Systèmes d'Exploitation 1

Série TD N°7
Gestion de la mémoire : Allocation de mémoire non contiguë (3)

Exercice 1: 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 2: Pagination

Soit un système d'exploitation gérant la mémoire au moyen du mécanisme de pagination (gestion par blocs de taille fixe). La machine dispose de 600 unités de mémoire centrale (RAM). Le système d'exploitation utilise des pages de 200 unités, il est configuré de façon à pouvoir gérer 14 pages de swap (mémoire virtuelle simulée au moyen du disque dur).

Le système est soumis aux demandes d'accès décrites dans le tableau suivant.

1. Suivez le fonctionnement du système en supposant que les pages de la mémoire centrale sont vidées dans l'ordre où elles ont été chargées (stratégie FIFO). Complétez le tableau sachant que dans un système de gestion de mémoire par page, une adresse mémoire est convertie en une adresse de page et un déplacement (offset) à l'intérieur de la page. Simulez le fonctionnement en utilisant la première colonne de l'annexe 1. Notez a) le nombre de défauts de pages (chargement d'une page de la mémoire virtuelle vers la mémoire centrale—c'est-à-dire non présence en mémoire centrale de la page demandée) jusqu'à la douzième demande (comprise) et b) le nombre total de défauts de pages.

2. Suivez le fonctionnement du système avec une stratégie qui décharge prioritairement la page utilisée il y a le plus longtemps. Utilisez la deuxième colonne de l'annexe 1. Notez le nombre de défauts de pages (mêmes conditions que la question 1). Comparez les résultats obtenus. Que s'est-il passé à partir de la douzième demande ?

3. La dernière simulation utilise une stratégie de déchargement qui sélectionne en priorité les pages ayant le moins servi. En cas d'égalité, on utilise la stratégie de la question 2. Utiliser la colonne 3 de l'annexe 1 pour effectuer la simulation. Afin de maintenir un compteur d'utilisation des pages, le système maintient une liste (tableau horizontal colonne3). Comptabilisez à nouveau les défauts de pages, comparez-les avec les résultats précédents. Qu'en déduire ? Cette stratégie est rarement utilisée, pourquoi ?

4. Supposons que les demandes sont mises en attente et que l'ordre de traitement n'est pas fondamental. Quelle serait la stratégie optimale ?

5. Refaire la question 2 en considérant une taille de page de 300 unités. Expliquez les résultats. Utilisez l'annexe 2 pour simuler le système.

6. Que pensez vous d'une taille de page de 10 unités (justifier) ?

Demande	**Conversions d'adresse= (page, offset)**			**Mémoire Centrale**	
	Adresse	**Page**	**Offset**	**Cadre**	**Adresse**
1	657	3	57	0	57
2	523	2	123	1	323
3	170	0	170	2	570
4	725				
5	1133				
6	145				
7	190				
8	658				
9	573				
10	56				
11	598				
12	888				
13	1134				
14	170				
15	472				