

Processi e Risorse nei Sistemi Operativi

Che cos'è un processo

Un **processo** è un **programma in esecuzione**, ovvero un'entità **dinamica** creata dal **sistema operativo** quando un programma viene caricato nella **memoria principale (RAM)**. È l'unità di **esecuzione** del sistema operativo, dotata di **risorse proprie** come memoria, CPU e file aperti. Quando un programma viene eseguito:

- il **sistema operativo** crea un processo;
- gli assegna un **identificativo univoco** chiamato **PID (Process IDentifier)**;
- gli destina le **risorse necessarie** al corretto funzionamento.

Il processo può essere **visibile** (programma attivo, ad esempio un browser) o **in background** (processi del sistema operativo).

Invece, un **programma** è un'entità **statica**, formata da una sequenza di **istruzioni** per dire al computer cosa fare e non utilizza direttamente **risorse**.

Tipi di processi

Processi indipendenti

Sono processi che **non condividono dati o risorse** con altri. Operano in modo **autonomo** e non vengono influenzati dall'esecuzione di altri processi.

Esempio: l'esecuzione di due programmi distinti come un **editor di testo** e un **browser web**.

Processi cooperanti

Sono processi che **collaborano** tra loro condividendo **dati, risorse o obiettivi comuni**.

Esempio: un processo produce **dati** (produttore) e l'altro li **elabora** (consumatore).

Processi concorrenti

Sono più processi che si alternano nell'uso della **CPU**, dando l'impressione di essere eseguiti **simultaneamente**. Si parla di **concorrenza** perché i processi condividono la CPU nel tempo, grazie al principio del **multitasking**. I processi **competono per avere a disposizione le risorse** come la memoria RAM e soprattutto il processore.

In architetture **multiprocessore**, possono anche eseguire realmente in **parallelo**.

I 5 stati del processo

Durante la propria vita, un processo attraversa diversi **stati**, che rappresentano la sua interazione con le **risorse** del sistema operativo:

1. **New (Nuovo)** – Il processo è stato creato, ma non è ancora caricato completamente in memoria.
2. **Ready (Pronto)** – È in memoria, in attesa che la **CPU** lo esegua.
3. **Running (In esecuzione)** – Il processo utilizza la **CPU** ed è effettivamente in esecuzione.
4. **Waiting o Blocked (In attesa)** – Il processo è **sospeso** in attesa di una **risorsa** o di un **evento esterno** (es. I/O, completamento di operazioni).
5. **Terminated (Terminato)** – L'esecuzione è **completata** e il sistema operativo rilascia le **risorse**.

Lo **scheduler** è il componente del sistema operativo che gestisce le **transizioni tra stati**, assegnando la CPU secondo **criteri di priorità** o politiche di **scheduling** (come Round Robin, FCFS, Priorità, ecc.).

Risorse e molteplicità

Risorsa

Una **risorsa** è qualsiasi componente **hardware** o **software** necessaria al processo per **evolvere**, come CPU, RAM, stampante o disco. Una risorsa è ogni componente **riusabile** o **meno**, sia hardware che software necessario al processo o al sistema. Senza almeno una **risorsa disponibile**, un processo non può proseguire.

Classe e istanza di risorsa

- Le **classi di risorse** raggruppano risorse dello stesso **tipo**.
- Ogni **istanza** è un elemento della classe. Le risorse di una classe vengono dette **istanze della classe**.

Esempio: la classe *RAM* contiene singole **celle di memoria** come istanze. La **capacità totale di memoria** è la molteplicità della risorsa.

Molteplicità della risorsa

Indica il **numero massimo di processi** che possono usare contemporaneamente la **risorsa**. Il numero di risorse in una classe viene detto **molteplicità del tipo di risorsa**.

- Se la molteplicità è 1, la risorsa è **seriale** (es: stampante, CPU singola). Le **risorse seriali** non possono essere assegnate a più processi contemporaneamente, ma questi devono **attendere il loro turno** per poterle utilizzare.
- Se la molteplicità è maggiore di 1, la risorsa è **non seriale** (es: RAM multicanale o file a sola lettura accessibili da più processi). Le **risorse non seriali** ammettono l'**accesso contemporaneo** di più processi e quindi possono considerarsi risorse di **molteplicità infinita**.

Esempio:

- In un PC con **una singola CPU**, la molteplicità del processore è 1 → solo un processo alla volta può evolvere.
- La molteplicità della **RAM** invece dipende dal numero di **celle fisiche di memoria** che la compongono.

Condivisione e gestione

Se i processi che richiedono una **risorsa** superano la sua **molteplicità**, la risorsa deve essere **condivisa** secondo regole gestite dal **sistema operativo**. **Condividere** è sinonimo di «**avere in comune**» e quindi le risorse di elaborazione devono essere **assegnate alternativamente** ai singoli processi che le richiedono.

Il **sistema operativo** mette a disposizione:

- Un **gestore di risorsa** (o gestore della risorsa), che è un **programma** che ne regola l'utilizzo;
- Un **protocollo di accesso** alla risorsa, che consiste nella **procedura** con la quale un processo effettua la **richiesta** della risorsa, la **ottiene** e quindi la **utilizza** e alla fine la **rilascia** affinché gli altri processi la possano utilizzare.

Classificazione delle risorse

Tra processi e risorse esiste un **legame molto stretto**. In merito all'interazione tra risorse e processi possiamo effettuare una classificazione in base:

- Al **tipo di richieste**;
- Alla **modalità di assegnazione**.

Classificazione per tipo di richieste

- **Richiesta singola:** un processo richiede una singola risorsa di una classe definita (caso normale);
- **Richiesta multipla:** un processo richiede contemporaneamente almeno due risorse per poter evolvere.

Classificazione per modalità di assegnazione

L'assegnazione delle risorse avviene in due modalità:

- **Assegnazione statica:** la risorsa viene assegnata al momento della creazione del processo e rimane a esso «dedicata» fino alla sua terminazione;
 - **Assegnazione dinamica:** i processi richiedono le risorse, le utilizzano quando sono loro assegnate e le rilasciano quando non sono più necessarie oppure alla loro terminazione.
-

Il grafo di Holt e la riducibilità

Il **Grafo di Holt**, o **grafo delle attese**, rappresenta la relazione tra **processi** e **risorse**.

- I **nodi** rappresentano i processi (P) e le risorse (R);
- Gli **archi diretti** indicano relazioni di **richiesta** o **assegnazione**:
 - Da **P a R**: processo che **richiede** una risorsa;
 - Da **R a P**: risorsa **assegnata** al processo.

Il grafo è **riducibile** se è possibile applicare regole di riduzione che permettono di rimuovere tutti i **nodi** del grafo (tutti i processi completano la loro esecuzione e rilasciano le risorse).

Un grafo che contiene un **ciclo** e non è riducibile indica una situazione di **deadlock**, in cui i processi restano **bloccati** aspettando risorse detenute da altri.
