

**Contrôle continu - 19/11/2021****Durée : 1h30****Une feuille A4 R/V manuscrite (cours et/ou td) autorisée.****Barème donné à titre indicatif : Ex.1 : 4 - Ex. 2 : 7 - Ex. 3 : 5 - Ex 4 : 4**

**Exercice 1.** Dans le cadre d'une stratégie FIFO, pour un processus accédant à ses pages selon la séquence suivante

$$0, 1, 2, 3, 0, 1, 4, 0, 1, 2, 3, 4$$

pour un nombre de cadres physiques égal à 3 puis égal à 4, donner l'état de la mémoire centrale à chaque défaut de page, ainsi que l'état de la liste chaînée des pages présentes en mémoire. On suppose qu'initialement, la mémoire est vide. Combien y a-t-il de défauts de page dans chaque cas ? Qu'observez-vous ?

**Exercice 2.** On considère un système de mémoire virtuelle ayant les caractéristiques suivantes :

- la taille d'une page (et d'un cadre) est de 1 Ko
- la taille de la mémoire centrale est de 32 Mo
- la taille de mémoire virtuelle est de 512 Mo
- on utilise la technique de la segmentation paginée : l'espace d'adressage virtuel d'un processus est composé de segments contigus. Chaque segment peut contenir entre 1 et 128 pages. La numérotation des pages d'un segment est relative au segment.

1. Calculer le format d'une adresse virtuelle et d'une adresse physique en spécifiant le nombre de bits réservés pour chaque champ.
2. Un processus  $p$  a un segment de code de 9 Ko et un segment de données de 3 Ko. Dans l'espace virtuel de  $p$ , le segment de code est suivi du segment de données. Si le segment de code débute à l'adresse 0, à quelle adresse débute celui des données ?
3. Sachant que le segment de données de  $p$  est chargé totalement en mémoire centrale dans les cadres contigus 4096, 4097 et 4098, calculer l'adresse réelle qu'occupe en mémoire centrale une donnée qui se trouve à l'adresse logique 10728, relative au début de l'espace d'adressage. On demande un entier décimal.
4. Soit la séquence de références de pages suivante :

$$0, 1, 0, 1, 2, 3, 4, 2, 3, 4, 5, 6, 7, 8$$

effectuée par  $p$ . Les opérandes référés par les instructions dans les pages 0, 1 et 2 se trouvent dans la page  $0_d$  du segment de données ; les opérandes des instructions des pages 3, 4 et 5 sont dans la page  $1_d$  ; les opérandes des instructions des pages 6, 7 et 8 sont dans la page  $2_d$ . Supposons que toutes les instructions de  $p$  ont des opérandes qui réfèrent en mémoire. Au départ, 4 cadres contigus sont alloués pour le code du processus à l'adresse  $X$  et 2 cadres contigus sont alloués pour ses données à l'adresse  $Y$  ( $X$  et  $Y$  ne sont pas nécessairement contigües). Le chargement des pages est réalisé à la demande. De plus, aucun cadre supplémentaire n'est alloué à  $p$  durant son exécution.

- (a) Représenter l'état d'occupation de la mémoire centrale à chaque instant  $t_i$  (c'est-à-dire  $t_0, t_1, \dots$ ) où une nouvelle page est chargée en appliquant l'algorithme de remplacement de pages LRU.
- (b) Calculer le nombre de défauts de page généré par l'algorithme LRU. Ce nombre est-il optimal ?

**Exercice 3.** Un système qui implante la pagination à la demande dispose de 4 cadres physiques qui sont tous occupés, à un instant donné, par des pages de mémoire virtuelle. La table suivante donne, pour chaque case mémoire, le moment  $t_{\text{chargement}}$  du chargement de la page qu'elle contient, le temps  $t_{\text{dernier\_accès}}$  du dernier accès à cette page, et l'état des bits référence (R), modifié (M) et présence (P). Les temps sont donnés en tops d'horloge.

Case	$t_{\text{chargement}}$	$t_{\text{dernier\_accès}}$	R	M	P
0	126	270	0	0	1
1	230	255	1	0	1
2	110	260	1	1	1
3	180	275	1	1	1

Indiquer quelle page sera remplacée dans le cas d'un défaut de page si l'algorithme de remplacement de pages est :

1. NRU
2. FIFO
3. horloge

**Exercice 4.**

1. Quel est l'intérêt de la pagination à double niveau ?
2. Qu'est ce qu'une *commutation de contexte* ? Quand a-t-elle lieu ? Quelles sont les opérations qu'elle nécessite ?