Architettura di Von Neumann
I sistemi operativi sono software complessi progettati per gestire le risorse hardware di un computer e fornire un'interfaccia efficiente e sicura agli utenti e alle applicazioni. Tuttavia, per comprendere il funzionamento e le responsabilità di un sistema operativo, è fondamentale partire dall'architettura hardware su cui esso opera.
L'architettura di von Neumann, proposta nel 1945 dal matematico John von Neumann, è il modello concettuale alla base della maggior parte dei computer moderni. Essa descrive un sistema in cui i dati e le istruzioni sono memorizzati nella stessa memoria principale, e l'elaborazione è svolta da un'unità centrale di elaborazione (CPU) che esegue sequenzialmente le istruzioni.
A partire da questa struttura logica, possiamo identificare le principali componenti hardware:

Processore (CPU - Central Processing Unit)
È l'unità centrale di elaborazione. Interpreta ed esegue le istruzioni dei programmi, elabora i dati e coordina il funzionamento generale del sistema. È composto da unità di controllo, unità aritmetico-logica (ALU) e registri interni.

Scheda madre (Motherboard)
È la scheda elettronica principale su cui sono installati tutti i componenti hardware. Fornisce le connessioni elettriche e logiche tra CPU, memoria, schede di espansione, dispositivi di archiviazione e periferiche. Contiene anche chipset, BIOS/UEFI e controller di comunicazione.

RAM (Random Access Memory)
È la memoria principale, di tipo volatile, usata per conservare temporaneamente dati e istruzioni in fase di elaborazione. È caratterizzata da velocità elevate e accesso diretto ai dati. La capacità e la frequenza della RAM influenzano direttamente le prestazioni generali del sistema.

Memoria di archiviazione (HDD / SSD / NVMe)
Serve per la memorizzazione permanente dei dati. I dischi rigidi (HDD) utilizzano piatti magnetici, mentre le unità a stato solido (SSD) si basano su memoria flash, molto più veloce e resistente. Le interfacce più comuni includono SATA e PCIe/NVMe.

Scheda grafica (GPU - Graphics Processing Unit)
Elabora e genera immagini da visualizzare sullo schermo. Può essere integrata nella CPU o presente come componente dedicato. Le GPU moderne sono altamente parallele e possono anche essere impiegate per calcoli complessi oltre all'elaborazione grafica.

Alimentatore (PSU - Power Supply Unit)
Trasforma la corrente alternata (AC) della rete elettrica in corrente continua (DC) utilizzabile dai componenti interni. Deve fornire tensioni stabili e adeguate per tutte le componenti, e la sua potenza complessiva va dimensionata in base al consumo dell'intero sistema.

Interfaccia di rete (NIC - Network Interface Card)
Consente la connessione a reti locali (LAN) o a Internet, tramite cavo Ethernet o connessioni wireless (Wi-Fi). Può essere integrata nella scheda madre o disponibile come scheda dedicata. Supporta protocolli di comunicazione standard come Ethernet, TCP/IP e Wi-Fi.

Sistema operativo
Per comprendere appieno il funzionamento di un sistema operativo, è utile definire i concetti fondamentali di **kernel** e **BIOS/UEFI**.

Il **kernel** è il nucleo centrale di un sistema operativo. Si tratta di un componente software che funge da intermediario tra l'hardware del computer e i programmi applicativi. Il kernel gestisce le risorse fondamentali del sistema, come la CPU, la memoria, i dispositivi di input/output e i processi, assicurando che ogni componente funzioni correttamente e in modo coordinato. È il primo modulo del sistema operativo a essere caricato in memoria durante l'avvio del computer e rimane attivo fino allo spegnimento. Esistono diversi tipi di kernel, come il **monolitico**, il **microkernel**, il **kernel ibrido** e altri, ciascuno con una propria architettura e filosofia di progettazione:
Microkernel: è un software ridotto ai minimi termini che include un numero limitato di funzionalità del sistema operativo. Un esempio di Microkernel è MINIX.
Monolitico: è un unico file binario che include la maggior parte delle funzionalità di un sistema operativo; un esempio di Kernel monolitico è Unix.
Modulare: può essere visto come un'estensione del Kernel Monolitico, con la capacità di aggiungere o rimuovere moduli a seconda delle necessità.
Ibrido: insieme di Monolitico e Microkernel. (Windows ha un kernel ibrido)

Il **BIOS** (Basic Input/Output System) e l'**UEFI** (Unified Extensible Firmware Interface) sono due tipologie di firmware installate sulla scheda madre che si occupano della fase iniziale di avvio del sistema e della gestione a basso livello dell'hardware.
Alla prima accensione del computer, il BIOS esegue una serie di test diagnostici chiamati **POST** (Power-On Self Test) per verificare che le componenti hardware fondamentali siano funzionanti. Dopodiché, individua un dispositivo di avvio (ad esempio un hard disk o una chiavetta USB) e carica il bootloader, ovvero il codice responsabile dell'inizializzazione del sistema operativo.
L'**UEFI**, evoluzione moderna del BIOS, offre numerosi vantaggi: un'interfaccia grafica più avanzata, supporto per dischi di dimensioni superiori ai 2 TB, avvio più rapido e funzionalità avanzate di sicurezza come il **Secure Boot**. A differenza del BIOS, l'UEFI può leggere direttamente i file system moderni e caricare file eseguibili direttamente dalla memoria di massa.

Non appena il computer viene avviato, il firmware (BIOS o UEFI) esegue i test iniziali, identifica un dispositivo di boot e carica il bootloader, che a sua volta carica il **kernel** del sistema operativo dalla memoria di massa alla memoria RAM.
Una volta caricato, il kernel rileva e configura i **driver** necessari per permettere al sistema di interagire con i dispositivi hardware, come tastiera, disco rigido, scheda di rete e altri componenti essenziali. A questo punto, il kernel **monta il file system principale**, contenente i dati necessari per completare l'avvio del sistema, come la configurazione dei **demoni di sistema**, la gestione della rete, le impostazioni di sicurezza e i servizi legati all'interfaccia utente.
Infine, viene caricata l'interfaccia utente (desktop) rendendo il sistema operativo pienamente operativo.
Da questo momento ogni volta che viene eseguita un'applicazione, il kernel crea un **nuovo processo**, assegnandogli risorse specifiche come memoria e tempo di CPU. Per determinare quale processo deve essere eseguito in un dato momento, il kernel utilizza un sistema di **scheduling**, garantendo così che le operazioni vengano eseguite in maniera ordinata ed efficiente.
Quando un processo richiede l'accesso a un dispositivo, ad esempio, leggere un file o visualizzare un'informazione a schermo, il kernel riceve la richiesta e la gestisce tramite driver appropriati, i quali traducono le istruzioni software in comandi comprensibili per l'hardware.

Memoria centrale
Possiamo distinguere quattro tipi principali di memoria centrale:
La memoria ROM (Read-Only Memory) è un tipo di memoria non volatile, cioè, mantiene i dati anche quando il computer è spento. Come suggerisce il nome, è una memoria di sola lettura: i dati in essa contenuti non possono essere modificati facilmente o durante il normale funzionamento del sistema. La ROM viene utilizzata per memorizzare istruzioni fondamentali per l'avvio del computer, come il firmware del BIOS o dell'UEFI dato che il processore non ha ancora accesso al sistema operativo installato sul disco.

La **Cache** è una memoria utilizzata per memorizzare temporaneamente dati che sono già stati utilizzati, con l'obiettivo di rendere più rapido un eventuale accesso futuro agli stessi dati. La sua funzione principale è quindi quella di velocizzare le operazioni del sistema riducendo i tempi necessari per recuperare le informazioni più frequentemente richieste.
Queste aree di memoria, tuttavia, sono piuttosto limitate in termini di capacità: possono contenere solo una piccola quantità di dati. Per questo motivo, la cache è progettata per essere estremamente selettiva, conservando solo le informazioni che, con maggiore probabilità, saranno riutilizzate a breve termine. Nonostante le sue dimensioni ridotte, il suo impatto sulle prestazioni del sistema è significativo, perché consente di accedere più rapidamente ai dati rispetto alla memoria principale.

I registri della CPU sono memorie ad accesso molto rapido che il processore utilizza per conservare istruzioni e dati per brevissimi periodi di tempo durante l'esecuzione delle operazioni. La loro funzione è quella di fornire alla CPU un accesso immediato alle informazioni più critiche, necessarie nell'immediato per eseguire calcoli, confronti o gestire il flusso delle istruzioni.
Grazie alla loro estrema velocità, i registri permettono alla CPU di operare in modo estremamente efficiente, evitando qualsiasi ritardo che deriverebbe dal dover accedere a memorie più lente, come la cache o la RAM. Sebbene siano molto limitati in numero e dimensione, i registri svolgono un ruolo essenziale nell'architettura del processore, rappresentando il primo livello di memoria che entra in gioco durante l'elaborazione delle istruzioni.

La RAM (Random Access Memory) è una memoria volatile utilizzata dal sistema per memorizzare temporaneamente dati e istruzioni durante l'esecuzione dei programmi. Essendo volatile, i dati contenuti nella RAM vengono cancellati automaticamente quando il computer viene spento, a differenza delle memorie permanenti come il disco rigido.
La RAM ha un ruolo fondamentale nel garantire che il sistema funzioni in modo fluido ed efficiente: quando si avvia un programma, questo viene caricato dalla memoria di massa alla RAM, dove il processore può accedere rapidamente alle informazioni necessarie per eseguire le operazioni. Una delle caratteristiche principali della RAM è infatti la possibilità di accedere direttamente a qualsiasi cella di memoria, senza dover seguire un ordine sequenziale. Questo accesso diretto e rapido permette una gestione efficiente dei dati e delle istruzioni, migliorando notevolmente le prestazioni complessive del sistema durante l'uso.

Gestione della RAM
Dal punto di vista strutturale, la RAM è composta da **una matrice di celle**, ognuna delle quali è **identificata da un indirizzo univoco espresso in notazione binaria**. Questi indirizzi permettono al sistema di localizzare e accedere con precisione ai dati memorizzati, assicurando velocità e accuratezza nelle operazioni di lettura e scrittura.
Per poter gestire al meglio la memoria disponibile della RAM senza lasciare spazi di memoria inutilizzati, viene utilizzata una tecnica chiamata "**paging**" che consiste nell'organizzare la RAM in tanti "frame" che sono delle porzioni di memoria vuota. Quando un programma viene eseguito, non viene caricato interamente nella RAM, ma solo alcuni frame. Se il programma dovesse aver bisogno di altri frame rispetto a quelli che sono stati precedentemente assegnati, verrebbero assegnati altri frame. Se i frame della RAM finiscono, avviene quello che si chiama "file swap" che consiste nello spostare i datti che dovrebbero essere inseriti nella RAM piena, sul disco fisso sul quale rimangono fino a quando non vengono liberati dalla RAM i frame necessari.

Gestione dei processi
Un processo è un'istanza (un contenitore), identificata da un codice univoco(PID), che contiene tutte le istruzioni del programma insieme a tutte le informazioni necessarie per eseguirlo.
Ad ogni processo vengono assegnati degli stati:

Esecuzione che sta ad indicare che le sue istruzioni vengono eseguite dal processore

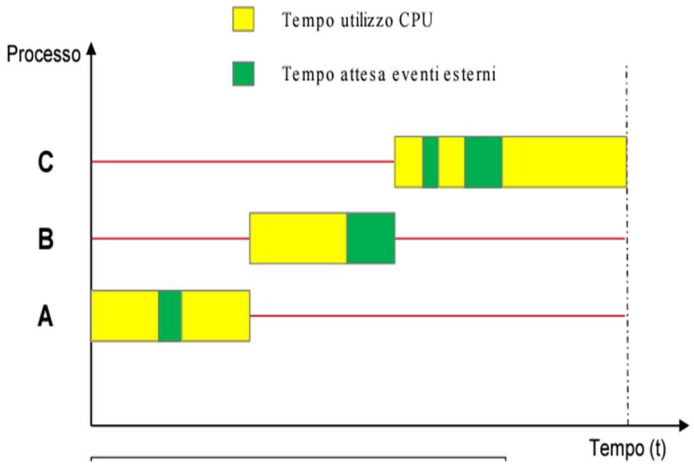
Attesa dove il processo attende che si verifichi qualche evento (attende input utente tramite tastiera).

Pronto il processo attende di essere assegnato ad un processore per passare in Esecuzione.

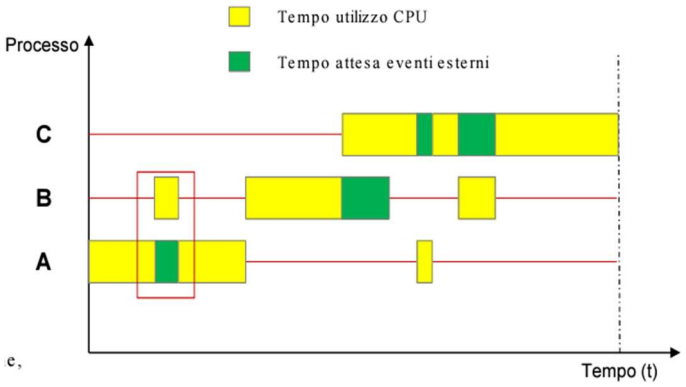
Terminato il processo ha terminato la sua esecuzione

La gestione dei processi prende il nome di **Scheduling** che si suddivide in 3 metodi:

sistemi mono tasking sono quelli che consentono l'esecuzione di un solo processo o programma alla volta. In questi sistemi, la CPU è dedicata esclusivamente a un singolo compito fino al suo completamento. Questo significa che, durante l'esecuzione di quel programma, nessun altro processo può essere eseguito. I sistemi mono tasking erano comuni nei primi computer, dove le risorse hardware erano limitate e non era possibile o pratico gestire più processi contemporaneamente. Sebbene siano semplici da implementare, questi sistemi risultano poco efficienti quando si devono gestire più operazioni o utenti, perché tutte le attività devono aspettare il completamento del processo in esecuzione.



I sistemi multi tasking, invece, permettono l'esecuzione contemporanea di più processi. Questo non significa che la CPU esegua più processi esattamente nello stesso istante (a meno che non si tratti di un sistema con più core), ma che il sistema operativo gestisce il tempo della CPU suddividendolo tra i processi attivi. Grazie a questa divisione, ogni processo sembra "girare" simultaneamente, anche se in realtà la CPU passa rapidamente da un processo all'altro. Il multi tasking migliora notevolmente l'efficienza e la produttività del sistema, consentendo all'utente di eseguire più programmi contemporaneamente, come ascoltare musica mentre si naviga su internet o si scrive un documento.



Il time sharing è una particolare forma di multi tasking che enfatizza la condivisione equa del tempo di CPU tra più utenti o processi. Nei sistemi time sharing, la CPU assegna a ciascun processo un intervallo di tempo breve e fisso, chiamato "quantum" o "time slice". Quando questo intervallo scade, il sistema operativo sospende temporaneamente il processo in esecuzione e passa la CPU al processo successivo nella coda. Questo cambio avviene così rapidamente che gli utenti percepiscono tutti i programmi come attivi contemporaneamente e reattivi. Il time sharing è particolarmente utile in ambienti multiutente, come i server o i mainframe, dove è importante garantire una risposta rapida e un accesso equo alle risorse del sistema.