# Sistemi Operativi

Federico Zotti

2021-04-20

# Indice

## Introduzione

Un sistema operativo (abbreviato in SO, OS in inglese) è un software di base (composto normalmente da più sottosistemi o componenti software: kernel, scheduler, file system, gestore della memoria, gestore delle periferiche, interfaccia utente e spooler di stampa), che gestisce le risorse hardware e software della macchina, fornendo servizi di base ai software applicativi.

## Funzioni principali

1. Controllo e della gestione delle risorse di sistema (CPU e Memoria primaria) e delle componenti hardware che costituiscono il computer (processi di Input/Output da e verso le periferiche collegate al sistema).
2. Esecuzione dei programmi (processi) che su di esso vengono eseguiti, assegnando ad essi le necessarie risorse per l'avanzamento dei processi.
3. gestire l'archiviazione e l'accesso ai file. I programmi possono gestire l'archiviazione dei dati su memoria di massa, servendosi delle procedure messe a disposizione del sistema operativo.

## Il kernel

Un kernel è un software che ha il compito di fornire ai moduli che compongono il sistema operativo e ai programmi in esecuzione (processi) sul computer le funzioni fondamentali ed un accesso controllato all'hardware.

Esso fornisce dunque le funzionalità di base per tutte le altre componenti del sistema operativo, che assolvono le loro funzioni servendosi dei servizi che esso offre.

A seconda del tipo di sistema operativo il kernel può inglobare altre parti o fornire solo funzioni base delegando più funzioni possibile a oggetti/gestori esterni.

Un kernel tradizionale, detto monolitico, integra dentro di sé la gestione della memoria virtuale, la CPU, lo scheduler e i gestori di file system, nonché i driver necessari per il controllo di tutte le periferiche collegate.

## La shell

La shell è la componente fondamentale di un sistema operativo che favorisce la comunicazione e l'interfacciamento dell'utente con il kernel.

Tramite la shell è possibile impartire comandi e richiedere l'avvio di altri programmi. Il nome "shell" proviene dall'inglese e significa "guscio", a conferma del fatto che questa componente viene considerata l'involucro, la parte visibile del sistema ed è dunque definibile come interfaccia utente.

L'interfaccia di una shell può essere grafica (GUI), testuale (TUI) o tramite riga di comando (CLI).

## Lo scheduler

Lo scheduler è un programma che implementa un algoritmo di *scheduling* il quale, dato un insieme di richieste di accesso ad una risorsa (tipicamente l'accesso al processore da parte di un processo da eseguire), stabilisce un ordinamento temporale per l'esecuzione di tali richieste, privilegiando quelle che rispettano determinati parametri secondo una certa politica di scheduling.

L'obiettivo dello scheduler è quello di sfruttare al massimo le risorse hardware ed il parallelismo per minimizzare i tempi di risposta e aumentare il *throughput*, ossia il numero di programmi eseguiti per unità di tempo.

## Programma e processo

Un **programma** è l'insieme di istruzioni memorizzate su una memoria di massa.

Un **processo** è un entità astratta e dinamica, caricata in RAM da un programma. Esso è una sequenza di attività (task) controllata da un programma (scheduler) che si svolge su un processore in genere sotto la gestione o supervisione del rispettivo sistema operativo.

Un processo è suddiviso in due parti:

- Codice (composto dalle istruzioni);
- Dati del programma:
  - Variabili globali/statiche, allocate nella RAM;
  - Variabili locali, memorizzate in uno stack;
  - Variabili temporanee, memorizzate nei registri della CPU;
  - Variabili dinamiche, memorizzate in un Heap.

L'insieme di tutti i dati di un processo prende il nome di **testo del processo**.

### Modelli di processo

Esistono 3 diversi modelli di processo:

- dipendente;
- indipendente;
- in competizione.

Due processi dipendenti cooperano per raggiungere un obiettivo; per potersi evolvere, i due processi devono scambiarsi informazioni.

Due processi indipendenti evolvono in modo autonomo senza bisogno di aiutarsi.

Due processi in competizione posso invece ostacolarsi compromettendo la loro elaborazione; Nel caso in cui due processi usano una stessa risorsa limitata, o uno o entrambi i processi bloccano la loro elaborazione (blocco individuale o critico).

### Fasi di un processo

Essendo un entità dinamica, un processo si evolve nel tempo.

Quando viene creato gli viene assegnato un identificatore e viene inserito nell'elenco dei processi pronti (*ready list*) per essere eseguiti dal processore.

Durante l'esecuzione ci possono essere 3 conclusioni:

- Il codice del processo viene eseguito e termina;
- Il suo tempo di CPU termina e torna nella ready list;
- Una risorsa necessaria per l'evoluzione del processo manca, quindi viene sospeso (entra nella *waiting list*). Un processo sospeso non può passare direttamente in esecuzione ma deve venire reinserito nella ready list.

Lo scheduler, oltre a provvedere alla gestione della coda dei processi pronti (RL) e quella dei processi sospesi (WL), può far risvegliare un processo in attesa e re-inserirlo nella RL. Questa azione è detta cambio di contesto (context-switch), ed è realizzata dal dispatcher (un modulo dell'OS a cui lo scheduler dà il controllo di gestire determinati processi).

Infine, non tutti i processi posso essere sospesi ed inseriti in una lista di attesa, ma devono essere per forza completati senza essere interrotti. Questi processi sono chiamati processi non pre-emptive (sono rocessi non pre-emptive tutti i programmi eseguiti in modalità kernel).

## Gestione dei dispositivi di I/O

Ogni dispositivo I/O è formato da due parti: La parte vera e propria (nel caso degli HDD, è il disco vero e proprio) ed il controller, un software che permette di estendere le funzionalità del sistema operativo,

permettendogli di interfacciarsi con il device (Device Driver). Durante la fase di bootstrap, viene caricato nella RAM solo un grosso file contenente le funzioni principali dell'OS, ovvero il Kernel; I driver, invece, vengono caricati in seguito.

Il registro dei device è chiamato spazio di I/O.

Per essere utilizzato, il driver deve essere inserito nel sistema operativo in modo che possa essere eseguito in modalità kernel.

Ci sono tre modi in cui un driver può essere inserito nel kernel:

- Ricollegare il kernel con il nuovo driver e quindi riavviare il sistema;
- Inserire nel sistema operativo un file che gli dice che ha bisogno del driver e poi riavviare il sistema;
- Fare in modo che il sistema operativo sia in grado accettare nuovi driver durante l'esecuzione e installarli al volo.

L'I/O può essere gestito in tre modi:

### Busy waiting

Un programma utente emette una chiamata di sistema, che il kernel poi traduce in una chiamata di procedura al driver appropriato. Il driver, quindi, avvia l'I/O e interroga continuamente il device per vedere se è completo. Quando l'I/O è stato completato, il driver inserisce i dati dove sono necessari e li restituisce. Il sistema operativo restituisce quindi il controllo al chiamante. Questo metodo è chiamato busy waiting e ha lo svantaggio di impegnare la CPU che esegue il polling (indica la verifica ciclica di tutte le unità o periferiche di input/output da parte del sistema operativo di un PC tramite test dei bit di bus associati ad ogni periferica) del dispositivo fino al termine.

### Interrupt

Chiamata al driver e gli ordina di fare qualcosa; il driver dà un "segnale" (interrupt) quando il dato è pronto, mentre il processore fa altro. Se sono presenti più interrupt, ci seve essere un protocollo che decide chi deve essere controllato per primo.

### DMA

Inserire all'interno del computer stesso un processore che gestisca gli interrupt (DMA, intermediario tra il processore e i device).

## Il bootstrap

Il boot indica, in generale, l'insieme dei processi che vengono eseguiti da un computer durante la fase di avvio, in particolare dall'accensione fino al completo caricamento in memoria primaria del kernel del sistema operativo a partire dalla memoria secondaria.

I passi compiuti dal sistema dopo l'accensione sono i seguenti:

1. il POST (Power On Self Test), una serie di test diagnostici per verificare il corretto funzionamento dell'hardware e della scheda madre: se tutti i dispositivi controllati sono funzionanti emette un solo "beep" dall'altoparlante di sistema e prosegue;
2. cerca una scheda video installata ed esegue il POST video che si trova nella ROM interna della scheda video;
3. esegue altri test, come il conteggio della memoria e lo stato della tastiera. Se incontra degli errori, non ricorre al codice sonoro dei bip ma mostra un messaggio a video.
4. configura automaticamente i dispositivi Plug and Play presenti e mostra un messaggio a video per ciascuno di essi.
5. si interfaccia con la memoria CMOS, contenente i parametri di configurazione suscettibili di modifica.
6. Infine, cerca un'unità a disco da cui caricare il kernel del sistema operativo. Se c'è, carica in RAM il primo settore del disco (cilindro 0, testina 0, settore 1), che corrisponde al master boot record (MBR) e l'esecuzione continua da lì.