



Chapitre VI : La gestion des disques et des fichiers

- Les supports de stockage**
- Les fichiers**
- L'organisation des fichiers**



Introduction

- La gestion des fichiers recouvre deux activités :
 - **Une gestion statique** : taille, date de création, localisation
 - **Une gestion dynamique** : ouverture, gestion des adresses de zones allouées sur le disque



6.1 Supports de stockage

- Les anciens supports de stockage étaient essentiellement à accès séquentiel
- **Exemples :**
 - analogie avec les K7 de magnétoSCOPE,
 - bandes magnétiques,
 - rubans perforés etc.
- Aujourd'hui, il existe toujours des bandes



Supports de stockage

- On préfère utiliser des disques magnétiques
 - Rapidité des accès
 - Fiabilité
 - Grandes capacités
- **Exemple** : capacité actuelle d'un disque sur un PC 20Go, accès 5-10ms, transfert 20Mo/s



Supports de stockage

- Ces supports permettent un accès direct aux fichiers au moyen de leur adresse
- On spécifie son premier bloc
- **Problème :**
 - l'utilisateur ne connaît pas l'emplacement du fichier sur le disque : il ne connaît que son nom



Méthode Naïve

- On intègre au début de chaque fichier son nom
- On place le fichier sur le disque au premier endroit où on trouve de la place
- La recherche d'un fichier est alors séquentielle :
 - Étant donné le nom, on lit les blocs du disque à partir du début et on recherche la première occurrence du nom du fichier
- **Inconvénient** : on n'utilise alors pas du tout l'accès direct permit par la structure physique du disque dur

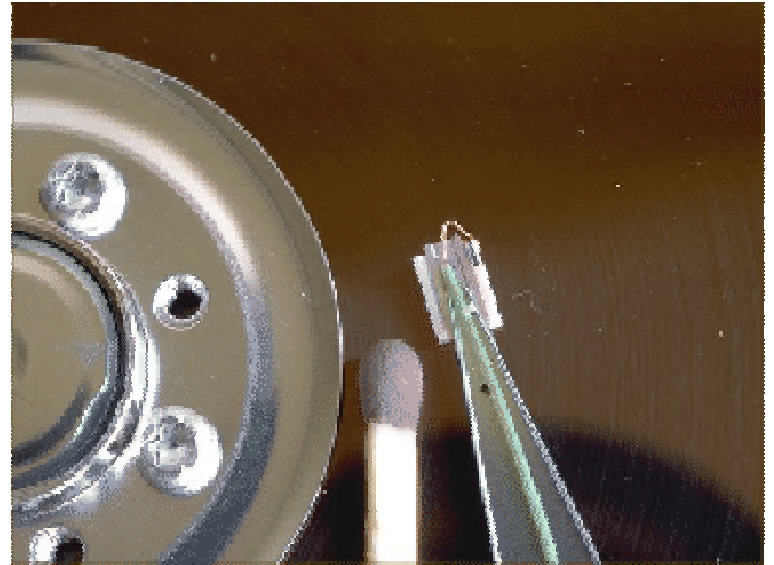
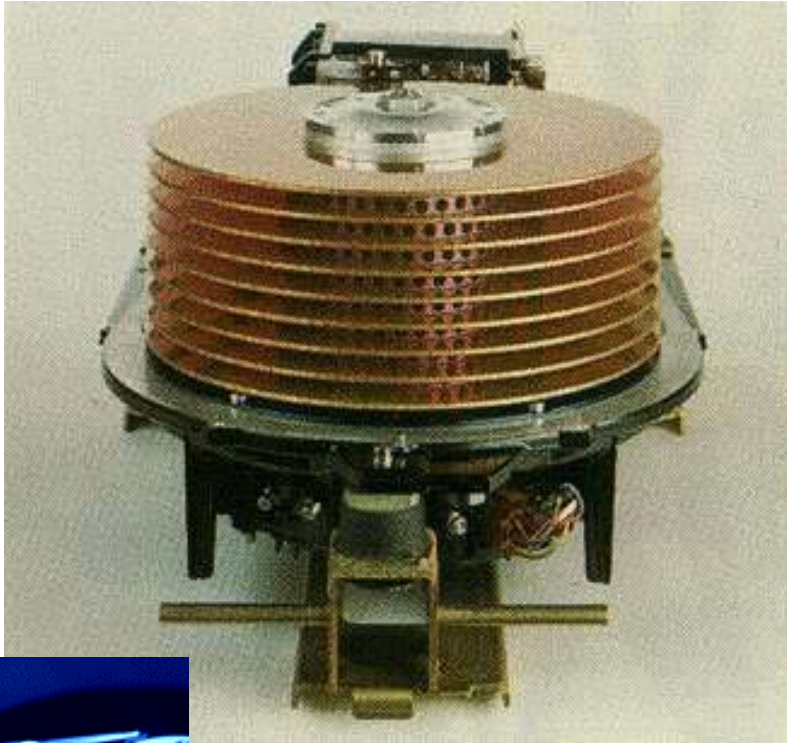
Le disque dur



- 1: Plateaux
- 2: Moteur (axe)
- 3: Mécanisme de déplacement des bras
- 4: bras supportant les têtes de lecture
Écriture (5)



Le disque dur

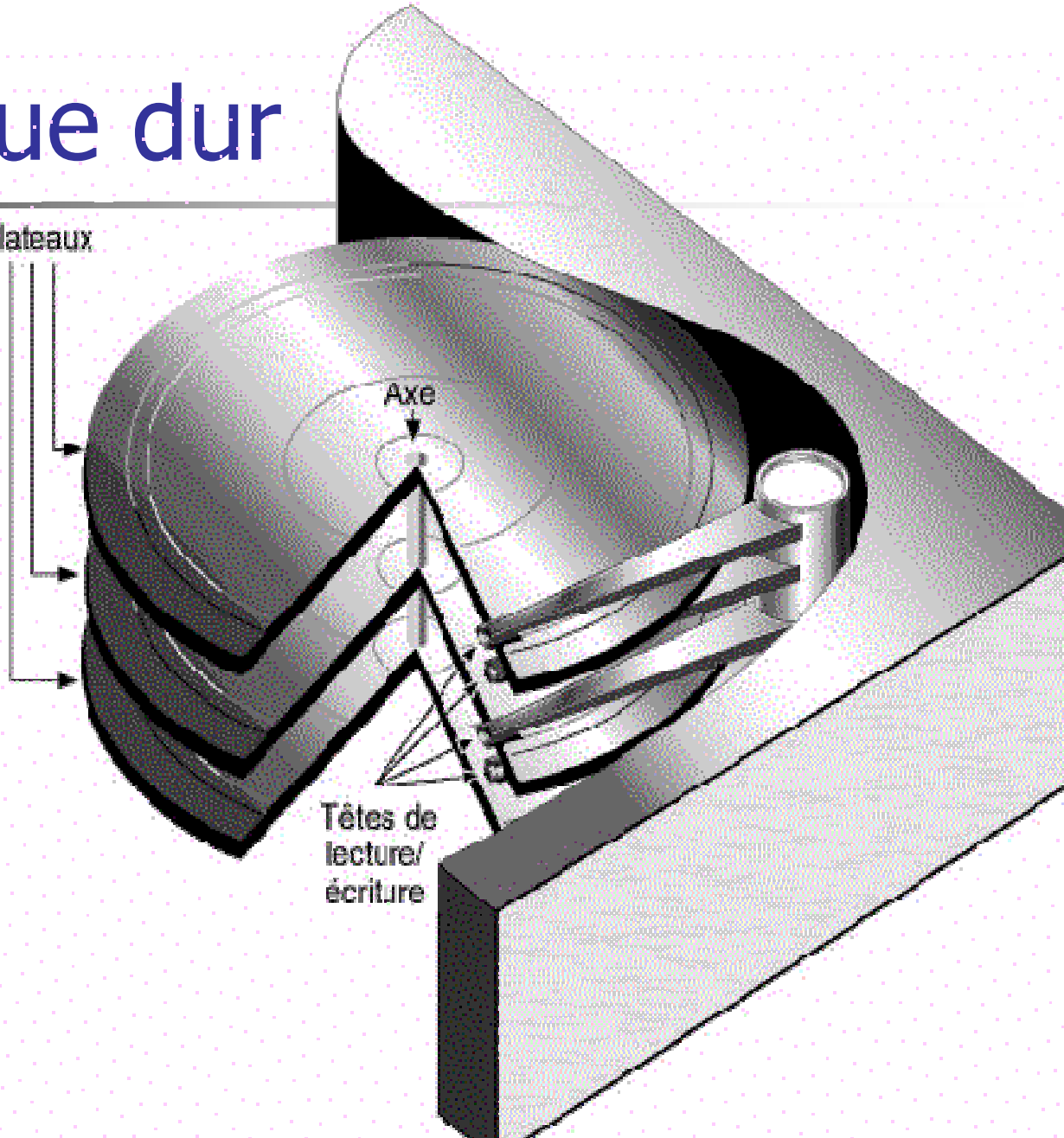


Le disque dur

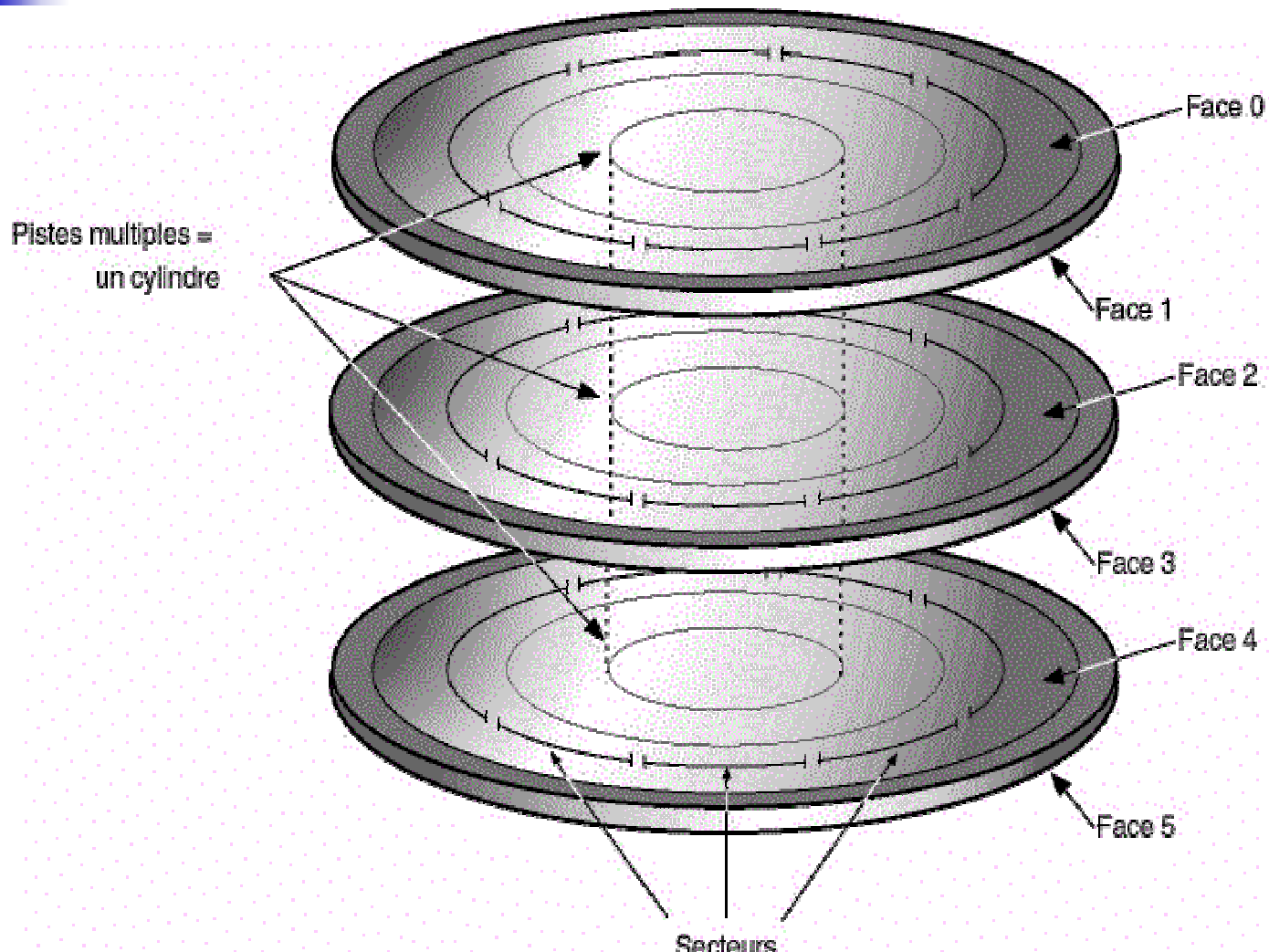
Plateaux

Axe

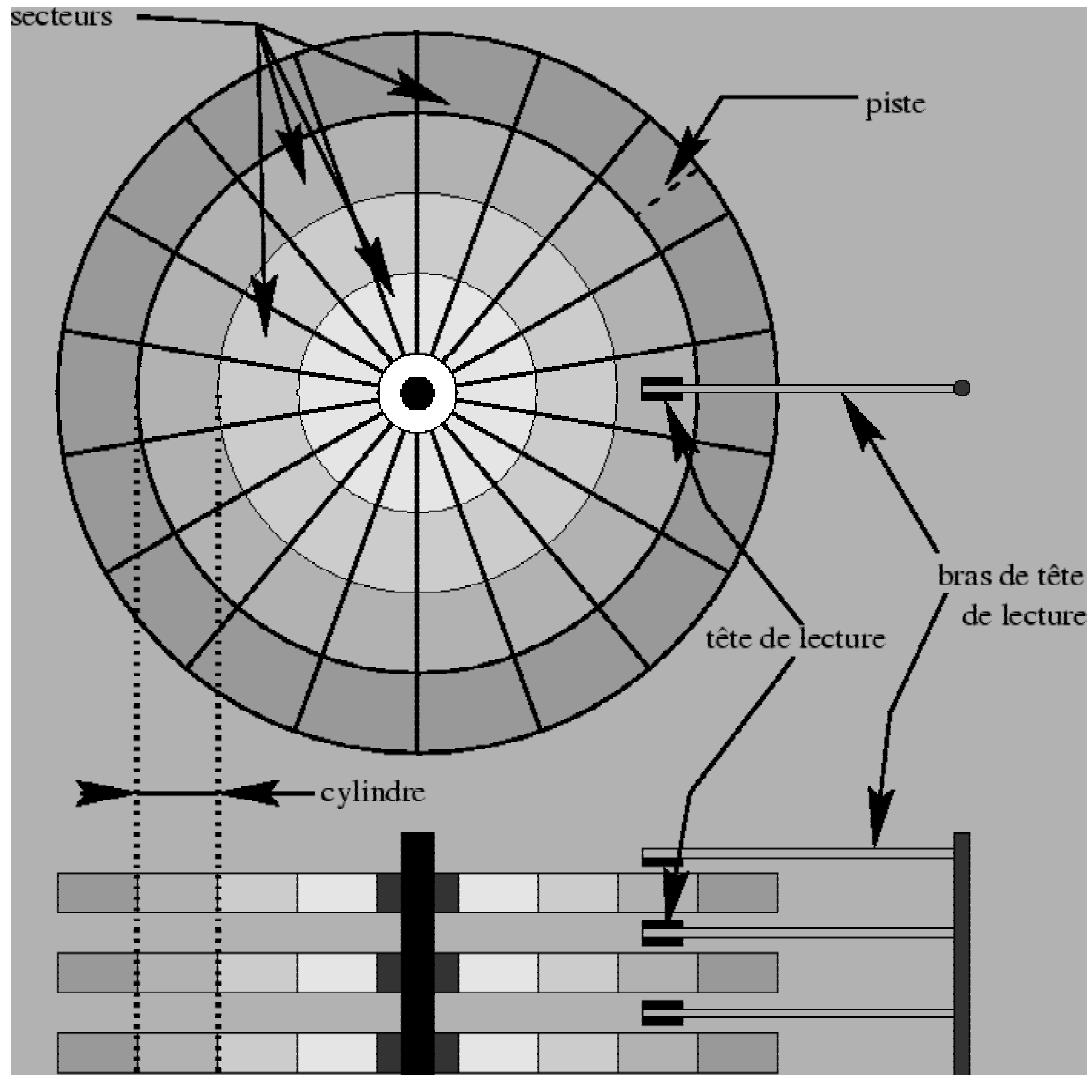
Têtes de lecture/écriture



Le disque dur



Le disque dur



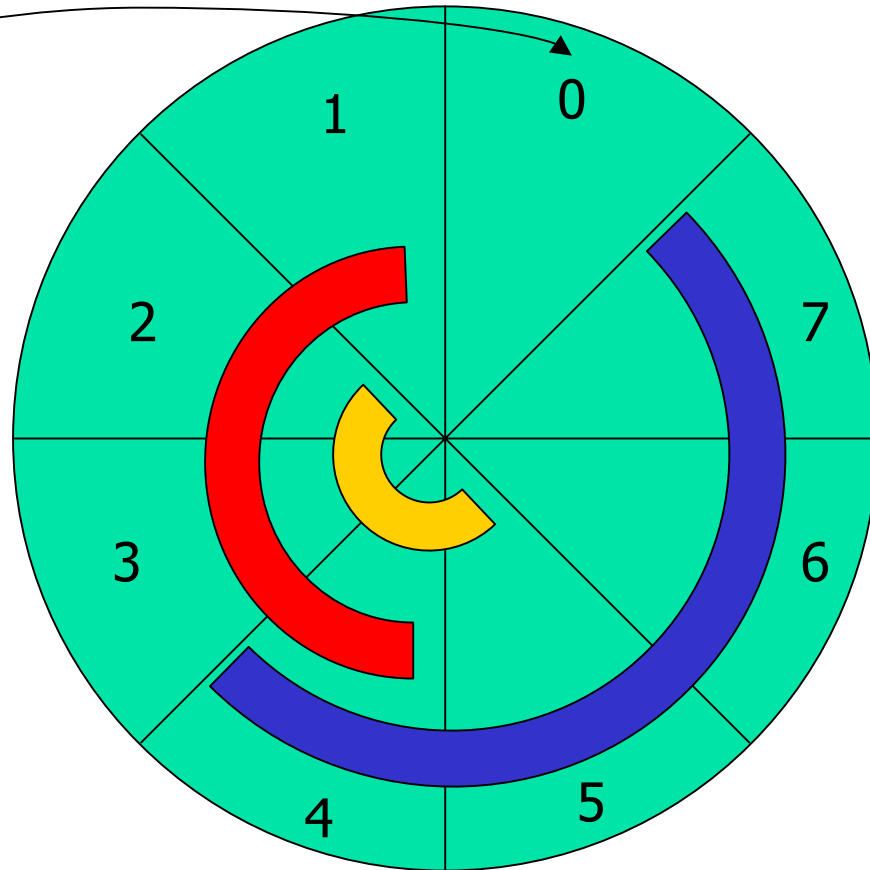


Méthode par indexation

- Formatage : le disque est découpé en zones logiques appelées **secteurs** regroupés sur des **pistes**
- L'ensemble des pistes de même numéro sur différents plateaux forme un **cylindre**
- **Solution** : le premier secteur contient un tableau T qui s'étend sur la globalité de la première piste et qui associe à chaque nom de fichier le numéro du secteur qui le contient
- **Inconvénients** : ...

Méthode par indexation

Nom	Sect
f1	2
f2	1





Méthode par indexation

- Étant donné un nom M, le positionnement de la tête de lecture sur le premier secteur (secteur 0) permet :
 - De charger l'index
 - De le parcourir à la recherche de la chaîne de caractères M
 - De déterminer le secteur qui contient le début du fichier
 - La tête de lecture du disque se positionne alors sur le secteur en question et recherche plus précisément le fichier



6.2 Les fichiers

6.2.1 Niveau de structuration d'un fichier

- Un fichier regroupe un ensemble de données appelé **contenu**
- Un fichier est associé à un nom
- Le contenu d'un fichier est un ensemble structuré d'articles (appelés aussi **enregistrements ou records**)
- Chaque article représente une information unitaire (unité d'information)
- **Exemple** : fichier du personnel d'une entreprise



6.2 Les fichiers

- Les articles se décomposent en **champs**
- Chaque champ représente une **propriété** de l'article
- **Exemple** : dans le fichier du personnel le champ nom représente le nom d'une personne
- **Les champs sont l'association**
 - entre un attribut (Nom, Adresse, etc.)
 - et une valeur ('toto', 'rue Quimonte' etc.)



6.2 Les fichiers

- Le plus souvent la structure des articles est identique dans une même fichier :
 - nombre d'attributs fixe
 - champs de taille fixe
- Ainsi tous les articles ont la même taille
- On parle alors d'article de longueur fixe par opposition aux fichiers comportant des articles de longueur variable
- Quand peut-on rencontrer des articles de taille variable ?
- Avantages/Inconvénients de ces deux formats



6.2 Les fichiers

6.2.2 Points de vue sur les fichiers

- Les fichiers peuvent être considérés selon différents points de vue :
 - Unité centrale
 - Système d'exploitation
 - Utilisateur
- On parle alors : de fichier physique, de fichier logique et de fichier ouvert



6.2 Les fichiers

- **Le fichier physique** : est le fichier dans la réalité de son stockage sur le support physique (disque)
 - La répartition physique de son contenu dépend des conditions de stockage
 - Les articles sont regroupés en blocs
 - Les articles sont contigus dans un bloc



6.2 Les fichiers

- **Le fichier logique** : c'est une abstraction du fichier physique qui fournit un moyen simple d'utiliser le fichier physique
 - Groupe structuré d'information qui définissent le fichier : nom, taille, date, propriétaire etc.
- Des fichiers logiques identiques peuvent correspondre à des fichiers physiques différents



6.2 Les fichiers

- **Le fichier ouvert** : correspond à une variable en mémoire centrale qui permet à l'utilisateur d'accéder au fichier physique (au moyen d'un programme Pascal par exemple)
- La mémoire contient alors la partie du fichier traitée par le programme en cours



6.2 Les fichiers

- Le déroulement d'un accès à un fichier s'effectue en 3 phases :
 1. Le processus demandeur **crée une variable référence au fichier** au moyen d'une fonction à laquelle on fournit le nom du fichier
 2. La partie gestion des fichiers du SE **établit la correspondance avec le fichier physique**
 3. La partie gestion de la mémoire se charge des **transferts des portions de fichier vers la mémoire** centrale en utilisant un zone tampon (buffer)



6.2 Les fichiers

6.2.3 La gestion statique

- La gestion statique des fichiers consiste à fournir des procédures pour organiser les fichiers sur le disque en vue de leur stockage et accès. Il s'agit :
 - D'une gestion **statique externe** : association du nom du fichier à l'endroit où est stocké le fichier
 - D'une gestion **statique interne** : permettant de repérer les différents blocs du disque constituant le fichier



6.2 Les fichiers

6.2.4 La gestion dynamique

- La gestion dynamique permet de gérer les fichiers lors des phases d'utilisation :
 - **Fichier ouvert** : on conserve son état
 - Le mode d'utilisation : lecture, écriture
 - Le descripteur sur le bloc courant
 - **La gestion des accès concurrents**
 - Mécanisme de verrouillage au niveau : du fichier de l'enregistrement, du bloc
 - Principe : un seul processus utilise le fichier, l'enregistrement ou le bloc
 - Avantages/Inconvénients ?
 - **Les transferts d'information** : la gestion de la zone tampon



6.3 Organisation des fichiers

6.3.1 La vision de l'utilisateur

- Au niveau logique l'utilisateur perçoit des fichiers organisés dans des répertoires
- Cette **structure arborescente** est souple et conserve la même structure quelque soit le niveau (**récursive**)
- Les répertoires sont considérés comme des fichiers particuliers : cette abstraction permet d'unifier les procédures de traitement de fichier avec celle de traitement de répertoire



6.3 Organisation des fichiers

- Sous UNIX : tout est géré à partir d'une seule et même racine ('/')
- Les utilisateurs sont tous possesseurs d'une branche dans laquelle ils sont maîtres (home directory)
- Ils peuvent la développer : créer de nouveaux fichiers et de nouveaux répertoires



6.3 Organisation des fichiers

- Sous UNIX, contrairement à DOS, l'utilisation de disquettes n'est pas vue comme l'ajout d'un nouveau système de fichiers, mais comme le raccordement d'un système de fichiers à l'arborescence existante
- On appelle cette opération un **montage**
 - **Exemple :** `mount /dev/fd0 /mnt/floppy`
- Les disquettes ou tout autre support de stockage sont greffés sur l'arbre principal par une commande `mount`



6.3 Organisation des fichiers

- **Notion d'alias, de raccourcis, de liens symboliques**
- Quelques fois il est utile de nommer les fichiers de plusieurs façons (alias) ou bien de pouvoir accéder à un fichier depuis plusieurs emplacements dans l'arborescence.
- On a alors recours aux :
 - Liens symboliques sous UNIX :
`ln /home/eric/toto18.pas /home/eric/prog/toto.pas`
 - Aux raccourcis sous Windows (cf TP)
 - Aux alias sur MACOS



6.3 Organisation des fichiers

6.3.2 Stockage des informations relatives à un fichier

- Deux informations sont fondamentales :
 - Le nom du fichier
 - L'adresse du premier bloc sur le disque
- On ajoute en général :
 - Les droits d'accès ou protection
 - La date de création
 - La date et heure de la dernière modification
 - La taille du fichier
 - Le propriétaire



6.3 Organisation des fichiers

- Sous Windows ces informations sont regroupées dans l'arborescence au niveau du répertoire dans lequel figure le fichier
- Un déplacement de fichier implique alors une recopie de ces informations
- Sous Unix ces informations sont enregistrées de façon indépendante dans de petites tables descriptives de fichier appelées i-nodes ou i-nœuds
- Un déplacement sous UNIX = un déplacement du i-node



6.4 Organisation du contenu

- **Gestion statique interne**
- La mémoire secondaire est composée d'un ou plusieurs supports de stockage
- L'allocation de la mémoire secondaire consiste à organiser le stockage des fichiers sur ces supports en fonction des zones disponibles
- L'organisation peut être réalisée suivant différents critères



6.4 Organisation du contenu

Généralement l'organisation doit optimiser :

- **la restitution du contenu des fichiers**
 - Même lorsque les données sont complexes il est nécessaire que les accès en lecture soient rapides
 - **Exemple** : dictionnaire, plan de ville, images satellitales
- **la modification du contenu des fichiers :**
 - Comment ne pas laisser de trous après une suppression
 - Comment insérer un nouvel article dans un fichier
 - Comment supprimer des articles
 - Comment replacer un information qui augmente de volume



6.4 Organisation du contenu

- **le remplissage du support :**

- Éviter la fragmentation

6.4.1 Organisation en fonction d'un accès par adresse

- Tout accès réel à un article se fait en spécifiant son adresse :
 - Implantation contiguë
 - Table d'allocation par fichier
 - Table des zones



6.4 Organisation du contenu

- **Implantation contiguë :**
 - La méthode de stockage consiste à déposer séquentiellement les blocs dans l'ordre ou ils apparaissent dans le fichier
 - Les problèmes sont nombreux :
 - **Accès** : facile lorsque les blocs sont de taille fixe (on peut calculer la position du n-ième bloc)
 - **Ajout d'information** : l'ajout d'information dans un fichier nécessite de pousser les bloc suivants
 - **Suppression** : nécessite un retassement (suppression des trous)

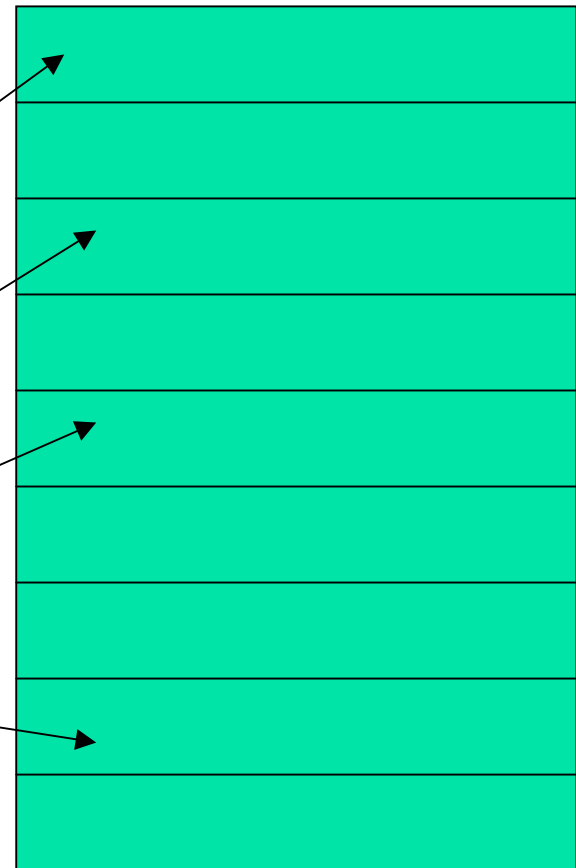
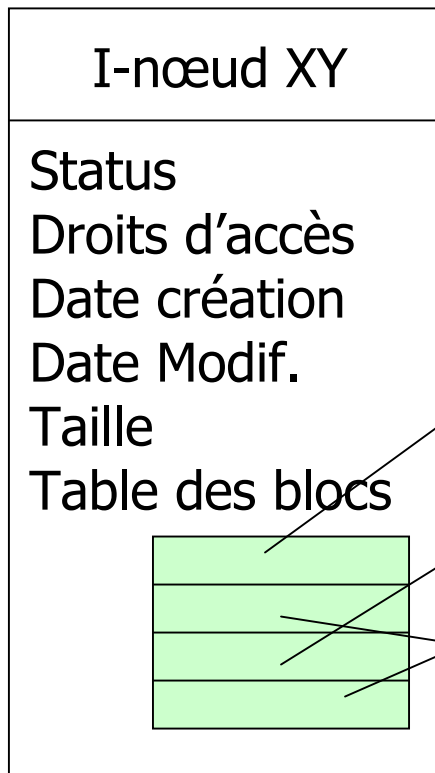


6.4 Organisation du contenu

- **Table d'allocation par fichier :**
 - Pour chaque fichier, on conserve une liste des zones utilisées par ce fichier
 - L'information reste relative à un seul fichier
 - On doit ajouter un structure pour la gestion des espaces libres
 - Stratégie choisie dans UNIX avec les i-noeuds

6.4 Organisation du contenu

Mémoire secondaire
Blocs du disque dur





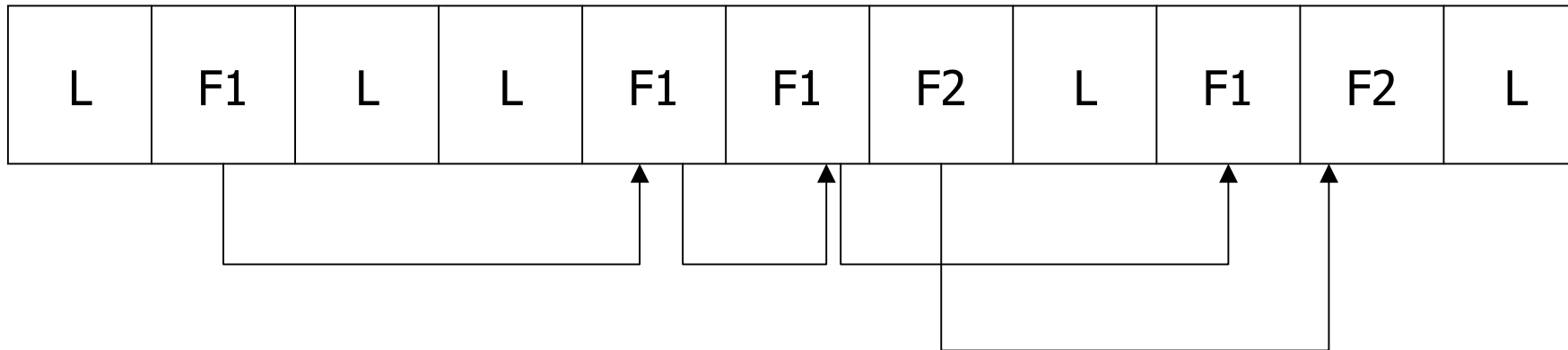
6.4 Organisation du contenu

■ **Table générale des zones**

- Les enregistrements sont insérés aux emplacements libres, sans contrainte de contiguïté
- Des blocs homogènes sont constitués, ils sont soit :
 - Occupés par un même fichier
 - Ou uniformément vides
- Une table générale des blocs et de leur affectation est maintenue à un endroit bien défini (dans chaque répertoire ou à la racine du disque)
- Les blocs appartenant au même fichier sont chaînés
- **Exemple** : MS-DOS, WIN9x et système FAT, FAT32

6.4 Organisation du contenu

Table générale des zones et chaînage





6.4 Organisation du contenu

- Comparaison UNIX / FAT :
 - Copie, déplacement des fichier
 - Modification d'un fichier
 - ...



6.4 Organisation du contenu

6.4.2 Organisation en fonction d'un accès au contenu du fichier

- On peut souhaiter demander au système l'accès à un article qui a certaines propriétés mais dont on ne connaît pas l'adresse (fichier très grand)
- **Exemple** : recherche du numéro de téléphone d'une personne en fonction de son nom



6.4 Organisation du contenu

- C'est par l'organisation du contenu du fichier que l'on peut retrouver rapidement les informations
- Certains attributs permettent de discriminer totalement une information
- Ces attributs sont appelés des **clés**
- Deux valeurs différentes de la clé font référence à des articles différents
- Ainsi dans un fichier il ne peut pas y avoir deux articles différents ayant la même valeur de clé



6.4 Organisation du contenu

- Conclusion : chaque valeur de clé correspond à une seule adresse dans le fichier
- Comment utiliser cette propriété pour accéder au contenu d'un article en se basant sur la valeur de la clé ?
- Différents types d'organisations ont été étudiés depuis plus de 40 ans :
 - Séquentielle (fichier triés)
 - Indexée
 - B-Arbre
 - Aléatoire



6.4 Organisation du contenu

- **Organisation séquentielle :**

- L'organisation séquentielle comme l'allocation continue des blocs ne permet pas de mettre en œuvre des techniques d'optimisation d'accès très poussées
- Lors d'un accès par adresse il est possible de sauter les articles (blocs) qui précèdent celui cherché
- C'est impossible lorsqu'on recherche un article (bloc) contenant un information particulière



6.4 Organisation du contenu

■ **Indexation :**

- Le principe consiste à bâtir une table de correspondances (table d'index) entre
 - les valeurs de la clé et
 - les adresses des articles sur le disque
- Pour faciliter les accès (rapidité) ce tableau doit être trié et chargé en mémoire centrale
- On parle alors de fichier séquentiel indexé

6.4 Organisation du contenu

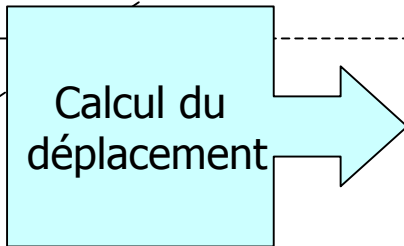
■ Exemple d'index

Fichier

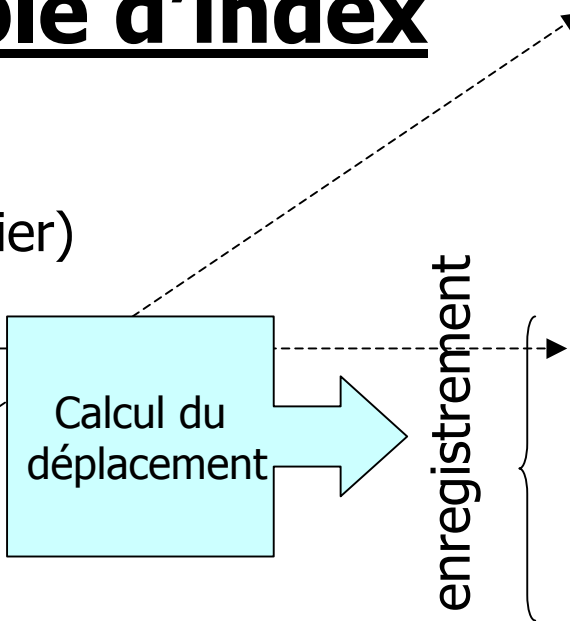
Sherpa
Paul
Rue Mouffetard
33 34 56 12
Asterix
Rue Maître Albert
45 67 98 22

Table d'index (fichier)

Asterix	2
Sherpa	1
Zazie	120



Un enregistrement



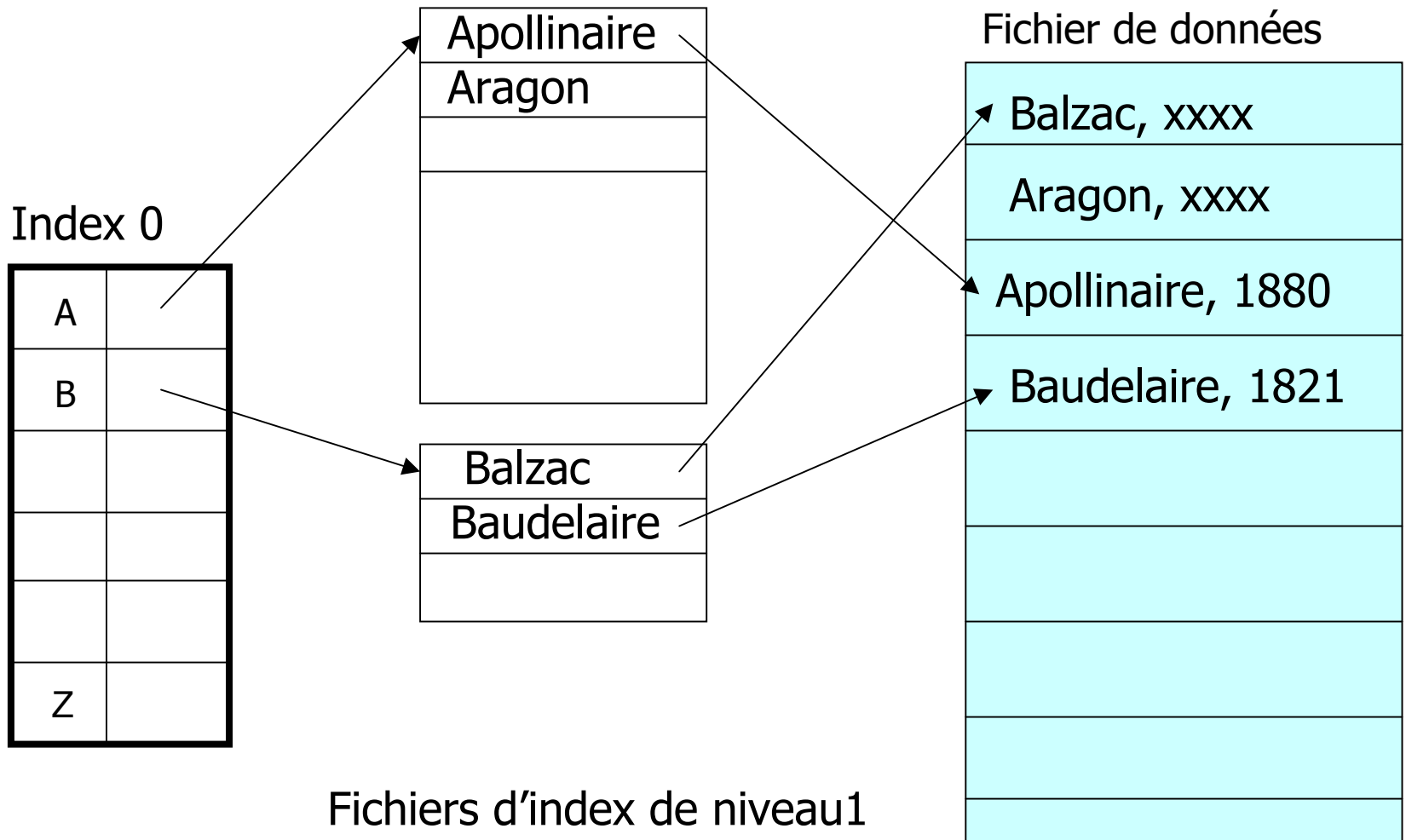


6.4 Organisation du contenu

- Le principe de l'indexation peut être complété de plusieurs manières :
 - Au lieu de la clé on peut utiliser un attribut assez discriminant (qui n'est pas une clé stricte)
 - **Exemple** : mots d'un dictionnaire (1 mot = plusieurs sens donc plusieurs définitions)
 - On peut créer et combiner plusieurs index suivant les recherches effectuées (par nom, par adresse etc.)
 - On peut constituer une clé à partir de plusieurs attributs (par exemple nom + prénom s'il y a des homonymes)
 - Pb quand la table d'index est trop grande

6.4 Organisation du contenu

Fichier associé à plusieurs niveaux d'index





6.4 Organisation du contenu

■ **Les B-arbres**

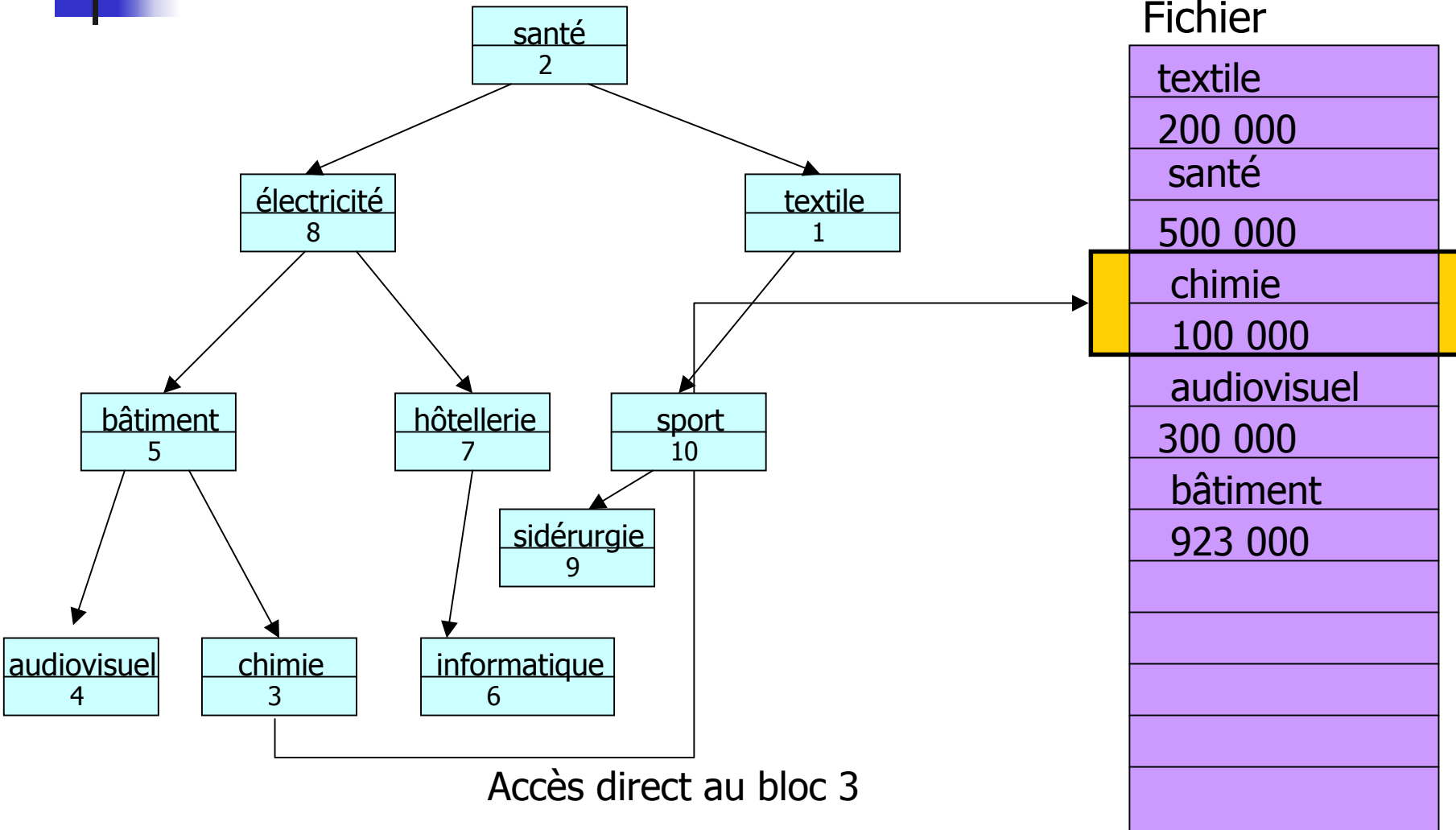
- Inventé chez Boeing : B pour Boeing ou Balanced
- L'idée est d'utiliser une clé pour retrouver rapidement un article
- Au lieu d'une table d'index, on utilise une structure plus adaptée aux recherches : un arbre
- Cette structure est très performante lorsqu'on recherche une valeur



6.4 Organisation du contenu

- Principe de B-Arbre :
 - Les articles dont la valeur de la clé est inférieure (respectivement supérieure) à l'article du nœud courant sont placés à sa gauche (respct. à sa droite)
 - Cette structure comme l'index ne contient pas les articles mais seulement la valeur de la clé et la position de l'article dans le fichier
 - Cette stratégie est efficace si l'arbre est bien organisé : s'il a autant de fils gauche que droits

6.4 Organisation du contenu





6.4 Organisation du contenu

- Cette technique est intéressante car la parcours de l'arbre revient à une recherche dichotomique
- Mais ce n'est vérifié que si l'arbre est équilibré (balanced)
- Chaque nœud doit avoir autant de descendants gauches que droit
- Le principe est de construire dès le départ un arbre équilibré puis de maintenir l'équilibre au fur et à mesure de sa croissance
- On doit donc couper des branches et les greffer ailleurs

6.4 Organisation du contenu

- Le B-ARBRE est lui même stocké sous la forme d'un fichier

clé Adresse fichier Fils gauche Fils droit

1	Santé	2	8	4
2	Hôtellerie	7	9	Nil
3	Sport	10	Nil	Nil
4	Textile	1	3	7
5
6	Chimie	3	Nil	Nil
7	Sidérurgie	9	Nil	Nil
8	Électricité	8	10	2
9	Informatique	6	Nil	Nil
10	Bâtiment	5	11	6
11	Audiovisuel	4	Nil	nil



6.4 Organisation du contenu

- **Organisation aléatoire : *hash-coding***
- Les techniques précédentes consistaient à voir différemment le fichier
 - Sans le modifier
 - En lui associant une structure complémentaire
- La technique de *hash-coding* consiste à réorganiser le fichier
- On réserve une zone contiguë pour ranger des enregistrements
- Cette zone est divisée en **p paquets** numérotés de **taille fixe t**



6.4 Organisation du contenu

- L'adresse réelle d'un paquet peut être obtenue :
 - à partir de son numéro
 - et de l'adresse de début de la zone
- Le principe de *hash-coding* est d'utiliser une fonction dite de **hachage** qui fait correspondre à chaque valeur de clé un numéro de paquet
- Les clés peuvent être de type variés et le résultat est impérativement un petit nombre entier



6.4 Organisation du contenu

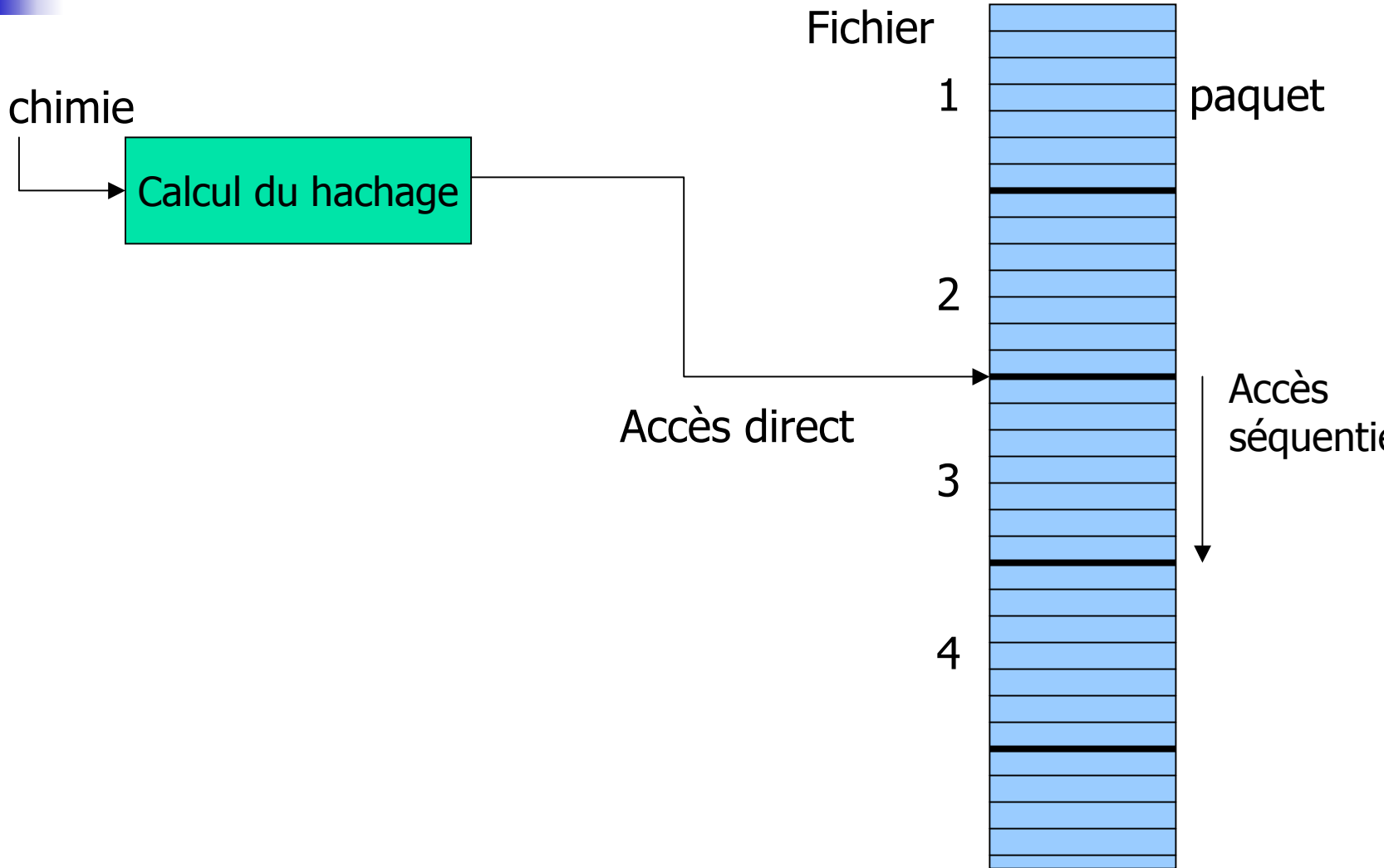
- **Exemple** : la fonction de hachage doit faire correspondre un nombre entier en 0 et 500 à l'attribut Domaine du cas précédent
- L'accès s'effectue en appliquant la fonction de hachage à la valeur de la clé présentée pour obtenir un numéro de paquet



6.4 Organisation du contenu

- **Au sein d'un paquet, la recherche est séquentielle**
 - Les articles sont rangés par ordre d'arrivée
- Pour définir une fonction de hachage, il faut d'abord transformer les valeurs de la clé en une valeur numérique (un nombre entier)
- On se sert souvent du codage interne de la valeur des caractères qui est toujours numérique
- On utilise abondamment la fonction modulo

6.4 Organisation du contenu





6.4 Organisation du contenu

- Lorsque le système fonctionne bien, les accès sont très rapides pour une recherche par le contenu
 - Il y a seulement un accès direct
 - Ensuite un petit déplacement (séquentiel)
- Le problème du hash-coding provient de ce que :
 - En fonction des données présentes, certains paquets peuvent rester vides (mémoire sous utilisée)
 - D'autres paquets peuvent être en état de débordement (mémoire saturée)
 - Il n'est pas possible de garantir à l'avance ce qui se produira



6.4 Organisation du contenu

- Pour pallier ces difficultés **en amont** on doit faire en sorte que la fonction de hachage :
 - Reste le plus possible représentative de la valeur de la clé. Il y a alors peu de valeurs de clé différentes pour un même numéro de paquet
 - **Exemple** : le nombre de caractères du mot est moins caractéristique que le code ASCII de la première lettre
 - Donne une répartition statistique uniforme des valeurs de clés sur les numéros de paquets



6.4 Organisation du contenu

- En aval en essaie de gérer la saturation des paquets en utilisant des techniques de redirection des articles surnuméraires :
 - Adressage ouvert
 - Chaînage
 - Rehachage
 - Dans tous les cas le débordement détériore les performances du système