Organisation des fichiers

IFT320 - Système d'exploitation, *Michael Fortin*

Mardi 1er et mercredi 2e d'octobre, 2019

Notes de cours prises par *Lucien B. Regout*.

Tableau résumé

Type	**Représentation emplacement physyque**	**Nombre d'accès direct**	**Fragment-ation**	**Gestion espace libre**	**Changement de taille**	**Protection de dommanges**
Contigu	Pos. début Taille **O(1)**	Début + Déplacement **O(1)**	Externe	Liste espaces libres [Position,Taille] triée par grandeur	• **O(1)** Si espace libre contigu • **O(n)** Si un trou libre assez grand existe • **O(Taille Disque)** Si fragment	Copies de sûretées
Liste chainée	• Numéro bloc départ • Taille (n) • Un numéro bloc suivant par bloc (n/taille bloc) • position dernier bloc • **O(n)**	• Nb de blocs • **O(n/2)** • Si fin fichier, **O(1)**	Interne	1. Liste chaînée 2. Liste chaînée d'index 3. Bitmap	**O(1)** Ajuster un bloc a la fin de la chaîne	• Chainage vulnerable ◦ Chainage double

---

Type	Représentation emplacement physyque	Nombre d'accès direct	Fragment-ation	Gestion espace libre	Changement de taille	Protection de dommanges
Index	• Un no de bloc par bloc • **O(n)** • + Un no de bloc par bloc d'index • **O(Log(n))**	$O(Log_{TI}N)$ O(nombre de niveau d'index)	$O(Log_{TI}N)$	Interne Également dans les blocs d'index	• Index • Bitmap	Bloc d'index vulnérables ?Autres? Chainage entre les blocs
Fichier Image	un numero de bloc par bloc	• **O(n)** acces RAM • **O(1)** acces disque	**O(1)**	Interne	Fichier Image	Fichier Image vulnerable 2 copies du fichier image

---

Arbres cycliques et acycliques. (À completer)

Un répertoire typique

F = Fichier, **R** = Répertoire, **S** = lien Symbolique

Nom	Empl. Phys.	Type (F/R/S)
R1	F7	F
...	...	...

Liens symboliques

- Il est possible d'empêcher les cycles en bloquant les liens physiques sur les répertoires.
- Une **table de montage** est un petit fichier, chargé au démarage qui sert à déterminer les partition d'un même disque dur.
- C'est non optimale pour établire des liens physiques avec des disques amovibles (tels clefs USB) ou disques réseaux, car ceux-ci peuvent être déconnecté en tout temps.

Contigu

Magnetic Tape File System (*MTFS*)

Répertoire:	**F1**	**Libre**	**F3**	**F4**	**Libre**	**F5**	**Libre**	**F6**	**F7**	**Libre**

Répertoire:		**F1**	**Libre**	**F3**	**F4**	**Libre**	**F5**	**Libre**	**F6**	**F7**	**Libre**
Emplacement:	0	1024	5020	6044	14206	14334	14846	16894	19966	20666	22714
Taille:	1ko	4ko	1ko	8ko	128o	512o	2ko	3ko	700o	8ko	182o

Fichiers sur un répertoire donné

Nom	**Empl. Phys.**	**Taille**
F1	1024	4ko
Libre	5020	1ko
F3	6044	8ko
F4	14206	128o
Libre	14334	512o
F5	14846	2ko
Libre	16894	3ko
F6	19966	700o
F7	20666	8ko
Libre	22714	182o

Liste d'espace libre

Début	**Taille**
5020	1k
14334	512o
16894	3k
22714	1862o

Remarque

- Il est super complexe d'agrandir la taille d'un fichier dans se système.

Retour au Tableau résumé

Algorithmes de gestion d'espaces libres

Best-Fit

> Connait la taille du répertoire et la taille est **statique**. Ceci maximise mes courbes.
> - Louis 2019.

Worst-Fit

> Taille appeller à **augmenter**. Comme choisir du linge pour de jeunes enfants. Pour que ca dure le plus longtemps.

Firs-Fit

> Pas d'information sur les changements de tailes. Ce veut être très **rapide**.

Fragmentation

Retarder : le temps que ca prends avant d'avoir la necessité d'utilisé les opperation Éliminer

Eliminer : l'espace perdu causé par la fragmentation

- #### Externe
  - Retarder
    - Algo First-Fit, Best-Fit, Worst-Fit
  - Eliminer (l'espace perdu causé par la fragmentation)
    - Squeeze : Consolider l'espace libre.
- #### Interne
  - Retarder
    - 2 différentes tialles de bloc et on alloue les petits blocs aux petits fichiers.
  - Eliminer
    - Tail-packing (difficile) --> Plus simple c'est le systeme à 2 tailles de blocs
    - Stocker des données (compactes) dasn l'en-tête ou, pire encore, dans l'entré de répertoire.

Retour au Tableau résumé

Fragmentation Externe

> Assez de capacité pour l'utilisation. Cependant, il est séparé en petits morceaux.

Frag Externe: Ca laisse de l'espace entre les blocs.

Fragmentation Interne

> Taille Disque = 2^{40} Taille Bloc = 8ko = 2^{13} Si, en moyenne, la taille d'un fichier est de 1ko, et qu'on en réserve 75ko, on perds énormément d'espace.

Système à 2 tailles de blocs

> Minimise la perte d'espace dû a la fragmentation interne. Plus simple d'implémenter 2, voir 3, tailles de blocs pour minimiser la perte d'espace avec la fragmentation interne. Plus simple que la technique Tail-Packing.

Si la taille est moins d'un petit bloc, assigne un petit bloc. Si la taille est plus grande qu'un petit bloc, assigner un gros bloc.

Liste chaînée

7	⇄	11	⇄	13	⇄	3	⇄	14	⇄	25	⇄	12	⇄	20

TB == Taille de bloc TI == Taille Index N == Nombre de blocs TF == Taille Fichier

N = N/TB TI = 4 N/TI = Bloc d'Index

Complexité = O((N/TB)/TI)

Exemple, avec un fichier de 1Go, puis des blocs de 1ko:

```
TF = 2^30
N = 2^10
TI = 2^8

(2^30)/(2^12) = 2^18
-> 2^18 / 2^10 = 2^8
```

Retour au Tableau résumé

Index

- Taille d'un bloc (TB)
- Taille d'un numéro de bloc {TNB
- Taille d'index = TB/TNB}

Si un seul niveau d'index, alors taille max = TI*TB

Ces exemples n'ont qu'un de

> Ex. 1: 128o = 2^7 32bits = 2^2 2^7 / 2^2 = 2^5 numero de blocs

> Ex. 2: TD = 4To = 2^{42} TB = 8ko = 2^{13} 2^{42} / 2^{13} = 2^{29} numero de blocs

Taille de fichier max = Taille max = TI^N * TB

> Nombre de niveau d'index = Log_{TI}(TF / TB)

Fichier Image

Fichier

3	⇄	7	⇄	1	⇄	10

Disque

Fichier Image	0	1	2	3	4	5	6	...

Fichier Image

	Val
0	
1	10
2	
3	7
4	
5	
6	
7	1
8	
9	
10	x
...	

> FAT12 : Disquettes
> FAT16 : Disques durs (mo)
> FAT32 : Gros disques durs (Go)
> 12, 16 et 32 sont les tailles de numeros de blocs, effectivement 28 pour le FAT32.