

SYSTÈME DE GESTION DE FICHIERS

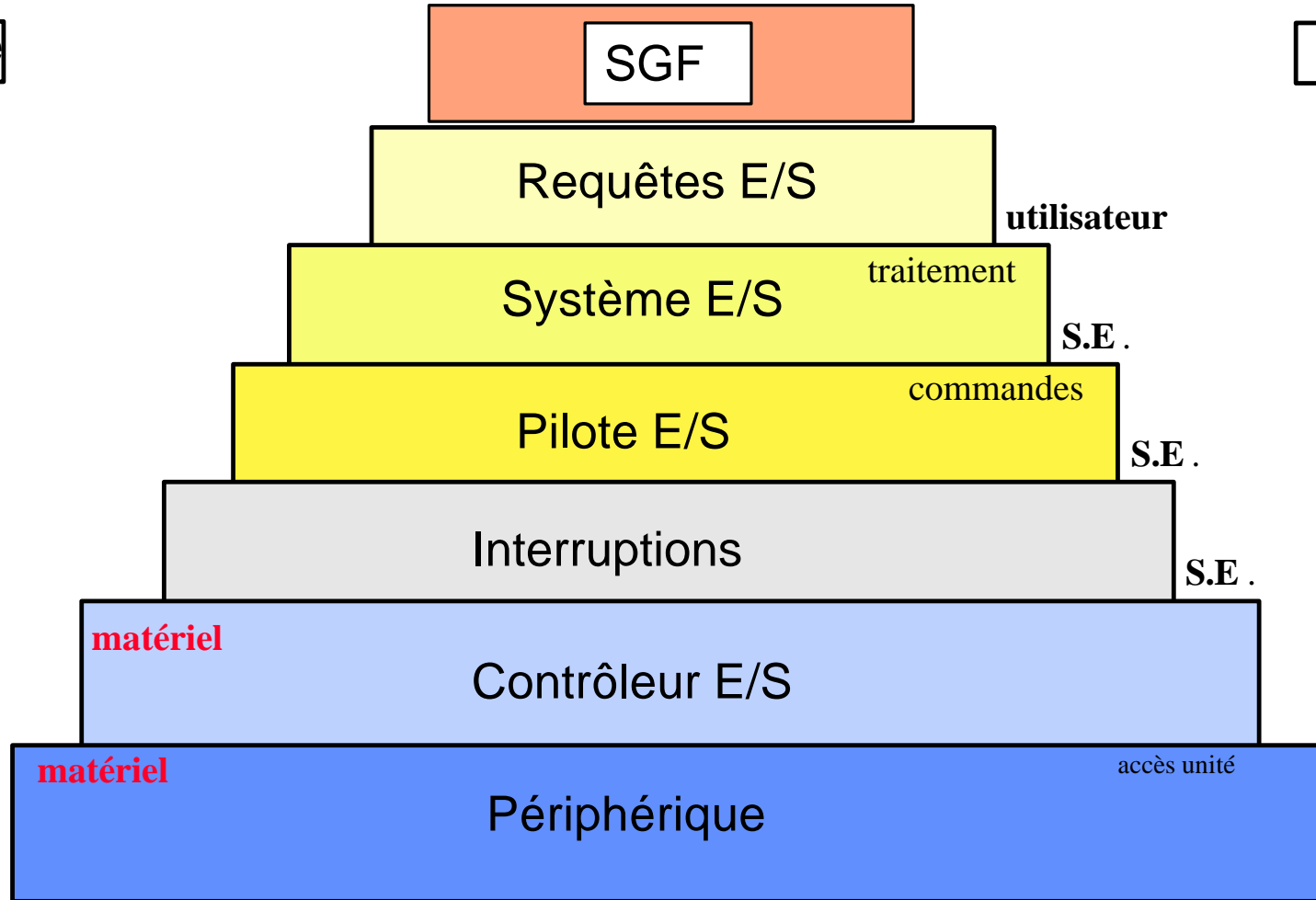
SGF - DISQUE

Les couches logicielles

réponse

SGF

requête



Introduction

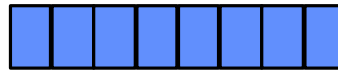
- **Partie visible du système d'exploitation**
 - interface
 - commandes
 - requêtes
 - fonctionnalités
 - fiabilité
- **Partie cachée**
 - gestion des données: fichiers
 - organisation
 - accès

Les concepts logiques

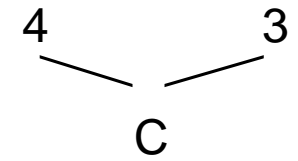
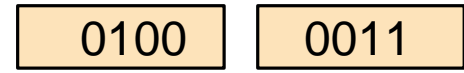
- **Les concepts:**
 - **Fichier**
 - **Enregistrement**
- **Les concepts logiques ont pour but:**
 - de ranger de manière rationnelle les données sur disque
 - de rendre transparents à l'utilisateur les concepts physiques
 - de gérer les fichiers indépendamment des périphériques
- **Les concepts logiques sont gérés par le SGF**
 - Le SGF s'appuie sur la gestion des concepts physiques

La construction du concept 'fichier'

octet



1 octet = 8 bits



1 caractère ascii

secteur

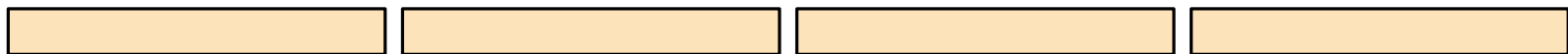


1 secteur = 512 octets



512 caractères

fichier



1 fichier

Les différents concepts 'fichier'

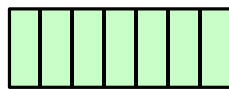
Selon le SGF, le fichier est :

- une simple suite d'octets
- une suite d'enregistrements
- une arborescence de blocs

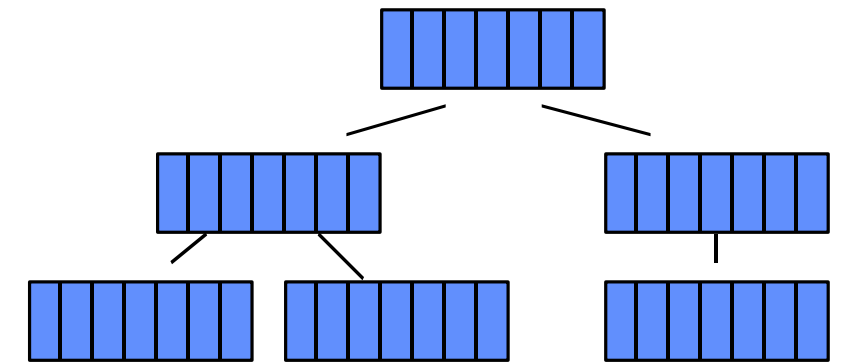
→ gestion des secteurs
→ gestion des enregistrements
→ gestion des blocs



fichier



fichier



fichier

Les fichiers sur disque

regroupement des informations par entité



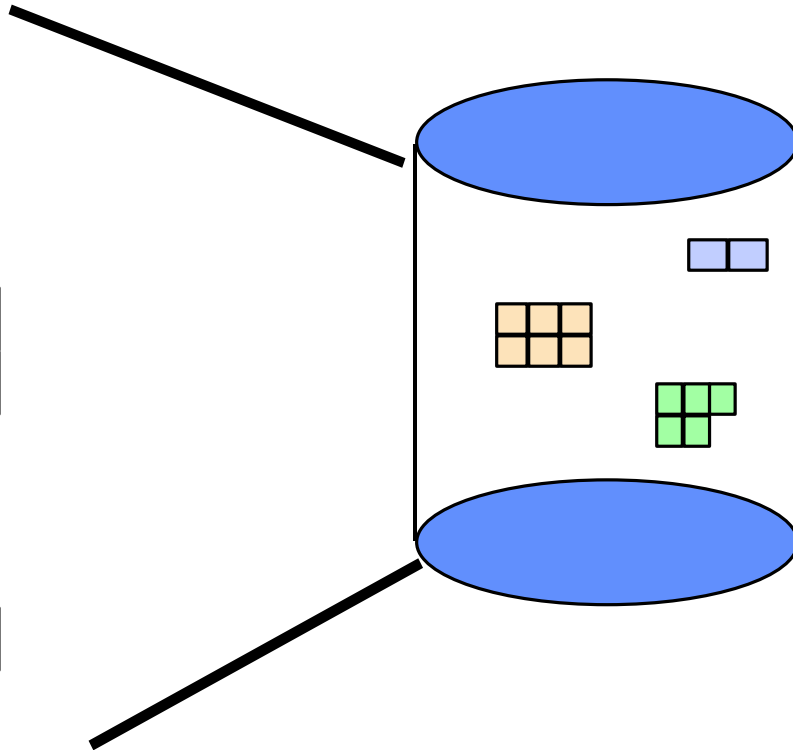
informations de l'entité 1



informations de l'entité 2



informations de l'entité 3



La nomination des fichiers

Les fichiers sont repérés sur le disque par un **nom**:

FICHER.C	programme source en C
FICHER.PAS	programme source en pascal
FICHER.BIN	programme binaire exécutable
FICHER.LIB	fichier librairie
FICHER.TXT	fichier texte
FICHER.DAT	fichier données

Les noms de fichiers respectent la syntaxe imposée par le SGF:

- longueur maxi d'un nom
- jeu de caractères autorisé pour le nom
- extension différenciant la nature du fichier....

Le concept 'enregistrement'

L'enregistrement:

Ensemble de données décrivant un objet d'une même entité

1 ligne de programme source

1 ligne de texte

1 instruction codée binaire

L'accès aux données d'un fichier se fait en fournissant au SGF:

- le nom du fichier
- le n° d'enregistrement dans ce fichier

Ce concept d'enregistrement n'existe pas toujours:

- accès à partir d'un pointeur dans le fichier

L'enregistrement structuré

Plusieurs types d'enregistrements sont possibles selon le SGF:

- enregistrement = bloc = données
- enregistrement = bloc = zones de données

Zone:

Chaque zone est une représentation d'un objet de l'entité

Entité: article

n°	nom_article	référence_article
----	-------------	-------------------

clé

Zone nom_article

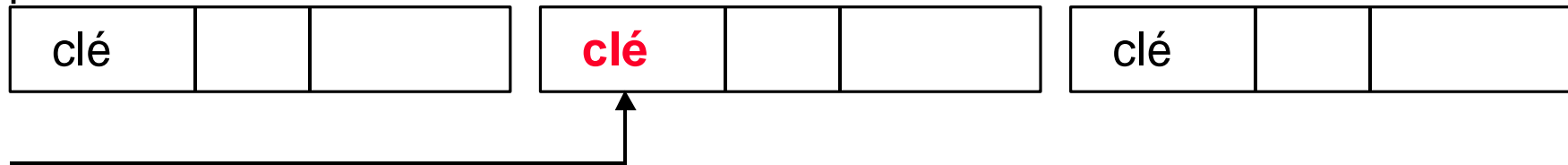
Clé:

Donnée identifiant de manière unique chaque entité

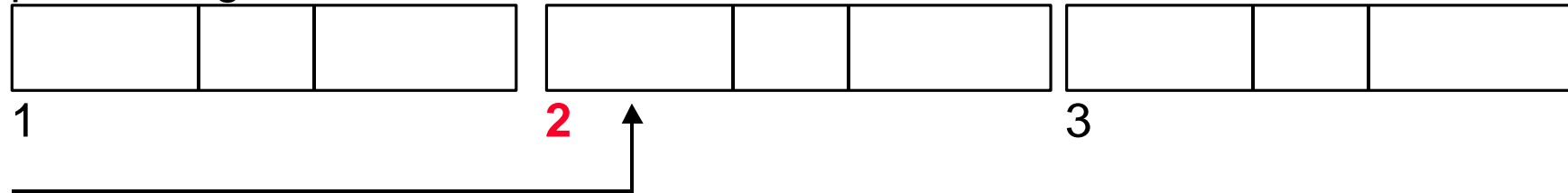
L'enregistrement structuré

Plusieurs moyens pour repérer l'emplacement physique d'un enregistrement:

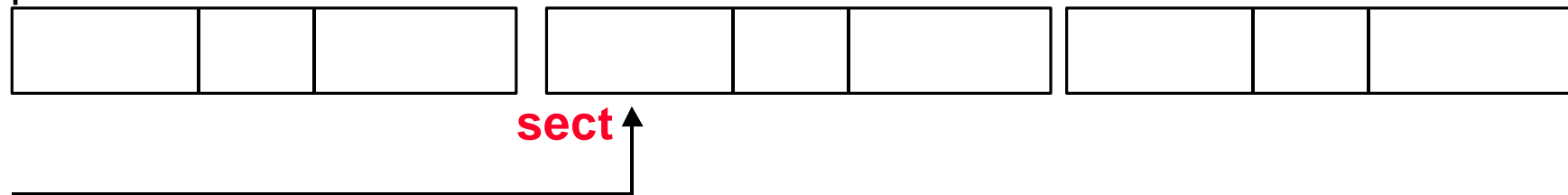
par la clé



par le rang



par l'adresse



La longueur de l'enregistrement

La longueur fixe:

- tous les enregistrements d'un même fichier sont de même longueur



Facilité de gestion

Risque de perte de place importante

La longueur variable:

- chaque enregistrement a sa propre longueur



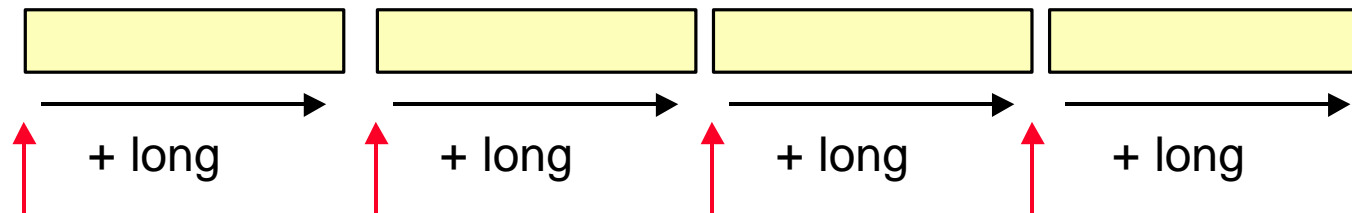
Complexité de gestion

Occupation optimisée de l'espace

La gestion des enregistrements

Accès aux enregistrements de longueur fixe:

- connaître l'emplacement du premier enregistrement
- connaître la longueur



Accès aux enregistrements de longueur variable:

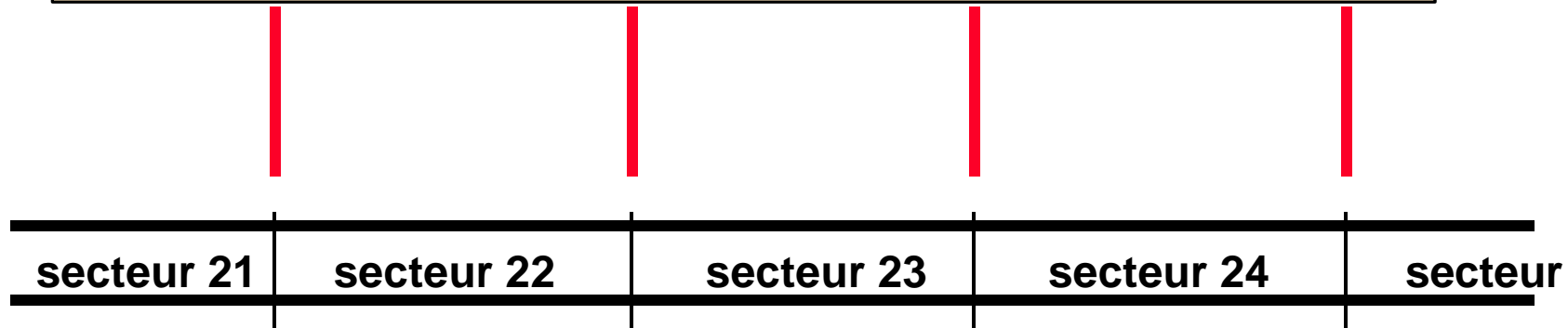
- connaître l'emplacement du premier enregistrement
- connaître la longueur de chaque enregistrement



Les concepts logiques - les concepts physiques

enregistrement

ceci est un enregistrement d' une ligne de texte saisie au clavier



ceci est un enregistrement d' une ligne de tex te saisie au clavier

suite de secteurs

Le calcul de l'adresse physique de chaque enregistrement

Connaître la position du **1er enregistrement** du fichier:

- soit l'adresse logique: n° secteur logique - n° bloc
- soit l'adresse physique: n° face, n° piste, n° secteur

Connaître le **déplacement** pour chaque enregistrement:

- soit le nombre d'octets
- soit un chaînage sur l'enregistrement suivant

chaînage:

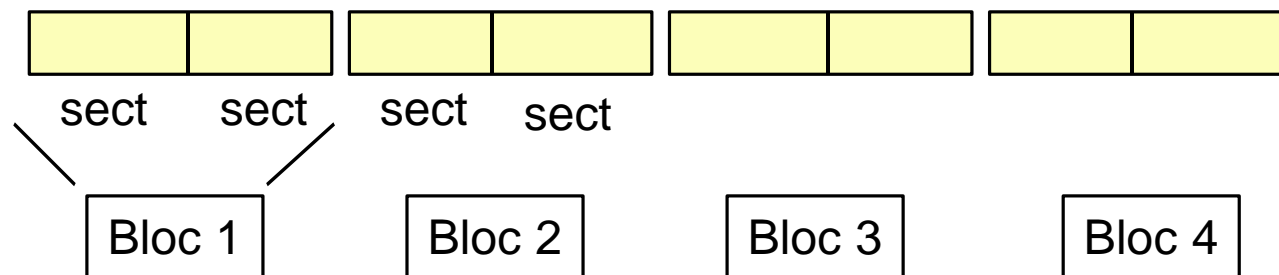
- adresse de l'enregistrement suivant

Le bloc physique

Pour un meilleur compromis: temps d'accès et capacité de stockage
certains SGF organisent l'espace du disque en blocs de secteurs

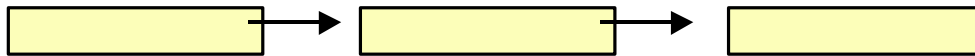
ex: **1 bloc = 2 secteurs de 512 octets soit 1KO**

Les opérations de lecture et d'écriture du SGF se font bloc par bloc



Gestion des blocs libres

chaînage des blocs entre eux => adresse du bloc suivant



liste chaînée des n° des blocs libres

14	56	[]
2016	566	
258	254	
311	81	
1965	388	
784	3677	
69	550	

Arrows point from the bottom of the first and second columns to the right.

table de bits des blocs libres

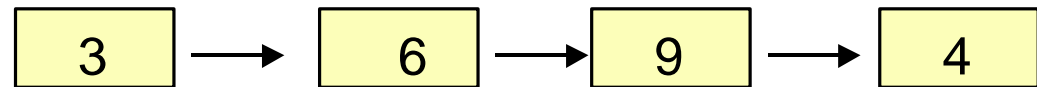
1001111000100001111
0001101010001101100
0110100101111111000
0001111100000111111
0000001111111011111
1010111000111011001

Exemple de MS-DOS

La **FAT** File Allocation Table
Table d'allocation des fichiers

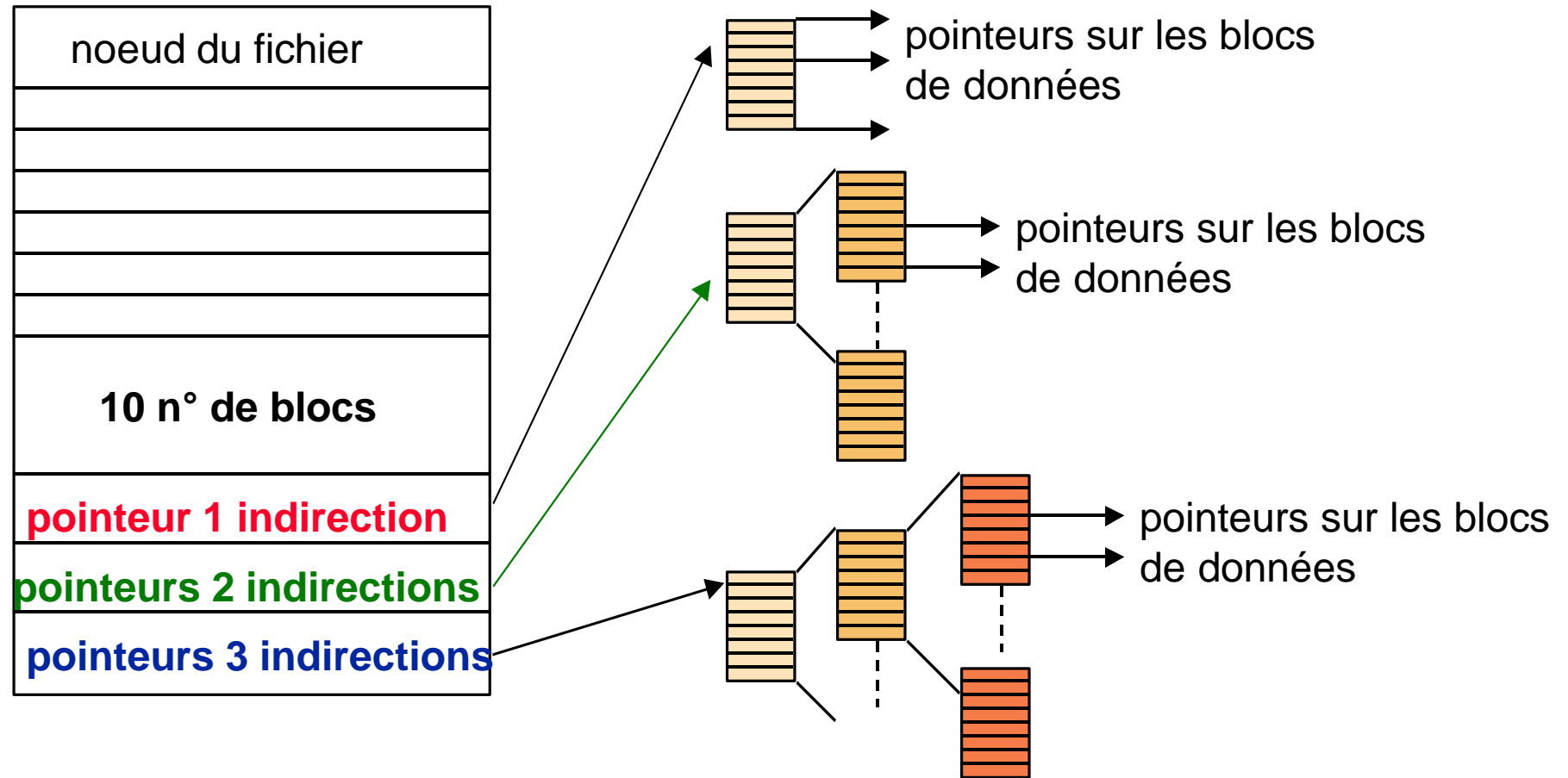
0	x
1	x
2	EOF
3	6
4	EOF
5	
6	9
7	mauvais
8	
9	4

fichier_A:



Exemple d'Unix

Une **table i-node** par fichier sur le disque



Le blocage d'enregistrements

1 bloc logique = n enregistrements

- Lorsque plusieurs enregistrements sont transférés ensemble lors d'un échange, on dit qu'ils sont bloqués.
- Le **facteur de blocage** est le nombre n d'enregistrements par bloc logique

Enregistrement n

Enregistrement n+1

Enregistrement n+2

v

BLOC logique

Le facteur de blocage est calculé de manière à occuper au mieux les blocs physiques

Le fichier

- **LE FICHER**

- Un fichier est le regroupement de tous les enregistrements décrivant le même type d'entité

FICHER:

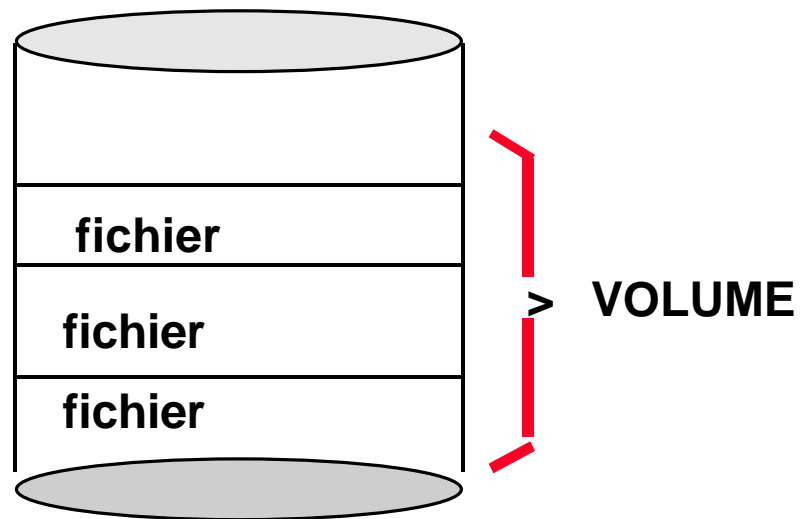


Avantage:

Rapidité de sélection des enregistrements concernés par un traitement

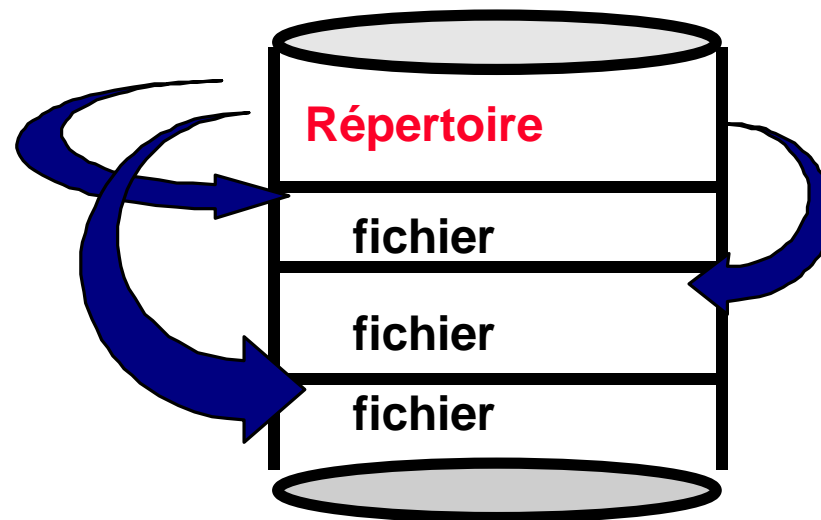
- **LE VOLUME**

- le volume est l'ensemble des fichiers se trouvant sur un même support



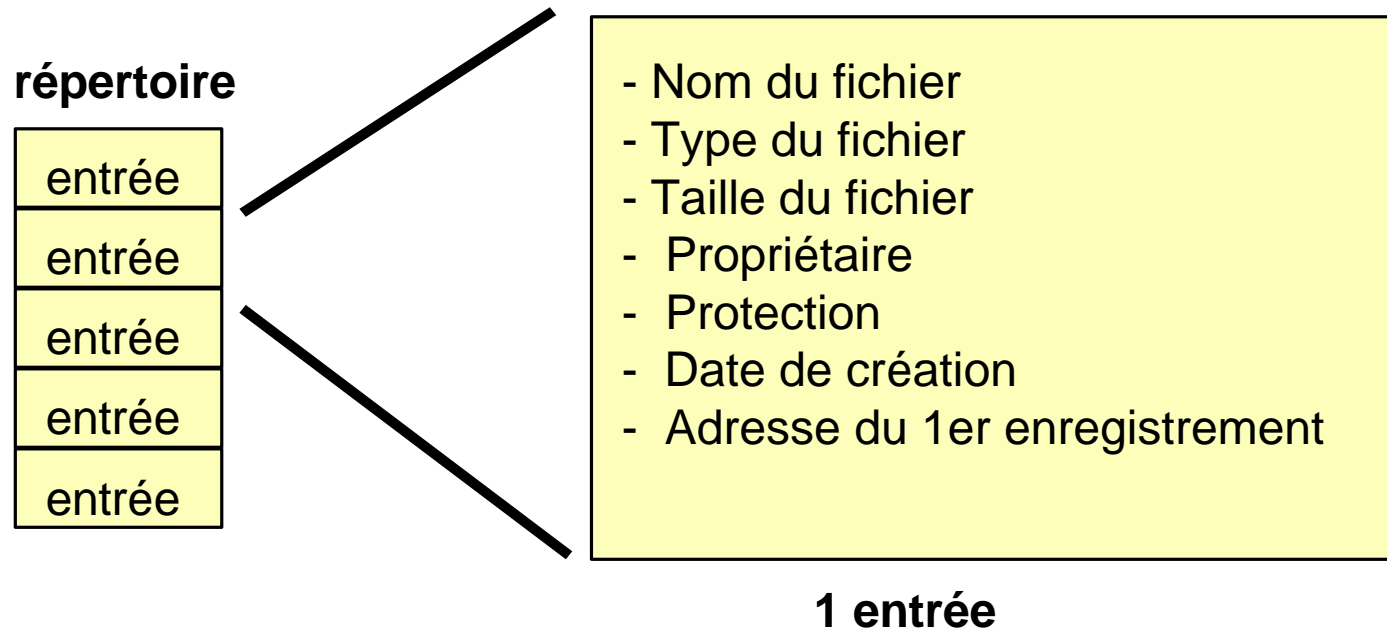
Le répertoire

- Le répertoire est une table sur le support permettant de référencer tous les fichiers existants du volume avec leur nom et leurs caractéristiques principales
- Le répertoire stocke pour chaque fichier l'adresse du premier enregistrement

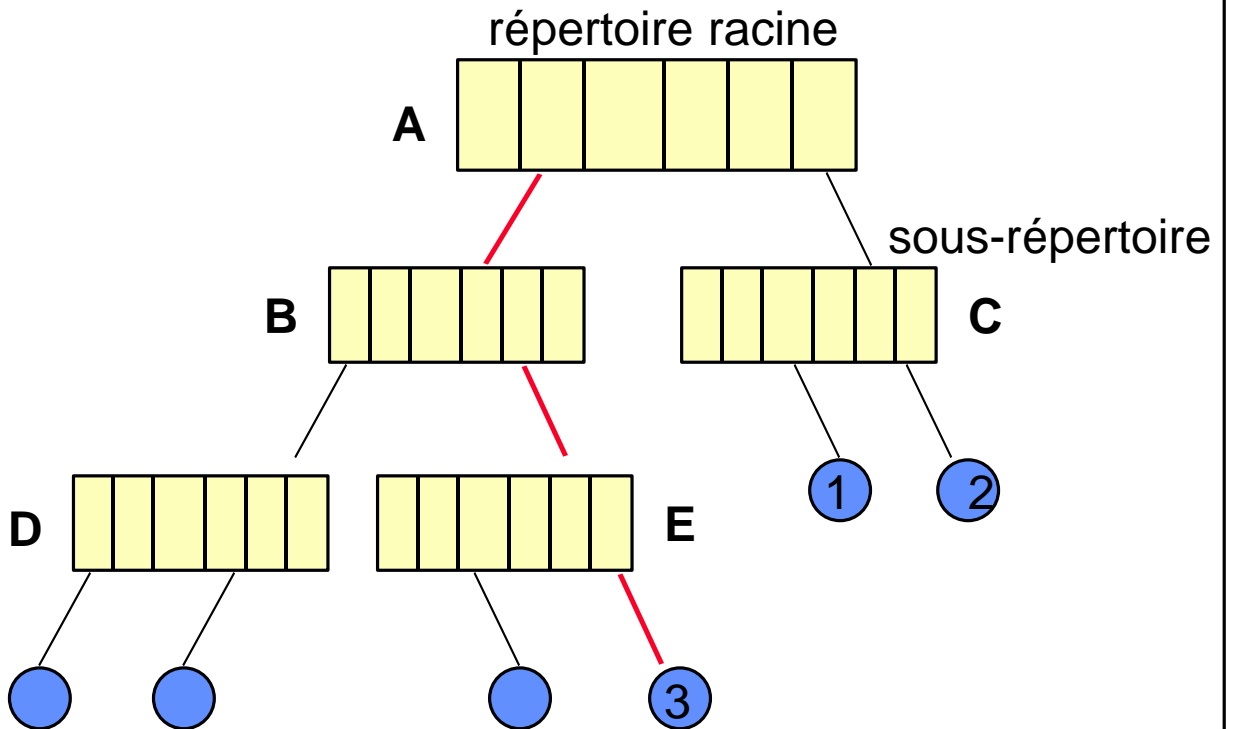
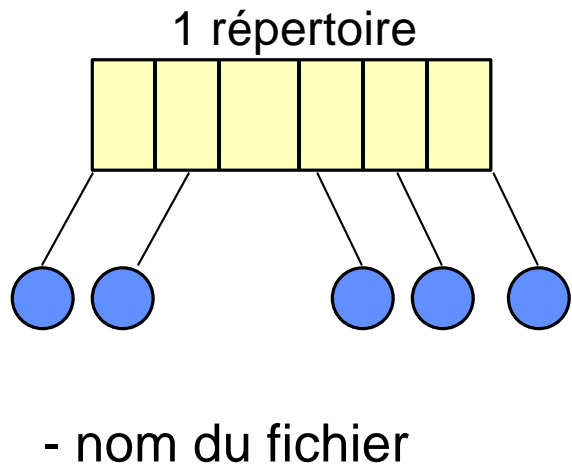


Composition du répertoire

Un répertoire est une zone disque réservée par le SGF.
Le répertoire comprend un certain nombre d'entrées.
Une entrée est allouée à chaque fichier du disque

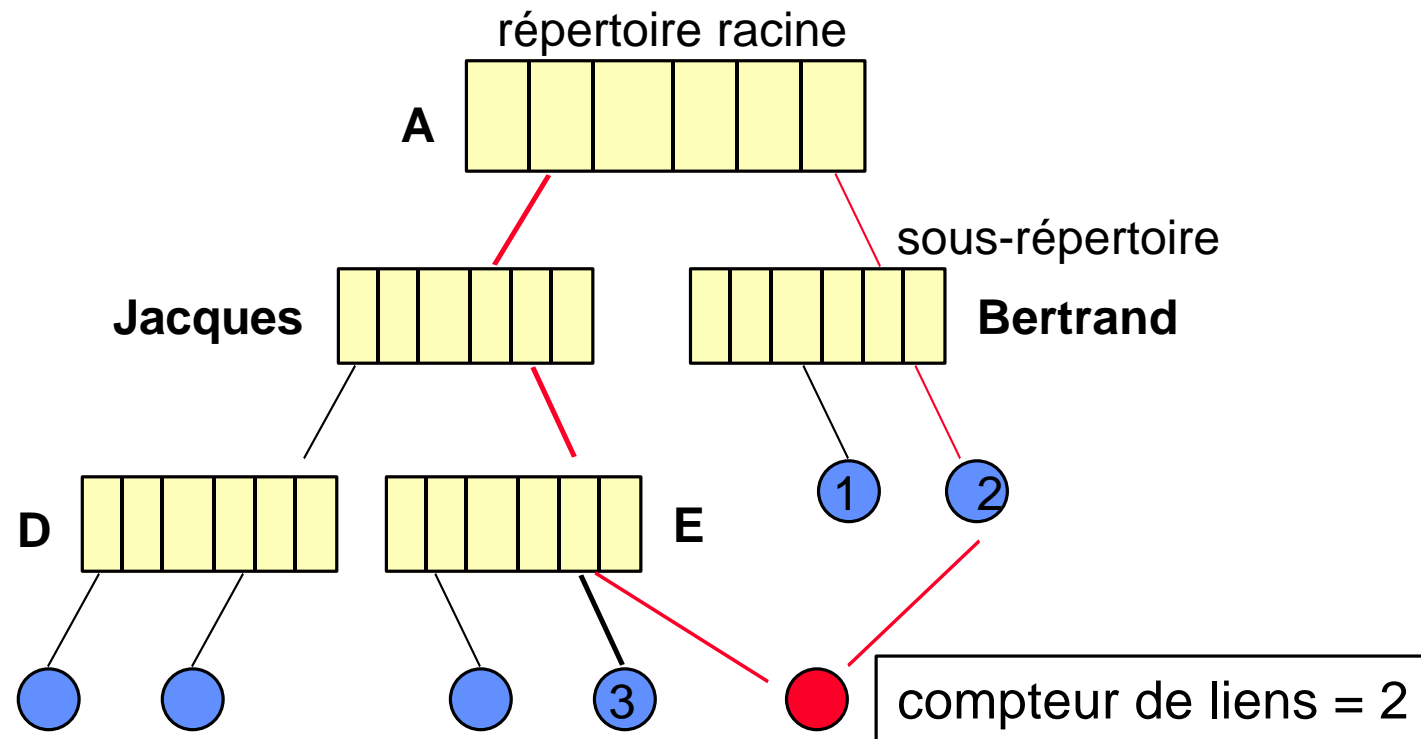


Organisation des répertoires



- chemin d'accès **A/B/E/**
- nom du fichier **FIC_3**

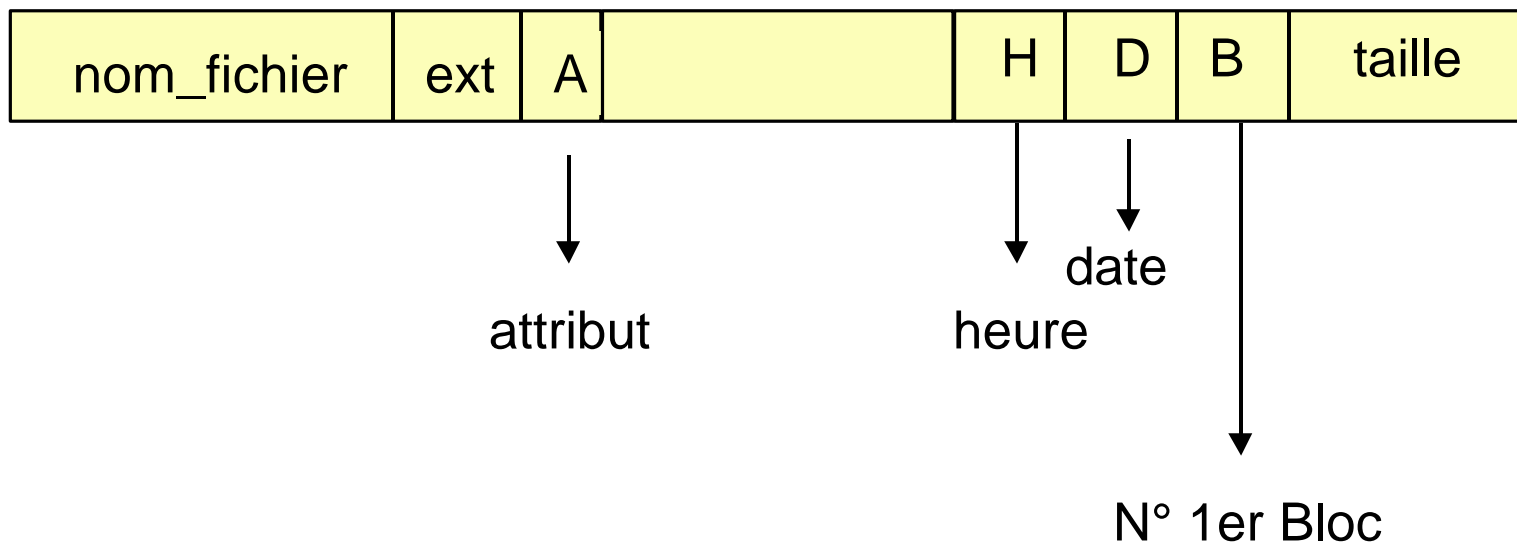
Les fichiers partagés



- Propriété des fichiers par sous-répertoire
- Partage d'un même fichier par deux usagers

Exemple de MS_DOS

Chaque fichier occupe 32 octets dans le répertoire



Un sous-répertoire occupe une entrée comme un fichier
le répertoire racine a 112 entrées
les autres sous-répertoires n'ont pas d'entrées limitées

Exemple d'Unix

répertoire racine

1	▪
1	▪▪
4	bin
7	dev
5	lib
3	etc
9	usr
8	tmp

↓ ss-répertoire

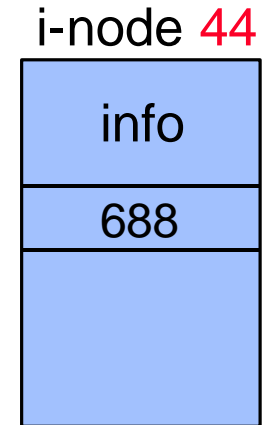
n° i-node



/usr

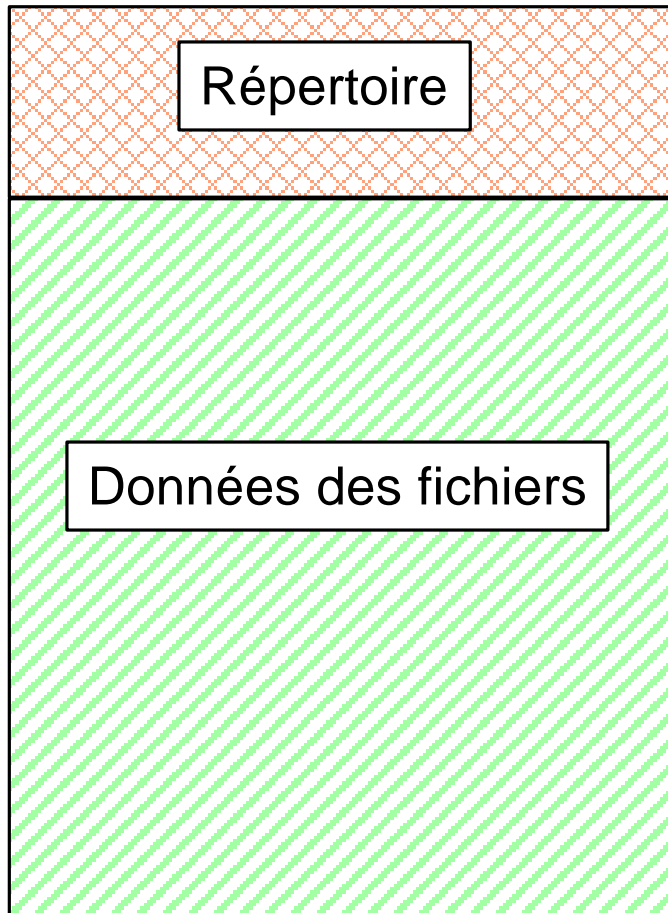
bloc 125

9	▪
1	▪▪
17	claudé
57	luc
44	prog



/usr/prog

Allocation de l'espace du disque



Répertoire

allocation **statique** ou dynamique

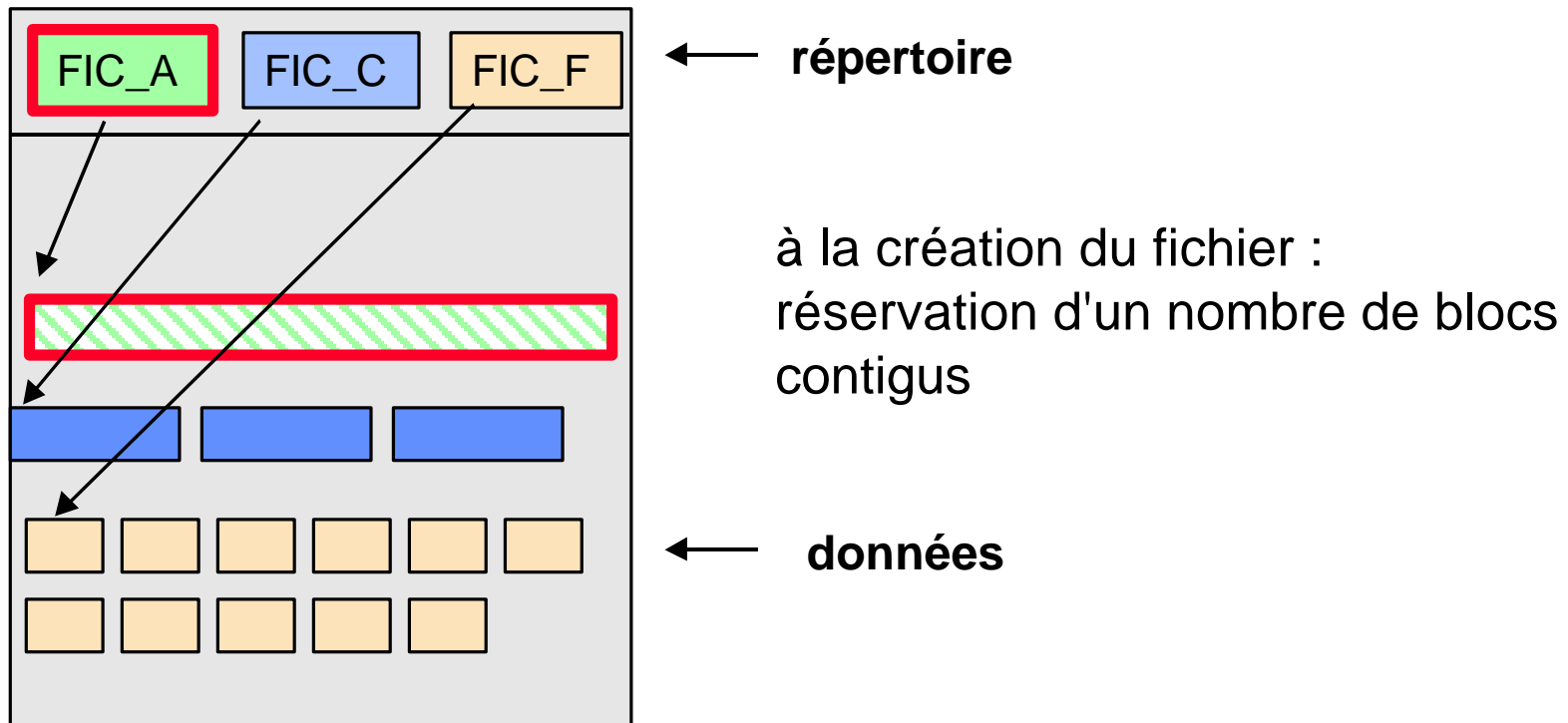
Données des fichiers

allocation **statique** ou **dynamique**

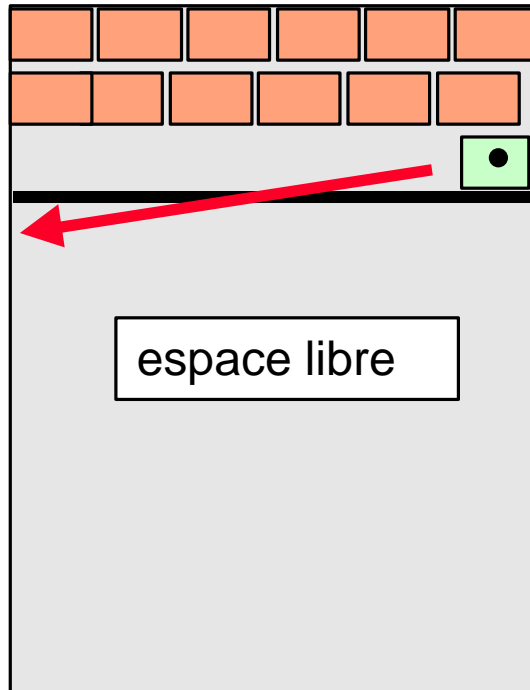
Allocation de l'espace d'un fichier

allocation statique de l'espace du fichier

Les enregistrements d'un même fichier sont contigus



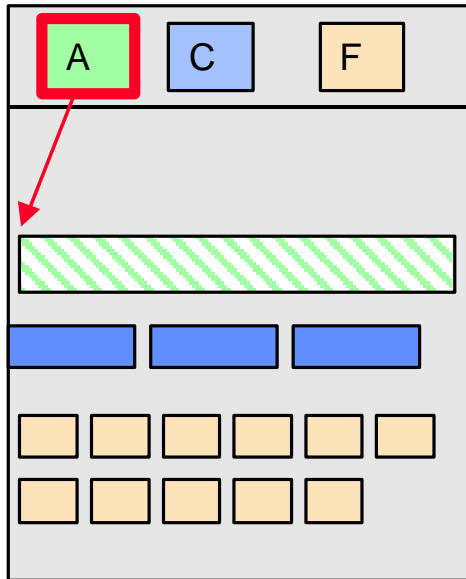
Allocation statique



Phase d'**initialisation** SGF:

- réservation de l'emplacement du **répertoire**
les entrées sont initialisées comme étant libres
- pointeur vers le 1er emplacement libre

Allocation statique

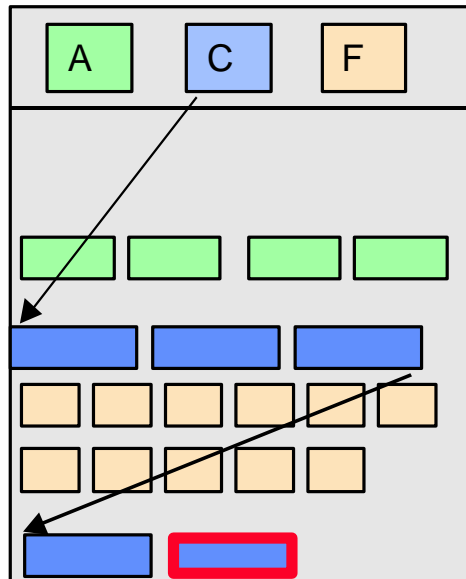


A la création du fichier:

- calcul de l'emplacement nécessaire:
 $\text{nb d'enregistrements prévus} * \text{long d'un enregistrement}$
- recherche d'un emplacement libre correspondant
- si oui création d'une entrée dans le répertoire avec
maj du pointeur 1er emplacement

Cet emplacement ne fait plus partie de l'espace libre
- maj du pointeur vers le 1er emplacement libre

Allocation statique



débordement

A la **création d'un enregistrement**:

- place disponible dans l'espace réservé du fichier?

si **oui**

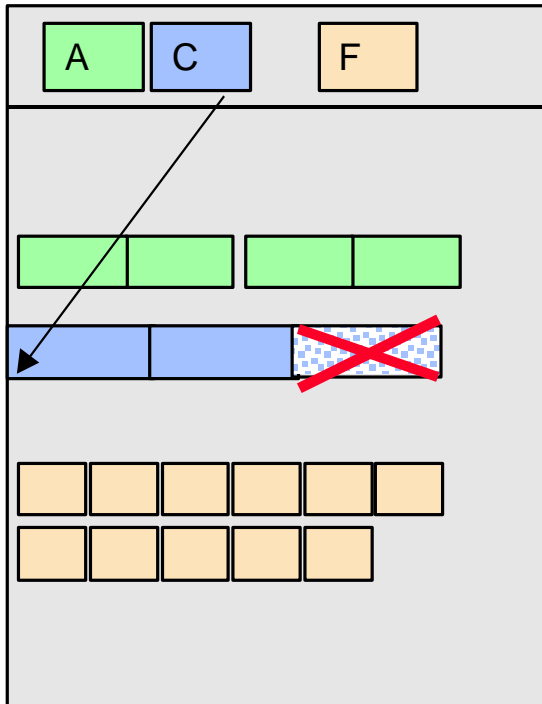
- occupation d'un emplacement dans l'espace réservé

si **non**

- soit allocation d'un emplacement supplémentaire (débordement)

- soit déplacement dans un espace plus grand (réorganisation)

Allocation statique



A la **suppression d'un enregistrement**:

- aucune récupération d'espace

mais:

- possibilité de récupération de l'emplacement de l'enregistrement supprimé: marquage

Compteur:

un compteur d'enregistrements est mis à jour dans le répertoire

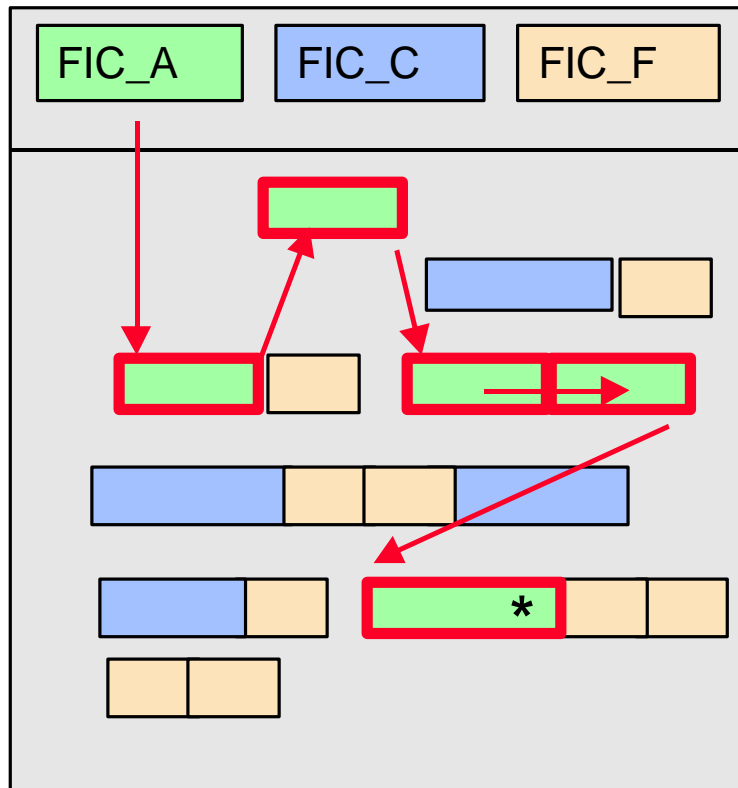
- si 0 aucun enregistrement

- si $\neq 0$ n enregistrements

Allocation de l'espace d'un fichier

allocation dynamique de l'espace du fichier

Les enregistrements d'un même fichier sont chaînés entre eux

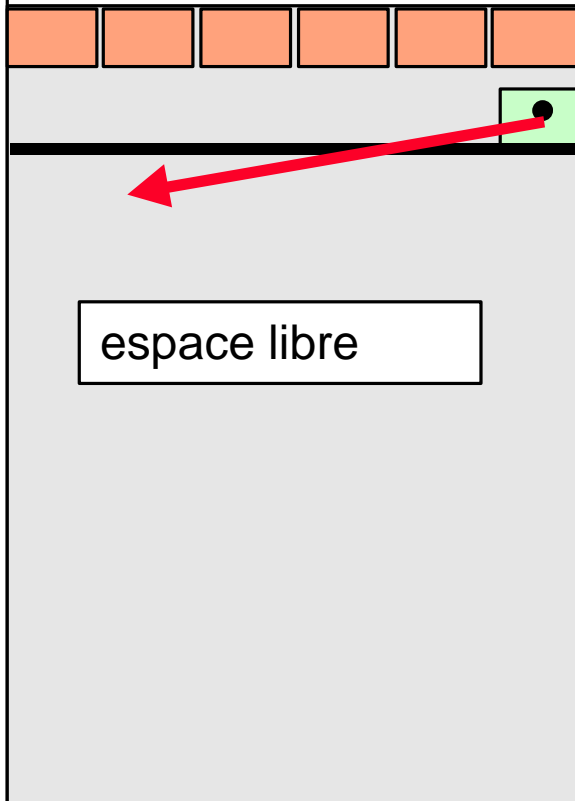


← répertoire

occupation de l'espace au fur et à mesure des créations d'enregistrements

← données

Allocation dynamique

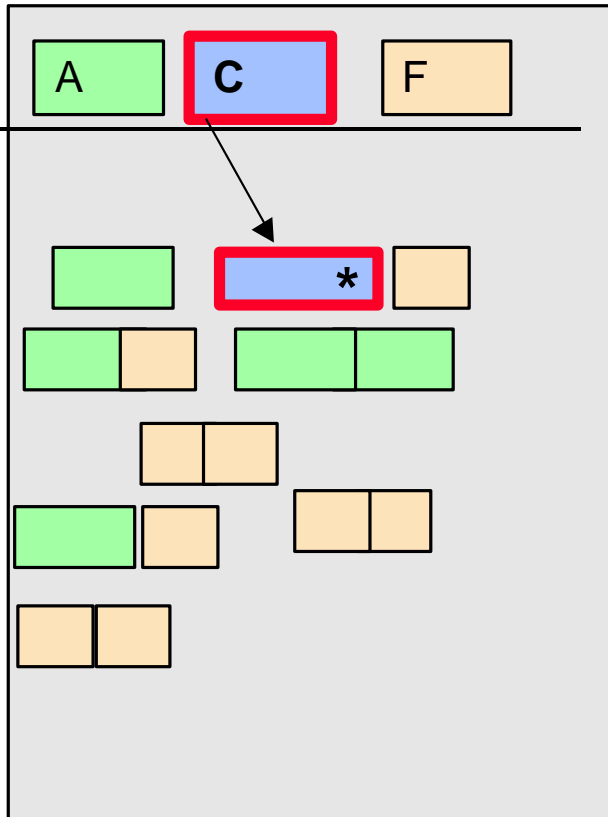


Phase d'**initialisation** SGF:

- réservation de l'emplacement du **répertoire**
les entrées sont initialisées comme étant libres

- pointeur vers le 1er emplacement libre

Allocation dynamique



création du fichier:

- à la création du 1er enregistrement
- pas de réservation préalable d'un emplacement

A la création d'un enregistrement:

- place disponible dans l'espace libre?

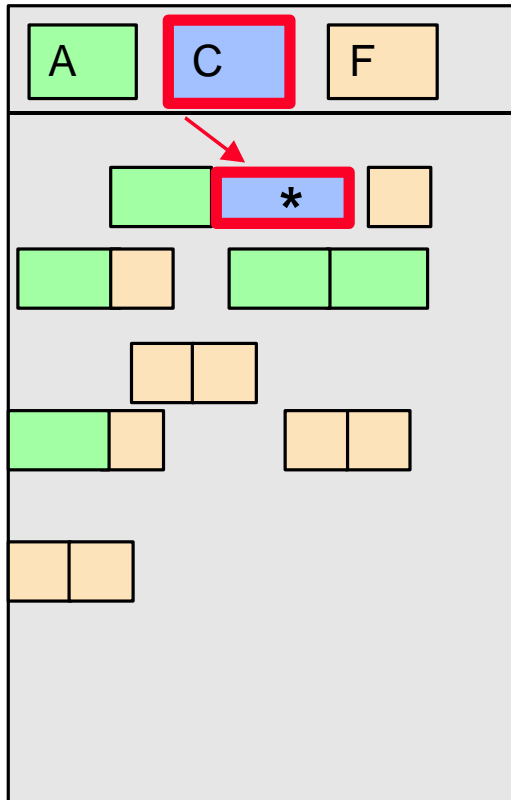
si oui

- occupation d'un emplacement dans l'espace libre
- maj du pointeur vers 1er emplacement libre

si non

- plus de place libre

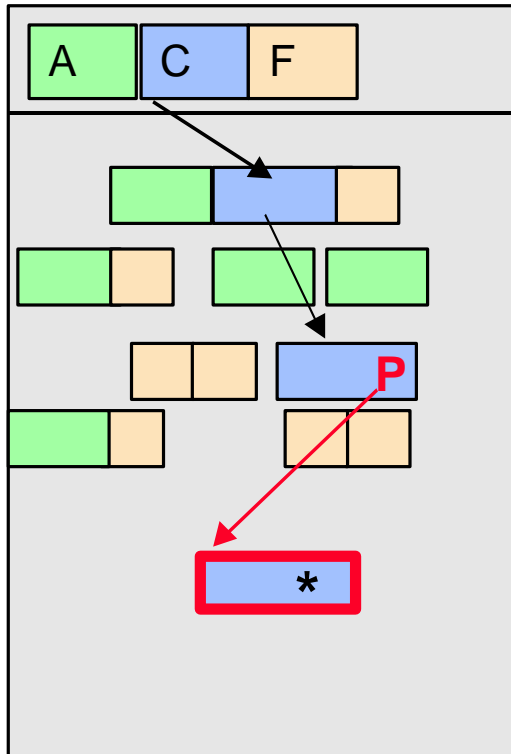
Allocation dynamique



Création d'un enregistrement:

- utilisation du premier enregistrement libre
- . maj pointeur 1er libre
- si **1er enregistrement** du fichier
- . création entrée dans répertoire
- . maj pointeur du 1er enregistrement dans répertoire
- . maj pointeur dernier enregistrement dans répertoire
- . chaînage de l'enregistrement: * (dernier)

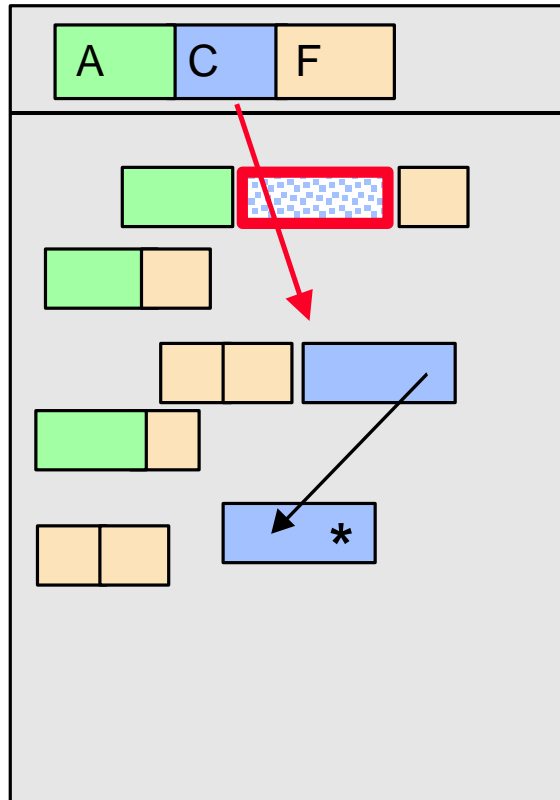
Allocation dynamique



Création d'un enregistrement:

- si **nième enregistrement** du fichier
- . maj pointeur dernier enregistrement dans répertoire
- . chaînage dernier enregistrement vers nouvel enregistrement
- . chaînage de l'enregistrement: * (dernier)

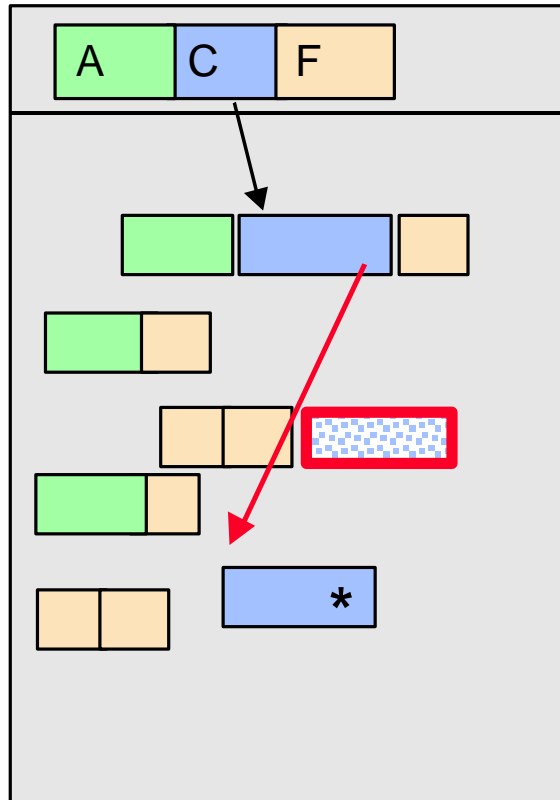
Allocation dynamique



Suppression d'un enregistrement:

- si **1er enregistrement** du fichier
- . maj pointeur du 1er enregistrement dans répertoire
- chaînage enregistrement libéré avec le dernier libre

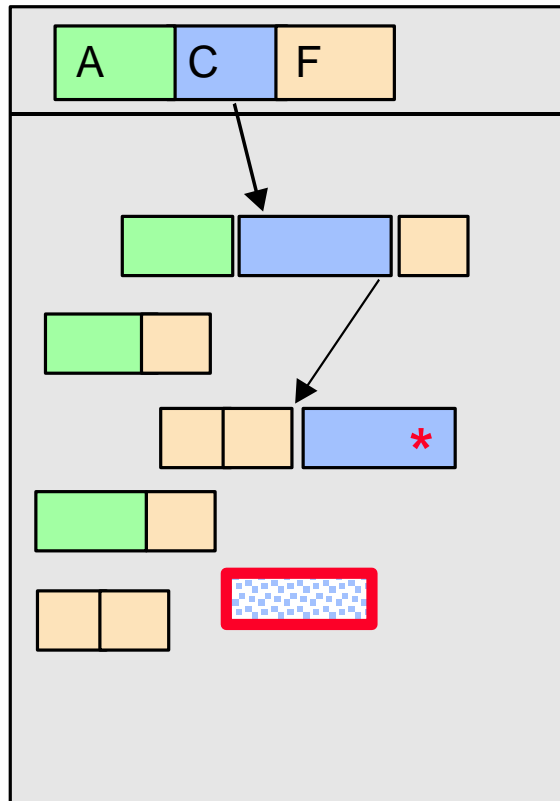
Allocation dynamique



Suppression d'un enregistrement:

- si **nième enregistrement** du fichier
- chaînage enregistrement précédent vers enregistrement suivant
- chaînage enregistrement libéré avec le dernier libre

Allocation dynamique



Suppression d'un enregistrement:

- si **dernier enregistrement** du fichier
 - . maj pointeur dernier enregistrement dans répertoire
 - . chaînage avant-dernier enregistrement : * (dernier)
-
- chaînage enregistrement libéré avec le dernier libre

Les organisations de fichiers

Selon l'allocation de l'espace disque qu'utilise le SGF plusieurs organisations de fichiers sont possibles

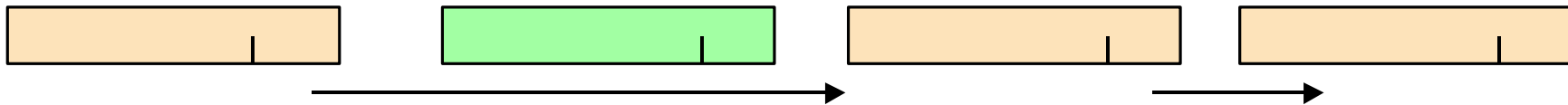
- **séquentielle**

allocation statique



- **séquentielle chaînée**

allocation dynamique



Les organisations de fichiers

Organisation **séquentielle indexée**:

- table d'index
- fichier des données

Principe:

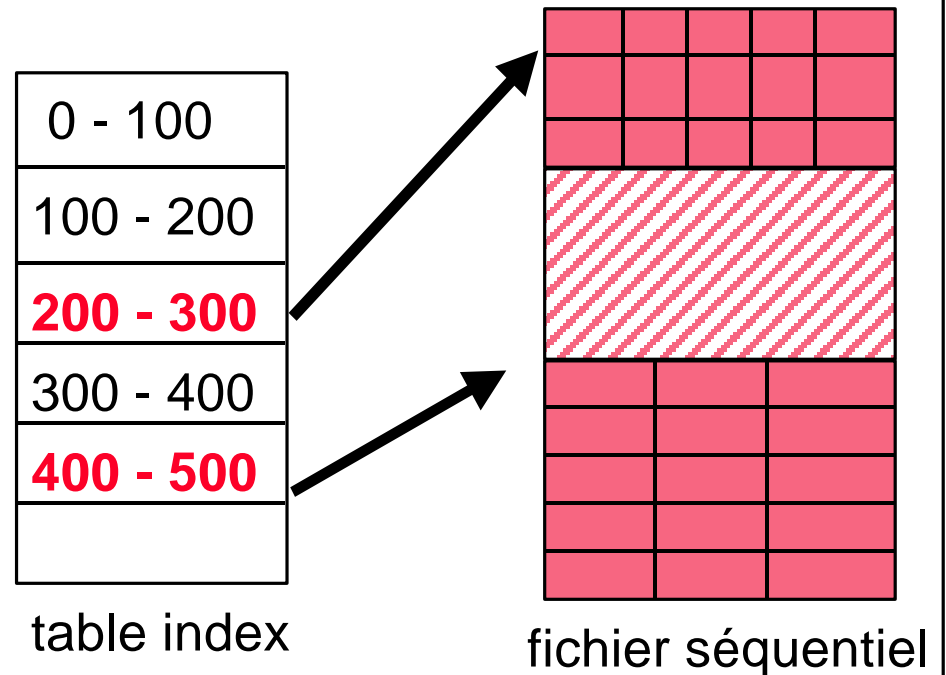
- recherche d'un enregistrement dans une partie du fichier

avantages:

- rapidité d'accès

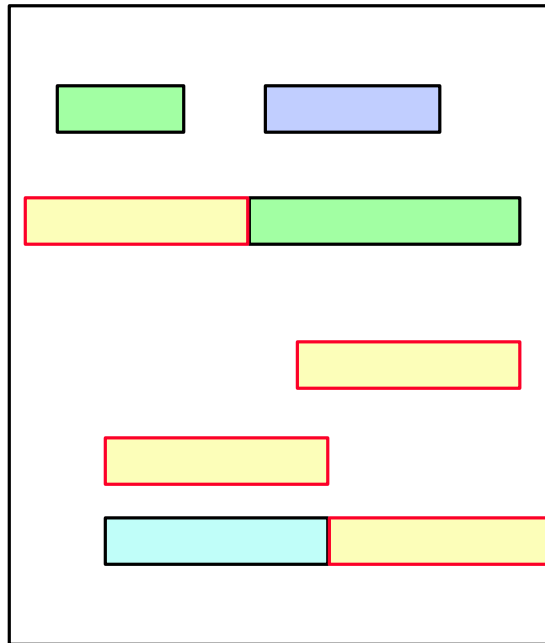
inconvénients:

- double accès
- stockage des index



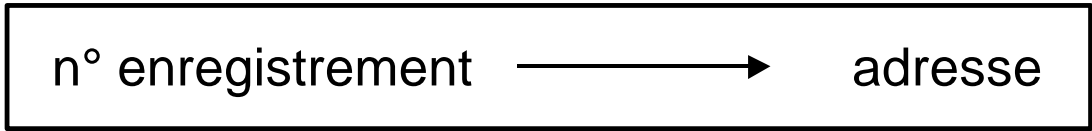
Les organisations de fichiers

Organisation **aléatoire**:



Aucun lien ne relie les enregistrements entre eux

Les emplacements des enregistrements sont calculés: H-coding



Les accès

Comment accéder aux emplacements des enregistrements?

Quelques méthodes d'accès:

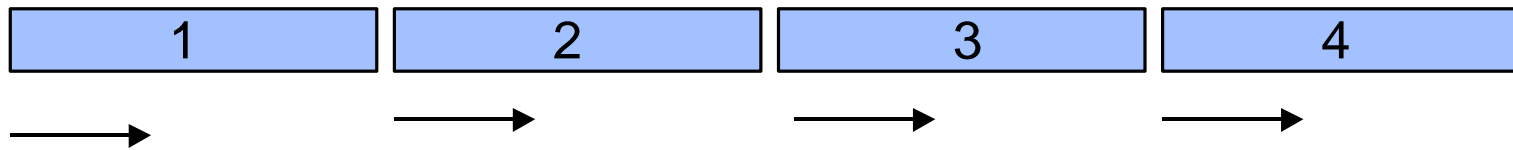
- **séquentielle** si organisation séquentielle
- **séquentielle-indexée** si organisation séquentielle-indexée
- **séquentielle chaînée** si organisation séquentielle-indexée
- **calculé** si organisation aléatoire et H-Coding
- **relatif** si organisation séquentielle et format fixe
- **direct** si adresse physique connue
- **indirect** si 1er accès donne adresse du 2scd

Un SGF gère quelques méthodes
ces méthodes sont parfois particulières au SGF
Elles dépendent des organisations

Accès séquentiel

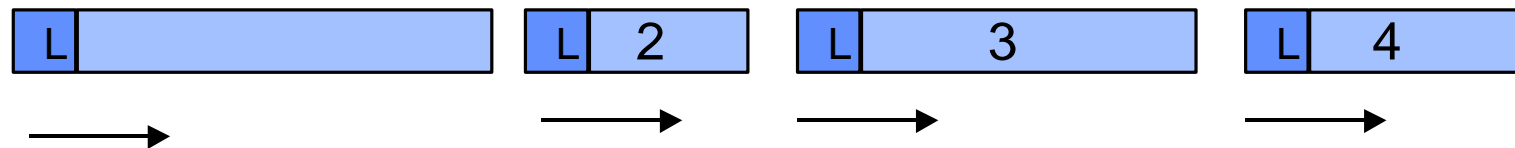
- Accès au premier enregistrement: adresse dans le répertoire

- **organisation séquentielle**



si longueur **fixe**:

adresse suivant = adresse + longueur fixe

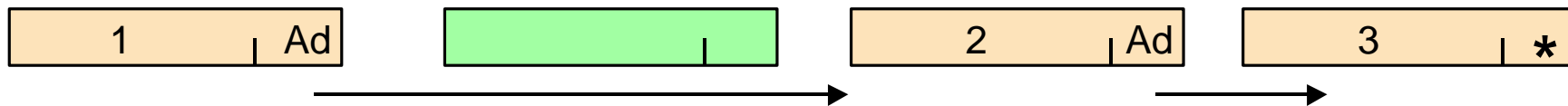


si longueur **variable**:

adresse suivant = adresse + longueur variable pour chaque enregistrement

Accès séquentiel chaîné

- séquentielle chaînée



longueur fixe ou variable

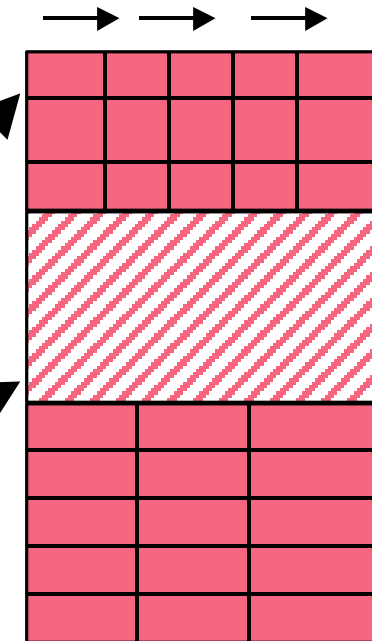
Accès séquentiel-indexé

- Accès au premier élément de la table d'index: répertoire
- la table d'index est transférée en mémoire

- recherche plage dans table
- accès à la première adresse
- lecture séquentielle jusqu'à l'enregistrement recherché

100	Ad
200	Ad
300	Ad
400	Ad
500	Ad

table index



fichier séquentiel

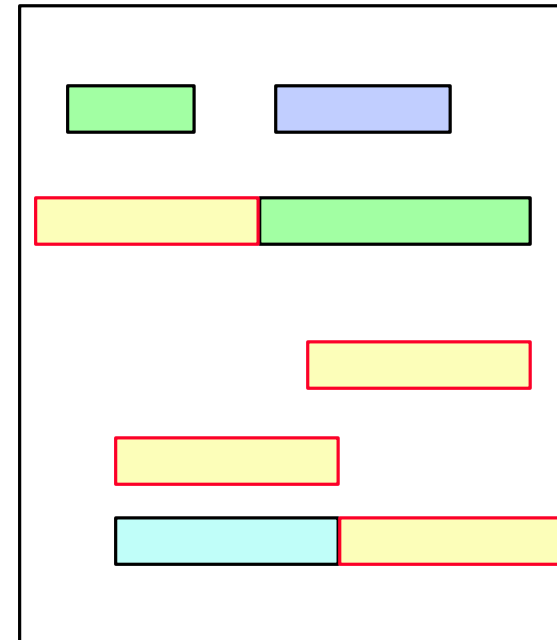
Accès calculé

Le calcul de l'emplacement de l'enregistrement se fait par une fonction connue du SGF

- fournir la clé de l'enregistrement
- le SGF applique une fonction (H_Coding)
- le résultat donne l'adresse

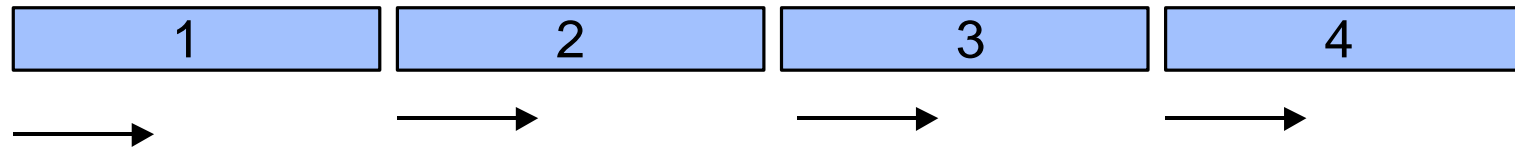
ex:

clé:011203 -> face 01
-> piste 12
-> secteur 03



Accès relatif

- organisation séquentielle

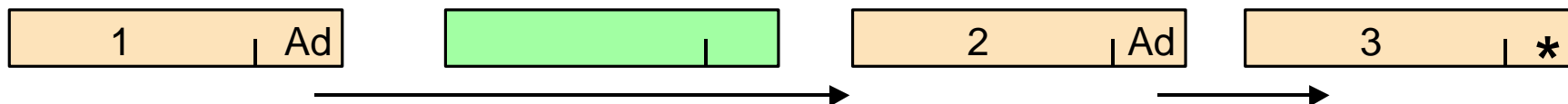


si longueur **fixe**:

si N: nième enregistrement

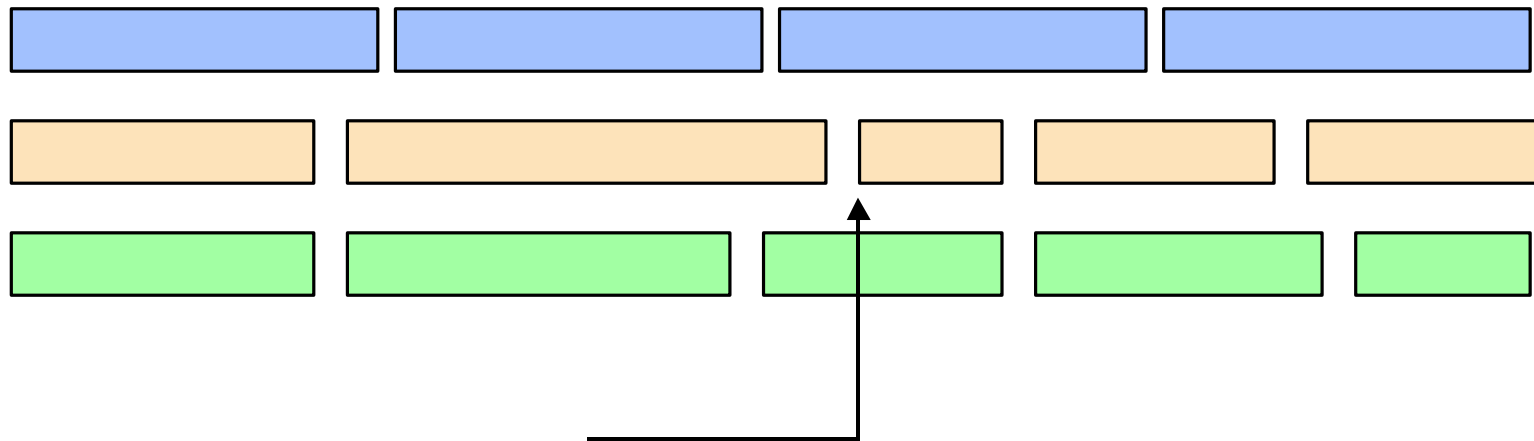
adresse = adresse début + (N-1) * longueur fixe

- séquentielle chaînée



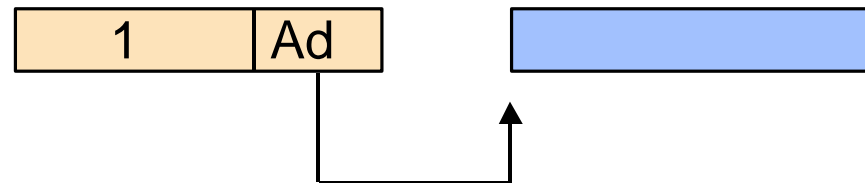
lecture des enregistrements jusqu'à celui recherché

Accès direct



adresse physique du début de l'enregistrement

Accès indirect



- à la lecture d'un premier enregistrement le SGF récupère l'adresse de l'enregistrement à lire
- l'emplacement de l'Adresse doit être connu du SGF

Les commandes et les requêtes

les commandes:

- interface SGF utilisateurs
- langage de commande ex: shell d'unix, commandes MS-DOS
- niveau fichiers et non contenu des fichiers

>> DIR	liste du répertoire courant MS-DOS
\$ ls -la	liste du répertoire courant Unix

les requêtes:

- interface SGF logiciel
- appel dans un programme aux fonctions du SGF (langage C, Cobol...)
- niveau fichier et enregistrements

read (fic, tampon, nb_oct)

Les commandes

Les commandes du S.E. incluent un certain nombre de commandes SGF



Langage de commandes

Quelques commandes du SGF:

- liste du répertoire
- changement de répertoire
- création sous-répertoire
- suppression sous-répertoire
- création fichier (allocation statique)
- suppression fichier
- modification d'attributs d'un fichier
- changement de nom de fichier...

Au lancement d'une commande -> appel à la fonction du SGF

Les requêtes

Les requêtes sont des appels aux fonctions SGF dans un programme.
Les requêtes permettent de modifier le contenu des fichiers

Quelques requêtes du SGF:

- ouverture fichier
 - fermeture fichier
- et
- lecture d'enregistrement
 - écriture d'enregistrement
- ou
- création d'un enregistrement
 - suppression d'un enregistrement
 - modification d'un enregistrement

Ouverture fichier

OPEN (fic, n°, descripteur, L/E)

L'ouverture

- vérifie l'existence du fichier dans le répertoire du disque
- transfère les éléments dans le descripteur correspondant en mémoire

Le descripteur:

- nom du fichier
- n° logique correspondant
- pointeur sur 1er enregistrement du fichier
- autorisation d'accès: L lecture E: écriture.....

Le descripteur est conservé et mis à jour en mémoire jusqu'à la fermeture

Lecture fichier

READ (n°fic, accès, tampon, n°enregistrement)

La lecture d'un fichier nécessite:

- n° logique du fichier du descripteur
- mode d'accès demandé compatible avec l'organisation
- l'adresse mémoire de la zone réceptrice de l'enregistrement
- n° de l'enregistrement qui peut:
 - . ne pas exister si lecture séquentielle
 - . être une clé
 - . être le rang si accès relatif.....

Cette fonction n'agit pas sur le contenu du descripteur (lecture)

Les données lues sont transférées dans le tampon mémoire

Plusieurs requêtes peuvent exister avec des fonctionnalités différentes

Écriture fichier

WRITE (n°fic, accès, tampon, n°enregistrement)

L'écriture dans un fichier nécessite:

- n° logique du fichier du descripteur
- mode d'accès demandé compatible avec l'organisation
- l'adresse mémoire de la zone émettrice de l'enregistrement
- n° de l'enregistrement qui peut:
 - . ne pas exister si écriture séquentielle
 - . être une nouvelle clé (contrôle unicité)
 - . être le rang si accès relatif.....

Cette fonction met à jour certains éléments du descripteur

Gestion des pointeurs ou tables d'allocation des espaces

Les données sont transférées du tampon mémoire vers le fichier

Plusieurs requêtes peuvent exister avec des fonctionnalités différentes

Fermeture fichier

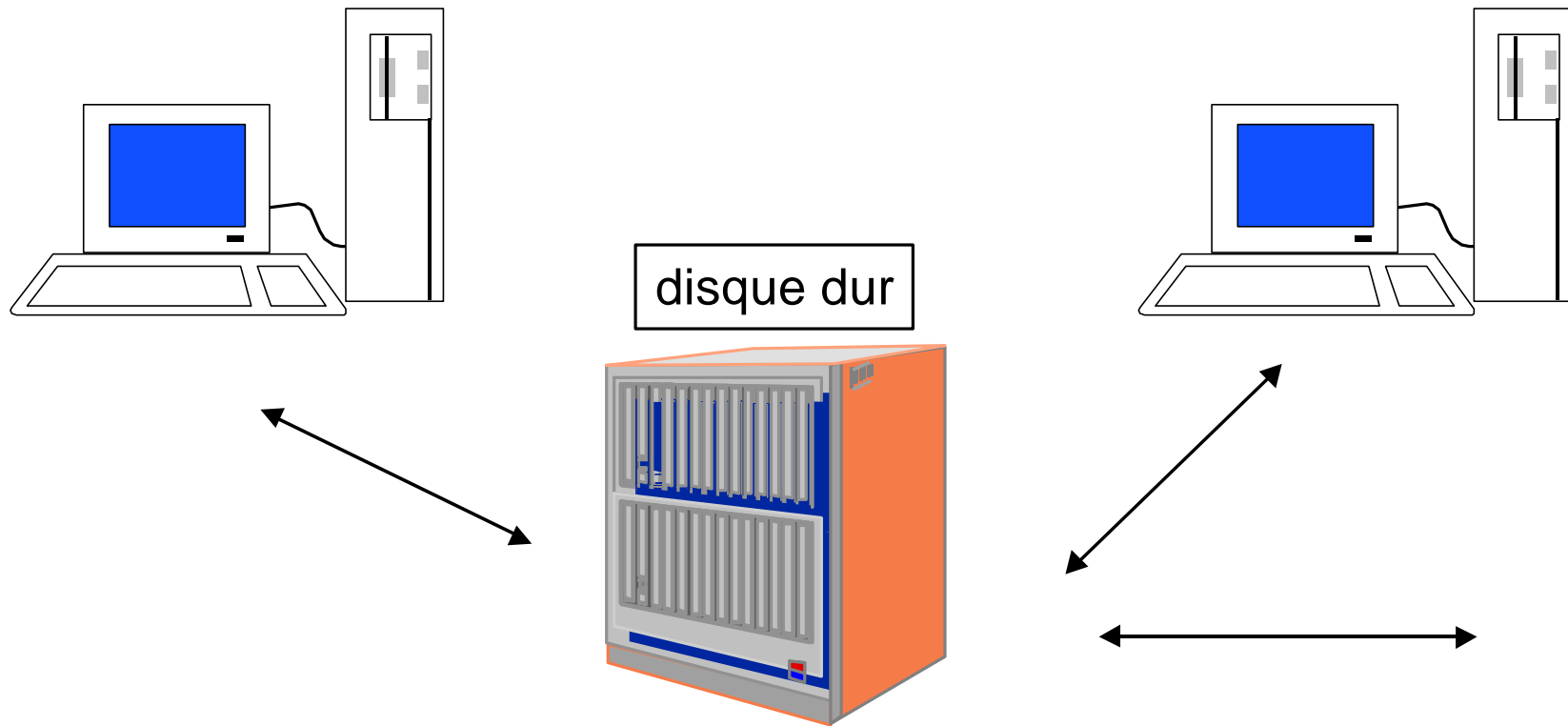
CLOSE (n°fic, descripteur)

La fermeture:

- transfère les éléments du descripteur vers le répertoire sur le disque
- les pointeurs vers l'espace libre sont mis à jour
- d'autres accès sont alors possibles

Les serveurs de fichiers

Partage de fichiers à distance



Le disque déporté

Allocation d'un espace du disque pour chaque utilisateur: **disque virtuel**

- même interface qu'un SGF local
- simulation du contrôleur par le réseau

Accès à des **fichiers à distance**:

- création du fichier
- retour par le serveur de fichiers d'un n° de fichier: identificateur
- accès à ce fichier uniquement avec le n° identificateur

SGF serveur de fichiers:

- le serveur offre un SGF complet
- transparence total de la localisation des répertoires, des fichiers...

La mise à jour

Pour assurer la cohérence des données

Mise à jour atomique:

- lecture ou écriture d'un enregistrement complète et non partielle

Sauvegarde stable:

- deux disques sont mis à jour à des emplacements physiques identiques
- contrôle périodique par le serveur de l'égalité de chaque bloc
 - . si bloc endommagé recopie du bloc

Fichiers à plusieurs versions:

- les mises à jour sont effectuées sur une copie
- fichiers datés

Les accès concurrents

Ex: maj simultanées d'un enregistrement dans un fichier

- $Z=500$

- maj P1 -> $Z = Z + 200$ et

P2 -> $Z + Z + 300$ $\Rightarrow Z = 1000$

- P1 lit $Z=500$

- P2 lit $Z=500$

- P1 $Z = 500 + 200 \Rightarrow Z = 700$

- P2 $Z = 500 + 300 \Rightarrow Z = 800$

$\Rightarrow Z = 800$ faux

SOLUTIONS:

- sérialiser les mises à jour

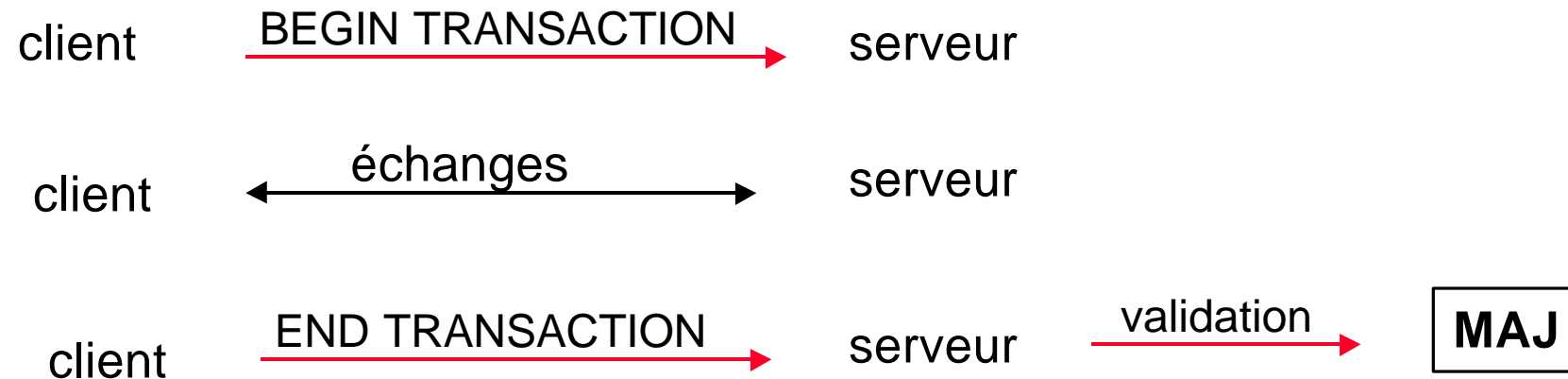
- verrouillage du fichier lors d'un accès



Algorithmes de contrôle d'accès concurrents

Les transactions

Une transaction s'exécute à terme de l'opération



Les transactions

BEGIN TRANSACTION:

- Serveur
- crée un enregistrement de la transaction
 - verrouille l'accès au fichier
 - copie du fichier dès la première tentative d'écriture
- les maj se font sur la copie

END TRANSACTION:

- Serveur
- crée une liste d'intentions
 - copie la liste dans l'enregistrement de la transaction
 - interdit toute opération jusqu'à la fin de la transaction
 - met à jour tous les blocs du fichier (maj atomique)
 - déverrouille le fichier
 - validation

Les fichiers dupliqués

Certains SGF dupliquent n fois les fichiers

Nom_Fichier	noeud1	noeud2	noeud3	noeud4
Nom_Fic1	45	865	659	84
Nom_Fic2	56	54	75	99
Nom_Fic3	55	59	26	77

En cas de modification:

- maj des blocs des copies

ou

- nouvelles copies

La sécurité

Protection des informations contre des accès non autorisés

Les pertes de données:

- catastrophes: incendie, feu....
- erreurs matérielles et logicielles: panne, bogues
- erreurs humaines: saisie erronée, mauvais programme,



prévention: sauvegarde des fichiers

Les intrus:

- indiscretion
- contournement des obstacles d'accès par plaisir...
- gain financier: transfert de comptes bancaires
- espionnage

Quelques cas

Failles du SGF des anciennes versions d'Unix:

- lpr impression puis effacement du fichier mots de passe
- core fichier lié au fichier des mots de passe:
 effacement et remplacement
- mkdir création d'un répertoire: uid = racine puis uid = propriétaire
 par chown
 entre mkdir et chown se mettre propriétaire

Le cheval de Troie:

- modification d'un programme de manière à fonctionner normalement mais en effectuant en plus des opérations illicites.

Les moyens de sécurité

Les droits d'accès:

- identificateur des utilisateurs: **mot de passe**

> Unix

nom d'utilisateur

mot de passe associé

> autres méthodes:

- quelques questions précises (2/3) avec réponses sont stockées

le S.E. pose une question au hasard

- algorithme connu de l'utilisateur: $5 * N$

Le S.E. donne 4

l'utilisateur doit répondre 20

- **Identification physique**

> carte magnétique, signature, voix,

La protection

La protection des fichiers par des **droits d'accès**

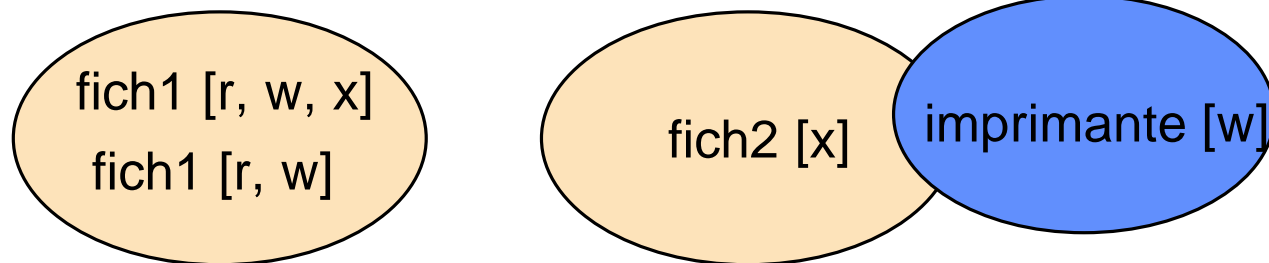
fich1	droits
fich2	droits
fich3	droits

fich1 [r, w, x]

fich2 [x]

fich1 [r, w]

Par leur appartenance à des **domaines**



ex: Unix domaines d'accès: propriétaire, ,groupe ou autre
domaines d'exécution: utilisateur ou noyau

CONCLUSION

Le SGF du S.E est une partie complexe mais necessaire:

- il permet de stocker des données sur un support externe
- il utilise des concepts logiques faciles à mettre en oeuvre
- il décharge l'utilisateur de tout concept physique
- il peut être utilisé en mode commande ou dans un applicatif en mode requête

Chaque S.E. à son propre SGF avec ses caractéristiques:

- organisations
- modes d'accès
- enregistrement plus ou moins structuré
- multiutilisateurs
- niveau de protection et sécurité