Systèmes d'exploitation
TD 5: Gestion de la mémoire, pagination.

Frédéric Vivien, bureau 306, e-mail: fvivien

11 mars 1997

Adresses virtuelles et physiques

Une mémoire virtuelle a une taille de page de 1024 mots, huit pages virtuelles et 4 pages physiques. Sa table des pages est la suivante:

Page virtuelle	Page réelle
0	3
1	1
2	pas en mémoire physique
3	pas en mémoire physique
4	2
5	pas en mémoire physique
6	0
7	pas en mémoire physique

1. Donnez la liste des adresses qui provoqueront un défaut de page.
2. Quelles sont les adresses physiques de 0, 3728, 1023, 1024, 1025, 7800 et 4096?

Importance de la taille de la mémoire physique

La mémoire centrale comprend 25 cadres. 5 processus s'exécutent simultanément. Chaque processus dispose de 5 cadres et produit la liste des références (en numéro de page) suivante:
processus i: $0 + 5 \times i$, $1 + 5 \times i$, $2 + 5 \times i$, $3 + 5 \times i$, $4 + 5 \times i$, $0 + 5 \times i$, $1 + 5 \times i$, ... pour «i» variant de 0 à 4.

L'algorithme de remplacement utilisé est LRU (*least recently used*) et la mémoire est initialement vide.

1. Quel est le nombre de défauts de page provoqués par cette exécution?
2. On introduit un sixième processus, analogue aux cinq autres quant à la suite de références produites (pour i égal à 5). Chaque processus dispose maintenant de 4 cadres. Calculer le taux de défauts de page.

Importance de la taille de la mémoire physique (suite)

On considère un programme qui a cinq pages virtuelles numérotées de 0 à 4. Les pages sont référencées dans l'ordre suivant: 0, 1, 2, 3, 0, 1, 4, 0, 1, 2, 3, 4. On suppose que l'algorithme de remplacement des pages est l'algorithme FIFO.

Combien de défauts de pages seront engendrés par cette séquence si le programme dispose de 3 cadres? de 4? Vaut-il mieux que la mémoire physique soit de grande ou de petite taille?

Comparaison de stratégies de remplacement

Un programme a un espace virtuel de 512 mots. On considère la suite d'adresses virtuelles: 34, 123, 145, 510, 456, 345, 412, 10, 14, 12, 234, 236, 412.

1. Donnez la suite des numéros des pages référencées, sachant qu'elles comportent 100 mots.
2. Le programme dispose de 300 mots en mémoire centrale. Calculez le taux de défauts de page pour les algorithmes FIFO, LRU, et pour l'optimal.

Mémoire associative

Un ordinateur dont les processus ont 1024 pages dans leur espace d'adressage garde toute sa table des pages en mémoire. Il faut 500 nano-secondes pour lire un mot dans la table des pages. Pour réduire ce temps, l'ordinateur possède une mémoire associative qui contient 32 paires (page virtuelle, cadre). La recherche dans cette mémoire associative ne prend que 100 nano-secondes. Quel doit être le pourcentage de pages trouvées dans la mémoire associative pour que le temps de recherche global soit de de 200 nano-secondes?

Importance de la taille de la mémoire vive

Supposez qu'un ordinateur a 2 Mo de mémoire vive dont 512 Ko sont occupés par le système d'exploitation et où chaque programme occupe 512 Ko. Si tous les programmes attendent des entrées-sorties en moyenne pendant 60% du temps, quelle serait l'amélioration de rendement si on ajoutait 1 Mo de mémoire?

Stratégies d'allocations du swap

Considérez un système à va-et-vient (*swap*) dont la mémoire contient des zones libres (ordonnées en fonction des adresses mémoire) de 10 Ko, 4 Ko, 20 Ko, 18, Ko, 7 Ko, 9 Ko, 12 Ko et 15 Ko. Quelles zones l'algorithme sélectionne-t-il pour les demandes d'allocations de segments de 12 Ko, 10 Ko, et 9 Ko pour les algorithmes de première zone libre (*first fit)*, meilleur ajustement (*best fit*), plus grand résidu (*worst fit*) et de la zone libre suivante (*next fit*)?