ORGANISATION LOGIQUE DES FICHIERS

Organisation des fichiers

- Une *organisation de fichier* est une manière de disposer les enregistrements dans un fichier stocké sur le disque.
- Un fichier peut être accédé et modifié de différentes façons, et une organisation de fichier peut être *bonne* pour un type d'accès et *mauvaise* pour un autre type.
- Un fichier trié sur le nom des employés n'est pas une bonne organisation quand on veut avoir les employés ayant un salaire supérieur à 100.
- Un SGBD offre plusieurs organisations possibles. C'est à l'administrateur que revient le choix de l'organisation adéquate.

Modèle de coût

Pour pouvoir comparer différentes organisations, il nous faut un modèle de coût. Pour cela, on va considérer les paramètres :

- **P** : Le nombre de pages contenant des données
- **E** : Le nombre d'enregistrements par page
- **T** : Le temps "moyen" pour lire ou écrire une page

Ainsi le coût d'une opération est exprimée en fonction de ces paramètres.
Dans ce modèle *simplifié*, on ne considère pas le coût **C** des traitements en unité centrale (en général, nous avons $\mathbf{C} = \frac{\mathbf{T}}{\mathbf{25}}$).

Trois types d'organisations :

1. Fichier tas
2. Fichier trié sur un champ
3. Fichier "haché" sur un champ

Opérations considérées :

- **Balayage :** (Scan) parcourir tous le fichier
- **Recherche avec égalité** : On cherche les enregistrements ayant un champs X égal à une valeur particulière
- **recherche avec intervalle** : On cherche les enregistrements ayant un champs X compris dans un intervalle particulier
- **Insertion** : (i) On doit identifier la page qui doit contenir l'enregistrement à insérer (ii) la modifier en mémoire et la réécrire sur disque.
- **Suppression** : Suppression d'un enregistrement identifié par son *IdE*. (i) ramener la page correspondant en mémoire, (ii) la modifier ensuite la réécrire sur disque.

Fichiers hachés

Un fichier haché est caractérisé par

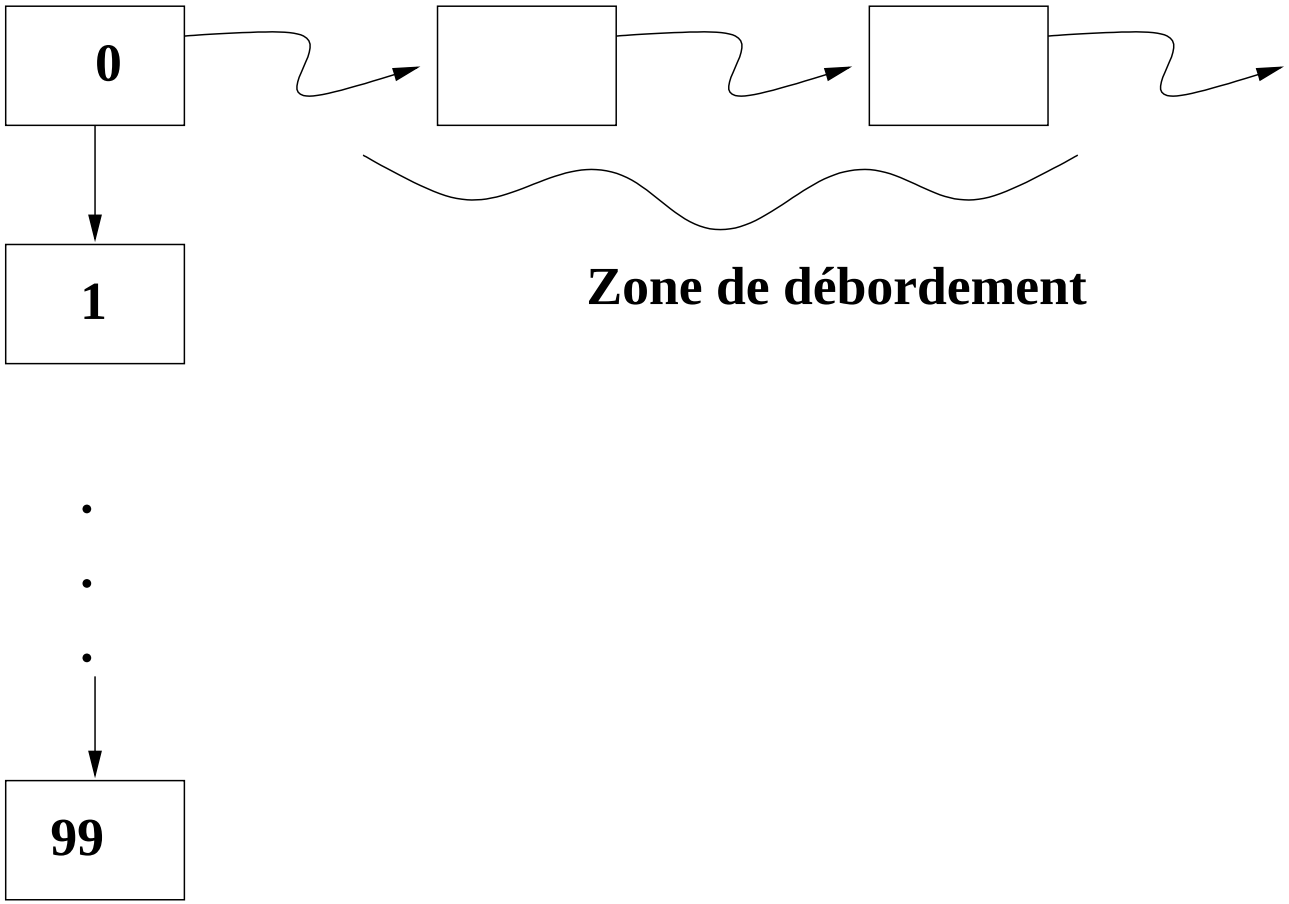
1. un champ particulier appelé *clé de recherche*[1]
2. une fonction de hachage h qui associe à chaque valeur de la *clé de recherche* un entier.
   Exemple : $X = X_1 X_2 \ldots X_n$ une String
   $h : String \longrightarrow \mathbf{N}.\ h(X) = \sum_{i=1}^{n} ASCII(X_i) modulo(100)$

L'entier associé à une valeur correspond au numéro de la page où l'enregistrement correspondant doit se trouver.

[1] Rien a voir avec la notion de "clé" dans une table.

Fichiers hachés

Plusieurs valeurs de la clé peuvent être associées au même entier.
Nous avons donc une zone de débordement.
Exemple : $h(abc) = h(bca)$.



Fichier tas

- **Balayage :** coûte $\boxed{\mathbf{P \times T}}$. Il faut transférer toutes les pages en mémoire
- **Recherche avec égalité :**
  1. Si on sait qu'il y a un seul enregistrement, alors en moyenne on lit la moitié du fichier ; le coût est $\frac{\mathbf{P \times T}}{\mathbf{2}}$. Au pire, on doit lire tout le fichier donc $\boxed{\mathbf{P \times T}}$
  2. Si l'on ne sait pas le nombre d'enregistrements, alors le coût est $\boxed{\mathbf{P \times T}}$
- **Recherche avec intervalle :** On doit balayer tout le fichier ; le coût est $\boxed{\mathbf{P \times T}}$
- **Insertion :** On suppose que l'insertion se fait à la fin du fichier. (i) On lit la dernière page (1 transfert), puis (ii) on la modifie en mémoire et on la réécrit sur disque (2em transfert) ; le coût est $\boxed{\mathbf{2 \times T}}$.

Fichiers tas (Suite)

- **Suppression :** Comme on a l'IdE de l'enregistrement à supprimer, on accède directement à sa page pour la charger en MC (1 transfert), ensuite (ii) on fait la modification en mémoire et on réécrit la page ; le coût est $\boxed{\mathbf{2 \times T}}$.
  On a supposé que les pages ne sont pas compactées.
  Quel serait le coût si au lieu de l'IdE, on a une condition sur une valeur associée à (aux) enregistrement(s) à supprimer ?
  e.g. Supprimer les employés dont le salaire est inférieur à X.

Fichiers triés

Balayage : Le coût est $\boxed{\mathbf{P} \times \mathbf{T}}$

Recherche avec égalité :

1. Si le champs de recherche n'est pas celui du tri, alors même coût que le fichier tas : $\boxed{\mathbf{P} \times \mathbf{T}}$
2. Sinon, avec une recherche dichotomique, on peut retrouver l'enregistrement en $\log_2(P)$ transferts ; le coût est $\boxed{\log_2(\mathbf{P}) \times \mathbf{T}}$.

Ici, on n'a considéré qu'un seul enregistrement trouvé

Recherche avec intervalle : On fait d'abord une recherche dichotomique jusqu'à trouver une valeur dans l'intervalle spécifié ; le coût est $\boxed{\log_2(\mathbf{P}) \times \mathbf{T}}$.

Là aussi, on n'a considéré qu'un seule valeur.

S'il y a N enregistrements, alors il nous faudra transférer au max $\frac{\mathbf{N}}{\mathbf{E}} + 1$ autres pages.

Le coût est majoré par $\boxed{\left(\log_2(\mathbf{P}) + \frac{\mathbf{N}}{\mathbf{E}} + \mathbf{1}\right) \times \mathbf{T}}$.

Fichiers triés (Suite)

- **Insertion :** Il faut d'abord chercher la bonne page où le placer. Au pire, ceci peut nécessiter la lecture de $\log_2(P)$ pages (c'est la situation où la page est au début ou à la fin du fichier). La page peut être déjà remplie.
  1. Si elle est à la fin, alors il faut y insérer l'enregistrement et créer une nouvelle page, i.e 2 écritures
  2. si elle est au début, au pire il faut décaler tous les enregistrements (toutes les pages sont pleines). Ce qui fait à peu près $2 \times P$ accès.

  Au pire, nous aurons donc un coût

$$\boxed{\Big(\log_2(P) + (2 \times P)\Big) \times T}$$

- **Suppression :** On retrouve la page de l'enregistrement en un transfert. On la modifie et on la réécrit sur le disque. Ca donne un coût $\boxed{2 \times T}$.
  Noter qu'on n'a pas considéré la situation où l'on doit réorganiser le fichier si la page devient vide

Fichiers hachés

On suppose qu'il n'y a pas de débordement

- **Balayage :** Pareil que pour le fichier tas ; le coût est donc $\boxed{\mathbf{P \times T}}$
- **Recherche avec égalité :**
  - Si X est la valeur recherchée, alors on peut savoir la page en calculant $h(X)$ et on fait un accès direct ; coût est $\boxed{\mathbf{T}}$
  - S'il y a plusieurs enregistrements, comme on a supposé qu'il n'y a pas de débordement, alors tous les autres enregistrements sont dans la même page
- **Recherche avec intervalle :** On est obligé de balayer tout le fichier ; le coût est $\boxed{\mathbf{P \times T}}$
- **Insertion :** Le coût est $\boxed{\mathbf{2T}}$; on accède directement à la page
- **Suppression :** On lit la page puis on la réécrit ; le coût est donc $\boxed{\mathbf{2T}}$

Conclusion

Il n'y a pas une organisation meilleure que toutes les autres pour tous les types d'opérations.

	Tas	Trié	Haché
Balayage	$P \times T$	$P \times T$	$P \times T$
Recherche $=$	$P \times T$	$\log_2(P) \times T$	T
Recherche [ ]	$P \times T$	$\left(\log_2(P) + \frac{N}{E} + 1\right) \times T$	$P \times T$
Insertion	$2 \times T$	$\left(\log_2(P) + (2 \times P)\right) \times T$	$2T$
Suppression	$2 \times T$	$2 \times T$	$2T$