

ED 2 : Mémoire centrale et système de gestion de fichiers

NFP136

Exercice 1 — *Espace d'adressage paginé*

On considère 3 processus A, B, C.

- Le processus A dispose de 3 pages P1, P2 et P3 dans son espace d'adressage.
- Le processus B dispose de 3 pages P1, P2 et P3 dans son espace d'adressage.
- Le processus C dispose de 2 pages P1, P2 dans son espace d'adressage.

Les cases mémoires numérotées de 0 à 19 sont au nombre de 20. Une case mémoire a une capacité de 1 Ko. A l'instant t, l'état d'allocation en mémoire centrale des pages des 3 processus est le suivant :

- Les pages P1 et P2 du processus A sont chargées respectivement dans les cases 3 et 7 de la mémoire centrale ;
- Les pages P3 et P2 du processus B sont chargées respectivement dans les cases 5 et 12 de la mémoire centrale ;
- La page P1 du processus C est chargée dans la case 8 de la mémoire centrale ;

1. Représentez les structures de données allouées (mémoire centrale et tables des pages).
2. Dans chacun des deux cas suivants, calculez l'adresse paginée, puis l'adresse physique correspondant à l'adresse linéaire qui vous est donnée. En cas de défaut de page, vous supposerez que la page manquante est chargée en mémoire centrale dans la première case libre de plus petit numéro.
 - (a) Adresse linéaire 1152 dans l'espace d'adressage du processus A ;
 - (b) Adresse linéaire 1536 dans l'espace d'adressage du processus C.

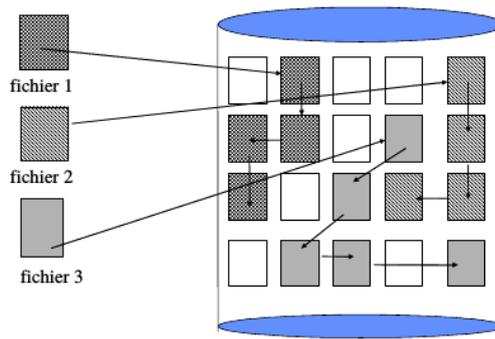
Exercice 2 — *Algorithmes de pagination*

On s'intéresse aux systèmes de pagination à la demande. Au cours de son exécution, un programme accède successivement aux pages 0, 1, 4, 2, 0, 1, 3, 0, 1, 4, 2, 3.

1. On utilise l'algorithme FIFO et le système alloue à ce programme un espace de 3 pages (initialement vides). Donner la suite des pages présentes en mémoire et en déduire le nombre de défauts de page.
2. On utilise l'algorithme LRU et le système alloue à ce programme un espace de 3 pages (initialement vides). Donner la suite des pages présentes en mémoire et en déduire le nombre de défauts de page.

Exercice 3 — *Gestion de fichiers Windows*

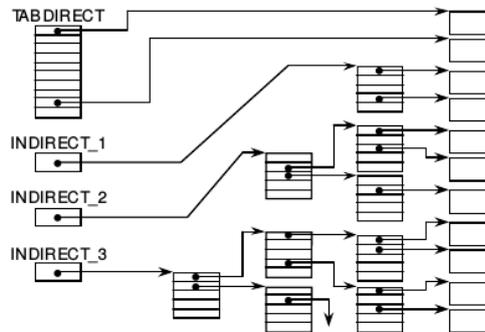
Soit le schéma suivant représentant l'allocation du disque C sur un système Windows pour lequel l'allocation est géré selon un système de FAT. Les blocs sont numérotés de 1 à 20, ligne à ligne, de gauche à droite.



Représentez le contenu des 20 premières entrées de la table en fonction de l'allocation figurée par le schéma. Les blocs sont numérotés à partir de 1 par ligne de gauche à droite.

Exercice 4 — Gestion de fichiers UNIX

La localisation d'un fichier UNIX est assurée par la structure suivante :



- une table, que nous appelons *TABDIRECT*, donnant les numéros de blocs pour les 10 premiers blocs de l'objet externe,
- un numéro de bloc, que nous appelons *INDIRECT₁*, qui contiendra les p numéros des blocs suivants de l'objet externe,
- un numéro de bloc, que nous appelons *INDIRECT₂*, qui contiendra les p numéros de blocs contenant les numéros des p^2 blocs suivants de l'objet externe,
- un numéro de bloc, que nous appelons *INDIRECT₃*, qui contiendra les p numéros de blocs contenant, au total, p^2 numéros de blocs contenant les numéros des p^3 blocs suivants de l'objet externe.

On considère un fichier de type UNIX. Le fichier a une taille de 16 Moctets. Les blocs disque sont de 1024 octets. Un numéro de bloc occupe 2 octets.

1. Quel est le nombre de blocs de données du fichier ?
2. Représentez sur un schéma l'allocation des blocs de données du fichier.
3. Combien comporte-t-il de blocs d'adresses ?
4. Si le système dispose d'un cache disque, combien d'accès disque sont nécessaires pour lire le fichier ?