

Partiel – 26 octobre 2015

N. Sabouret, R. Bonaque, M. Gleize

Nom :

Prénom :

Signature :

- *L'épreuve dure 1h30. Tous les documents sont autorisés.*
- *Le sujet comprend 4 exercices indépendants.*
- *Toutes vos réponses doivent être justifiées.*

Exercice 1 – Mémoire paginée (5 points)

On se place dans un système de mémoire de 4Go de mémoire géré de manière paginée avec des cadres de page de 4Ko. Chaque processus peut utiliser jusqu'à 64Mo de mémoire. Le système d'exploitation autorise jusqu'à 1024 processus.

1. Quelle est la taille (en bits) de l'adresse logique ? (0.5 point)

2. Quelle est la taille (en bits) de l'adresse physique ? (0.5 point)

3. Combien y a-t-il de cadres de page dans la RAM ? (0.5 point)

4. Combien chaque processus peut-il contenir de pages ? (0.5 point)

5. On suppose que la pagination se fait sur un seul niveau. Quelle quantité de mémoire est consommée par les tables des pages ? (1 point)

6. On suppose que la pagination se fait sur deux niveaux avec un répertoire sur 6 bits. Quelle quantité de mémoire est consommée par un processus qui utilise les plages d'adresses logiques suivantes ?

Remarque : les adresses sont représentée en hexadécimal sur 32 bits (les bits en trop sont mis à 0).

de 00 00 1B 07 à 00 00 2C 08
de 00 03 CA 00 à 00 0E 00 00
de 00 0D AB 10 à 00 0F B1 CC
de 00 2B 10 21 à 00 2B 3A 72

(2 points)

Exercice 2 page suivante...

Exercice 2 – Ordonnancement (8 points)

On considère les processus suivants, définis par leur durée (réelle ou estimée) et leur date d'arrivée :

Processus	P1	P2	P3	P4	P5
Durée	6	3	1	5	2
Date d'arrivée	0	1	2	4	8

- On suppose que l'ordonnancement est fait suivant l'algorithme du **plus court temps restant non-préemptif**. Indiquez dans chaque case du diagramme de Gantt ci-dessous quel processus est affecté à l'UC à chaque pas de temps de l'exécution, puis indiquez en dessous le temps d'attente moyen. (1 point)

Date	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17
UC																	

Temps moyen et explications:

- On suppose que l'ordonnancement est fait suivant l'algorithme du **plus court temps restant préemptif**. Indiquez dans chaque case du diagramme de Gantt ci-dessous quel processus est affecté à l'UC à chaque pas de temps de l'exécution, puis indiquez en dessous le temps d'attente moyen. (1,5 points)

Date	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17
UC																	

Temps moyen et explications:

- On suppose que l'ordonnancement est fait suivant l'algorithme « round robin » avec un quantum de temps fixé à 2. Indiquez dans chaque case du diagramme de Gantt ci-dessous quel processus est affecté à l'UC à chaque pas de temps de l'exécution, puis indiquez en dessous le temps d'attente moyen. (1,5 points)

Date	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17
UC																	

Temps moyen et explications:

4. On définit ainsi un algorithme d'ordonnancement à deux niveaux de priorité :
- Le niveau 0 obéit à un ordonnancement « round robin », quantum 2 : entre eux, les processus de priorité 0 suivent cet ordonnancement.
 - Le niveau 1 obéit à un ordonnancement « plus court d'abord » non préemptif (en cas d'égalité, le plus ancien dans la file obtient le processeur)
 - Le niveau 0 a priorité sur le niveau 1 : tous les processus de priorité 0 sont toujours prioritaires sur ceux de priorité 1, et ce de manière préemptive.

Les priorités de nos processus sont définis de la manière suivante :

P1 durée : 6, date 0, **priorité 1**

P2 durée : 3, date 1, **priorité 0**

P3 durée : 1, date 2, **priorité 1**

P4 durée : 5, date 4, **priorité 1**

P5 durée : 2, date 8, **priorité 0**

Indiquez dans chaque case du diagramme de Gantt ci-dessous quel processus est affecté à l'UC à chaque pas de temps de l'exécution, puis indiquez en dessous le temps d'attente moyen. (1,5 points)

Date	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17
UC																	

Temps moyen et explications:

5. Quel est le meilleur algorithme suivant le critère du temps d'attente moyen ? (0,5 point)

6. Quel est le meilleur algorithme suivant le critère du temps d'attente min-max ? (0,5 point)

7. Quel est le meilleur algorithme parmi les trois algorithmes préemptifs que nous avons proposés (plus court d'abord préemptif, round-robin et algorithme à deux niveaux de priorité) pour le critère du **temps de rotation**? (1,5 points)

Exercice 3 page suivante...

Exercice 3 – Synchronisation (4 points)

On propose les fonctions suivantes, écrites en C, pour implémenter un MUTEX sur un système à deux processus :

```
int[] sc = {0,0};
void entrer_section_critique(int id) {
    sc[id]=1;
    while(sc[1-id])
        ;
}
void sortir_section_critique(int id) {
    sc[id]=0;
}
```

1. Donner un exemple d'exécution à deux processus pour lequel il se produit un inter blocage. (1 point)

2. Récrivez les fonctions `entrer_section_critique` et `sortir_section_critique` de telle sorte qu'il n'y ait plus d'interblocage possible. (1 point)

3. Quelle propriété parmi l'exclusion mutuelle, le déroulement et la vivacité n'est pas vérifiée par le mécanisme que vous avez proposé en question 2 ? (0.5 point)

4. Nous souhaitons écrire un nouveau mécanisme « équitable » qui garantit les trois propriétés pour un système à deux processus. En plus du MUTEX, de quoi avons-nous besoin ? (0.5 point)

5. Écrivez en langage C les fonctions de votre mécanisme d'exclusion équitable. (1 point)

Exercice 4 – Question de cours (3 points)

1. Quelle est la différence entre la multiprogrammation et le temps partagé (1 point)

2. Qu'est-ce qui permet, dans un OS, de mettre en œuvre la multiprogrammation et le temps partagé ? (1 point)

3. Quelle est la fonction de l'édition de liens ? (1 point)