Système d'Exploitation 2

Chapitre 3 : La gestion de la mémoire secondaire

(partie 4)

1 ère Année Second Cycle

Dr. M. Baba Ahmed

Types de systèmes de fichiers

- Il existe plusieurs systèmes de fichiers mais qui ne sont pas tous largement utilisés

Les systèmes les plus courants sont :

- FAT16, FAT32, exFAT et NTFS (Windows)
- HFS+ et APFS (macOS/Mac OS X).
- Linux utilise actuellement ext4 (successeur de ext3 et ext2)

Types de systèmes de fichiers

Le system de fichiers définit différentes tailles de cluster (groupe de bloc, secteurs), qui peuvent affecter la taille sur le disque

Exemple :

Soit deux SF différents, SF1 = taille cluster (2Ko), SF2 = taille cluster (32Ko), un fichier de 1Ko doit être stocké dans les deux SF

La taille occupé pour SF1 sera de 2 Ko sur le disque, la taille occupé pour SF2 est de 32 Ko

L'espace restant sera inutilisé

Remarque : lorsque la taille du cluster est petite, il y'aura moins de perte d'espace mémoire

Et vice versa

Types de systèmes de fichiers

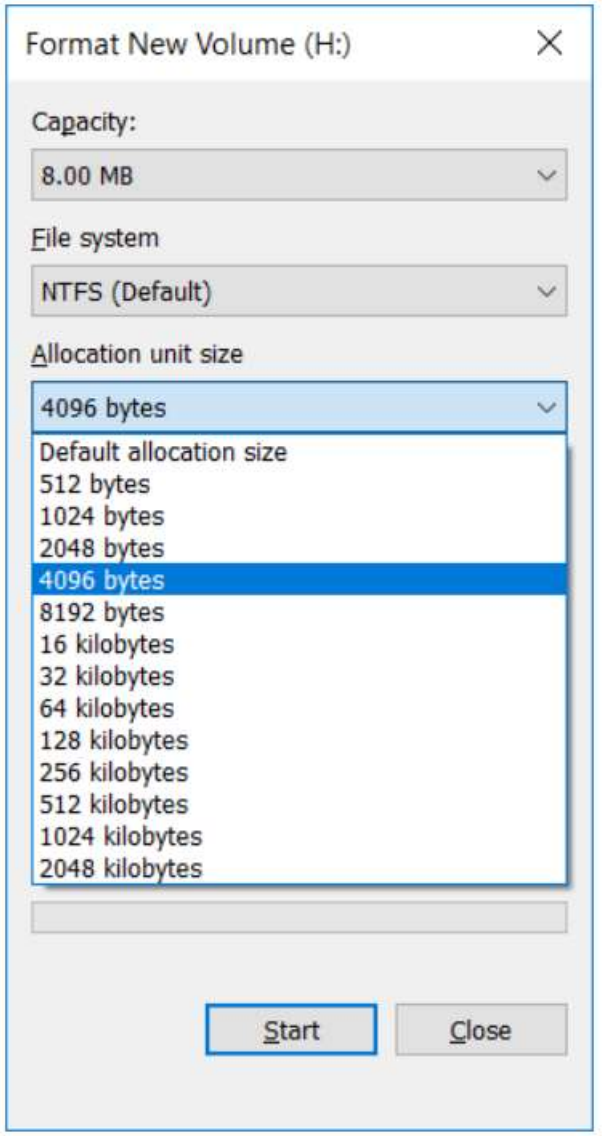
- Si l'utilisateur ne précise pas la taille de bloc (cluster) demandée,

la valeur par défaut est en fonction de la taille du volume

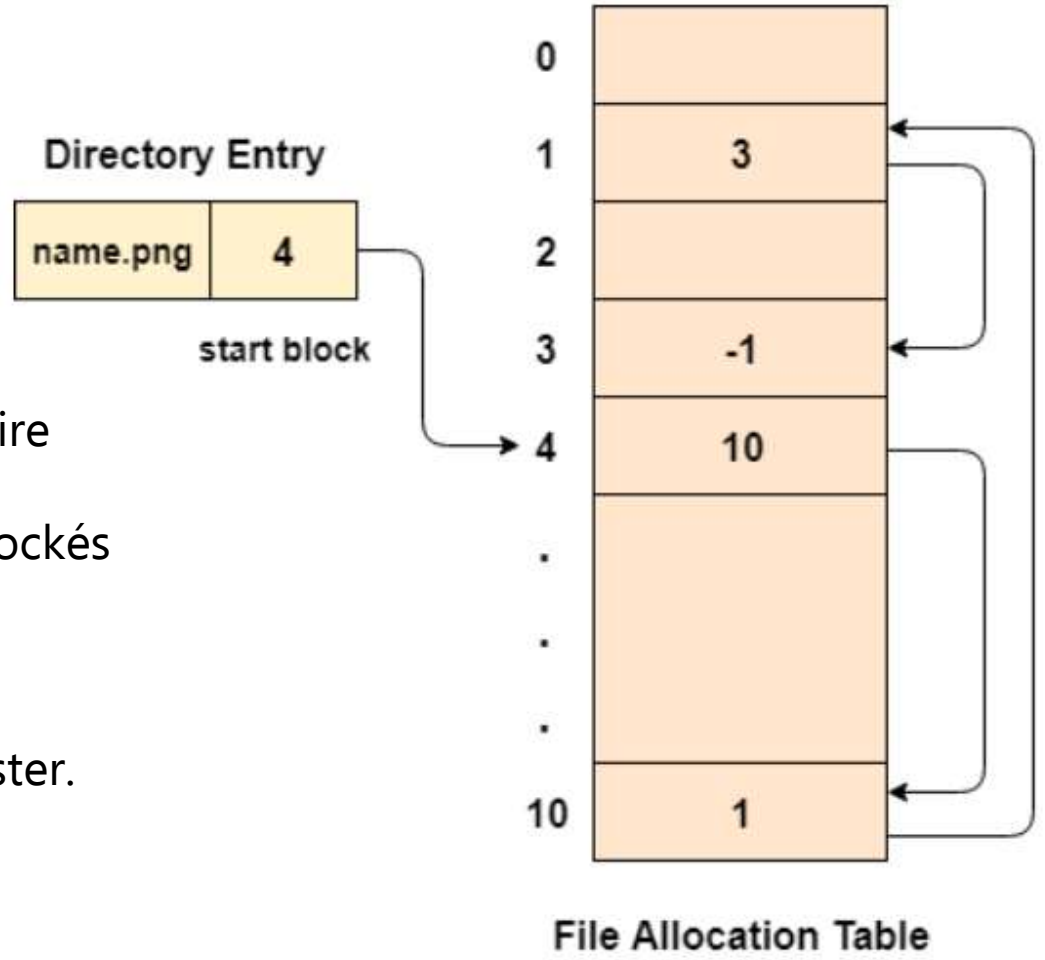
- Plus la taille du volume augmente plus la taille du cluster doit

augmenter

Volume size	Windows OS *
7 MB–512 MB	4 KB
512 MB–1 GB	4 KB
1 GB–2 GB	4 KB
2 GB–2 TB	4 KB
2 TB–16 TB	4 KB
16TB–32 TB	8 KB
32TB–64 TB	16 KB
64TB–128 TB	32 KB
128TB–256 TB	64 KB



File Allocation Table (FAT)



FAT utilise deux tables :

- Utilise un répertoire racine qui stocke une table de répertoire

contenant des entrées qui décrivent les fichiers et dossiers stockés

sur le volume (métadonnées)

- La table d'allocation (FAT), décrit l'utilisation de chaque cluster.
- La FAT est indexé par les numéros de blocs (cluster)
- Les entrées contiennent également des pointeurs vers le cluster suivant dans un fichier, permettant au FAT de suivre la séquence des clusters qui composent un fichier.
- Comme l'allocation chaînée, mais les liens sont stockés dans une table séparée

File Allocation Table (FAT)

Exemple :

Type de FAT	Nombre de cluster	Taille de cluster	Taille disque
FAT 12	2^12 = 4096	4 Ko – 64 Ko	16 Mo – 256 Mo
FAT 16	2^16 = 65536	4 Ko – 64 Ko	256 Mo – 4 Go
FAT 32	2^32 = 4294967296	512 octet – 4 Ko	2 To - 16 To

Taille disque = nombre_cluster * taille_cluster

Remarque : la taille de cluster 512 octets est possible pour FAT12 et FAT16

La taille de cluster de FAT32 peut aller jusqu'à 64 Ko

FAT 12 (12 bits pour représenter les blocs (clusters) ….. etc

New Technology File System (NTFS)

- NTFS utilise Master File Table qui stocke les indexations et informations sur les fichiers présents dans le support
- Elle inclut : l'emplacement des fichiers dans le répertoire, l'emplacement physique des fichiers sur le lecteur et les métadonnées des fichiers ainsi la liste de contrôle d'accès au fichier (ACL)
- La taille d'un cluster peut varier de 512 octets jusqu'à 64 Ko
- La suppression d'un fichier (corbeille) rend le fichier moins accessible, mais ne le supprime pas

réellement. Le système de fichiers de l'ordinateur supprime le chemin d'accès à ce fichier.

Cluster size	Largest volume and file
4 KB (default size)	16 TB
8 KB	32 TB
16 KB	64 TB
32 KB	128 TB

FAT vs NTFS

FAT	NTFS
• Supporte une taille maximale de fichier 4 Go	• Supporte une taille plus de 4 Go
• Pas de permission sur les fichier	• Contrôle les accès au fichiers
• Aucun chiffrement de données	• Chiffre les données (Encrypting File System)
• Aucun	• Permet la gestion des Quotas de disque
• Pas de compression	• Compression de fichiers

Extended file system Ext2, Ext3, Ext 4

- Un système de fichiers journalisés est un système de fichiers tolérant aux pannes qui permet d'assurer l'intégrité des données en cas de problème matériel, de panne ou d'arrêt brutal du système.
- Cette fonctionnalité est assurée par la tenue d'un Journal référençant les opérations d'écriture sur le support physique avant que ce dernier ne soit réellement mis à jour.
- NTFS possède une fonction de journalisation

Extended file system Ext2, Ext3, Ext 4

❑ Allocateur multibloc : allocateur qui décide quels blocs libres seront utilisés pour écrire les données.

- Si le système doit écrire un nombre important de données, il devra appeler l'allocateur de blocs plusieurs fois (allocateur multibloc)

❑ Allocation différée : L'attribution différée, n'alloue pas les blocs immédiatement lorsque le processus écrits, mais retarde l'allocation des blocs pendant que le fichier est conservé dans le cache, jusqu'à ce qu'il soit réellement écrit sur le disque.

Extended file system Ext2, Ext3, Ext 4

Ext 2 :

- Développé pour surmonter la limitation du système de fichiers **ext** d'origine.
- Pas de fonctionnalité de journalisation.
- La taille maximale des fichiers individuels peut aller de 16 Go à 2 To
- La taille globale du système de fichiers ext2 peut aller de 2 To à 32 To

Ext 3 :

- Permet la journalisation.
- La taille maximale des fichiers individuels peut aller de 16 Go à 2 To
- La taille globale du système de fichiers ext3 peut aller de 2 To à 32 To

Extended file system Ext2, Ext3, Ext 4

Ext 4 :

- La taille maximale des fichiers individuels peut être comprise entre 16 Go et 16 To
- La taille globale maximale du système de fichiers ext4 est de 1 Eo (exaoctet). 1 Eo = 1024 Po (pétaoctet). 1 Po = 1024 To (téraoctet).
- Le répertoire peut contenir un maximum de 64 000 sous-répertoires (contre 32 000 en ext3)
- Allocation multibloc, allocation différée
- Possibilité de désactiver la fonction de journalisation.

Sécurité et protection des fichiers

- La protection des fichiers ou processus de sécurisation des fichiers contre l'accès, la modification ou la suppression non autorisés.
- De garantir la confidentialité.

Divers mécanismes et techniques sont fournit :

- Mot de passe
- Les autorisations de fichiers (droits d’accès)
- Le chiffrement (cryptage)
- Les listes de contrôle d'accès (ACL)
- Audit et journalisation

Sécurité et protection des fichiers

- Autorisations de fichiers Les autorisations de fichiers permettent à l'administrateur système d'attribuer des droits d'accès spécifiques aux utilisateurs et aux groupes (r, w, x)
- Cryptage Le cryptage est le processus de conversion du texte brut en texte chiffré pour protéger les fichiers contre tout accès non autorisé.
- Les fichiers cryptés ne sont accessibles qu'aux utilisateurs autorisés qui disposent de la clé de cryptage correcte pour les décrypter.

Sécurité et protection des fichiers

- Listes de contrôle d'accès (ACL) Les listes de contrôle d'accès (ACL) sont des listes d'autorisations attachées à des fichiers et des répertoires qui définissent quels utilisateurs ou groupes y ont accès et quelles actions ils peuvent effectuer sur eux.

```
carole@carole-UX430UAR:~/Documents/Systeme Avance M2$ getfacl toto.txt
# file: toto.txt
# owner: carole
# group: carole
user::rw-
group::r--
other::r--
```

- Audit et journalisation L'audit et la journalisation sont des mécanismes utilisés pour suivre et surveiller l'accès, les modifications et les suppressions de fichiers.

Cela implique la création d'un enregistrement de tous les accès et modifications aux fichiers, y compris qui a accédé au fichier

Contrôle d’accès par classe utilisateurs

Création d’utilisateurs et groupes :

#**sudo useradd -m <user_name> -p <password>** (creation d’un utilisateur>

#**sudo userdel -r <user_name>** (suppression d’un utilisateur)

sudo getent passwd (affichage de tout les utilisateurs Linux)

sudo groupadd <group_name> (création d’un groupe)

sudo usermod –a -G <group_name> <user_name> (ajout d’un utilisateur au groupe)

members <group_name> (afficher les membres du groupe avec members)

groupdel <group_name> (suppression d’un groupe)

#**cat /etc/group** (affichage de l’ID des groupes)

Contrôle d’accès par classe utilisateurs

Linux : (affichage des utilisateurs)

- **Les utilisateurs du système** sont des entités créés automatiquement lors de l'installation du système d'exploitation ou lors de l'installation de logiciels. Ces utilisateurs sont généralement utilisés pour exécuter certaines tâches système
- **Les utilisateurs normaux** sont des utilisateurs humains créés par *root* ou un autre utilisateur avec des privilèges *root.*

```
ubuntu-user@HP:~$ getent passwd
root:x:0:0:root:/root:/bin/bash
daemon:x:1:1:daemon:/usr/sbin:/usr/sbin/nologin
bin:x:2:2:bin:/bin:/usr/sbin/nologin
sys:x:3:3:sys:/dev:/usr/sbin/nologin
sync:x:4:65534:sync:/bin:/bin/sync
games:x:5:60:games:/usr/games:/usr/sbin/nologin
man:x:6:12:man:/var/cache/man:/usr/sbin/nologin
lp:x:7:7:lp:/var/spool/lpd:/usr/sbin/nologin
mail:x:8:8:mail:/var/mail:/usr/sbin/nologin
```

Contrôle d’accès par classe utilisateurs

```
root : x : 0 : 0 : root : /root : /bin/bash

démon : x : 1 : 1 : démon : /usr/sbin : /usr/sbin/nologin
```

root : nom d'utilisateur utilisé pour se connecter au système.

x : mot de passe crypté de l'utilisateur

0 : Il s'agit de l'ID utilisateur (UID). 0 (zéro) est toujours réservé à root.

0 : Il s'agit de l'ID de groupe (GID)

root : infos. supplémentaires sur l'utilisateur. Par exemple, d'autres noms, etc

/root : Le répertoire personnel

/bin/bash : Il s'agit du chemin du shell de ligne de commande utilisé par l'utilisateur.

Contrôle d’accès par classe utilisateurs

Windows :

- Utilise une Liste de contrôle ACL
- Gere les autorisation sur :

Utilisateur, utilisateurs authentifiés, système, et admin

- Type d’autorisation :

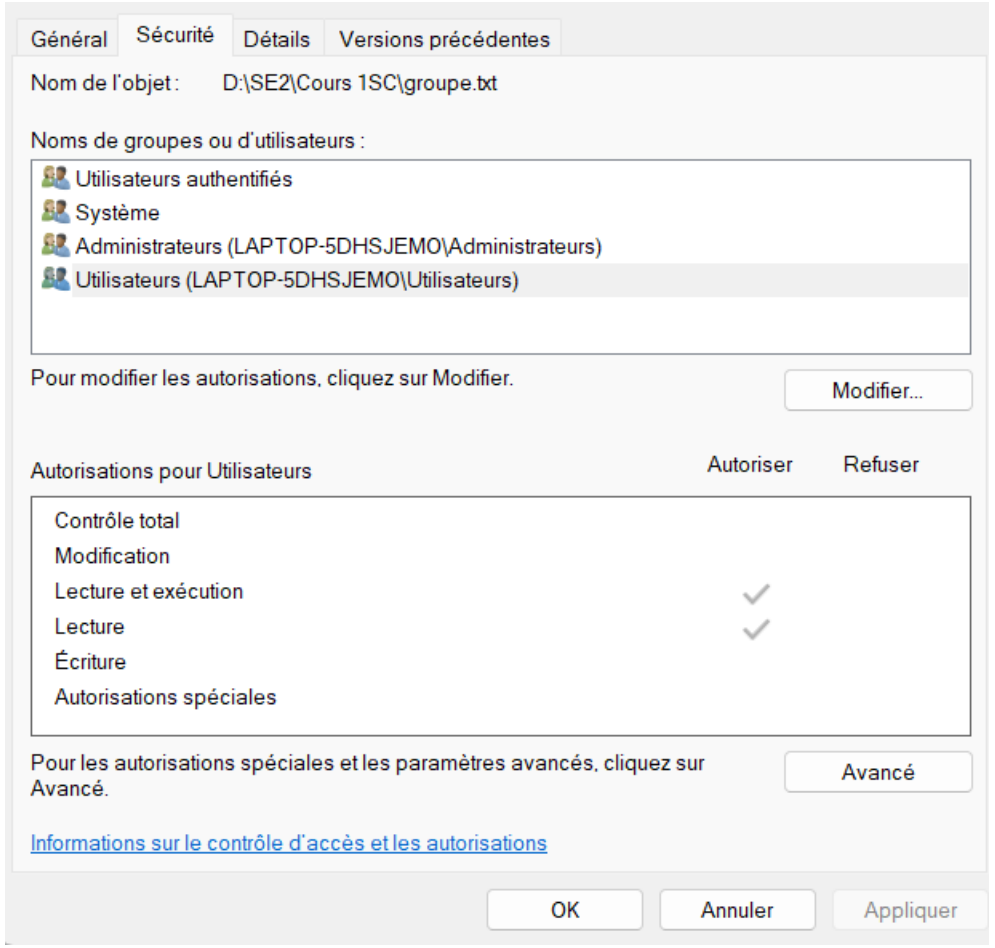
Contrôle total : (lecture, écriture …..)

les utilisateurs peuvent modifier les paramètres

des autorisations pour tous les fichiers et sous-répertoires

Modification, lecture, écriture ….etc

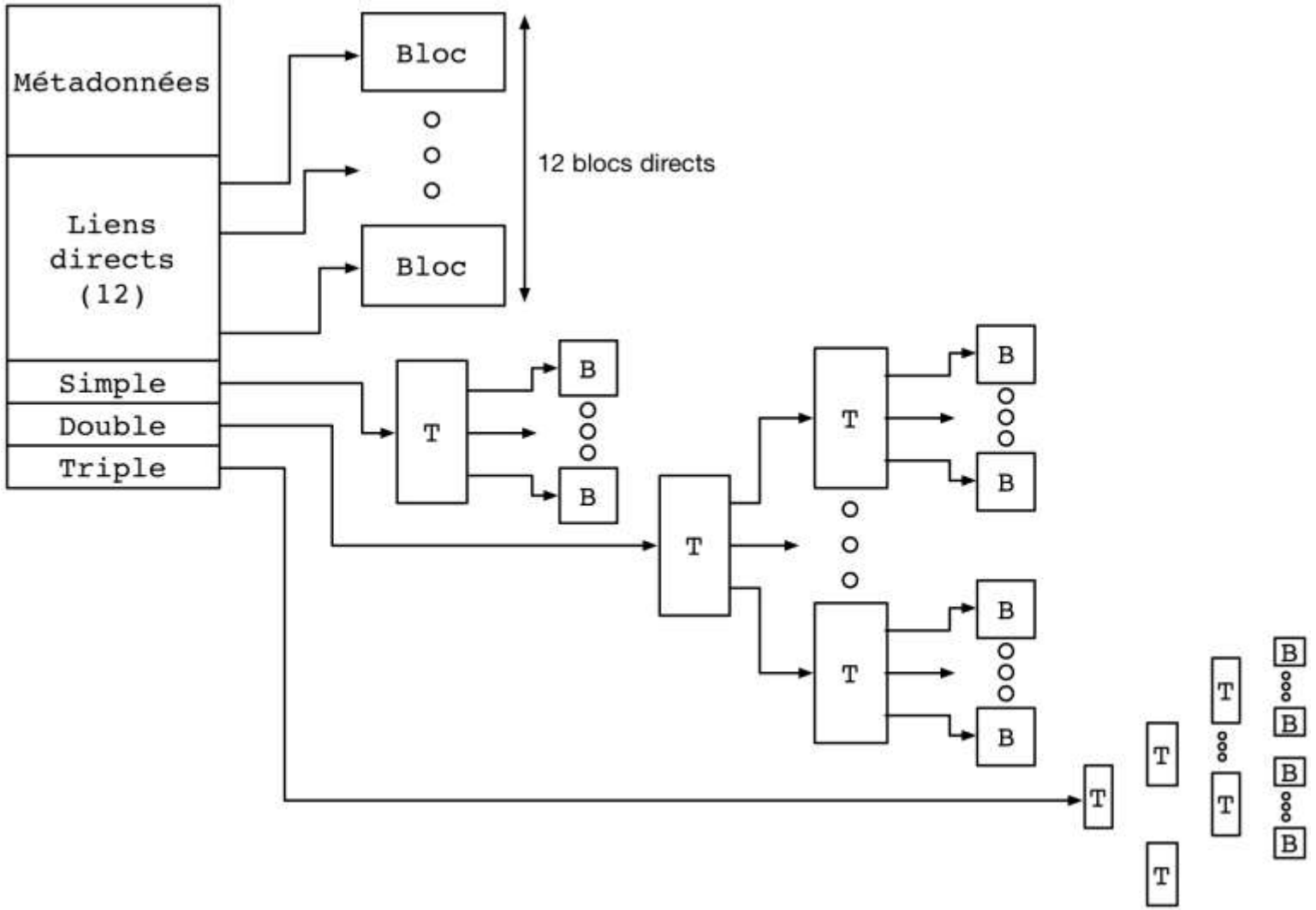
Autorisations spéciales : (suppression, possession, ….)



Indexation a plusieurs niveaux (Linux, UNIX)

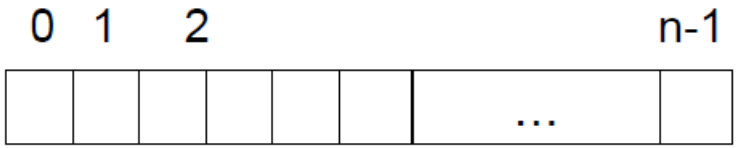
- un i-node contient un nombre limité de liens directs vers des blocs de données (une configuration standard est de 12 liens “directs” de ce type)
- La structure de l’i-node permet d’utiliser des liens supplémentaires et donc des fichiers plus volumineux en utilisant plusieurs niveaux d’indirections.
- Les premiers blocs d’un fichier sont accessibles directement
- Si le fichier contient des blocs additionnels, les premiers sont accessibles à travers un niveau d’indices
- Les suivants sont accessibles à travers 2 niveaux d’indices, etc.
- Permet accès rapide à petits fichiers, et au début de tous les fichiers

Indexation a plusieurs niveaux (Linux,UNIX)



Gestion des espaces libres (vecteur de bits Bitmap)

- Vecteur de bits (n blocs)



0 ⇒ block[i] libre

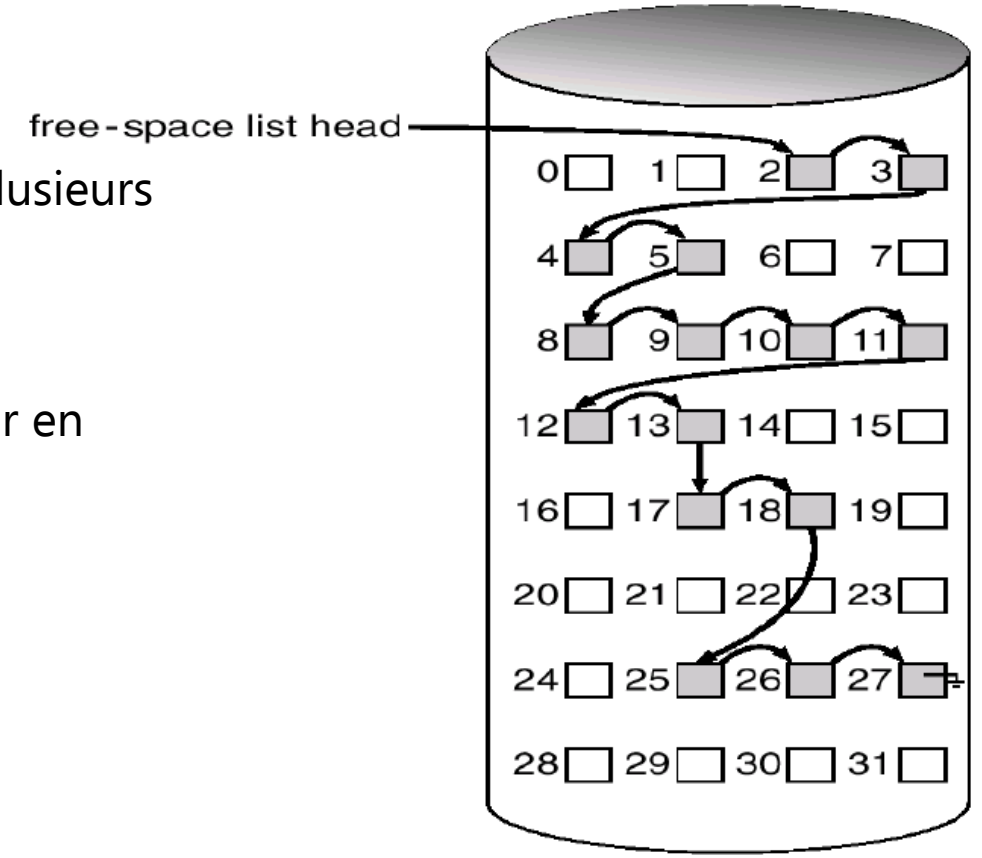
1 ⇒ block[i] occupé

Exemple d'un vecteur de bits où les blocs 3, 4, 5, 9, 10, 15, 16 sont occupés: 0001110001100011…

- Bitmap gardée en mémoire principale rapide (demande de l'espace de mémoire principale)
- Bitmap gardée en mémoire secondaire (temps de lecture de mémoire secondaire)

Gestion des espaces libres (liste chainée)

- Pour trouver plusieurs blocs de mémoire libre, plusieurs accès de disque.
- Pour augmenter l'efficacité, nous pouvons garder en mémoire centrale l'adresse du 1er bloc libre



Les matrices de contrôle d’accès

- Introduit en 1971 par **Butler Lampson**
- Mise en œuvre d’un modèle de protection.
- Contient des lignes qui représentent le domaine (utilisateur, processus)
- Les colonnes, représentent les objets ou les ressources

❑ **O**, l’ensemble des entités objet qui sont impliquées dans le système.

❑ **S**, un ensemble de sujet qui se compose d’entités actives

❑ **R**, un ensemble de droits. (lecture, écriture, exécution, autre)

Les matrices de contrôle d'accès

Exemple :

		F1	**F2**	**F3**	**Printer**	**D1**	**D2**	**D3**	**D4**
Sujets (S)	**D1**	r		r			switch		
	D2				prints			switch	switch
	D3		r	x					
	D4	rw		rw		switch			

Objets (O)

- La matrice d'accès fournit un mécanisme pour définir le contrôle d'accès entre Domaine (Sujets)/Ressources (Objets)
- Les processus doivent pouvoir basculer d'un domaine (Di) vers un autre domaine (Dj) si et seulement si un droit de basculement est autorisé