

TD2 INF104

Pagination de la mémoire

Rappels : On appelle « bloc libre » une page libre en mémoire physique. Lorsqu'une page virtuelle référencée par un processus n'est pas présente en mémoire (elle n'a pas été chargée précédemment sur un bloc libre), il y a *défaut de page*. Pour traiter ce défaut de page le système d'exploitation cherche un bloc libre en mémoire, et s'il n'y en a aucun, il applique un algorithme de remplacement de page (FIFO, LRU).

Exercice 1

On va voir l'évolution du contenu des blocs libres en mémoire au cours de l'exécution d'un processus dont l'espace d'adressage (i.e. l'espace mémoire utilisée par ce processus) fait 5Ko, la taille des pages étant de 1Ko.

Il y a 3 blocs libres (de 1 Ko) en mémoire : B1, B2, et B3.

La suite des numéros de pages virtuelles référencées par ce processus est la suivante:

4, 0, 1, 2, 0, 1, 0, 3, 1, 4, 1, 3.

Notation : NPV = Numéro De Page virtuelle

Question 1 : Représenter (canevas ci-dessous) l'évolution des entrées de la table de pages, c'est-à-dire dans quel bloc se trouve la page référencée. On initialisera l'utilisation des blocs dans l'ordre suivant : B1, B2, puis B3. L'algorithme de remplacement est FIFO.

NPV référencées	4	0	1	2	0	1	0	3	1	4	1	3
0												
1												
2												
3												
4	B1											

Question 2 : Reprendre l'exercice 1 en supposant que l'algorithme de remplacement est LRU.

Question 3 : Combien comptez vous de défauts de page dans chaque cas (FIFO et LRU) traités précédemment.

Exercice 2

En supposant la table des pages suivante :

NPV référéncées	Numéro de bloc
0	4
1	2
2	5
3	9
4	3
5	6
6	1

Question 1 : On considère qu'un bloc permet de stocker 1 Kilo-octet (1024 octets), quelle est l'adresse physique correspondant à l'adresse virtuelle 0x042F (en format hexadecimal)?

Indication : $2^{10}=1024$

Question 2 : On considère maintenant qu'un bloc permet de stocker 0.5 Kilo-octet (512 octets), quelle est l'adresse virtuelle correspondant à l'adresse physique 0x072E ?

Indication : $2^9=512$

Ordonnancement

On va suivre l'exécution de deux processus, P1 et P2, dans des conditions d'ordonnancement différentes.

Dans les scenarios donnés ci-dessous, S0 et S1 sont des verrous.

Les opérations sur ces verrous se font en temps nul : quand un processus travaille pendant 1s puis fait une opération sur un verrou, celle-ci se fait pendant cette seconde.

Scénario pour P1 :	Scénario pour P2 :
Au temps 0s, il démarre il travaille pendant 2s, P(S1), il travaille pendant 2s, P(S0), il travaille pendant 1s, V(S0), il travaille pendant 1s, V(S1), il travaille pendant 3s, fin	Au temps 1s, il démarre il travaille pendant 1s, P(S0), il travaille pendant 1s, P(S1), il travaille pendant 1s, sleep (3s); il travaille pendant 1s, V(S1), il travaille pendant 1s, V(S0), fin

Exercice 1 :

Représenter sous la forme d'un chronogramme l'ordonnancement de P1 et P2 dans le cas suivant:

- l'ordonnancement se fait suivant un algorithme du type tourniquet (quantum de temps de 1s)
- en donnant des priorités égales aux deux processus, quand un processus démarre, il est mis au début de la file d'attente des prêts.

Exercice 2 :

Représenter sous la forme d'un chronogramme l'ordonnancement de P1 et P2 dans le cas suivant:

- politique d'ordonnancement : le plus prioritaire d'abord, avec préemption, sans quantum de temps,
- la priorité du processus P1 est inférieure à celle du processus P2.