

Programmazione e Architetture (Modulo B)

Lezione 12

Creazione di processi con fork

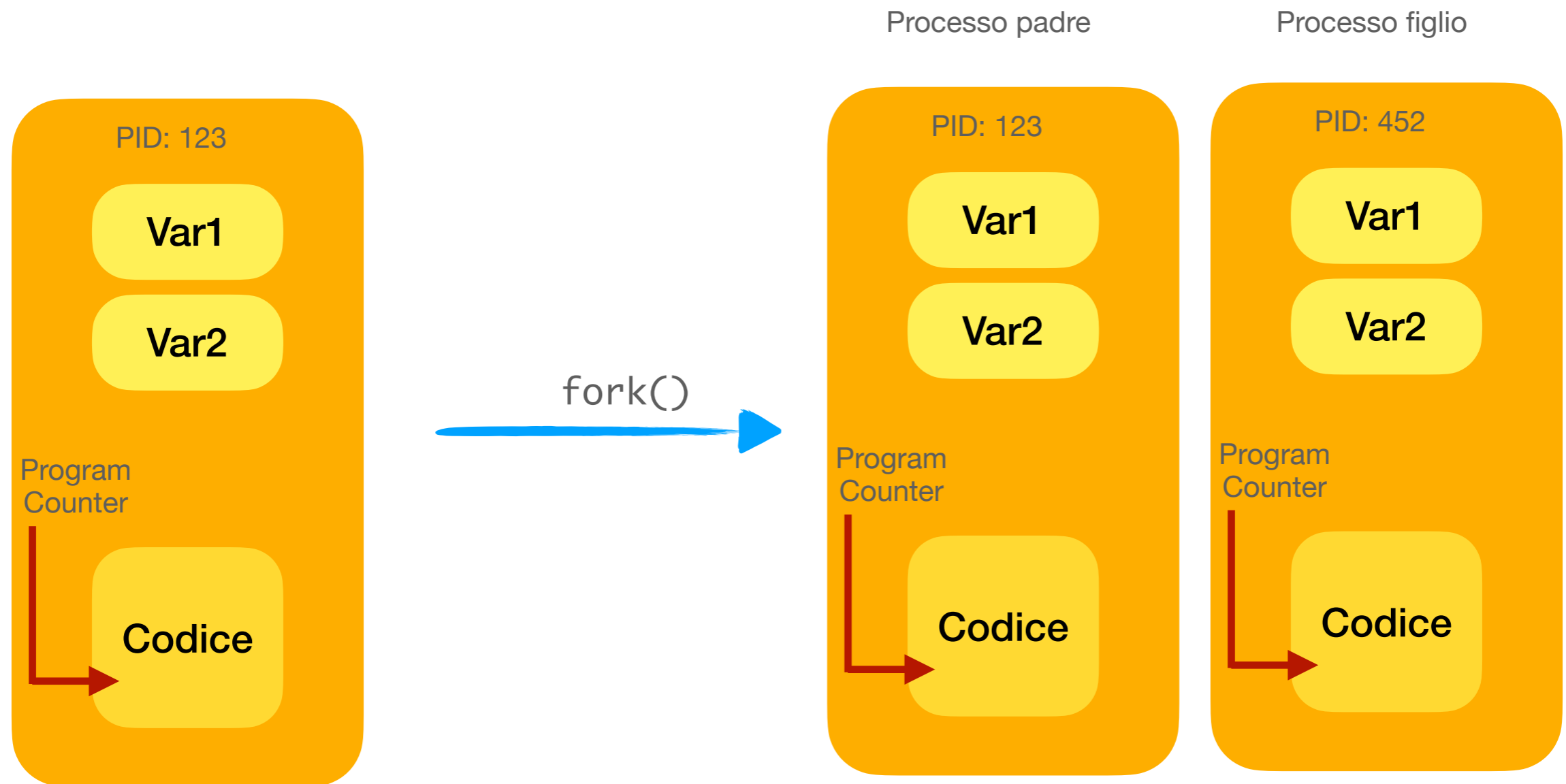
Processi in unix

E creazione di nuovi processi

- In unix ogni processo ha un identificativo detto PID - **P**rocess **I**Dentifier
- Il primo processo che viene avviato ha solitamente PID 1
- È possibile creare altri processi tramite la chiamata di sistema “fork”
- Fork duplica un processo (e tutte le risorse associate):
 - Una copia manterrà il PID originario (questo è il processo padre)
 - Una copia avrà un nuovo PID (questo è il processo figlio)
 - Esclusa questa differenza le due copie sono identiche

Fork

Rappresentazione grafica



L'intero spazio degli indirizzi è duplicato. Questo significa che ogni processo ha una sua copia di tutta la memoria (ed un suo program counter indipendente)

Fork

E creazione di nuovi processi

- Per utilizzare fork in C è necessario includere unistd.h
- Questo rende disponibile la funzione fork()
- Il valore ritornato dalla chiamata a fork, di tipo pid_t è:
 - -1 se la chiamata è fallita
 - 0 nel processo figlio
 - Il pid del processo figlio nel processo padre
- Notate come fork ritorni **due** volte in due processi diversi con valori diversi

Wait

Interazione tra processi

- È possibile richiedere a un processo padre di attendere che un processo figlio termini
- Includendo `sys/wait.h` diventa disponibile la funzione `wait`
- “Wait” prende come argomento un puntatore a un intero
- “Wait” ritorna quando uno dei processi figli è terminato con valore di ritorno:
 - `-1` se vi è stato un errore
 - Altrimenti è il PID del processo figlio che è terminato