

Algorithmique de fichiers

CM4 – Stockage physique et Systèmes de fichiers

Adrien Krähenbühl

UFR Mathématiques-Informatique

9 Mars 2015

Organisation logique

Stockage physique

Système de fichiers

Notion

La notion de fichier **logique**
est le dual de
la notion de fichier .

La notion de fichier **logique
est le dual de
la notion de fichier **physique**.**

La notion de fichier **logique
est le dual de
la notion de fichier **physique**.**

La notion de fichier logique est :

- une représentation/organisation des données
- Indépendante du support de stockage
- Indépendante du système de fichier
- Indépendante du codage du fichier (ASCII, ISO, UTF-8, ...)

La notion de fichier **logique
est le dual de
la notion de fichier **physique**.**

La notion de fichier logique est :

- une représentation/organisation des données
- Indépendante du support de stockage
- Indépendante du système de fichier
- Indépendante du codage du fichier (ASCII, ISO, UTF-8, ...)

C'est "*comment doit être perçu*" le fichier.

Exemple

```
1 Name,Given Name,Additional Name,Family Name,Yomi Name,Given Name Yomi,Additional Name Yomi,Family Name Yomi,Name Prefix,
  Name Suffix,Initials,Nickname,Short Name,Maiden Name,Birthday,Gender,Location,Billing Information,Directory Server,Mileage,
  Occupation,Hobby,Sensitivity,Priority,Subject,Notes,Group Membership,E-mail 1 - Type,E-mail 1 - Value,E-mail 2 - Type,E-
  mail 2 - Value,Phone 1 - Type,Phone 1 - Value,Phone 2 - Type,Phone 2 - Value,Website 1 - Type,Website 1 - Value,Jot 1 -
  Type,Jot 1 - Value
2 Adeline MERIA,Adeline,,MERIA,,,,,,,,,,,,,* My Contacts ::: Doctorants,* ,adeline.meria@srsmc.uhp-nancy.fr,,
  Mobile,06 47 84 44 85,,,,,
3 Adrien AMBERT,Adrien,,AMBERT,,,,,,,,,,,,,Telecom Nancy ::: Chercheur ::: * My Contacts,,ambert@loria.fr,* ,adrien
  .ambert@loria.fr,,,,,Profile,http://www.google.com/profiles/107695441374683290304,,
4 Adrien MONIN,Adrien,,MONIN,,,,,,,,,,,,,Fac ::: Doctorants ::: * My Contacts,* Home,aved88@gmail.com,,Mobile,+33
  6 80 87 79 92,,Profile,http://www.google.com/profiles/114887965991793014444,,
5 Aladin,Aladin,,,,,,,,,,,,,* My Contacts,,,,,Mobile,03 38 93 94 22,,,,,
6 Albert DUPONT,Albert,,DUPONT,,,,,,,,,,,,,1931-08-20,,,,,,,,,,,,,* Family ::: * My Contacts,* Home,albert.dupont@orange.fr,,
  Home,0384734467,,,,,
7 Alexandre JUNGLE,Alexandre,,JUNGLE,,,,,,,,,,,,,Doctorants ::: * My Contacts,* Other,alex84110@msn.com,,Mobile,
  0641560445,,,,,User,Alex
8 Amandine LAROCHE,Amandine,,LAROCHE,,,,,,,,,,,,,* My Contacts ::: Doctorants,* ,amandine.laroche@loria.fr,,Mobile
  ,06 29 00 89 16,,,,,
9 Amina NEGAME,Amina,,NEGAME,,,,,,,,,,,,,Doctorants ::: * My Contacts,,,,,Home,0636442934,,,,,
10 Anais DUTIGRE,Anais,,DUTIGRE,,,,,,,,,,,,,* My Contacts ::: * Family,,,,,Mobile,+33750133868,Other,0670434447,,,,
11 Anne-So,Anne-So,,,,,,,,,,,,,Fac ::: * My Contacts,asmalgras@gmail.com,* Home,as.malgras@gmail.com,Mobile,
  0665974731,,Profile,http://www.google.com/profiles/100177595497611629500,,
```

Exemple : liste de contacts au format CSV.

- Ensemble de données
- En entrée ou sortie d'un programme
- Conçu hiérarchiquement
- Critères de subdivision appropriés

Organisation logique

Stockage physique

Système de fichiers

CD-Rom



CD-Rom



- Support **optique** lu par un laser
- Contient un ensemble de **pistes**
- Stockage et lecture **séquentiels**

Disque dur



Disque dur



- Support **magnétique** lu par des têtes de lecture
- Empilement de **plateaux** autour d'un même axe
- Succession de **pistes** sur chaque plateau
- Découpage d'une piste en **secteurs**

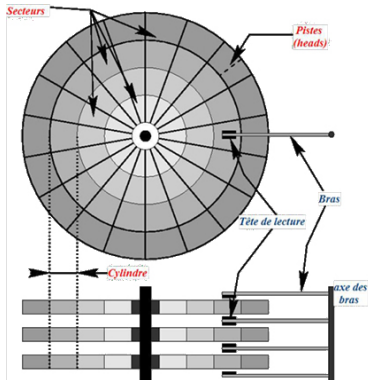
Disque dur



Fonctionnement :

- Les plateaux tournent à vitesse constante.
- Les têtes de lecture tournent autour de leur propre axe pour accéder aux différentes pistes

Disque dur



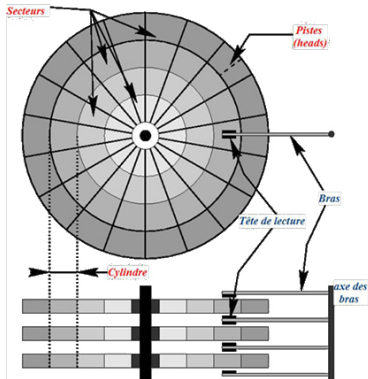
Accès à une donnée par ses coordonnées CTS :

Cylindre Numéro de la piste.

Tête Numéro de la tête (2 par plateaux).

Secteur Position angulaire autour de l'axe central.

Disque dur



Unité de lecture/écriture sur le disque : le **bloc** (ou cluster).

Taille en général 512 octets.

Contrainte Impossible de lire ou écrire moins d'un bloc de données.

Stockage physique d'un enregistrement

Exemple :

- On dispose d'un disque dur avec des blocs de taille 512 octets.
- On a défini un fichier avec des enregistrements de taille fixe 600 octets.

Lorsqu'on stocke 1 enregistrement :

1. Combien de blocs sont nécessaires ?
2. Combien perd-t-on d'espace ?

Et pour un fichier avec 10000 enregistrements ?

Stockage physique d'un enregistrement

Exemple :

- On dispose d'un disque dur avec des blocs de taille 512 octets.
- On a défini un fichier avec des enregistrements de taille fixe 600 octets.

Lorsqu'on stocke 1 enregistrement :

1. Combien de blocs sont nécessaires ? **2**
2. Combien perd-t-on d'espace ?

Et pour un fichier avec 10000 enregistrements ?

Stockage physique d'un enregistrement

Exemple :

- On dispose d'un disque dur avec des blocs de taille 512 octets.
- On a défini un fichier avec des enregistrements de taille fixe 600 octets.

Lorsqu'on stocke 1 enregistrement :

1. Combien de blocs sont nécessaires ? **2**
2. Combien perd-t-on d'espace ? $2 * 512 - 600 = \mathbf{424 \text{ octets}}$

Et pour un fichier avec 10000 enregistrements ?

Stockage physique d'un enregistrement

Exemple :

- On dispose d'un disque dur avec des blocs de taille 512 octets.
- On a défini un fichier avec des enregistrements de taille fixe 600 octets.

Lorsqu'on stocke 1 enregistrement :

1. Combien de blocs sont nécessaires ? **2**
2. Combien perd-t-on d'espace ? $2 * 512 - 600 =$ **424 octets**

Et pour un fichier avec 10000 enregistrements ?

1. **20000 blocs**

Stockage physique d'un enregistrement

Exemple :

- On dispose d'un disque dur avec des blocs de taille 512 octets.
- On a défini un fichier avec des enregistrements de taille fixe 600 octets.

Lorsqu'on stocke 1 enregistrement :

1. Combien de blocs sont nécessaires ? **2**
2. Combien perd-t-on d'espace ? $2 * 512 - 600 = \mathbf{424 \text{ octets}}$

Et pour un fichier avec 10000 enregistrements ?

1. **20000 blocs**
2. $424 * 10000 = \mathbf{4\ 240\ 000 \text{ octets}} \approx 4 \text{ Mo}$

Stockage physique d'un enregistrement

Exemple :

- On dispose d'un disque dur avec des blocs de taille 512 octets.
- On a défini un fichier avec des enregistrements de taille fixe 600 octets.

Lorsqu'on stocke 1 enregistrement :

1. Combien de blocs sont nécessaires ? **2**
2. Combien perd-t-on d'espace ? $2*512-600 = \mathbf{424 \text{ octets}}$

Et pour un fichier avec 10000 enregistrements ?

1. **20000 blocs**
2. $424*10000 = \mathbf{4\ 240\ 000 \text{ octets}} \approx 4 \text{ Mo}$

... 10Mo occupés pour seulement 6Mo de données utiles !

Stockage physique d'un enregistrement

Exemple :

- On dispose d'un disque dur avec des blocs de taille 512 octets.
- On a défini un fichier avec des enregistrements de taille fixe 600 octets.

Lorsqu'on stocke 1 enregistrement :

1. Combien de blocs sont nécessaires ? **2**
2. Combien perd-t-on d'espace ? $2 \cdot 512 - 600 = \mathbf{424 \text{ octets}}$

Et pour un fichier avec 10000 enregistrements ?

1. **20000 blocs**
2. $424 \cdot 10000 = \mathbf{4\ 240\ 000 \text{ octets}} \approx 4 \text{ Mo}$

... 10Mo occupés pour seulement 6Mo de données utiles !

Des idées pour limiter la perte de données ?

Le Bucket

- Le Bucket est la plus petite quantité d'information qui circule entre un logiciel et le disque dur.
- C'est un nombre entier d'enregistrements qui sont transférés en une seule fois par le système de fichiers.

1 bucket = N blocs = M enregistrements + pertes

Exemple de calcul de bucket

- On dispose d'un disque dur avec des blocs de taille 512 octets.
- On définit un fichier avec des enregistrements de taille fixe 600 octets.

Nombre d'enregistrements	Nombre de blocs	Perte
1		
2		
3		
4		
5		
6		

Exemple de calcul de bucket

- On dispose d'un disque dur avec des blocs de taille 512 octets.
- On définit un fichier avec des enregistrements de taille fixe 600 octets.

Nombre d'enregistrements	Nombre de blocs	Perte
1 (600 o)	2 (1024 o)	424 o
2 (1200 o)	3 (1536 o)	336 o
3 (1800 o)	4 (2048 o)	248 o
4 (2400 o)	5 (2560 o)	160 o
5 (3000 o)	6 (3072 o)	72 o
6 (3600 o)	8 (4096 o)	496 o

Exemple de calcul de bucket

- On dispose d'un disque dur avec des blocs de taille 512 octets.
- On définit un fichier avec des enregistrements de taille fixe 600 octets.

Nombre d'enregistrements	Nombre de blocs	Perte
1 (600 o)	2 (1024 o)	424 o
2 (1200 o)	3 (1536 o)	336 o
3 (1800 o)	4 (2048 o)	248 o
4 (2400 o)	5 (2560 o)	160 o
5 (3000 o)	6 (3072 o)	72 o
6 (3600 o)	8 (4096 o)	496 o

- Perte minimale de 72o pour 5 enregistrements

Exemple de calcul de bucket

- On dispose d'un disque dur avec des blocs de taille 512 octets.
- On définit un fichier avec des enregistrements de taille fixe 600 octets.

Nombre d'enregistrements	Nombre de blocs	Perte
1 (600 o)	2 (1024 o)	424 o
2 (1200 o)	3 (1536 o)	336 o
3 (1800 o)	4 (2048 o)	248 o
4 (2400 o)	5 (2560 o)	160 o
5 (3000 o)	6 (3072 o)	72 o
6 (3600 o)	8 (4096 o)	496 o

- Perte minimale de 72o pour 5 enregistrements
- Le bucket est donc de 5 enregistrements

Exemple de calcul de bucket

- On dispose d'un disque dur avec des blocs de taille 512 octets.
- On définit un fichier avec des enregistrements de taille fixe 600 octets.

Nombre d'enregistrements	Nombre de blocs	Perte
1 (600 o)	2 (1024 o)	424 o
2 (1200 o)	3 (1536 o)	336 o
3 (1800 o)	4 (2048 o)	248 o
4 (2400 o)	5 (2560 o)	160 o
5 (3000 o)	6 (3072 o)	72 o
6 (3600 o)	8 (4096 o)	496 o

- Perte minimale de 72o pour 5 enregistrements
- Le bucket est donc de 5 enregistrements
- C'est le système de gestion des fichiers qui définit le bucket

Exercice

- On dispose d'un disque dur avec des blocs de taille 1024 octets.
 - On définit un fichier avec des enregistrements de taille fixe 720 octets.
1. Quel bucket devrait choisir le système de gestion des fichiers ?
 2. Quelle place gagne-t-on pour un fichier de 10000 enregistrements ?

Exercice

- On dispose d'un disque dur avec des blocs de taille 1024 octets.
 - On définit un fichier avec des enregistrements de taille fixe 720 octets.
1. Quel bucket devrait choisir le système de gestion des fichiers ?

Nombre d'enregistrements	Nombre de blocs	Perte
1 (720 o)	1 (1024 o)	304 o
2 (1440 o)	2 (2048 o)	608 o
3 (2160 o)	3 (3072 o)	912 o
4 (2880 o)	3 (3072 o)	192 o
5 (3600 o)	4 (4096 o)	496 o
6 (4320 o)	5 (5120 o)	800 o
7 (5040 o)	5 (5120 o)	80 o

2. Quelle place gagne-t-on pour un fichier de 10000 enregistrements ?

Exercice

- On dispose d'un disque dur avec des blocs de taille 1024 octets.
 - On définit un fichier avec des enregistrements de taille fixe 720 octets.
1. Quel bucket devrait choisir le système de gestion des fichiers ?

Nombre d'enregistrements	Nombre de blocs	Perte
7 (5040 o)	5 (5120 o)	80 o

2. Quelle place gagne-t-on pour un fichier de 10000 enregistrements ?

Exercice

- On dispose d'un disque dur avec des blocs de taille 1024 octets.
 - On définit un fichier avec des enregistrements de taille fixe 720 octets.
1. Quel bucket devrait choisir le système de gestion des fichiers ?

Nombre d'enregistrements	Nombre de blocs	Perte
7 (5040 o)	5 (5120 o)	80 o

2. Quelle place gagne-t-on pour un fichier de 10000 enregistrements ?

$$1024 * 10000 = 10240000 \text{ o} = 10,24 \text{ Mo}$$

$$1429 * 5120 = 7316480 \text{ o} \approx 7,32 \text{ Mo}$$

$$10,24 - 7,31648 = 2,92352 \text{ Mo} = 28,55\% \text{ de gain de place !}$$

Organisation logique

Stockage physique

Systeme de fichiers

Keskecé ?

Un **Système de fichiers** ou **Système de gestion des fichiers** est :

Une façon de stocker les informations et de les organiser dans des fichiers.

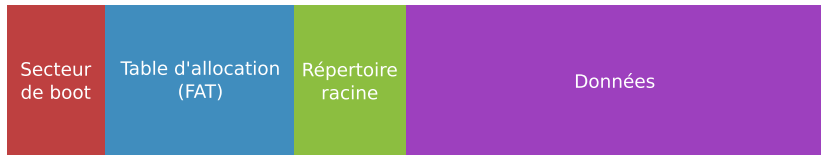
- Pleins de façons de faire → pleins de systèmes différents
- Il doit fournir la possibilité de créer/ouvrir/enregistrer/déplacer/supprimer un fichier.
- Deux notions importantes : le **fichier** et le **répertoire** (ou dossier)

FAT



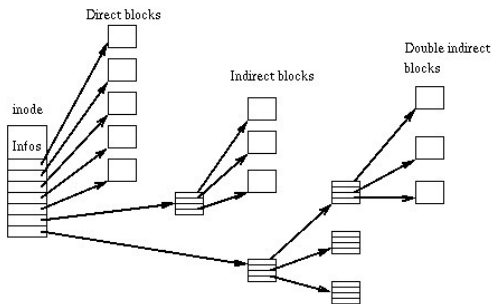
- Le système FAT (File Allocation Table) a été proposé par Bill Gates et Marc McDonald en 1977.
- Plusieurs évolutions : FAT-12, FAT-16, FAT-32
- Principe de stockage : la FAT contient pour chaque fichier une liste chaînée des blocs

FAT

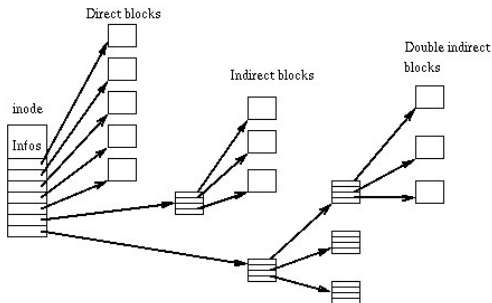


Quelques limitations (FAT 16/FAT 32) :

- Taille d'un fichier : 4 Go / 4 Go
- Nombre de fichiers : 65 518 / + 250 millions
- Taille du disque : 2 Go / 2 To
- Taille d'un nom de fichier : 8,3 octets / 255 caractères
- Pas de gestion des droits
- Fragmentation



- Le système **EXT** (Extended File System) a été proposé par Rémy Card en 1992.
- Plusieurs évolutions : Ext 2, Ext 3, Ext 4 et pleins d'autres.
- Principe de stockage :
 - ▶ la table d'inode contient pour chaque fichier son identifiant unique, le numéro d'inode.
 - ▶ un **inode** est un bloc contenant les méta-données d'un fichier.



Quelques limitation :

- Taille d'un fichier : 16 Go à 2 To
- Nombre de fichiers : variable
- Taille du disque : 2 To / 32 To
- Taille d'un nom de fichier : 255 octets
- Gestion des droits
- Pas de fragmentation