

COMPTE RENDUE DU TP4 DE SE

TD4 : Systèmes de Gestion de fichier

Souleymane DIOP et Mouhamed Abdallah SAKHO

DSTI2B

Exercice 1:

En supposant qu'un secteur contient 512 octets, calculons la taille maximale que peuvent utiliser ces systèmes :

$$T_{\max} = 512 * 63 * 16 * 1024 = 528\,482\,304 = 528 \text{ Mo}$$

Exercice 2:

Déterminons le déplacement total de la tête de lecture/écriture après que toutes les demandes aient été satisfaites si l'algorithme de planification des déplacements de la tête est :

- Premier arrivé premier servi

$$DT = (29-15) + (100-15) + (100-30) + (30-27) + (55-27) + (55-16) + (122-16) + (122-44) + (63-44) + (63-56) = 449$$

$$DT = 449$$

- Le plus petit déplacement d'abord (SSTF)

$$DT = (29-15) + (16-15) + (27-16) + (30-27) + (44-30) + (55-44) + (56-55) + (63-56) + (100-63) + (122-100) = 121$$

$$DT = 121$$

Exercice 3 :

1) La + grande taille que le SDF peut supporter :

Chaque bloc de données peut contenir 256 pointeurs (numéros de blocs). Il faut : - (12 * 1k) pour direct, - (256 * 1K) pour ind. simple, - (256 * 256 * 1K) pour ind. double, - (256 * 256 * 256 * 1K) pour ind. Triple Taille = (12 * 1K) + (256 * 1K) + (256 * 256 * 1K) + (256 * 256 * 256 * 1K) = (12 + 256 + 256*256 + 256*256*256) K = 16,843,020 K octets.

2) Le nombre de blocs de données nécessaires pour représenter ce fichier :

Si on considère un fichier contenant 100,000 octets, on voit que : $100,000 = 97 * 1024 + 672$. Il faudra donc 98 blocs pour conserver les données de ce fichier.

Exercice 4:

1. Montons que la création du fichier f1 conduit à une fragmentation interne.

Il faut d'abord comprendre que la fragmentation interne se produit lorsque des données sont stockées sur des blocs de données disjoints sur le disque, plutôt que sur des blocs contigus. Dans le cas de f1, avec une taille de bloc de 4K et 12 pointeurs directs, 1 pointeur indirect simple, 1 pointeur indirect double et 1 pointeur indirect triple, il est possible que des données soient stockées sur des blocs de données disjoints, car le nombre total de caractères dans f1 est de 20.000.000, ce qui dépasse la capacité de stockage des pointeurs directs. Les données supplémentaires seront stockées sur des blocs de données supplémentaires, créant ainsi de la fragmentation interne.

2. Calculer alors la fragmentation interne totale sur le disque résultant de la création de f1.

Pour calculer la fragmentation interne totale sur le disque résultant de la création de f1, il est nécessaire de déterminer le nombre total de blocs de données utilisés pour stocker les 20.000.000 de caractères. Avec une taille de

bloc de 4K, chacun de ces blocs peut contenir 4096 octets de données. Il faut donc diviser 20.000.000 par 4096 pour déterminer le nombre total de blocs de données utilisés pour stocker les 20.000.000 de caractères.

$$20.000.000 / 4096 = 4883.984375$$

Il est donc nécessaire de 4884 blocs de données pour stocker les 20.000.000 de caractères.

Exercice 5:

1. Calculons la taille maximale en nombre de blocs que pourrait avoir en précisant les nombres d'index et de données

- Chaque bloc d'index pointe vers 2047 blocs de données soit (8 KO / 4 O)

Le nombre maximal de numéros de blocs est 232 - Si m est le nombre maximal de blocs d'index et tous les blocs sont alloués au fichier (données +index), on a : $3 \ 232 = m + (2047 * m) \Rightarrow m = 232 / 2048 \Rightarrow m = 221$ Le nombre de blocs de données est : $2047 * 221$.

La taille maximale d'un fichier serait de 2 32 blocs dont 2 21 blocs d'index et $2047 * 221$ blocs de données.

Exo6 :

2) Quantité maximale de données que peut contenir un fichier Qmax On
 $ah=1024$

$$Q_{\max} = 4 + (h-1) + (h-1) + h \text{ au carré} + h \text{ au cube} = 1074792450 \text{ blocs}$$

3) Le nombre de blocs nécessaires pour stocker un fichier dont la taille est de 10blocs

$$NB = 4 + 1 = 5 \text{ blocs}$$

4) Le nombre total de blocs nécessaires pour stocker un fichier dont la taille est de 1095 blocs

$$NB = 4 + 1 + 1 + 1 + 1 + 67 + 1024 = 1099 \text{ blocs}$$

5) Le nombre total de blocs nécessaire pour stocker un fichier dont la taille est de 4000 blocs

$$NB = 1024 \times 3 + 924 + 4 + 1 + 1 + 4 = 4006 \text{ blocs}$$

6) Donnons la taille d'un pointeur $T@$:

$$T@ = 32768 / 4194304 = 7,8125.10^{-3} \text{ bits}$$

-calculons la taille maximale que peut atteindre un fichier T_{max}

$$T_{max} = 4 \times 32768 + 2(h \times 32768) + h^2 \times 32768 = 1,803.10^6 \text{ bits}$$

7) Donnons le nombre total de blocs pour stocker un fichier de 270.463.552 bits NB

$$NB = 270463552 / 32768 = 8253,89 \text{ blocs}$$

L'i-noeud ne dispose que de 12 pointeurs directs, il va donc falloir utiliser des blocs supplémentaires ($98 - 12 = 86$) pour conserver le reste des données.

Comme 86