

QCM1: contrôle continu IF-3-SYS de mars 2025

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

- ☐ Le rôle du noyau est, entre autres, d'interpréter les commandes que tape l'utilisateur dans le terminal
- ☐ Le rôle du noyau est, entre autres, d'exécuter les instructions assembleur qui composent les applications userland.
- ☐ Le rôle du noyau est, entre autres, de réagir aux évènements matériels signalés par les périphériques.
- ☐ Le rôle du noyau est, entre autres, de virtualiser les ressources matérielles de la machine : processeur, mémoire, etc.

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

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

Pour chaque élément de la liste ci-dessous, indiquez s'il s'agit typiquement d'un "*processus*" tel que défini au chapitre 1.

- ☐ *Shell*
- ☐ Cycle de von Neumann
- ☐ Émulateur de terminal
- ☐ Ordonnanceur
- ☐ Routine de traitement d'interruption i.e. "*ISR*"
- ☐ *Ready Queue*
- ☐ Compilateur
- ☐ Minuteur système i.e. "*System timer*"

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

```
void main()
{
    printf("A\n");
    execl("/bin/ls", "/bin/ls", "..", NULL);
    printf("A\n");
}
```

Réponse :

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

```
main()
{
    int r1 = fork()
    int r2 = fork()
    if(r1 * r2 == 0)
    {
        fork()
    }
    print('A')
}
```

Réponse :

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

- ☐ indépendant de P
- ☐ P
- ☐ 2P
- ☐ P*P

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

- ☐ Lorsqu'il quitte l'état "BLOCKED", un processus peut se retrouver dans le noyau.
- ☐ Si l'ordonnanceur est de type "non-préemptif" alors les processus ne sont jamais suspendus.
- ☐ Si tous les processus sont CPU-bound, alors la "Ready Queue" sera presque tout le temps vide.
- ☐ Lorsque qu'un processus s'exécute pendant un temps dépassant son "quantum", le noyau le passe dans l'état "READY" pour le faire attendre.

Pour chaque stratégie d'ordonnancement ci-dessous, indiquez si elle risque de souffrir d'un problème de famine:

- ☐ MLFQ
- ☐ SRTF
- ☐ RR
- ☐ SJF

Vrai ou faux : dans notre TD2, partie 1 (exercices 1, 2, et 3) le temps de séjour d'une tâche est *toujours* égal à la somme de sa durée d'exécution et de son temps d'attente.

- ☐ Vrai
- ☐ Faux

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 :