

### TD3 : Gestion de la mémoire secondaire

- I. On considère une variante de Unix classique (pre-BSD 4.2) utilisant le système de fichiers *sfs* (*System V File System*). L'information concernant les blocs de données de chaque fichier est donc accessible à partir du *inode* de celui-ci. On supposera qu'un pointeur occupe 4 octets:
- Le système de fichiers utilise des blocs de données de taille fixe 1K (1024 octets) ;
  - Le *inode* de chaque fichier (ou répertoire) contient 12 pointeurs directs sur des blocs de données, 1 pointeur indirect simple, 1 pointeur indirect double et 1 pointeur indirect triple.
1. Quelle est la plus grande taille de fichier que ce système de fichiers peut accommoder ?
  2. On considère un fichier contenant 100.000 octets. Combien de blocs :
    - 2.1. de données sont nécessaire pour stocker les 100.000 octets ?
    - 2.2. au total sont-ils nécessaires pour représenter ce fichier sur disque ?
- II. On considère un système fonctionnant sous une variante de Unix classique (pré-BSD 4.2) disposant du système de fichiers *sfs* avec une taille de blocs de données de 4K (4096 octets) et des adresses définies sur 4 octets. On supposera que le *inode* de chaque fichier compte 12 pointeurs directs, 1 pointeur indirect simple, 1 pointeur indirect double et 1 pointeur indirect triple. On désire créer un fichier contenant un total de 20.000.000 (vingt millions) de caractères (caractères de fin de ligne et de fin de fichier compris). Quelle est la fragmentation interne **totale** sur le disque résultant de la création de ce fichier.
- III. On considère un système fonctionnant sous une variante de Unix classique (pré-BSD 4.2) disposant du système de fichiers *sfs* disposant des caractéristiques suivantes :
- La taille des blocs est 2K.
  - Le *inode* contient 10 liens directs vers des blocs de données.
  - Un lien vers un bloc (une adresse) est représenté sur 4 octets.
1. Quelles sont la taille minimale et la taille maximale pour un fichier pour la représentation duquel exactement 500 blocs sont nécessaires pour conserver des liens vers d'autres blocs?
  2. On considère un fichier de texte contenant 2 500 000 caractères ASCII (y-compris les caractères de fin de ligne et de fin de fichier). Suite à un malencontreux accident, le *inode* de ce fichier est corrompu et la valeur des pointeurs indirects double et triple est détruit. Quelle proportion du contenu de ce fichier a-t-elle été perdue par suite de cet accident ?
- IV. Supposons que chaque piste d'un disque soit divisée en 12 secteurs et que la tête soit placée au-dessus du secteur 2, avec un sens de rotation correspondant au défilement de secteurs par numéros croissants. Si les requêtes pour les secteurs 11, 5, 8 et 7 sont en attente, quel serait l'ordre de traitement dans le cas de l'algorithme PCTL.
- V. Un disque contient 200 pistes (numérotées de 0 à 199). La tête de lecture sert une requête piste 143 et vient de terminer une requête piste 125. Calculez le nombre de déplacements de la tête pour les algorithmes suivants et en supposant que la file d'attente des requêtes contient les requêtes suivantes :  
86, 147, 91, 177, 94, 150, 102, 175 et 130  
PCTR, SCAN, C-SCAN et LOOK
- VI. Un disque contient 200 pistes numérotées de 0 à 199. La dernière requête traitée concernait la piste 112 et une requête en cours pour la piste 138. La liste des nouvelles requêtes dans l'ordre d'arrivée est la suivante : 91, 165, 67, 158, 43, 132, 28, 106 et 84  
Calculer le déplacement total de la tête pour les algorithmes suivants :  
FIFO, PCTR, SCAN, C-SCAN, LOOK et C-LOOK