EXAMEN UE LI324 – PRINCIPE DES SYSTEMES

LI324

JUIN 2010

2 HEURES – TOUT DOCUMENT PAPIER AUTORISE

BAREME DONNE A TITRE INDICATIF

I. SYNCHRONISATION (6,5 POINTS)

On considère *K* ressources partagées disponibles et *N* processus consommateurs avec N > K. Une ressource n'est consommée que par un processus consommateur (une fois consommée, une ressource n'est plus disponible).

Après avoir consommé une ressource un processus peut en consommer une autre. Cependant, s'il n'y a plus de ressources, le consommateur doit se bloquer, sauf le dernier. Celui-ci, au lieu de se bloquer, doit débloquer tous les autres avant de se terminer. Ceux-là en se réveillant doivent aussi se terminer.

Question I.1 (2 points)

Quels sont les sémaphores et variables partagées nécessaires à cette synchronisation. Vous préciserez les valeurs initiales.

```
Variables partagées :
int fin =0 ;   / *indique si fin ou non du programme */
int count =0 ; nombre de ressource + processus bloqué ;

Sémaphores :
SEM *mutex = CS(1) ;  / *acces à cont  et fin initialisé à 1 ;
SEM* Ressource= CS(K) ; / *contrôler le nombre de ressources disponibles ; initialisé à K */
```

Question I.2 (4,5 points)

Donnez le code des processus consommateurs, variables partagées ainsi que leur initialisation.

Observation : le type de la ressource partagée et son accès n'est pas important. Ajouter dans le code le commentaire : / *accès à la ressource */

```
void consommateur () {
  int i ;
```

```
while (true) {
   P(mutex) ;
    cont++ ;
    if (cont == K+N-1) {
       / *K ressources consommées e N-1 processus bloqués */
       fim =1 ;
        / *réveiller le N-1 autres processus */
        for (i=0 ;  i< N-1 ; i++)
          V(Ressource);
        V(mutex);
        exit (0);
    }
    V(mutex) ;
    P(Ressource) ; / *obtention d'un Ressource ou bloqué */

    P(mutex)
    if (fim)  {
      V(mutex);
       exit (0);
    }
    V(mutex) ;
  / *accès à la ressource */
}
```

II. REMPLACEMENT DE PAGES (5 POINTS)

L'algorithme NRU favorise le maintien en mémoire des pages récemment utilisées. La version de NRU proposée ici fonctionne de la manière suivante.

Deux bits sont associés à chaque page : un bit R de référencement, et un bit M de modification. Lorsqu'une page est utilisée, son bit R est positionné à 1 ; s'il s'agit d'un accès en écriture, alors son bit M est également positionné à 1. Une interruption horloge vient régulièrement remettre à 0 tous les bits R des pages chargées en mémoire. On obtient ainsi une hiérarchie des pages présentes en mémoire :

- Priorité 3 : référencée, modifiée
- Priorité 2 : référencée, non modifiée
- Priorité 1 : non référencée, modifiée
- Priorité 0 : non référencée, non modifiée

Lors d'un remplacement de page, la page victime est celle de moindre priorité. Si plusieurs pages tombent dans cette catégorie, c'est la moins récemment utilisée qui est déchargée.

On souhaite mettre cet algorithme en application sur la suite de références suivante :

`1W 0R 4R 2R 2W 1R 3R 4R 0R 1R 3R 4R 1W 2R 4R 0R`

Le programme qui effectue ces accès dispose de 3 cases mémoire. La référence 1W représente un accès en écriture à la page 1, tandis que la référence 0R représente un accès en lecture à la page 0. Les bits R sont remis à 0 tous les 4 référencements.

Question II.1 (4 points)

Remplissez le tableau suivant en indiquant les pages présentes en mémoire, ordonnées de la plus récemment à la plus anciennement utilisée. Vous indiquerez également leur priorité, en notant `X(P)` la page X de priorité P.

1W	
0R	
4R	
2R	
2W	
1R	
3R	
4R	
0R	
1R	
3R	
4R	
1W	
2R	
4R	
0R	

Correction :

1W	D	1(3)
0R	D	0(2) ; 1(3)
4R	D	4(2) ; 0(2) ; 1(3)
2R	D	2(2) ; 4(2) ; 1(3)
2W		2(3) ; 4(0) ; 1(1)
1R		1(3) ; 2(3) ; 4(0)

3R	D	3(2) ; 1(3) ; 2(3)
4R	D	4(2) ; 1(3) ; 2(3)
0R	D	0(2) ; 1(1) ; 2(1)
1R		1(3) ; 0(2) ; 2(1)
3R	D	3(2) ; 1(3) ; 0(2)
4R	D	4(2) ; 3(2) ; 1(3)
1W		1(3) ; 4(0) ; 3(0)
2R	D	2(2) ; 1(3) ; 4(0)
4R		4(2) ; 2(2) ; 1(3)
0R	D	0(2) ; 3(2) ; 1(3)

Question II.2 (1 point)

Combien de défauts de pages sont-ils causés par cette suite de références ?

Y a-t-il une page qui n'est jamais déchargée ? Si oui, laquelle ?

11 défauts de pages

La page 1 n'est jamais déchargée.

III. MEMOIRE (5,5 POINTS)

On considère une mémoire **simplement paginée** avec des pages de 512 mots.

Considérez la table des pages suivante :

No	Adresse physique	p	m	u	Droits (xrw)
0	3584	1	0	1	110
1	4096	1	1	1	011
2		0			
3		0			
4	7168	1	0	1	011

Question III.1 (2 points)

Quelle est la conséquence des accès suivants en mémoire paginée :

a. écriture à l'adresse <0, 284>

b. lecture à l'adresse <1, 614>

c. écriture à l'adresse <4, 168>

d. lecture à l'adresse <3, 233>

a. L'utilisateur ne possède pas les droits en écriture sur la page 0 => accès refusé

b. Dépassement de la taille de page => erreur

c. Accès autorisé et écriture de la donnée à l'adresse 7168+168 = 7336 ; bit m à 1

d. La page 3 n'est pas chargée en mémoire => défaut de page

On considère maintenant un processus A. Son code va des adresses virtuelles 0 à 1023 mots, sa pile de 1024 à 2047 et ses données de 2048 à 3072.

La page 1 du processus est chargée dans la case 10, la page 3 dans la case 20.

Question III.2 (1,5 point)

Donnez la table des pages du processus A. Vous préciserez, les bits de présence, les droits et les cases si la page est présente en mémoire.

<page, case, P, droits>

<0, -, 0, RX>

<1, 10, 1, RX>

<2,-,0,RW>

<3,20,1,RW>

<4, -, 0, RW>

<5,-,0, RW>

Question III.3 (1 point)

Que se passe-t-il lors d'un accès en lecture aux adresses 1600 et 500 ? Vous préciserez les actions faites par le processeur et éventuellement celles faites par le système.

a) 1600 :

page = 1600 div 512 = 3,

la page 3 est en mémoire, les droits sont correct

la MMU fait directement la traduction de l'adresse (le système n'intervient pas)

b) 500 :

Page = 500 div 512 = 0 ,

Page non présente la MMU lève un IT défaut de page, traitée par le système

Question III.4 (1 point)

Donnez l'adresse physique associée à l'adresse 1700

Page = 1700 div 512 : 3, Déplacement = 1700 mod 512 = 164

Adresse physique = 20 * 512 + 164 = 10404

IV. QUESTIONS DE COURS (3 POINTS)

Question IV.1. (1 point)

Dans une mémoire paginée quel est le rôle de la MMU ?

Traduction d'adresse, gestion de erreur et déclenchement des défauts de pages

Question IV.2. (2 points)

On considère un système de fichiers de type FAT, la configuration de la FAT et du répertoire courant est la suivante :

Répertoire

nom	débu	…
F1	5	
F2	7	

FAT

0	11
1	4
2	10
3	2
4	0
5	2
6	0
7	3
8	0
9	0
10	-1
11	-1

Quels sont les blocs composant le fichier F1 ?

Quel est le premier bloc libre du système de fichier ?

Y-a-t-il une incohérence entre les fichiers F1 et F2 ? Si oui laquelle ?

(a) : 5, 2, 10

(b) 4

(c) F2 partagent les blocs 2 et 10 avec le fichier F1