

Semaine 3 – Questions sur les processus légers (*threads*)

Question 1 : Qu'est-ce qu'un "changement de contexte" ? Quand a-t-il lieu ? Quelles sont les opérations qui y sont faites ?

Réponse :

La commutation de contexte est une procédure réalisée par le système d'exploitation lorsque le processus en cours est interrompu, pour une raison quelconque (fin de quantum, demande d'E/S, préemption due à l'arrivée d'un processus plus prioritaire, ...).

Les opérations réalisées durant cette procédure sont :

- Sauvegarde du contexte du processus interrompu (compteur ordinal, contenu des registres et des variables, liste des fichiers ouverts, etc) => PCB.
- Restauration du contexte du processus chargé

Question 2 : Décrivez la procédure complète à appliquer lorsqu'une interruption arrive.

Réponse :

- Interruption du processus en cours
- Sauvegarde du contexte du processus interrompu.
- Exécution de la routine d'interruption correspondante.
- Restauration du contexte du processus interrompu
- Poursuite de l'exécution du processus interrompu

Question 3 : Pour lancer en parallèle plusieurs traitements d'une même application, vous avez le choix entre les appels système `fork()` et `pthread_create()`. Laquelle des deux possibilités choisir ? Pourquoi ?

Réponse :

`pthread_create ()` car le `fork()` consomme beaucoup d'espace (duplication de processus). Cependant, avec `pthread_create()` il faut faire **attention** au conflit d'accès aux objets partagés (*race condition*) comme nous l'avons vu dans l'exercice 5 avec la variable `a`. Les temps de communication inter processus (IPC) sont beaucoup plus longs que les temps de commutation inter threads au sein d'un même processus.

Question 4 : Pour quelles raisons les programmes de l'utilisateur utilisent des fonctions du système d'exploitation (espace noyau) plutôt que des fonctions du programmeur (espace utilisateur) afin d'accéder aux périphériques d'un ordinateur?

Réponse :

Pour simplifier la vie du programmeur.

Pour que le système d'exploitation gère l'état des tâches.

Pour que le système d'exploitation fasse une abstraction du matériel : les programmeurs n'ont pas à se soucier de la nature exacte des périphériques.

Pour que le système d'exploitation gère les accès concurrents de plusieurs processus pour le même périphérique.