

## QCM moodle IF-3-SYS du 25 mars 2021

Donnez la forme longue de l'abréviation "IRQ"

Réponse :

Pour chacun des mécanismes proposés dans la liste ci-dessous, indiquez si ce mécanisme est implémenté à l'intérieur du noyau.

Veillez choisir au moins une réponse :

- ☐ Ordonnanceur
- ☐ Shell
- ☐ Cycle de Von Neumann
- ☐ Fonction `printf()`
- ☐ Fonction `exec()`
- ☐ Mémoire cache
- ☐ Émulateur de terminal
- ☐ Memory Management Unit

Que peut-il arriver si un processus invoque l'appel système `wait()` alors qu'il n'a jamais appelé `fork()` auparavant ?

Veillez choisir au moins une réponse :

- ☐ Le noyau suspend l'exécution de notre processus jusqu'à ce qu'un autre processus se termine.
- ☐ C'est un cas d'erreur, donc le noyau tue définitivement notre processus.
- ☐ C'est un cas d'erreur, donc le noyau nous rend la main immédiatement.
- ☐ Le noyau suspend indéfiniment l'exécution de notre processus, il faudra le réveiller manuellement.

Combien de fois ce programme affiche-t-il la lettre "A" ?

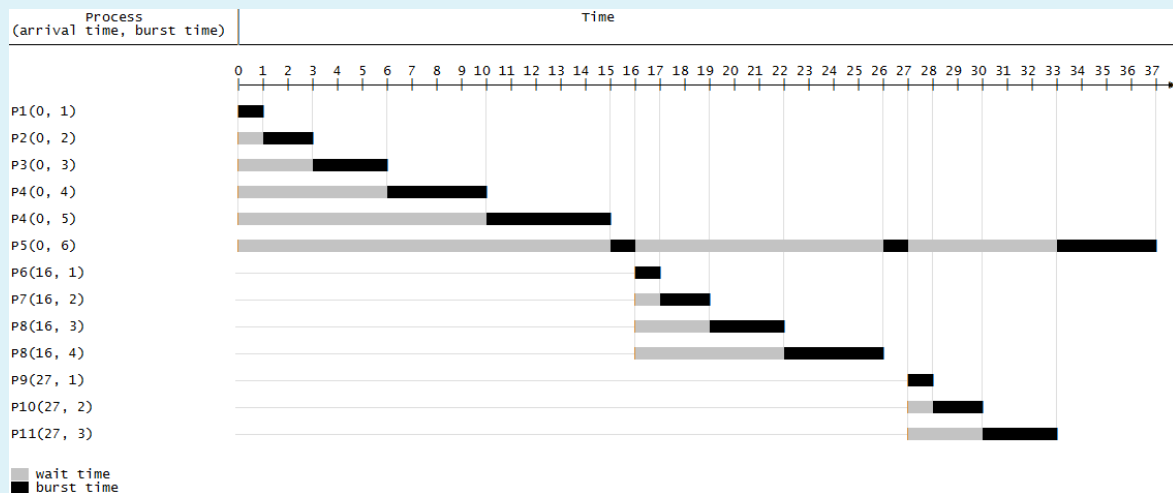
```
void main() {  
    int n = 2;  
    while(n > 0) {  
        fork();  
        n = n-1;  
        print('A');  
        fork();  
    }  
}
```

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"
- ☐ 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.
- ☐ Dans un ordonnanceur Round Robin, plus on augmente le "quantum" plus on augmente la fréquence du "context switch"

L'image ci-dessous représente l'exécution de 11 tâches P1 à P11 sur une machine monoprocesseur: le temps d'attente est représenté en gris, et le temps passé sur le CPU est représenté en noir. Chaque tâche est caractérisée par son instant d'arrivée et par sa durée d'exécution (indiqués entre les parenthèses). Mais quelle stratégie l'ordonnanceur applique-t-il dans cet exemple ? Pour chaque proposition ci-dessous, indiquez si elle vous paraît compatible avec ce chronogramme.



- ☐ Shortest Remaining Time First
- ☐ Shortest Job First
- ☐ Round Robin
- ☐ First Come First Served

On s'intéresse au même chronogramme que dans l'exercice ci-dessus. Indiquez quel est le *temps de séjour* de la tâche P5.

Réponse :

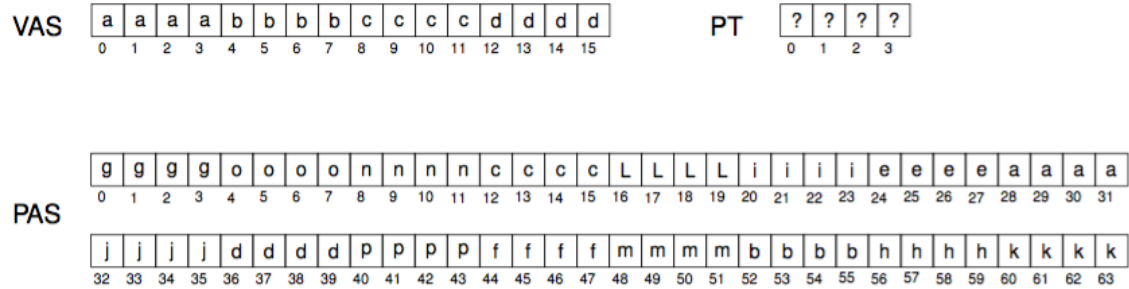
repose sur un mécanisme de traductions d'adresses: lors de chaque accès mémoire, l'adresse  émise par  est traduite par  en une adresse  avant d'être transmise à .

On s'intéresse à un système avec mémoire virtuelle. Le processeur travaille avec des adresses virtuelles de 16 bits, il peut donc adresser  $64 \text{ kio} = 65536 = 2^{16}$  octets. En supposant une taille de page de  $1 \text{ kio} = 1024$  octets, combien de bits sont nécessaires pour encoder un numéro de page ?

Réponse :

Le diagramme ci-dessous illustre le VAS d'un processus (16 octets) ainsi que le contenu de la mémoire principale (64 octets). La taille de page est de 4 octets. Remplissez la table de pages de ce processus (i.e. les points d'interrogation sur le diagramme) pour que la traduction d'adresses soit correctement réalisée:

PT 0:  1:  2:  3:



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

- ☐ En termes d'ordonnancement (cf chap 2) un appel système `sleep()` (cf chap 2) est équivalent à une "IO burst"
- ☐ À chaque appel système `fork()` (cf chap 1) le noyau crée une nouvelle table de pages (cf chap 3).
- ☐ En termes d'ordonnancement (cf chap 2) une "faute de page" (cf chap 3) est équivalente à une "IO burst"
- ☐ Les opérations de memory-mapped input/output (cf chap 1) utilisent des adresses virtuelles (cf chap 3)