





#### Test n°2

Processus, signaux, ordonnancement, mémoire, disques

#### Exercice 1 – Processus

Soit le programme suivant :

```
int main () {
  int i, x, f;
  x = 1;
  for (i = 0; i < 3; i++) {
    f = fork ();
    if (f == 0) {
        x = x * 2;
        printf ("pid %d : %d\n", getpid(), x);
    }
}
return 0;
}</pre>
```

Combien de lignes seront imprimées lors de l'exécution du programme ? Indiquez une des suites possibles. Par exemple :

```
pid 500 : 2
```

### Exercice 2 – Signaux

Soit le programme suivant :

```
1 int main (void)
2 {
3    if ( fork() ) wait (NULL);
4
5    printf ("[% d ] fini \ n " , getpid ());
6    return 0;
7 }
```

- 1) Expliquez le fonctionnement de ce programme.
- 2) En sachant qu'à la terminaison d'un fils le signal SIGCHLD est envoyé au père, réécrire ce programme sans utiliser l'appel système wait(). Par exemple, en utilisant l'appel système pause(), on pourra écrire une solution qui met le processus dans l'état endormi jusqu'à ce qu'un signal SIGCHLD soit reçu.

#### Exercice 3 – Ordonnancement

Soit trois processus pour lesquels la durée des calculs et la date à laquelle on attend au plus tard les résultats (échéance) sont données dans le tableau suivant :

Processus	Durée de traitement	Echéance
P1	4	5
P2	5	12
P3	2	7

1) Que donne la politique Round-Robin avec un quantum de 1 unité de temps, sachant que tous les processus ont la même priorité et qu'ils se trouvent dans l'ordre P1, P2, P3 dans la file d'attente des processus prêts ? Dessiner le chronogramme d'exécution et indiquer les temps de rotation, les temps d'attente et les rendements individuels des processus.

### Répondre sur la page 4 du sujet.

Les échéances sont-elles respectées ?

2) Même question en appliquant la politique d'ordonnancement SJF non préemptif.

Répondre sur la page 4 du sujet.

3) En utilisant un algorithme d'ordonnancement à priorités fixes, indiquez comment attribuer les priorités de façon à obtenir tous les résultats au plus tard à leur échéance ? Donner le nouvel ordonnancement produit par cette politique.

Répondre sur la page 4 du sujet.

#### Exercice 4 – Mémoire

Un processus veut accéder aux pages ci-dessous sur un ordinateur équipé de 3 cadres (numérotés de 0 à 2) et de 8 pages (numérotées de 0 à 7) : 2, 1, 0, 7, 1, 6, 1, 5, 7, 6, 2, 6, 7, 0, 7, 1, 0, 2, 1, 7.

Faites un schéma qui indique, à chaque accès à une page, l'état de la table des cadres, s'il y a un défaut de page, et dans ce cas, quelle est la page victime qui sera choisie pour être stockée sur disque dur lorsque l'algorithme de remplacement des pages est :

- 1) OPT (algorithme de Belady)
- 2) FIFO
- 3) LRU

## Exercice 5 – Disques

On considère un disque dur présentant 128 pistes numérotées en ordre croissant de l'intérieur vers l'extérieur de 0 à 127. On suppose que la tête de lecture/écriture se trouve actuellement placée à la verticale de la piste 15, que sa position précédente était sur la piste 29 et que des requêtes arrivent pour des accès aux pistes suivantes (dans cet ordre):

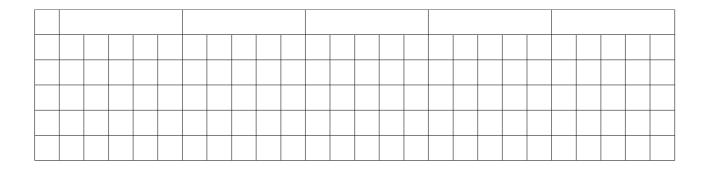
100, 30, 27, 55, 16, 122, 44, 63, 56.

Quel serait le déplacement total de la tête de lecture/écriture après que toutes les demandes aient été satisfaites si l'algorithme de planification des déplacements de la tête est :

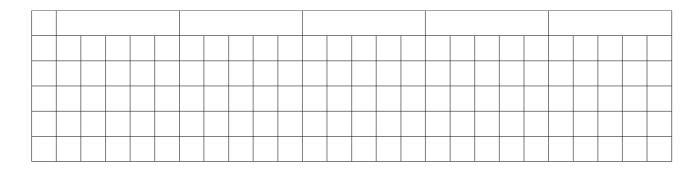
- 1) FIFO?
- 2) SSTF?

## Indiquez ci-dessous vos réponses aux questions de l'exercice 3 :

# Round-Robin (q=1)



# SJF non préemptif



## Ordonnancement à priorités

