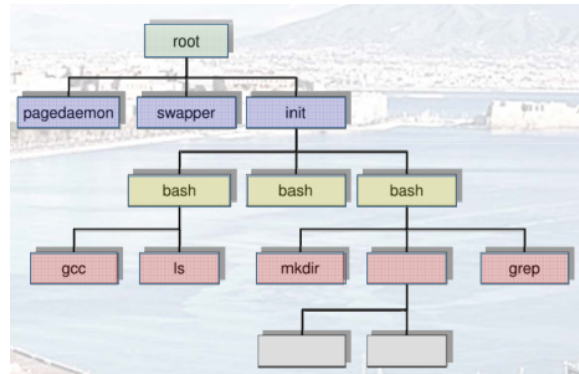




Thread



Processi e Thread

Processo (o task)

Programma in esecuzione con il suo spazio di indirizzamento ed il suo stato.

Un processo è ad esempio un'istanza di un qualsiasi programma. Viene anche definito processo pesante, in riferimento al contesto che si porta dietro

Thread

Singolo flusso sequenziale di controllo all'interno di un processo.

Un processo può contenere più thread. Tutti i thread di un processo condividono lo stesso spazio di indirizzamento. Viene anche definito processo leggero perché ha un contesto semplice

Parallelismo

Sistema parallelo

Architettura in cui sono presenti più unità di elaborazione (CPU) sulle quali sono in esecuzione processi e thread. In ogni istante di tempo posso avere più di un processo o più di un thread in esecuzione.

Sistema monoprocesso time-sliced

Vi è una sola unità di elaborazione. Il parallelismo di processi e thread viene simulato allocando a ciascuno una frazione del tempo di CPU (Time slice). Allo scadere di ogni unità di tempo si opera un cambio di contesto. I thread sono processi leggeri perché il cambio di contesto è veloce.

Preemption

Sistema non preemptive

Il cambio di contesto avviene quando il processo o il thread interrompe la propria esecuzione volontariamente

Sistema preemptive

Allo scadere del time slice il processo o thread viene forzatamente interrotto e viene operato il cambio di contesto

Motivazioni dei Thread

- Spesso risulta necessario dividere il programma anche in sotto-compiti indipendenti.
 - Un programma può avere diverse funzioni concorrenti:
 - operazioni ripetute nel tempo ad intervalli regolari
 - esecuzione di compiti laboriosi
 - attesa di messaggi da un altro programma
 - attesa di input da tastiera o dalla rete
 - L'alternativa al multithreading è il [polling](#), ma è scomodo da implementare e consuma risorse di CPU
-

Definizioni di Thread e Multithread

Un thread è un singolo flusso di controllo sequenziale all'interno di un programma

Ogni thread ha un inizio, una fine, una sequenza e, a ogni istante, un solo punto di esecuzione. Inoltre il contesto di un singolo thread prevede anche un proprio stack e un proprio Program Counter.

Vantaggi:

- Tempo di risposta breve
- Condivisione delle risorse (es: stesso spazio di indirizzi)
- Economia → è molto più economico creare un thread che un intero processo
- Uso di più unità di elaborazione → i thread possono essere eseguiti in parallelo

Thread a livello di utente → sono thread gestiti come uno strato separato dal nucleo del S.O. e realizzati tramite una libreria di funzioni senza alcun intervento del nucleo stesso.

Thread a livello di nucleo → sono thread gestiti direttamente dal sistema operativo

Modelli di programmazione dei thread

Modello da molti a uno → Fa corrispondere molti thread a livello di utente con un singolo thread a livello del nucleo

Modello da uno a uno → Mette in corrispondenza ciascun thread del livello di utente con un thread a livello del nucleo

Modello molti a molti → Mette in corrispondenza più thread a livello di utente con un numero minore o uguale di thread a livello di nucleo

← Capitolo 4: Processi

Capitolo 6: Scheduling della CPU

→
