

Il sistema operativo è un software che consente all'utente di interfacciarsi con il hardware.

Il SO genera delle **istruzioni** etc convention che "mascherano" la complessità dell'hardware. Un'istruzione che usa per gestire i dischi sono i **file**.

Si necessita di un'allocazione ordinata e controllata di risorse.

Con **multiplexing** intendiamo la **"condivisione di risorse"** in due modalità differenti: nel tempo e nello spazio.

nel tempo \Rightarrow programmi diversi fanno a turno per utilizzare la risorsa. La gestione del tempo e tempo di occupazione è a carico del SO.

nello spazio \Rightarrow invece di alternarsi prendono ogni parte della risorsa, ad esempio la memoria c'è suddivisione tra i file in esecuzione.

Evoluzione dei sistemi operativi

1^o generazione (1945 - 1955) \Rightarrow valvole termoioniche

Tutta la programmazione veniva effettuata in Fortran, assembly, ecc...

Svolgeva semplici calcoli matematici

2^o generazione (1955 - 1965) \Rightarrow transistor e sistema batch

Per eseguire un job (programma) il programma deve venire in cassetta e poi passare alle schede perforate.

Svolgeva equazioni diff. parziali

3^o generazione (1965 - 1980) \Rightarrow circuiti integrati e multiprogrammazione

Anche se non era efficiente, n'aveva un meccanismo di **time sharing** che consentiva ai più utenti di lanciare contemporaneamente

Mac os il primo sistema operativo basato su Unix

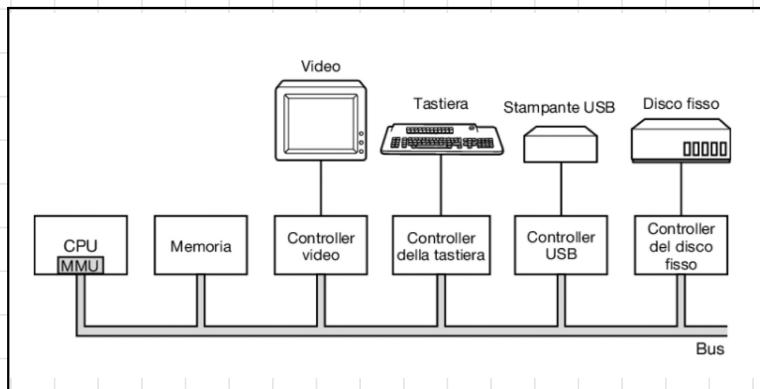
4^o generazione (1980 ad oggi) \Rightarrow personal computer

Mac os il **DOS** cioè il sistema operativo di disce. Mac os fa la **GUI** (graphical user interface). Si ha pure la

possibilità di eseguire anche applicazioni grafiche

5° generazione (1990 - oggi) → computer mobili e smartphone

Hardware



Un so è legato all'hardware del computer su cui gira.

Le componenti hardware più delicate sono il **processore** e la **memoria**.

Processore

Esegue ciclicamente le seguenti fasi:

- Fetch (prelievo)
- Decode (decodifica)
- Execute (esecuzione)

Il processore presenta dei registri speciali, tra i quali:

- PC (program counter)

Puntatore alla prossima istruzione da eseguire

- SP (stack pointer)

Puntatore alla cima dello stack attuale in memoria

- PSW (program status word) \Rightarrow essenziale per le system call

contiene informazioni sullo stato dell'ultimo processo eseguito dal processore
oltre ai bit relativi alle istruzioni di controllo + priorità della CPU,
modalità di esecuzione (Kernel o user).

Tecniche di programmazione più avanzate riguardano l'utilizzo
di pipeline e CPU superscalare

- Pipeline \Rightarrow esecuzione di più istruzioni in parallelo attraverso
la loro suddivisione in diverse fasi

- CPU superscalare \Rightarrow regole peraltro più istruzioni e messe
in un buffer in attesa che esse vengano eseguite
da un'unità di esecuzione libera. L'esecuzione non
avviene sequenzialmente ma il risultato è equivalente all'esecuzione ordinata

Per ottenere servizi del sistema operativo si usano le system call. Un'istruzione TRAP cambia la user mode in Kernel mode ed invoca il SO. Quando vengono eseguite tutte le istruzioni si ritorna in user mode e l'esecuzione riprende dall'istruzione che regge la chiamata di sistema.

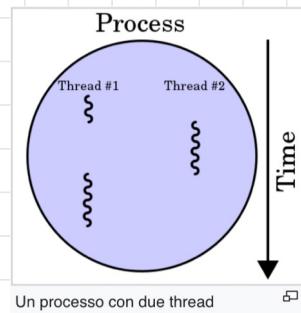
Multi Thread e multiette

multithread

Here all'interno della CPU ha lo stato di più Thread.

3 Thread sono blocchi di codice di un process che condividono la stessa area di memoria e che eseguono parti precise del process. Se ci fosse un solo thread, il process sarebbe molto più lento, mentre così si crea uno pseudoparallelismo che ottimizza il tempo di esecuzione dell'intero process. I vantaggi diretti sono identici quando si ha un solo core.

Divide l'esecuzione di un process in flussi che <= eseguono in parallelo parti diverse del process



Se un thread si dovesse bloccare a causa di una risorsa
da dove enere letta dalla memoria, viene eseguito un altro
thread che invece può andare avanti con l'esecuzione (scambio).
Risulta molto veloce scambiarsi con un altro thread della
stessa programma grazie al fatto che il processore riconosce
l'uso dei thread e ne tiene conto.

⇒ Ogni thread appare al SO come una CPU separata.
Se un sistema ha due CPU, ognuna delle quali con due
Thread, allora il SO le vedrà come quattro CPU reali.

multicore

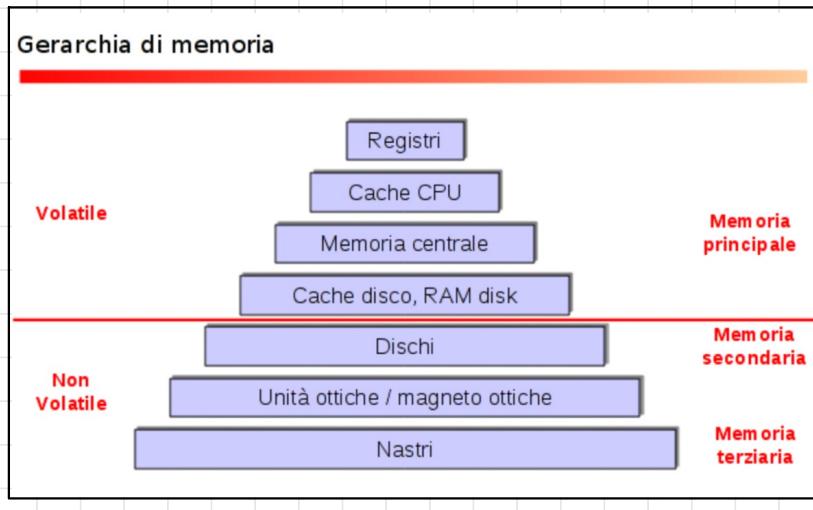
Con il termine multicore si intende una CPU dove sono
presenti più unità di esecuzione, detti **core** o **processori** che permettono
la reale esecuzione di diversi processi in parallelo.
In questo caso ogni core ha la propria cache dedicata.

Memoria

La memoria ideale ha una capacità molto grande ed un tempo di accesso nullo. Nel caso reale poi si cerca di ottenere una capacità il più grande possibile ed un tempo di accesso ridotto al minimo. Per ottenere una memoria con alta capacità e veloce allo stesso tempo, si suddivide la memoria in livelli gerarchici perché ogni memoria ha diversi tempi di accesso e diverse capacità.

CAPACITÀ

VELOCITÀ



La CPU utilizza le memorie per le proprie elaborazioni

Memoria ROM (read only memory)

è una memoria di sola lettura adatta ai programmi di inizializzazione del computer, nella quale i dati vengono memorizzati in modo permanente. Quando si avvia il calcolatore la RAM è nulla ed attraverso il BIOS (basic input/output system), la CPU esegue il codice necessario all'avvio e al funzionamento: questo programma si trova nella ROM.

Registri CPU

Memorie elettroniche interne alla CPU caratterizzate da un'alta velocità di risposta ma da una bassa capacità, oltre al fatto che sono **volatili**, ovvero non mantengono in maniera permanente i dati diversi da alimentazione.

Memoria RAM (random access memory)

La memoria RAM è una memoria elettronica. Essa è detta ad accesso casuale in quanto in grado di accedere a qualsiasi posizione in memoria con gli stessi tempi.

Memorie cache

memorie elettroniche basate su flip-flop e dunque sono **volatili**

Lo scopo di queste memorie è quello di "memorizzare" la latenza dell'accesso al disco rigido, infatti, esiste integrale tra la CPU e la RAM, esse compensano la differenza tra velocità di accesso, fornendo le informazioni più vicine alla CPU.

Hard-disk

Un disco fisso è composto da più platti di alluminio rotanti nello stesso senso di rotazione. Delle testine riescono a leggere informazioni dai dischi. Questa è una memoria **non-volatil**e ed ha una capacità molto elevata

Chi non conosce la struttura della memoria ha l'impressione di avere un'unica memoria veloce e con grandi capacità.

Per velocizzare l'esecuzione dei processi si sfrutta il vantaggio della cache dei processi frequenti eseguire un processo già eseguito in precedenza allo stesso processo che lo stava eseguendo precedentemente e quindi ha già in cache tutti i dati di quel processo