Systèmes d'exploitation, 2ème année

Multi-Threading

Yves Stadler

Université de Lorraine - IUT de Metz

29 janvier 2012

Agenda

Plan du cours

- Différence entre Threads et Processus
- Utilisation des threads
- Synchronisation et ordonnancement
- Implémentation du multi-threading avec pthread

Thread

Définition

- Exécution de code en parallèle.
- Partage de données
- Être plus efficace que les threads (changement de contexte coûteux)

Différence entre Threads et Processus

Points communs

- Permet d'obtenir plusieurs instructions s'exécutant en parallèle.
- Doivent se partager les ressources

Différences

- Les threads sont comme des processus au niveau d'une application
- Les threads partages leur mémoire
- L'ordonnancement des threads peut être contrôlé
- Les changments de contextes sont plus efficaces

Comportement du threads

Partage

- Partage : du tas, des fonctions
- Pas de partage de la pile
- Partage du temps alloué au processus entre les threads

Utilisation des piles

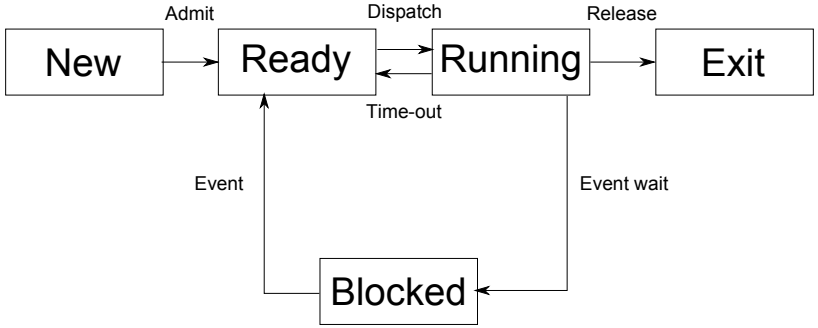
- Une pile applicative
- Une pile système
- Un thread est toujours créer par un autre thread (allocation + appel système)
- Les threads se détruisent eux-même (pas de retour de l'appel, ne peut pas désallouer sa mémoire applicative)

Thread et langages

Bibliothèques

- ADA : type task
- Java : class Thread
- C, C++ : pthread

État des threads



Ordonnancement

Remarque

- Un processus choisi comment répartir son temps entre les threads qui le composent.

Classe d'ordonancement

- Ancienneté (FIFO)
- Priorités (Fixes, variables)
- Quantum de temps durée max.
- Échéances
- Tourniquet
- Priorité et quantum