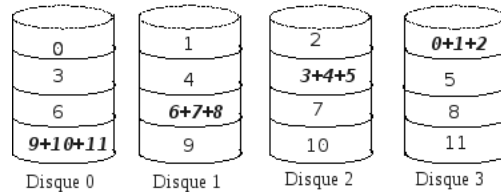
Striping + parité



mais goulet d'étranglement (accès au disque de parité)

### 4.5 RAID 5 : agrégation par bandes avec parité répartie

Striping + parité + répartition



RAID 5 : agrégation par bandes avec parité répartie

Objectif : fiabilité, performances. La charge en écriture est répartie.

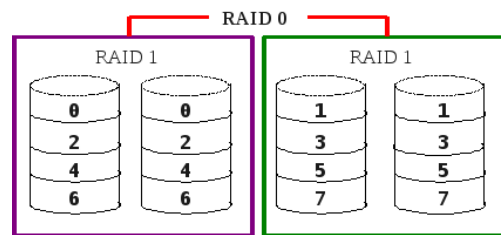
## 5 Combinaisons

- pour combiner les avantages
  - au prix de disques supplémentaires
- Quelques exemples courants :

### 5.1 RAID 10

RAID 10 ou 1+0 = grappe RAID 1 de paires de disques en miroir

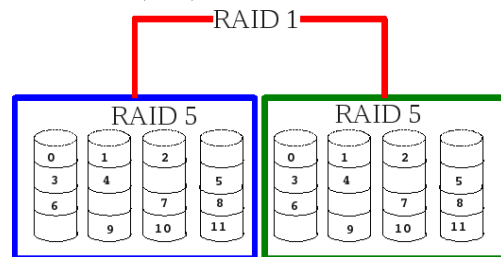
RAID 10 (1+0)



RAID 0 sur disques en miroir

### 5.2 RAID 51

RAID 51 (5+1)



miroir entre 2 grappes RAID 5

# Systèmes de fichiers 9 FAT

## 6 Définition

Système de fichiers = structure de données.  
*représentation* de *fichiers* et *répertoires* sous forme de 0 et de 1 stockés en mémoire secondaire.

## 7 Pour l'utilisateur

Fichier : contenu, méta-données.

- taille
- propriétaire
- droits d'accès
- date de création
- date de dernier accès
- ...

### 7.1 fonctions du SGF

- Système de Gestion de Fichiers, fonctions
- Manipulation des fichiers : créer/détruire des fichiers, ...
  - Allocation de la place sur mémoires secondaires
  - Localisation des fichiers : accès au contenu
  - Sécurité et contrôle des fichiers
  - Fiabilité en cas de panne
  - ...

ici : quelques idées sur la représentation

## 8 Catalogue de fichiers

(une méthode antique, pas de répertoires, fichiers contigus)

Volume Table of Contents (IBM)

- Au début du disque, *table* qui indique
  - le *nom* de chaque fichier
  - son emplacement (*position premier bloc, taille*)
- ensuite : *blocs de données* du premier fichier
- blocs du second
- ...

+ *liste des espaces non utilisés*

Table d'allocation. Fichiers non contigus : allocation plus facile à gérer. Table supplémentaire : index du bloc suivant.

### 9.1 Catalogue de fichiers

- Au début du disque, table qui indique
  - le nom de chaque fichier
  - son emplacement (*index premier bloc, taille*)
- ensuite : *table de chaînage* des blocs
- puis : blocs de données

## 10 Représentation des répertoires

### 10.1 Comme des catalogues de fichiers

Entrées spéciales de la table des fichiers qui renvoient vers d'autres parties de la table.

(CP/M, MS/DOS, Windows...)

Ne permet pas les *liens*

### 10.2 Comme des fichiers de données

SGF d'Unix/ Un système de fichiers contient

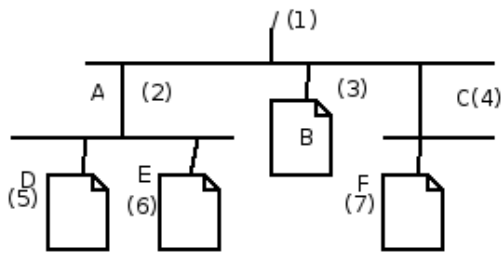
- une *table d'i-nodes* (noeuds d'information)
- des *blocs de données* liés à ces i-nodes

Un fichier/répertoire... est identifié par son numéro d'i-node

Différents types d'i-nodes

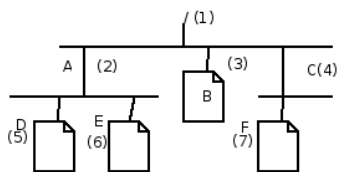
- fichiers. Blocs = contenu du fichier
- répertoires. Blocs = table de noms et numéros d'i-node
- liens symboliques
- périphériques
- ...

### 10.3 Exemple



Exemple d'arborescence

Table des i-nodes



N°	type	CR	contenu des blocs
1	d	4	..=1, .=1, 1=2, B=3, C=4
2	d	2	..=1, .=2, D= 5, E=6
3	f	1	"coucou"
4	d	2	.. :1, .=4, F=7
	...		

CR = compteur de références

### 10.4 Gestion des blocs libres

Le système possède

- une liste des blocs libres
- un tableau de marquage des blocs occupés

### 10.5 Vérification du système de fichiers

Utilitaire `fsck`, descente de l'arborescence :

1. vérification des i-noeuds, des blocs et des tailles
2. vérification de la structure des répertoires
3. vérification de la connectivité des répertoires
4. vérification des compteurs de référence
5. vérification de l'information du sommaire de groupe

## 11 Autres caractéristiques

### 11.1 Journalisation

**Journal** :

- garde une trace des opérations d'écriture non terminées
- point de reprise

Avantages

- pas de pertes d'informations
- reprise plus rapide (évite le `fsck`)

### 11.2 Snapshots (clichés)

*Cliché* : copie de l'état du système de fichiers à un moment donné