2 - INTERRUPTION HORLOGE

1. INTERRUPTION

1.1

Dans quel mode s'exécute le traitement d'une interruption ?

Le traitement d'interruptions utilise la pile utilisateur ou la pile système ?

1.2

Indiquez les différents types d'interruption et leur priorité

1.3

Soit le scénario suivant avec un seul processus P : on suppose qu'à t=5ms un appel système est appelé par P, cet appel doit s'exécuter pendant 4 ms, à t=7ms une interruption horloge a lieu dont le traitement dure 1ms, à t=9ms une interruption disque est déclenchée dont le traitement dure 1ms.

Dessinez un diagramme temporel où sont indiqués les traitements de l'appel système, l'interruption disque et de l'interruption horloge, aussi que les instants où ils se terminent.

1.4

On considère maintenant que le traitement de l'interruption horloge dure 3ms.

Dessinez le nouveau diagramme temporel.

2. GESTION D'UNE INTERRUPTION HORLOGE

On considère un processeur qui dispose d'un registre temporisateur RH, décrémenté automatiquement toutes les micro-secondes *en mode usager* et *en mode système*. Lorsque sa valeur atteint 0, RH provoque une interruption (interruption horloge).

L'interruption horloge est utilisée à la fois pour tenir à jour l'heure universelle dans une variable globale walltime, pour instaurer un tour de rôle avec quantum d'exécution, pour comptabiliser le temps CT utilisé par une tâche et pour gérer les alarmes.

Le système gère une table `proc` des descripteurs de tâches : proc[i] contient l'ensemble des informations relatives à la tâche i. Le système dispose d'une variable ITE qui contient l'indice de la tâche élue.

On suppose qu'une tâche élue se voit toujours attribuer un quantum entier, même si elle avait précédemment perdu le processeur en ayant utilisé partiellement un quantum de temps (suite à une demande d'E/S par exemple).

Dans un premier temps, on considère que le quantum a une valeur fixée à QX microsecondes.

2.1.

Quelles informations doit maintenir le système pour chaque tâche de la table *proc* afin de gérer la commutation et le temps ?

2.2.

Donnez l'algorithme de la routine de traitement de l'interruption horloge. Cette routine provoque une nouvelle élection.

2.3.

Pourquoi est-il important que la durée de traitement de l'interruption horloge ne soit pas trop longue ?

2.4.

Quels sont les avantages et les inconvénients de toujours initialiser le registre Rtempo à la même valeur ?

QUANTUM ET TICK

Sous Unix, un tick correspond au passage à 0 du registre temporisateur. Certaines opérations de l'interruption horloge sont effectuées à chaque tick, alors que d'autres ne sont gérées que tous les N ticks, N dépendant de la tâche à traiter. Par exemple, la commutation de tâches est traitée toutes les 100 ms alors que l'intervalle entre 2 ticks est de 10 ms.

Par contre, la mise à jour de l'heure universelle, du temps utilisé par la tâche ou le test des alarmes à déclencher (réveil de tâche, réémission de paquets, …) est effectué à chaque tick.

3.1.

Modifiez la routine de traitement de l'interruption horloge pour gérer une commutation toutes les 100 ms avec un intervalle entre 2 ticks de 10 ms.

3.2.

Lors du positionnement d'une alarme, l'instant de déclenchement peut être choisi à la micro-seconde près. Quelle est, en réalité, la précision du déclenchement ?

3.3.

Comment peut-on mesurer le temps passé par la tâche en mode utilisateur et en mode système ?