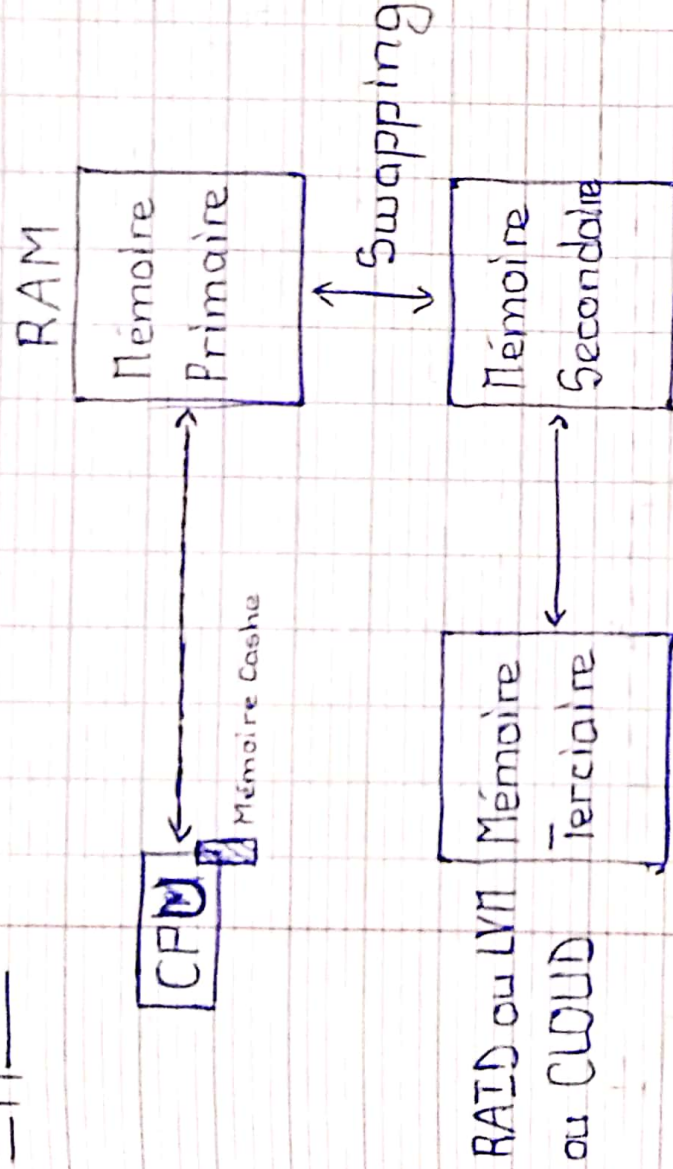


Chapitre I : Système de fichiers et fichiers

Rappel :



Swapping : Pour passer d'une mémoire à une autre

Fichier : Nom + Attributs (types, droits, taille, date...)
sous linux : `ls -al`

`- d c b l P s` } types de fichiers

Implantation par liste chaînée :

On fait un chaînage entre les blocs libres avec dans chaque bloc un pointeur contenant l'adresse du bloc suivant.

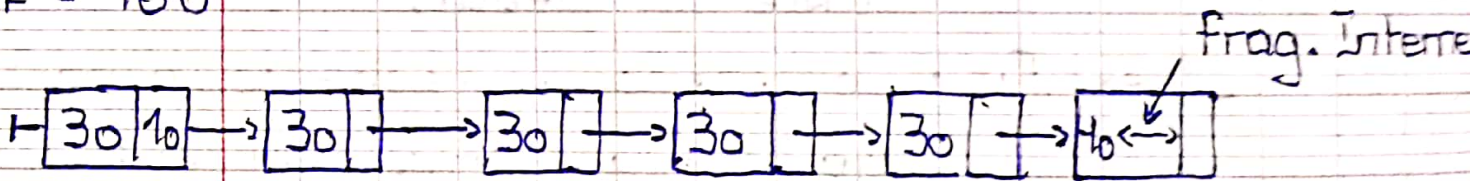
- avantage : Élimine la fragmentation externe.
- inconvénient : plus complexe à gérer.

Exemple :

Soit $T_B = 40$

Pointeur $T_P = 40$

$T_F = 460$



autre inconvénient : gaspillage espace pour les pointeurs.

Implantation FAT :

Création d'une structure de données contenant les adresses de tous les blocs : F.A.T

- avantages : accès rapide
bloc de données contenant entièrement la donnée

inconvenients: placée en mémoire (utilisation abusive de la RAM).

Implantation indexé:

On associe à chaque fichier un nœud d'index contenant les attributs du Fichier (propriétaire, date création...) et les adresses des blocs de données.

Gestion de l'espace Libre:

1. Par table de bits:

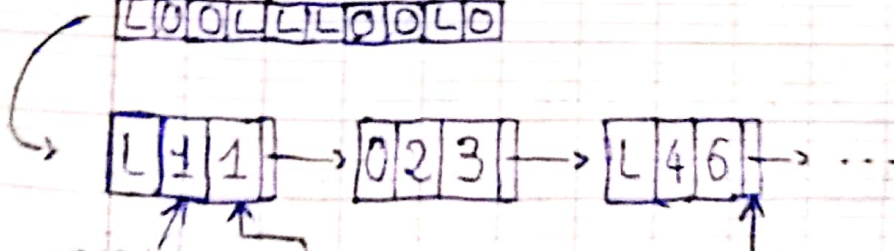
On associe chaque bloc à un bit.

$L = 1$ (10110010)
 $0 = 0$ (10110010)

2. Par liste chaînée:

On associe chaque bloc à un maillon.

Ex: 10011110010



N° bloc

Etendu

Pointeur

Système de fichier Ext2:

Boot	Super Bloc	i-node	blocs ...
------	------------	--------	-----------

$$512 \leq T_B \leq 4096$$

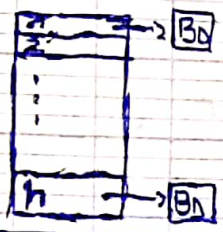
i-node :

Attribut
1
2
⋮
12
13
14
15

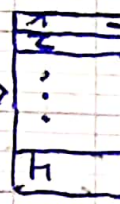
B_D

Accès Direct

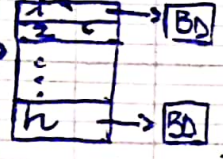
B_D



A.I.S



A.I.D



A.I.S : Accès Indirect Simple

A.I.D : " " Double

A.I.T : " " Triple

pour A.I.S : $T_F \leq 12 \times T_{BD} + h \times T_{BD}$

" A.I.D : $T_F \leq 12 \times T_{BD} + h \times T_{BD} + h^2 \times T_{BD}$

" A.I.T : $T_F \leq$ " " " " $+ h^3 \times T_{BD}$

EXT : $T_{MAX} = 12 \times T_{BD} + h \times T_{BD} + h^2 \times T_{BD} + h^3 T_{BD}$
 $= T_{BD} (12 + h + h^2 + h^3)$