

Università degli studi di Modena e Reggio Emilia
Dipartimento di Ingegneria Enzo Ferrari

Operating System Design

Anno Accademico 2024/25

Indice

1	Processi e Thread	1
1.1	Processi	1

Capitolo 1

Processi e Thread

1.1 Processi

Processo: è l'*astrazione* di un programma in esecuzione. Il processo è l'astrazione più elementare e più importante che ci può fornire il sistema operativo. Riuscendo ad emulare il comportamento di esecuzione concorrente nonostante la presenza di una singola CPUs.

Un altro paio di definizioni:

- **algoritmo:** in matematica ed informatica ci si riferisce ad algoritmo ad una sequenza finita di istruzioni rigorose matematiche che hanno l'obiettivo di risolvere una determinata classe di un problema specifico o di risolvere un calcolo.
- **programma:** è la sequenza o l'insieme di istruzioni in linguaggio macchina che può essere eseguito.

I moderni computer possono eseguire diverse operazioni nello stesso istante. Descrivendo in maniera rigida quello che effettivamente succede, però, è che ogni CPU in *ogni istante di tempo* esegue **uno e un solo** processo. Tendendo a 0 il tempo riservato a ogni singolo processo è però possibile simulare **parallelismo** definito anche come: ***pseudoparallelism***, che però va in contrasto con il vero parallelismo hardware (multi-CPU).

Il ***Process Model*** definisce che tutti gli eseguibili del computer, a volte includendo il sistema operativo, vengano organizzati in una serie di **processi sequenziali**. Il pro-

cesso è stato definito come l'istanza di un programma in esecuzione nel quale viene anche incluso il suo **PCB** (**Process Control Block**). Il **PCB**, anche noto come *process descriptor* è una struttura che permette di salvare tutte le informazioni che riguardano un determinato processo, ad esempio: *program counter*, i registri e le variabili. Concettualmente possiamo visualizzare che ad ogni processo è associata una CPU virtuale.

Process Switching

È quando l'*OS* cambia processo in esecuzione sulla CPU.

Per ora considereremo che esista un'unica **CPU**. Questa assunzione non tiene normalmente conto dei moderni *chip* che sono spesso multi-core.

Possiamo visualizzare inizialmente il processo come una tupla che contiene: il programma, degli input, degli output e uno **stato**. Un singolo processore può essere condiviso da n processi con un algoritmo di *scheduling* (*scheduler algorithm*) che viene utilizzato per determinare quando interrompere un processo (se può farlo) e servirne un altro.

Processo vs. Programma

Un programma è qualcosa che può essere salvato su disco, statico; mentre un processo è qualcosa di dinamico e che varia ad ogni sua istanza.

Un programma può essere eseguito da più processi che però sono distinti l'uno dall'altro.

La **creazione di un processo** può essere indotta da:

- inizializzazione di sistema
- un processo in esecuzione compie una *system call* che inizializza un nuovo processo
- un utente richiede l'esecuzione di un nuovo processo

I processi possono essere eseguiti in *foreground*, ovvero con i quali un utente può interagire, oppure in *background*, che sono “nascosti” all'utente e rispondono a certe specifiche funzioni. Su linux sono presenti decina di processi in background, alcuni anche noti come *daemons*.

In **UNIX** è presente una solo *system call* per creare un nuovo processo: **fork**. Dopo l'esecuzione della *syscall* i due processi, il padre e il figlio, hanno la stessa immagine della memoria, le stesse stringhe di environment e gli stessi file aperti. Normalmente, dopo il figlio, esegue **execve** o una *system call* simile per cambiare l'area di memoria ed eseguire un nuovo programma.

Alcune implementazioni di **UNIX** condividono la sezione *.text* tra i due visto che non può essere modificata. In alternativa altre implementazioni possono condividere tutta la memoria del padre, in questo caso la memoria è condivisa in maniera **copy-on-write**, ovvero ogni volta che uno dei due vuole modificare parte della memoria, quel specifico *chunk* viene copiato prima della modifica in una locazione privata della memoria.