Université Paris Descartes
UFR de Mathématiques et Informatique

L2 – Systèmes d'Exploitation
5 mai 2017 – Durée 1h30

EXAMEN
(aucun document autorisé)

Exercice 1 : MÉMOIRE VIRTUELLE (6 pts - 25 minutes)

Soit une machine à mémoire segmentée paginée. Les pages et cases de mémoire font 512 mots. Chaque page possède des bits de présence et de droits.

Il y a au plus 256 segments et les segments sont limités à 256 pages. La table des segments et chaque table des pages d'un segment sont rangées chacune dans une case à raison d'un mot par page contenant toutes les informations utiles au matériel sur le segment ou la page correspondante.

Soit une tâche divisée en 3 segments : le segment 0 contient le code, le segment 1 la pile et le segment 2 les données. Le segment 0 a 512 mots, le segment 1 a 1500 mots et le segment 2 en a 2050. Les tables des pages des segments 0, 1 et 2 sont déjà chargées en mémoire.

Toutes les pages du segment 1 sont en lecture et écriture, toutes celles des segments 2 en lecture.

Seules les pages suivantes ont déjà été chargées : <0,0> en A=5120, <1,0> en B=1536, <2,4> en C=1024, <1,1> en D=2560.

1.1 Donner le contenu de la table des segments, celui des tables des pages et celui de la MC. **(2,5 pts)**
1.2 Quels droits faut-il donner au segment 0 ? **(0,5 pts)**
1.3 Y a-t-il de la fragmentation ? Si oui, quelle fragmentation, où (<s,p>) et de combien de mots ?
1.4 À quelle adresse réelle est le mot d'adresse virtuelle <1,550> et <2,2049> ? Justifiez.
1.5 Décrire brièvement ce qui se passe sur une tentative d'écriture en <2,750>. Justifiez.

550 - 512 = 38

PS: *Les multiples de 512*
1 : 512; 2 : 1024; 3 : 1536; 4 : 2048; 5 : 2560; 6: 3072; 7 : 3584; 8 : 4096; 9 : 4608; 10 : 5120.

Exercice 2 : Algorithmes de remplacement de pages (5 pts – 25 minutes)

Algorithme *LRU (Least Recently Used)*
Lors d'un défaut de page, on remplace la page la moins récemment utilisée.

On suppose qu'un processus fait référence à ses pages dans l'ordre suivant:

3 4 0 1 4 2 4 5 1 2 4 2 1 0 1

2.1 Dans le cas où on alloue au processus 3 cases en Mémoire Centrale (0x0A ; 0x0B ; 0x0C), donnez l'évolution de la table de pages et le nombre de défauts de pages.

2.2 Sachant que chaque page fait 1Ko, donner l'adresse physique linéaire (en décimal) des 3 pages qui se trouvent en mémoire à la fin de l'exécution du processus.

PS: *Les multiples de 1024*
1 : 1024; 2 : 2048; 3 : 3072; 4:4096; 5 : 5120; 6 : 6144; 7 : 7168; 8 : 8192; 9 : 9216 ;
10 : 10240 ; 11 : 11264 ; 12 : 12288.

Exercice 3 : SYNCHRONISATION (5 pts - 20 minutes)

Un ensemble de N avec N > 4 processus partagent 3 imprimantes LP0, LP1, LP2. Pour éviter de mélanger les lignes de sortie de chaque processus, le processus Pi doit réserver l'imprimante avant de l'utiliser. Pour connaître l'état de chaque imprimante il y a un vecteur global `int LP[3]`.
Il y a deux fonctions : `prendre()` et `liberer()` :

```
int LP[3];

int prendre() {
int i=0;
    while(i<3) {
        if(LP[i]==0){
            LP[i] = getpid();
            break;
        }
        ++i;
    }
    if (i==3) return -1;
    else return i;
}
```

```
int liberer(int i) {
    LP[i]=0;
}
```

Compléter `prendre()` et `liberer()` pour synchroniser l'accès aux imprimantes en utilisant 2 sémaphores : `s_mutex` et `s_lp`. Préciser l'initialisation du compteur pour chaque sémaphore.

Exercice 4 : Questions de cours (4 pts - 20 minutes)

Répondez brièvement.

4.1 Définir un *défaut de page*. Quelles sont les actions entreprises par le SE dans ce cas ?
4.2 Qu'est-ce qu'une *table de pages* ? Où se trouve la *table de pages* et à quoi sert-elle ?
4.3 Qu'est-ce qu'une *mémoire virtuelle* ? Quels sont ses avantages et inconvénients ?
4.4 A quoi servent et comment fonctionne les *appels système* `wait` et `exec` ?