

## QCM Moodle IF-3-SYS de contrôle continu, mars 2022

Pour chaque syscall de la liste ci-dessous, indiquez s'il se comporte comme un appel de fonction dont on ne revient jamais.

- `fork()`
- `exec()`
- `exit()`
- `wait()`

Pour chaque affirmation ci-dessous, indiquez si elle est correcte.

Veillez choisir au moins une réponse :

- Lors d'un appel système, le processus actif passe d'abord en "mode superviseur" avant de sauter vers le noyau
- Seul le code du noyau est exécuté en "mode superviseur"
- Lorsqu'il reçoit une interruption, le noyau fait passer la machine en "mode superviseur"
- Lorsqu'il reçoit une interruption, le CPU passe automatiquement en "mode superviseur"

Pour chaque affirmation ci-dessous, indiquez si elle est correcte.

- L'implémentation de `fork()` est située dans le noyau.
- Chaque appel à `fork()` crée deux nouveaux processus.
- Après un `fork()`, les deux processus partagent le même espace d'adressage.
- `fork()` renvoie le PID du parent dans le processus enfant.

Combien de processus seront impliqués dans l'exécution du code ci-dessous ?

```
void main()
{
    print("A");
    execl("./rebours", "5");
    print("A");
}
```

Réponse :

Si on exécute le programme suivant, combien de fois la lettre A sera-t-elle affichée ?

```
void main()
{
    r1 = fork();
    print("A");
    r2 = fork();
    print("A");
    if ( r1 != r2 )
    {
        print("A");
    }
}
```

Réponse :

Les affirmations ci-dessous portent sur le chapitre 2 (ordonnancement). Pour chaque proposition, indiquez si elle est correcte.

- Dans un système grand-public, l'ordonnanceur est en général de type "Round Robin"
- Dans un ordonnanceur Round Robin, plus on augmente le "quantum" plus on augmente la fréquence du "context switch"
- Par définition, un ordonnanceur préemptif ne peut pas poser de risque de famine (starvation).
- Par définition, la Ready Queue ne contient jamais de processus suspendus.

On s'intéresse à une machine avec P processeurs. Sur ce système, combien de processus *au maximum* peuvent être simultanément dans l'état "READY" ?

- P\*P
- P
- 2P
- indépendant de P

On s'intéresse dans cette question à un ordonnanceur avec trois niveaux de priorités: haute, moyenne, et basse. Entre deux niveaux distincts, l'ordonnancement est strictement préemptif. Au même niveau, l'ordonnanceur applique la stratégie FCFS. On suppose quatre tâches de calcul A à D aux caractéristiques suivantes:

nom	arrivée	durée	priorité
A	0	5	basse
B	1	3	moyenne
C	2	8	basse
D	3	6	haute

Au brouillon, dessinez un chronogramme pour représenter l'exécution de ces tâches sur le CPU. Indiquez ensuite l'instant de terminaison de chaque tâche:

A:  B:  C:  D:

On s'intéresse dans cette question à un ordonnanceur Round Robin avec un quantum de 3 unités de temps. On suppose trois tâches A, B et C avec les caractéristiques suivantes:

nom	arrivée	durée
A	0	8
B	2	4
C	4	8

Au brouillon, dessinez un chronogramme pour représenter l'exécution de ces tâches sur le CPU. Puis calculez le temps d'attente (waiting time) de chaque tâche et indiquez le total ci-dessous.

Réponse :