



université
angers

FACULTÉ
DES SCIENCES
*Unité de formation
et de recherche*

UNIVERSITÉ
BRETAGNE
LOIRE

Système

Fabien GARREAU

sur la base du cours d'André Rossi

Université d'Angers

`fabien.garreau@univ-angers.fr`

L2 MPCIE, 2018–2019

Système de fichiers

Le fichier est le conteneur de l'information privilégié d'UNIX.

Il existe de très nombreuses extensions de fichiers : `prog.c` programme source C, `exec.bin` : fichier binaire exécutable, `save.zip` archive zip.

UNIX n'impose pas d'extension aux fichiers, qu'il considère comme des suites d'octets. Le système n'interprète donc pas le contenu des fichiers.

Mais UNIX distingue en particulier 3 types de fichiers :

- - Fichier ordinaire
- d Répertoire
- l Lien symbolique

On obtient cette information avec `ls -l` (premier caractère de chaque ligne)

Système de fichiers

Un même fichier peut avoir plusieurs noms (**liens physiques**), mais il a une seule représentation physique en mémoire. Il dispose d'un identifiant numérique entier unique, le **numéro d'inode**, que l'on obtient en tapant **stat monfichier**. Chaque fichier sur un périphérique est défini par une **entrée dans le répertoire** des fichiers, elle contient les informations suivantes :

- Nom du fichier
- Type du fichier
- Emplacement (pointeur vers un périphérique et vers l'emplacement dans ce périphérique)
- Taille (en octets, mots ou blocs)
- Droits (**r, w, x**)
- Date, heure (de la dernière modification, lecture etc...)

Méthodes d'accès à un fichier

Il existe deux méthodes d'accès à un fichier.

- **Accès séquentiel** utilisé par les systèmes de fichiers basés sur les bandes magnétiques.

Le fichier occupe un espace contigu.



- **Accès direct** rendu possible par les disques. Un fichier est une suite d'enregistrements (ou blocs) numérotés, et pas nécessairement contigus en mémoire. Il permet l'accès à n'importe quelle partie du fichier sans avoir à le parcourir entièrement depuis le début jusqu'à la position recherchée.

Méthode de gestion de l'espace libre

Lorsqu'un nouveau fichier est créé, ou qu'un fichier existant voit sa taille augmenter (fichier log), le système doit trouver de l'espace libre. On présente trois méthodes de gestion de l'espace libre :

- **Vecteur binaire.** Chaque bloc du disque est représenté par un bit valant 0 (libre) ou 1 (occupé). Cette méthode est inefficace en pratique car le vecteur binaire **occupe trop de place**.
- **Liste chaînée.** Chaque bloc libre contient l'adresse du prochain bloc libre. Exemple : les blocs 8, 10, 11, 14 sont libres. On maintient seulement l'adresse du bloc 8, on place dans le bloc 8 l'adresse du bloc 10, dans le bloc 10, l'adresse du bloc 11 etc. Inefficace si l'on cherche **un espace de plusieurs blocs contigus**.
- **Liste de blocs contigus.** On construit un tableau dans lequel chaque entrée contient l'adresse d'un bloc libre, et **le nombre de blocs libres qui le suivent**.

Méthode d'allocation d'espace aux fichiers

Deux types d'effets indésirables peuvent se manifester quand on alloue de l'espace à des fichiers.

- **Fragmentation interne** : elle a lieu quand on a réservé **trop d'espace pour un fichier**. Par exemple, si un bloc représente 512 octets sur le disque, un fichier de 515 octets conduit à une fragmentation interne car deux blocs sont nécessaires, mais le second n'utilise que **3 octets sur 512**.
- **Fragmentation externe** : quand, sur un disque, n blocs sont libres, et qu'on veut copier un fichier avec **l'allocation contiguë** de $n_1 \leq n$ blocs, il se peut qu'il n'existe aucune zone de **n_1 blocs contigus** pour y placer le fichier.

Méthode d'allocation d'espace aux fichiers

Il existe trois manières d'allouer de l'espace aux fichiers :

- 1 l'allocation contiguë
- 2 l'allocation chaînée
- 3 l'allocation indexée

Allocation contiguë : tout fichier est rangé dans une série de blocs adjacents.

La représentation du fichier dans l'entrée de répertoire contient alors :

- L'adresse du premier bloc
- Le nombre de blocs (la « longueur »)

Fichier	Début	Longueur
log	19	10
a.txt	4	8
dp.c	35	18

Méthode d'allocation d'espace aux fichiers

L'allocation pour un nouveau fichier de n blocs consiste alors à trouver dans la liste des blocs libres, une zone contenant au moins n blocs contigus. Différentes stratégies de recherche :

- **First-fit** : la première zone trouvée est choisie.
- **Best-fit** : on choisit la plus petite zone contenant n blocs contiguës. Inconvénient : création de nombreuses zones libres trop petites pour y placer un fichier. Encombrement de la liste des blocs libres, longue recherche avant de trouver de la place.
- **Next-fit** : identique à First-fit, mais on reprend la recherche à la position courante dans la liste des blocs libres au lieu de la parcourir depuis le début. On évite ainsi de concentrer les petites zones libres au début, on trouve plus rapidement une zone libre.

Bilan de l'allocation contiguë : Accès directs et séquentiels très faciles à mettre en œuvre. Mais il faut connaître la taille du fichier à sa création, ou le déplacer entièrement dans une zone plus grande si sa taille augmente.

Méthode d'allocation d'espace aux fichiers

La seconde méthode est l'**allocation chaînée**. Les blocs du fichier sont chaînés : chaque bloc contient **un pointeur vers le bloc suivant**, le pointeur du dernier bloc pointe vers **NULL**. La représentation du fichier dans l'entrée de répertoire contient alors :

- **Un pointeur vers le premier bloc**
- **Un pointeur vers le dernier bloc** (il sert à trouver la fin du fichier sans avoir à parcourir tous les blocs et ajouter plus facilement du contenu à la fin)

Bilan de l'allocation chaînée : La taille des fichiers n'est limitée que par la capacité du disque, plus de problème dûs à **la fragmentation externe**. Mais **perte de place** dans les blocs pour stocker le pointeur vers le bloc suivant. L'accès direct peut être très fastidieux pour les grands fichiers.

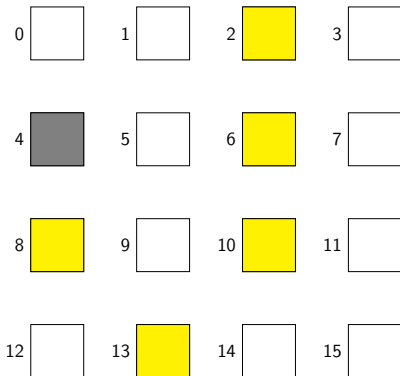
Méthode d'allocation d'espace aux fichiers

Dans l'allocation indexée à un seul niveau (« simple indirection »), tout fichier possède un bloc spécial, appelé **bloc d'index**, dont chaque élément est un pointeur vers un bloc du fichier.

La représentation du fichier dans l'entrée de répertoire contient alors un pointeur vers le bloc d'index du fichier.

Entrée du répertoire	
Fichier myfile	Bloc d'index

Contenu du bloc



Méthode d'allocation d'espace aux fichiers

Indexation à deux niveaux : dans l'entrée du répertoire du fichier, un pointeur pointe vers un bloc d'index dit de niveau 2. Chaque entrée du bloc d'index de niveau 2 est un pointeur vers un bloc d'index de niveau 1. Chaque entrée de bloc d'index de niveau 1 est l'adresse d'un bloc du fichier (taille maximale d'un fichier : $128 \times 128 = 16\,384$ blocs).

Indexation mixte (UNIX). Dans l'entrée du répertoire, on trouve 10 pointeurs directs vers les 10 premiers blocs du fichier, le 11^{ème} est un pointeur vers un bloc d'indexation simple, le 12^{ème}, un pointeur vers un bloc d'indexation double, le 13^{ème} un pointeur vers le bloc d'indexation triple.

Un fichier de moins de 10 blocs ne nécessite aucun bloc d'indexation. Or avec des blocs de 4096 octets, cela concerne les fichiers de moins de 40 ko, les plus nombreux d'après les études statistiques. Le nombre de blocs d'index augmente ensuite avec la taille du fichier.

Avec des adresses de blocs sur 32 bits, un bloc peut contenir 128 pointeurs. La taille maximale d'un fichier est donc de $10 + 128 + 128^2 + 128^3 = 2\,113\,674$ blocs.