	Examen de rattrapage	
T. Gherbi - J.A. Lorenzo – S. Yassa	Système d'exploitation	
ING1-GI-GM	Année 2017–2018	

Modalités

- Durée : 2 heures.
- Vous devez rédiger votre copie à l'aide d'un stylo à encre exclusivement.
- Toutes vos affaires (sacs, vestes, trousse, etc.) doivent être placées à l'avant de la salle.
- Une seule feuille manuscrite (pas de photocopies) est autorisée.
- Aucune question ne peut être posée aux enseignants, posez des hypothèses en cas de doute.
- Aucune machine électronique ne doit se trouver sur vous ou à proximité, même éteinte.
- Aucune sortie n'est autorisée avant une durée incompressible d'une heure.
- Aucun déplacement n'est autorisé.
- Aucun échange, de quelque nature que ce soit, n'est possible.

Exercice 1 : Programmation de processus (5 points = 3 + 2)

1. Écrivez un programme C qui crée 2 fils. Chaque processus doit afficher son PID à l'écran. Le père doit attendre la fin du 1^{er} fils. Lorsque le 1^{er} fils se termine, il envoie un code de retour, qui doit être récupéré par le père et affiché à l'écran.
2. Exécutez le code suivant en imaginant des PID pour les différents processus créés et donnez les messages affichés.

```
#include<unistd.h>

int main() {

    printf("Hello %d\n",getpid());

    fork();

    fork();

    printf("Hi %d:%d\n",getpid(), getppid());

}
```

Réponse :

1.

```
#include <stdio.h>
#include <stdlib.h>
#include <sys/wait.h>

int main()
{
    int pid1, pid2;
    int r;

    printf("\tje suis le pere %d\n", getpid());

    if ((pid1=fork()) == 0)
    {
        // Fils 1
        printf("\tje suis fils 1 de %d et mon pid %d\n", getppid(), getpid());
        exit(14);
    }
    else {
        if ((pid2=fork()) == 0)
        {
            // Fils 2
            printf("\tje suis fils 2 de %d et mon pid %d\n", getppid(), getpid());

        }
        else {
            // Père
            waitpid(pid1,&r, 0);
            printf("\tje suis le père %d de %d (mon 1er fils), mort pour : (%d)\n", getpid(), pid1, r);
        }
    }
}
```

2. Par exemple :

Coucou 7011

Hi 7013:7012

Hi 7012:7011

Hi 7014:7011

Hi 7011:5668

Exercice 2 : Gestion des fichiers (3 points)

Supposons un nœud d'information (i-node) en Unix (contenant 10 adresses directes et 3 adresses indirectes de blocs) qui représente un fichier f.

Quel est le nombre et le type d'adresses (directe, indirecte de niveau 1, indirecte de niveau 2, indirecte de niveau 3) nécessaires si la taille du fichier f est de 56320 octets, le bloc occupe 512 octets et l'adresse du bloc est donnée sur 16 bits ?

Réponse :

512 octets / 2 octets (16 bits) = 256 adresses de blocs dans un bloc.

10 adresses directes permettent d'atteindre 10 blocs de données

1 adresse indirecte de 1^{er} niveau permet d'avoir 256 adresses indirectes de 1^{er} niveau qui permettent d'atteindre 256 blocs de données

Taille du fichier = 56320 octets = $512 \times 110 = 512 \times 10 + 512 \times 100 \Rightarrow$ le fichier utilise donc les 10 adresses directes + 100 adresses des 256 adresses indirectes du 1^{er} niveau.

Exercice 3 : Ordonnanceur (5 points = $[1 + 0.25 + 0.5 + 0.25 + 0.25] \times 2$)

Soient cinq processus prêts A, B, C, D et E ; tel que : A arrive en premier, B arrive 2 unités de temps après A, C arrive 1 unité de temps après B, D arrive 2 unités de temps après C et E arrive 3 unités de temps après D. Les temps nécessaires pour l'exécution des processus A, B, C, D et E sont respectivement 8, 4, 2, 5 et 3 unités de temps. Le temps de commutation est supposé nul.

Processus	Temps d'exécution	Temps d'arrivée
A	8	0
B	4	2
C	2	3
D	5	5
E	3	8

Représentez sur un axe de temps horizontal, l'exécution de ces processus puis calculez :

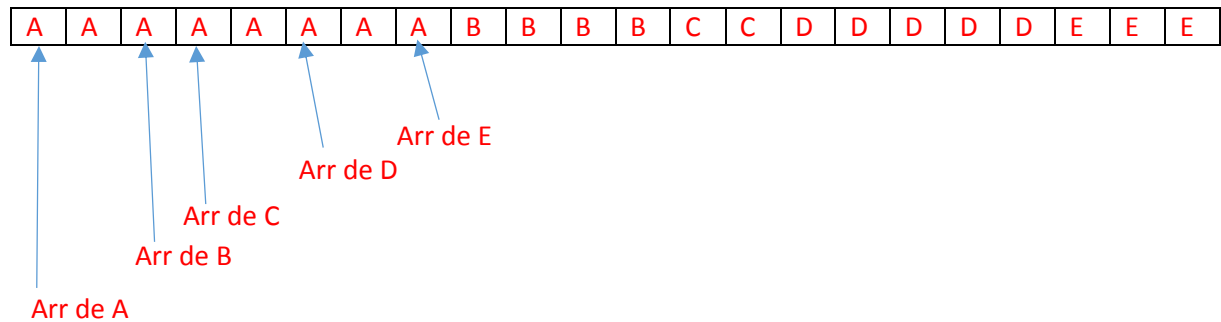
- le temps de séjour de chaque processus.
- le temps moyen de séjour.
- le temps d'attente : temps de séjour - temps d'exécution du travail.
- le temps moyen d'attente.
- le nombre de changements de contexte (Note : un changement de contexte se produit à chaque fois qu'un processus acquiert le processeur) en utilisant les techniques :

1. FCFS (First Come First Served)

2. SFJ (Shortest Job First)

Réponse :

1. FCFS :



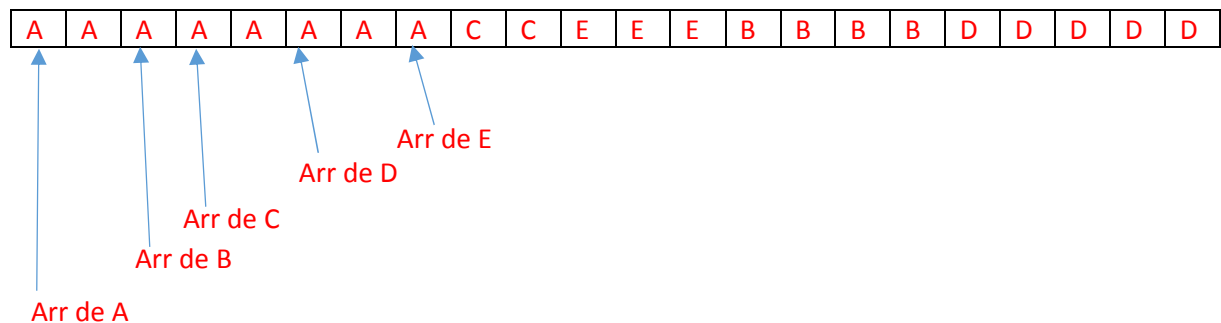
Processus	Temps de séjour	Temps d'attente
A	8	8-8=0
B	8-2+4=10	10-4=6
C	8-3+4+2=11	11-2=9
D	8-5+4+2+5=14	14-5=9
E	8-8+4+2+5+3=14	14-3=11

Temps moyen de séjour = $(8+10+11+14+14)/5 = 11.4$

Temps moyen d'attente = $(0+6+9+9+11)/5 = 7$

5 changements de contexte (en comptant le changement initial à l'arrivée de A)

2. SFJ :



Processus	Temps de séjour	Temps d'attente
A	8	8-8=0
C	8-3+2=7	7-2=5
E	8-8+2+3=5	5-3=2
B	8-2+2+3+4=15	15-4=11
D	8-5+2+3+4+5=17	17-5=12

Temps moyen de séjour = $(8+7+5+15+17)/5 = 10.4$

Temps moyen d'attente = $(0+5+2+11+12)/5 = 6$

5 changements de contexte (en comptant le changement initial à l'arrivée de A)

Exercice 4 : Mémoire virtuelle (3 points)

Supposons une machine avec 32 kOctets de mémoire, une plage d'adressage virtuelle de 16 bits et 8 kOctets de taille de page.

1. Dans l'adresse virtuelle, combien de bits faut-il utiliser pour le numéro de page virtuelle ? Quelle est la taille maximale de mémoire qui peut être gérée ? Justifiez.
2. Traduisez l'adresse virtuelle suivante en adresse réelle :

0 1 1 0 0 1 0 1 0 0 1 0 0 0 1 0

en utilisant la table de pages suivante :

	Physical address	Valid bit
7	00	0
6	00	0
5	11	0
4	00	0
3	01	1
2	00	0
1	10	1
0	00	1

3. Une fois l'adresse trouvée en mémoire physique, faudra-t-il remplacer la page ? Pourquoi ?

Réponse :

1. 8 koctets = $2^3 * 2^{10} = 2^{13}$ octets donc il faut 13 bits pour adresser les octets d'une page et il reste 3 bits (car $16-13=3$) pour adresser 8 pages. Par conséquent, la taille maximale de mémoire adressable est de $8 * 2^{13} = 8 * 2^3 * 2^{10} = 64$ Koctets
2. En utilisant les 3 bits du numéro de page virtuelle : 0 1 1 = page 3 (avec 01 en physical address et 1 en valid bit). Donc l'adresse sera 0 1 0 0 1 0 1 0 0 1 0 0 0 1 0 (NB : $2^{15} = 32$ Koctets de mémoire physique).
3. Non, parce que le valid-bit est à 1.

Questions courtes (4 points)

1. Quels sont les avantages et inconvénients de la méthode d'allocation contiguë de la mémoire secondaire ? A quel type de média est-elle adaptée?
Réponse : chap2 - page 16
2. Où se situe le MBR et que contient-il?
Réponse : secteur 0 (first-stage boot loader + table des partitions), voir chap3 – pages 5 et 6
3. Quelle est la différence entre apt-get et apt-get upgrade?
Réponse : chap3 - page 23
4. Qu'est-ce qu'un processus orphelin et qu'est-ce qu'un processus zombie?
Réponse : Orphelin (le père meurt avant son fils), zombie (le fils meurt sans (ou avant) que le père ne récupère la raison de sa fin via wait)