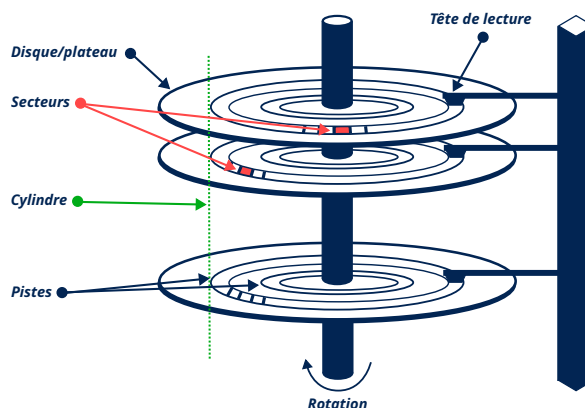


Exercice n°1 : Gestionnaire de mémoire

Dans cette première partie, on considérera comme média de stockage un disque dur à plateaux. Celui-ci est constitué de différents éléments : les **secteurs** correspondant à la plus petite unité de lecture, les secteurs sont contenus dans des **blocs** correspondant à une suite consécutive de secteurs, enfin ces blocs sont ensuite arrangés en cercle concentrique sur les **disques ou plateaux**, ces espaces de données sont appelés des **pistes**.



Question n°1 : Quels sont les sous-modules du gestionnaire de mémoire ?

Question n°2 : Expliquez les différentes opérations pour la lecture d'un secteur sur un stockage de type disque à plateaux (HDD).

Question n°3 : Considérons un disque où la taille des secteurs vaut 1024 octets, chaque piste contient 50 secteurs et chacune des surfaces à une capacité de 102400 octets, enfin le HDD contient deux plateaux (double face).

1. Quelle est la capacité d'une piste ? du disque ?
2. Combien y-a t-il de secteurs sur une surface ?

Question n°4 : Dans le cadre du gestionnaire de cache, quel est l'objectif du stockage du `dirty_bit` pour chaque page présente dans les cellules du cache ? Que se passe t-il si l'on cherche à remplacer une cellule contenant une page qui a un `dirty_bit` égal à 1

Question n°5 : Qu'est-ce que le `pin_count` ? Quand est-il incrémenté ? Décrémenté ?

Question n°6 : On considère la stratégie de remplacement FIFO, on considère dans cette exemple la lecture des pages seulement (qui sont automatiquement libérées ensuite). Quel est le nombre de lecture sur le disque de la séquences de demande de pages suivantes (pour un cache de 3 cellules) :

$$p_1 \rightarrow p_2 \rightarrow p_3 \rightarrow p_1 \rightarrow p_2 \rightarrow p_4 \rightarrow p_1 \rightarrow p_2 \rightarrow p_5 \rightarrow p_6$$

Question n°7 : Même question avec un remplacement en utilisant la stratégie LRU (la page la moins récemment utilisée)

Exercice n°2 : Disque et temps d'accès

On considère un disque ayant les caractéristiques suivantes : chaque face contient 2 000 pistes, chaque piste contient 50 blocs, et il a 5 plateaux double face. On suppose que le temps moyen de recherche d'une piste (mouvement du bras du disque) est de 10 msec et que la taille d'un bloc a été fixée à 1024 octets, considérons un fichier de 100 000 enregistrements de 100 octets chacun devant être stocké sur le disque (le chevauchement d'un enregistrement sur plusieurs blocs n'est pas autorisé).

Question n°1 : Combien d'enregistrements peuvent être stockés dans un bloc ?

Question n°2 : Combien de blocs sont nécessaires pour le stockage du fichier ?

Question n°3 : Si le fichier est placé séquentiellement sur le disque, combien de cylindres sont nécessaires pour le stockage du fichier ? Combien de temps est nécessaire pour lire le fichier ?

Question n°4 : Même question en supposant que le disque puisse lire en parallèle avec ses 10 têtes ?

Exercice n°3 : Fichiers et pages

On suppose qu'une relation est stockée sur 1000 pages et que chaque page contient 10 enregistrements.

Question n°1 : On souhaite accéder à l'enregistrement ayant pour $v_id = 1002$. Si toutes les pages contiennent 10 enregistrements, sur quelle page peut-on retrouver l'enregistrement si le fichier trié sur v_id ? Si le fichier est un tas ?

Question n°2 : On considère que une page p_{1002} ayant comme adresse de base dans le cache $0x1020$, qu'une page a une entête de 100 octets et que l'on cherche à accéder au 3ème enregistrement de la page. Donnez l'adresse du pointeur sur le début de l'enregistrement.

Question n°3 : On suppose que le fichier est trié, déterminez combien de lecture et d'écriture de blocs sont nécessaires pour d'ajout d'un enregistrement étant placé en 501ème position, en supposant que toutes les pages sont remplies.

Question n°4 : Même question mais cette fois-ci pour un tas.

Question n°5 : Quel est le temps maximal pour l'accès à l'enregistrement ayant pour $v_id = 512$ dans le cas d'un fichier trié sur v_id ? Dans le cas d'un tas ?

Exercice n°4 : Introduction aux index

Question n°1 : Quelles sont les alternatives pour les entrées d'un index ?

Question n°2 : Déterminer la structure de fichier la plus adaptée parmi le tas (*heap file*) ou le fichier trié (*sorted file*) si les opérations les plus fréquentes sont :

1. L'ajout de nouveaux enregistrements
2. Récupérer tout les éléments d'une instance de relation
3. La recherche d'un intervalle de valeurs

Expliquez pourquoi et donnez le nombre de chargement de page moyen approximatif (on considérera N pages dans le fichier) ?

Question n°3 : Quelle est la différence entre un index groupant et un index non groupant ? Est-il possible d'avoir plusieurs indexes groupants pour un fichier de données ?

Question n°4 : On dispose de l'instance de relation suivante :

NID	NOM	LOGIN	21	Moyenne
53831	Madrian	madayan@etu.fr	21	11.5
53832	Guads	guads@etu.fr	21	13
53833	Jean	jean@etu.fr	24	15.7
53838	Paul	Paul@etu.fr	22	13.4
53839	Jacques	jacques@etu.fr	22	10.4
53855	Jacques	jacques2@etu.fr	19	19.4

On obtient par un scan de la relation Etudiant la table précédente. Une page peut contenir au plus 3 enregistrements. Le rid d'un enregistrement est un couple numéro de page, numéro d'emplacement. Listez les entrées des index pour les caractéristiques suivantes :

1. Un index dense sur NID de type 2

2. Un index dense sur age de type 1
3. Un index dense sur age de type 2
4. Un index dense sur age de type 3
5. Un index non-dense de type 2 sur *NID*
6. Un index non-dense sur moyenne de type 2
7. Un index dense sur Nom de type 3

Question n°5 : On considère l'instance de relation précédente, on dispose d'un index trié sur la Moyenne (de type 2), en parcourant l'index pour les moyennes ≤ 16 . Quelles sont les pages successivement chargées? Expliquez les risques liés aux indexes non-groupant en terme de performance.

Question n°6 : Expliquez le principe du “bitmap index scan” et du “bitmap Heap Scan”. En considérant le tableau de la question 5 et un index de type 2 trié, combien de lectures de pages sont effectuées au maximum ?