

PCAD

Programmazione Concorrente e Algoritmi Distribuiti

DIBRIS

Laurea Triennale Informatica, a.a. 2021/22

Da Sequenziale a Concorrente

Programmazione Sequenziale

Il modello di esecuzione sequenziale prevede:

- Programmi formati da sequenze di istruzioni (quindi totalmente ordinate)
- Un'unica memoria (virtuale) per i dati dichiarati staticamente e creati dinamicamente
- L'esecuzione di una singola istruzione alla volta seguendo l'ordine del programma

Sistemi Concorrenti

Supponiamo che il nostro elaboratore fornisca una qualche forma di parallelismo

- Parallelismo reale: multi-processore, sistema distribuito
- Parallelismo virtuale: multi-programmazione con scheduling

Come possiamo sfruttare questa possibilità a livello di (linguaggio/sistema di) programmazione?

Programmazione Concorrente

- Rilassiamo il modello di calcolo in modo da prevedere l'esecuzione concorrente di un'insieme di istruzioni
- Nel nuovo modello le istruzioni di un programma possono essere parzialmente ordinate
- Esempi: Calcolare $(A + B) * (C + D)$, calcoli su matrici, schemi divide-et-impera, rendering di un'immagine sullo schermo

Vantaggi della Programmazione Concorrente

- Sfruttare le prestazioni di architetture multi-processore
- Migliorare la reattività delle interfacce grafiche
- Ottenere maggiore naturalezza per alcune tipologie di algoritmi e applicazioni (browser, server, giochi, simulatori)
- Migliorare design e comprensione del codice dei sistemi operativi

Modello di Esecuzione nei S.O.

Modello Concorrente dei Sistemi Operativi

- Un sistema operativo consiste in un gran numero di attività che vengono eseguite più o meno contemporaneamente dal processore e dai dispositivi di un elaboratore
- Senza un modello adeguato la coesistenza delle diverse attività sarebbe difficile da descrivere e realizzare
- Il modello che è stato realizzato a questo scopo prende il nome di **modello concorrente** ed è basato sul concetto di **processo** (programma in esecuzione)
- In tale modello i processi vengono eseguiti in parallelo con parallelismo reale (multiprocessori) o apparente (multiprogrammazione su una singola CPU)
- La concorrenza studia i problemi legati all'esecuzione parallela di processi

Processi e programmi

- Definizione di processo
 - È un'attività controllata da un programma che si svolge su un processore (o più semplicemente un programma in esecuzione)
- Un processo non è un programma:
 - Un programma specifica la sequenza di esecuzione di un'insieme di istruzioni ma non la distribuzione nel tempo dell'esecuzione
 - Un processo rappresenta il modo in cui un programma viene eseguito nel tempo

Processi e programmi

- Più processi possono eseguire lo stesso programma (ad es. più utenti possono usare un browser per il web, un singolo utente può eseguire varie istanze dello stesso programma)
- Ogni istanza viene considerata come un processo separato anche se possono condividere lo stesso codice mentre (in generale) i dati, l'immagine e lo stato rimangono separati

Processi e SO

- Il sistema operativo tiene traccia di tutti i processi del sistema tramite la **tabella dei processi**
- Le entry della tabella, Process Control Block (PCB), mantengono informazioni sui singoli processi:
- Lo scheduler gestisce l'avvicendamento dei processi in CPU
 - Decide quale processo deve essere in esecuzione ogni istante
 - interviene quando viene richiesta un'operazione di I/O e quando un'operazione di I/O termina, ma anche periodicamente per assicurare il buon funzionamento del sistema

Switch di Contesto

- Quando la CPU passa ad un altro processo, il sistema deve salvare lo stato del vecchio processo e caricare quello del nuovo processo.
- Lo stato del processo viene salvato nel PCB
- Il tempo di *context-switch* porta un certo overhead; il sistema non fa un lavoro utile mentre passa di contesto
- Può essere un collo di bottiglia per sistemi operativi ad alto parallelismo
- Il tempo impiegato per lo switch dipende dal supporto hardware (per il salvataggio dei valori dei registri, ecc.)