

Tecnologie e progettazione di sistemi
informatici e telecomunicazione -2

Lezione 4 -

Elaborazione sequenziale e concorrente

Itis L. Da Vinci – Borgomanero – AS 2018/19 – Docente Prof. Masetta

ELABORAZIONE SEQUENZIALE

Con il termine **elaborazione sequenziale** si intende l'esecuzione di un programma sequenziale che genera un processo sequenziale con un ordinamento totale alle azioni che vengono eseguite.

1. Le istruzioni vengono eseguite una alla volta
2. Le elaborazioni passano attraverso una **sequenza di stati**
3. Macchina di Von Neumann una sola CPU

Questo tipo di elaborazione in certi casi comporta uno spreco enorme di tempo ed uno scarso utilizzo delle risorse disponibili a causa dei tempi di attesa ed interruzioni.

Ad esempio i Server Web i Video Games a più giocatori necessitano di parallelizzazione delle attività.

PROGRAMMAZIONE CONCORRENTE

Con *programmazione concorrente* si indicano le tecniche e gli strumenti impiegati per descrivere il comportamento di più attività o processi che si intende far eseguire contemporaneamente in un sistema di calcolo (*processi paralleli*).

Siamo in una situazione di elaborazione contemporanea reale solo nel caso in cui l'esecutore sia dotato di una *architettura multiprocessore*, cioè con più processori che possono eseguire ciascuno un singolo programma: nei sistemi monoprocessori sappiamo che il *parallelismo* avviene solo virtualmente, grazie alla *multiprogrammazione*, e più processi evolvono "in parallelo" grazie al *quanto di tempo* che viene loro assegnato dalle politiche di scheduling del *sistema operativo*.

SISTEMA OPERATIVO E PROGRAMMAZIONE CONCORRENTE

Il **sistema operativo** è per eccellenza l'esempio più eclatante di **programmazione concorrente**: il suo compito è quello di assegnare le **risorse** hardware dell'elaboratore ai **processi** utente che ne fanno richiesta, cercando di massimizzarne l'efficienza nella loro utilizzazione.

Le attività del **sistema operativo** devono essere eseguite **concorrentemente** in modo da consentire l'esecuzione contemporanea di più programmi utente: ogni attività **interagisce** con le altre sia in **modo indiretto**, occupando delle risorse comuni, sia in **modo diretto**, scambiando informazioni in merito allo stato delle risorse e dei programmi di utente al fine di realizzare la multiprogrammazione.

Programmazione concorrente

SISTEMA CONCORRENTE

Per **sistema concorrente** intendiamo un sistema software implementato su vari tipi di hardware che "porta avanti" **contemporaneamente** una molteplicità di **attività diverse**, tra di loro correlate, che possono **cooperare** a un obiettivo comune oppure possono **competere** per utilizzare risorse condivise.

PROCESSO CONCORRENTE

Due **processi** si dicono **concorrenti** se la prima operazione di uno di essi ha inizio prima del completamento dell'ultima operazione dell'altro.

Programmazione concorrente

processi paralleli e grafo di precedenza

Nei **processi sequenziali** la sequenza degli eventi che costituisce il processo è *totalmente ordinata*

Un **grafo** che descrive l'ordine con cui le azioni (o gli eventi) si eseguono del tempo prende il nome di **grafo delle precedenze**.

Nei **processi paralleli**, invece, l'ordinamento non è completo, in quanto l'esecutore per alcune istruzioni "è libero" di scegliere quali iniziare prima senza che il risultato sia compromesso: possiamo affermare che nella **elaborazione parallela** l'esecuzione delle istruzioni segue un **ordinamento parziale**.

Programmazione concorrente

processi paralleli e grafo di precedenza

- **Processo sequenziale** => grafo delle precedenze che rappresenta una **lista ordinata**
- **Processo parallelo** => grafo **orientato aciclico** con percorsi alternativi che indicano la contemporaneità di esecuzione

Programmazione concorrente

Simbologia grafo di precedenza

Per la descrizione del **grafo delle precedenze** vengono utilizzati i seguenti simboli con i rispettivi significati:



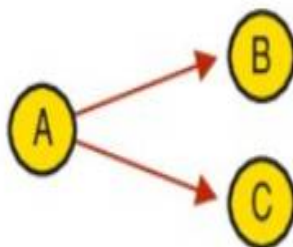
Attività (istruzione o sequenza di istruzioni)



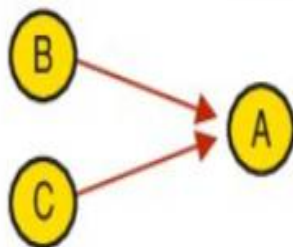
Vincolo di precedenza



Attività seriale (B inizia dopo il termine di A)



Inizio attività parallelizzabile (B e C possono essere eseguite in parallelo, dopo il termine dell'attività A)



Termine attività parallelizzabile (A può essere eseguita solo dopo il completamento sia di B che di C)

Programmazione concorrente

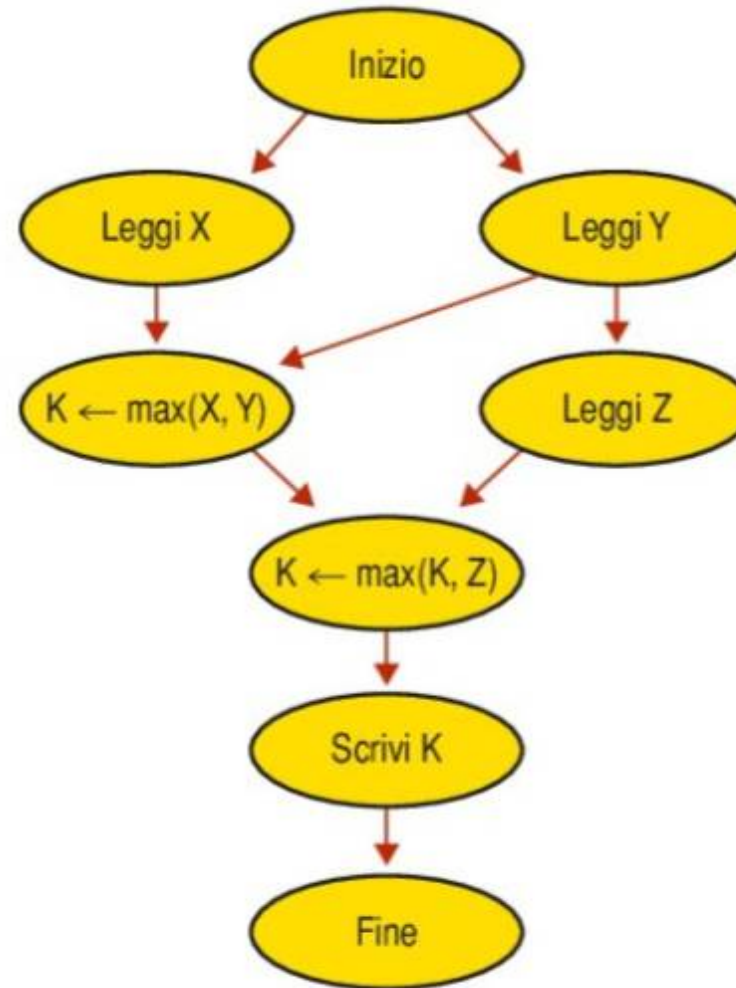
ESEMPIO *Grafo delle precedenze*

Vediamo un semplice esempio di un algoritmo che legge tre numeri e ne individua il maggiore. Per prima cosa scriviamo il codice sequenziale dell'algoritmo in *pseudocodifica* (algoritmo sequenziale):

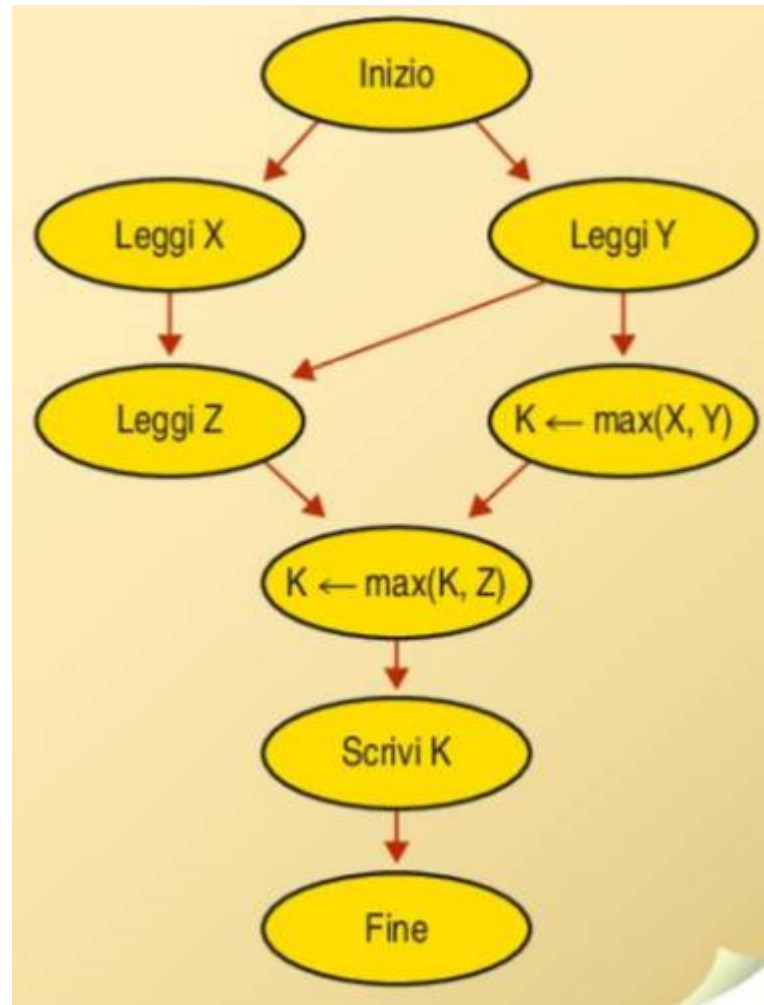
```
inizio
1. leggi X;
2. leggi Y;
3. leggi Z;
4.  $K \leftarrow \max(X;Y)$ ;
5.  $K \leftarrow \max(K;Z)$ ;
6. scrivi K;
fine
```

Grafi ad ordinamento parziale

Il grafo è così fatto: ►



Grafi ad ordinamento parziale



Anche questo grafo (diagramma)
è logicamente corretto

Esempio: ordinamento parziale e totale

Risolvere il seguente problema:

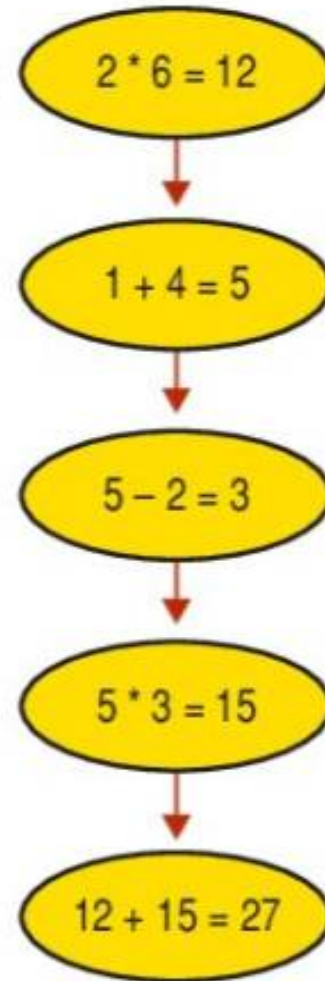
$$(2 * 6) + (1 + 4) * (5 - 2)$$

precedenza degli operatori (/, *, + e -).

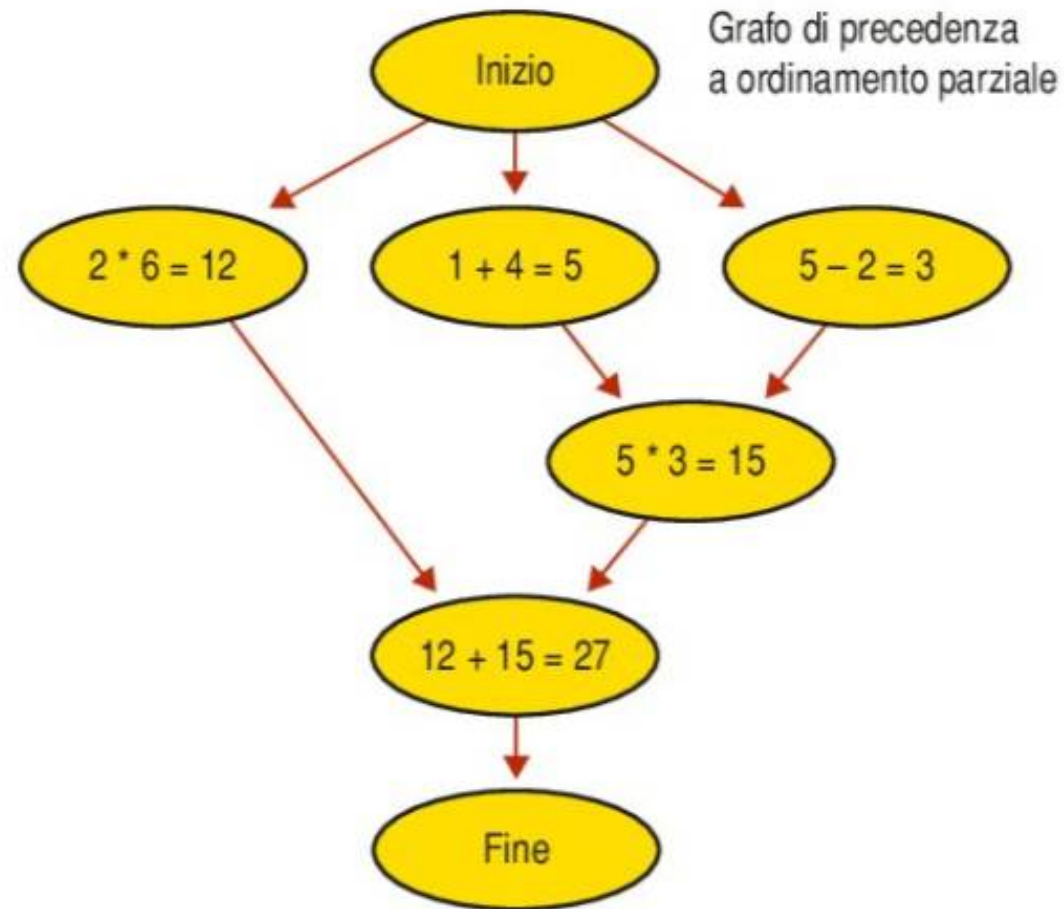
Non esiste l'obbligo di **un ordinamento totale** fra le operazioni da eseguire per il calcolo delle parentesi: potremmo indifferentemente eseguire prima $(1 + 4)$ piuttosto che $(2 * 6)$ o viceversa senza compromettere il risultato finale.

Esempio: ordinamento parziale e totale

Grafo di precedenza
a ordinamento totale



Esempio: ordinamento parziale e totale



Esercizio

- Grafo delle precedenze
- Ordinamento parziale e totale

Date le seguenti espressioni algebriche:

$$1 \quad 3+2-8*4-6/2+3 =$$

$$2 \quad (3+2)-8*(4-6)/2+3 =$$

$$3 \quad (3+2)-8*(4-6)/2+(3-8)*2 =$$

$$4 \quad 3*2-(8*4-6)/2+(3-8/2)*2 =$$

- A** disegnare il grafo sequenziale a ordinamento totale;
- B** disegnare il grafo sequenziale a ordinamento parziale.