

Gestion Mémoire (Pagination – segmentation)

Exercice 1 Adressage virtuel

Un ordinateur a un espace d'adressage virtuel codé sur 32 bits. Une adresse désigne un octet. Les pages ont une taille de 256 Ko.

1. Evaluer la taille de la mémoire virtuelle.
2. Pour une adresse virtuelle donnée comment est calculé le numéro de la page virtuelle et l'adresse (déplacement) dans cette page.

Exercice 2 Pagination

Un ordinateur a 16 pages d'adressage virtuelles mais seulement 4 pages physiques. Au départ, la mémoire est vide. Un programme référence les pages virtuelles dans l'ordre : 0, 7, 2, 7, 5, 8, 9, 2, 4, 2.

1. Quelles sont les références mémoire qui provoqueront des défauts de page : avec OPTIMAL, avec LRU, avec FIFO, avec FINUFO
2. Trouvez un exemple où FinuFo n'est pas optimal.
3. Donner l'algorithme FINUFO.
4. Proposez une amélioration de FinuFo en considérant qu'en plus du bit de référence, chaque entrée dans la table des pages dispose aussi d'un bit de modification.

Exercice 3 Structure des programmes

La pagination à la demande est souvent conçue pour être transparente à l'utilisateur, qui ignore souvent que la mémoire est paginée. Mais parfois, la performance du système pourrait être améliorée si l'utilisateur (ou le compilateur) était au courant de la pagination à la demande sous-jacente. Considérons un programme JAVA dont la fonction est d'initialiser à 0 les éléments d'un tableau de 128x128.

On suppose que le tableau est stocké en mémoire ligne par ligne :

```
A[0][0], A[0][1], ..., A[0][127], A[1][0], A[1][1], ..., A[127][127]
```

Les instructions suivantes sont exécutées :

```
int A[][] = new int[128][128];  
for (int j=0; j<128; j++)  
for (int i=0; i<128; i++)  
A[i][j] = 0;
```

Avec des pages de 128 mots, quel est le nombre de défauts de pages provoqués par cette exécution ?

Comment pourrait-on le réduire ?

Exercice 4 : Segmentation paginée

On considère une mémoire segmentée paginée pour laquelle les cases en mémoire centrale sont de 4 Ko. La mémoire centrale compte au total 15 cases numérotées de 1 à 15. Dans ce contexte, on considère deux processus A et B.

Le Processus A a un espace d'adressage composé de trois segments S1A, S2A et S3A qui sont respectivement de 8 Ko, 12 Ko et 4 Ko.

Le processus B a un espace d'adressage composé de deux segments S1B et S2B qui sont respectivement de 16 Ko et 8 Ko.

Pour le processus A, seules les pages 1 et 2 du segment S1A, la page 2 du segment S2A et la page 1 du segment S3A sont chargées en mémoire centrale respectivement dans les cases 4, 5, 10 et 6.

Pour le processus B, seules les pages 2 et 3 du segment S1B et la page 1 du segment S2B sont chargées en mémoire centrale respectivement dans les cases 11, 2 et 15.

Représentez sur un dessin les structures allouées (table des segments, tables des pages) et la mémoire centrale correspondant à l'allocation décrite.

Si 4098 et 12292 sont des adresses linéaires pour A, déterminez les adresses virtuelles et réelles correspondantes.

Même question avec 16389 pour A et 8212 pour B.

Exercice 5 : Une modélisation de la multiprogrammation

On suppose un système multiprogrammé à partitions fixes de 200 Ko. Le système ainsi que chaque programme tient dans une partition. On suppose que chaque processus passe en moyenne une fraction p de son temps à attendre ses entrées-sorties.

1. Calculer le pourcentage moyen d'utilisation du CPU avec n processus en mémoire.
2. Dessiner la courbe d'utilisation du CPU en fonction du nombre de processus en mémoire en supposant que les processus passent 20% de leur temps à attendre les entrées-sorties. Même chose pour 50 % et 80 %. Un ratio de 80 % d'attente est-il raisonnable ?
3. Un ordinateur de ce type possède 1 Mo de mémoire. Quel est le pourcentage moyen d'utilisation du CPU avec un ratio de 80 % ?
4. Que se passe-t-il avec 2 Mo de mémoire, 3 Mo de mémoire ? Que conclure ?