

Introduzione all'Informatica

In questa lezione:

- Architettura di un calcolatore
- Il Sistema Operativo
- La memoria secondaria
- Il "File System"

Architettura di un calcolatore

L'Architettura di un calcolatore

Come abbiamo visto, l'architettura dei moderni calcolatori è basata sul [modello di von Neumann](#), che prevede una [memoria](#), una [CPU](#), dei [dispositivi di input/output](#) e un [bus](#) di collegamento.

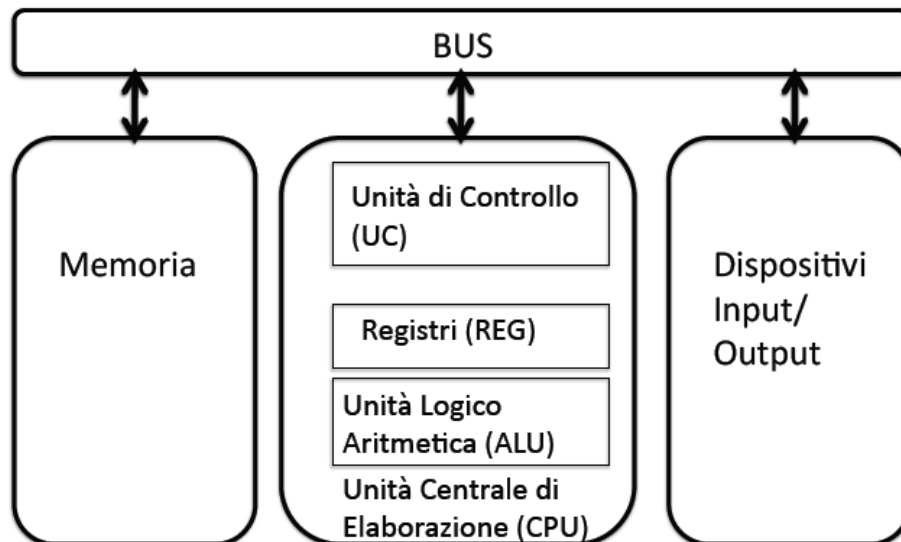
Anche se daremo dei dettagli in più sul funzionamento della CPU, un calcolatore non è solo hardware, ma ha dei componenti software, che, nel loro insieme, costituiscono il [Sistema Operativo](#).

Il Sistema Operativo permette di gestire le risorse hardware della macchina fornendo servizi di base ai programmi.

Di particolare interesse è la gestione della [memoria secondaria](#) (dischi, SSD), e l'organizzazione dei [file](#) (File System).

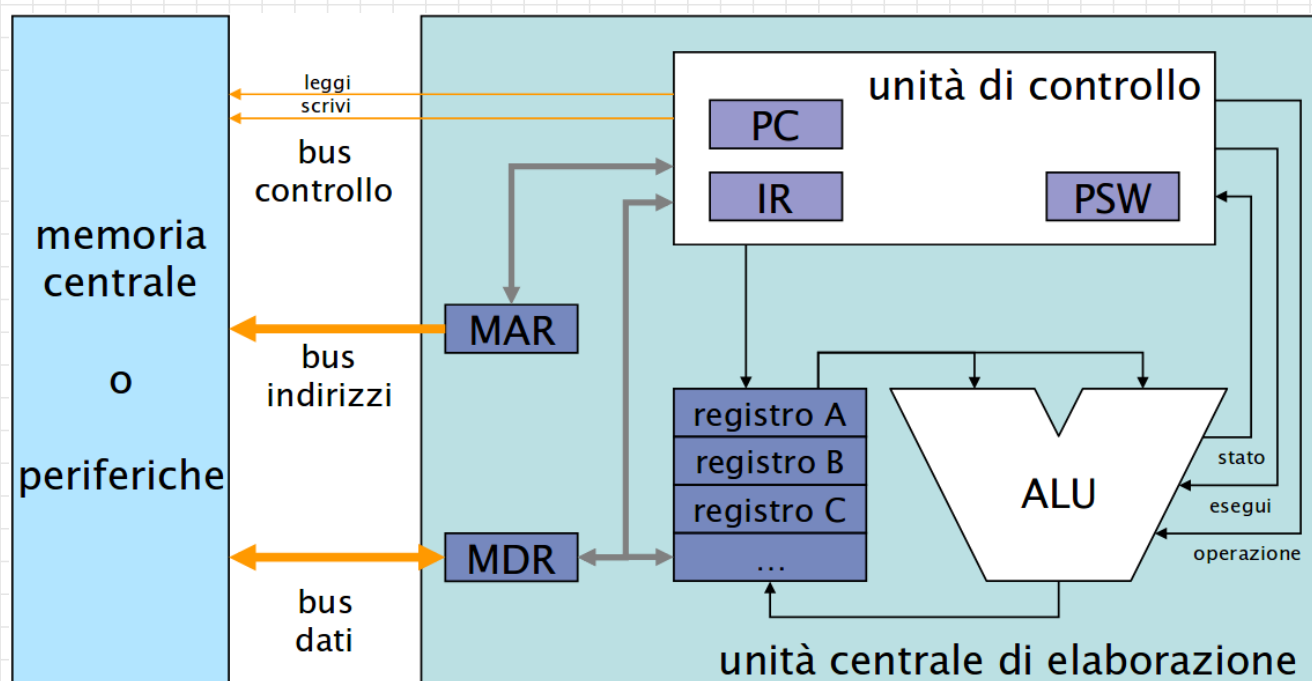
La comunicazione tra CPU e Memoria (1)

Ricordiamo l'architettura dei moderni calcolatori basata sul modello di Jhon von Neumann:



La comunicazione tra CPU e Memoria (2)

Uno sguardo più di dettaglio



La comunicazione tra CPU e Memoria (3)

Ciclo Fetch-decode-execute

Abbiamo già visto che la CPU esegue continuamente il [ciclo di fetch](#).

- fetch:** prelievo dalla memoria dell'istruzione da eseguire (l'indirizzo è nel registro Program Counter (PC)). L'istruzione viene messa nell'Instruction Register (IR). Il PC viene incrementato automaticamente per puntare alla prossima istruzione. Unità coinvolte: memoria e unità di controllo.
- decode:** decodifica dell'istruzione presente nell'IR e prelievo degli eventuali operandi che vengono messi nei registri. Unità coinvolte: memoria e unità di controllo.
- execute:** l'unità di calcolo esegue l'istruzione presente nell'IR. Eventuali risultati nei registri vengono trasferiti in memoria dall'unità di controllo. Se l'istruzione prevede un salto, il PC viene sovrascritto.

La comunicazione tra CPU e Memoria (4)

Registri speciali

Vediamo ora come avviene la comunicazione con la memoria per la scrittura e lettura dei dati.

Per supportare i trasferimenti di dati da e verso la memoria la CPU ha i seguenti registri speciali:

- **Memory Address Register (MAR)**, registro di indirizzamento della memoria
 - contiene l'indirizzo di memoria dove scrivere o da dove prelevare un dato
 - è direttamente collegato col [bus indirizzi](#)
- **Memory Data Register (MDR)**, registro dati di memoria
 - contiene una copia del dato da scrivere o prelevato dalla memoria
 - è direttamente collegato col [bus dati](#)

La comunicazione tra CPU e Memoria (5)

Segnali di controllo

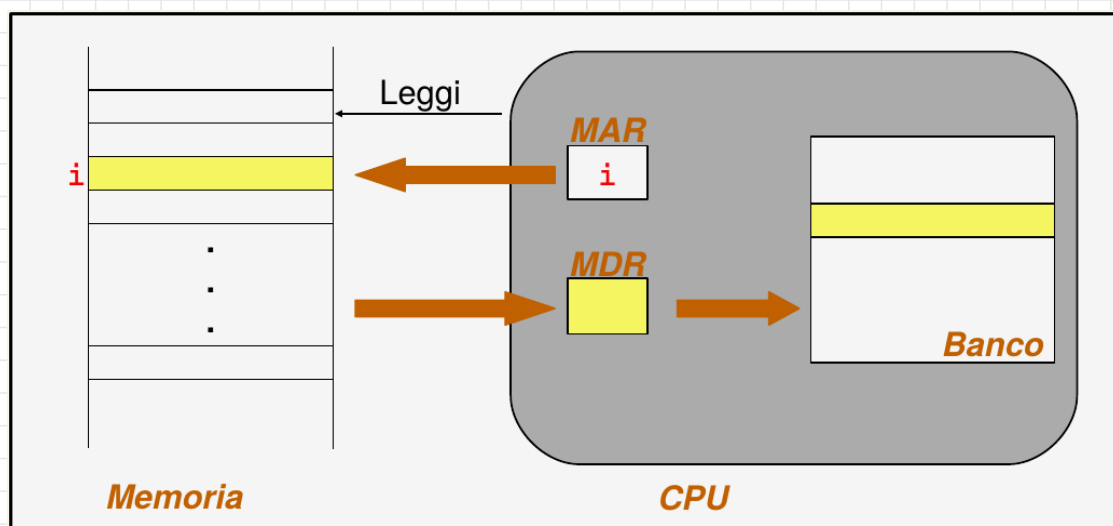
Per supportare le operazioni di trasferimento dati con la memoria sono previste due linee di controllo sul **bus controlli**:

- Segnale “leggi”:
quando questo viene attivato l'operazione richiesta è una lettura
- Segnale “scrivi”:
quando questo viene attivato l'operazione richiesta è una scrittura
- Se entrambi i segnali sono assenti la memoria non è chiamata in causa

La comunicazione tra CPU e Memoria (6)

Lettura

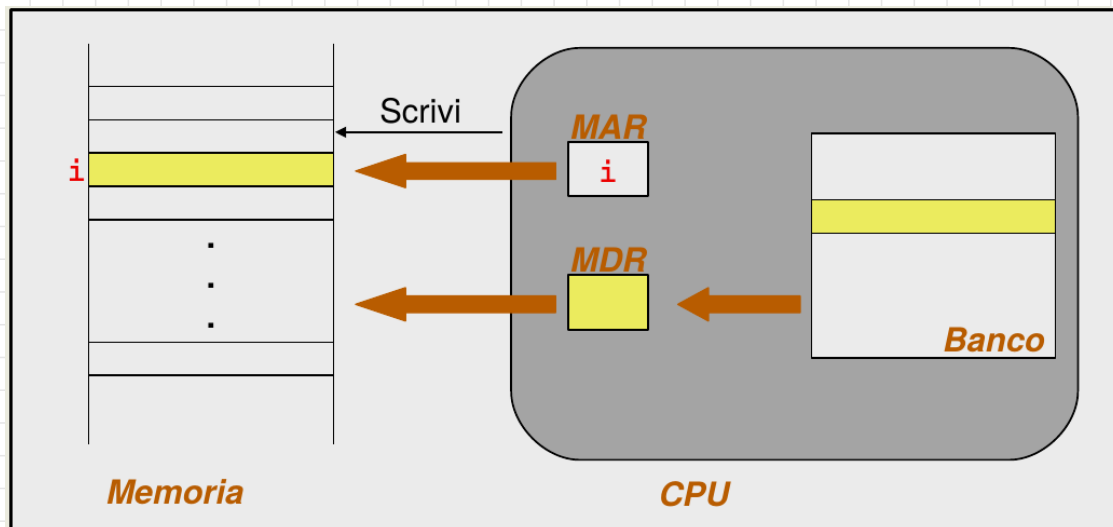
- La CPU scrive l'indirizzo **i** nel registro MAR (e quindi sul **bus indirizzi**) e invia il segnale “leggi” sul bus controlli
- La memoria risponde inviando sul **bus dati** il contenuto della cella di indirizzo **i**, che viene memorizzato nel registro MDR e quindi nel banco dei registri della CPU



La comunicazione tra CPU e Memoria (7)

Scrittura

- La CPU scrive l'indirizzo **i** nel registro MAR (e quindi sul **bus indirizzi**), il dato da memorizzare nel registro MDR (e quindi sul **bus dati**) e invia il segnale "scrivi" sul bus controlli
- La memoria scrive il dato nella cella di indirizzo **i**



Il Sistema Operativo

Il Sistema Operativo

I primi programmatori si trovavano a ripetere spesso alcune procedure per gestire l'hardware. I primi Sistemi Operativi erano una collezione di queste procedure. Attualmente:

Il **Sistema Operativo** (SO) è un insieme complesso di programmi che controlla le risorse del sistema e i processi che usano queste risorse.



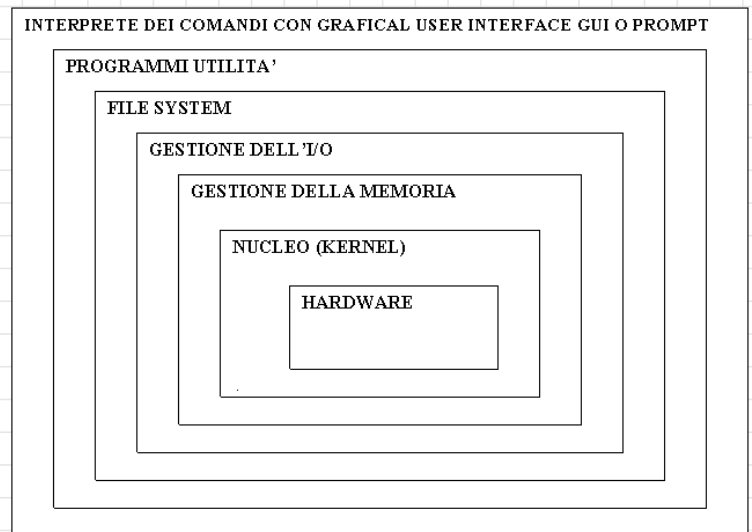
Logo di sistemi operativi:
MacOS, Linux, Windows,
iOS, Android, FreeBSD

Il Sistema Operativo

Gli attuali SO sono organizzati con una architettura a “buccia di cipolla”.

Per evidenziare le funzionalità delle parti del sistema e le interazioni fra esse si utilizza un modello basato su livelli di macchina virtuale (virtualizzazione delle risorse)

Il SO è organizzato in strati funzionali dove ciascun strato realizza una macchina virtuale che fornisce servizi allo strato superiore



Il Sistema Operativo

Il SO può essere visto come un'interfaccia software che si pone tra l'infrastruttura hardware e l'utente (o il software applicativo).

Il SO serve a facilitare l'uso delle risorse, nascondendo dettagli vicini alla macchina hardware (virtualizzazione).

Il SO è uno strumento software che virtualizza le caratteristiche dell'hardware sottostante, offrendo di esso la visione di una macchina astratta (virtuale), più potente e più semplice da usare di quella disponibile

Differenze tra applicazioni e SO

Applicazioni

- hanno accesso a un insieme ridotto di risorse
- possono utilizzare solo un sottoinsieme delle istruzioni del processore (esecuzione in **modalità utente**)
- non possono decidere autonomamente quando e come accedere alle risorse del sistema (devono chiedere l'intervento del sistema operativo)

SO

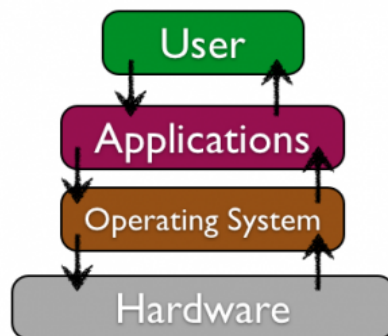
- ha accesso a tutte le risorse del sistema
- può utilizzare tutte le istruzioni del processore (esecuzione in **modalità supervisore**)
- stabilisce in che ordine e come devono essere soddisfatte le richieste di utilizzo delle risorse del sistema

Componenti del SO (1)

- **sistema di gestione del processore**: controlla la CPU vista come risorsa; definisce quale programma è da eseguire in un dato momento (**time sharing**).
- **sistema di gestione della memoria**: carica programmi (istruzioni e dati) nella memoria principale dalla memoria secondaria; controlla l'allocazione della memoria di lavoro ai diversi programmi che possono essere in esecuzione.
- **sistema di gestione delle periferiche**: garantisce l'accesso ai dispositivi di I/O; maschera i dettagli di basso livello e gli eventuali conflitti che possono insorgere quando arrivano contemporaneamente più richieste di accesso alla stessa risorsa; rileva i problemi provocati da guasti hw.

Componenti del SO (2)

- **sistema di gestione dei file (file system)**: consente l'archiviazione e il reperimento dei dati sfruttando i dispositivi che costituiscono la memoria di massa; predispone l'organizzazione logica dei dati sotto forma di “**cartelle**” (**directory**) e **file**.
- **sistema di gestione degli utenti**: fornisce un'interfaccia agli utenti (**interprete dei comandi**); permette agli utenti di accedere in modo semplice ed intuitivo alle funzionalità disponibili; permette la condivisione di risorse da parte di più utenti (sistema multiutente)



La memoria secondaria

La **memoria principale** di un calcolatore è veloce, ma **volatile** (se non alimentata si cancella). È quindi necessario memorizzare i dati su **supporti permanenti**.

Supporti storici sono le schede perforate, i tamburi, i nastri (ancora in uso).

Supporti attuali sono

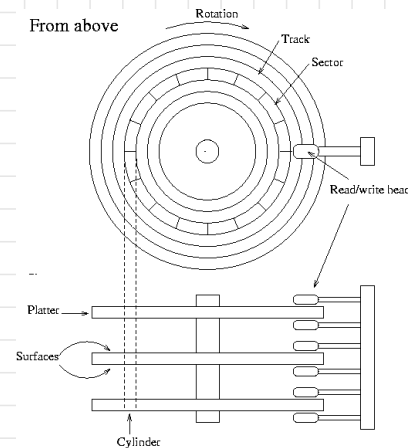
- **dischi magnetici**
- **CD** (Compact **D**isk)
- **DVD** (Digital **V**ersatile **D**isc)
- **BD** (Blu-ray **D**isc)
- **SSD** (Solid **S**tate **D**rive), impropriamente detti ***dischi a stato solido***

I dischi magnetici

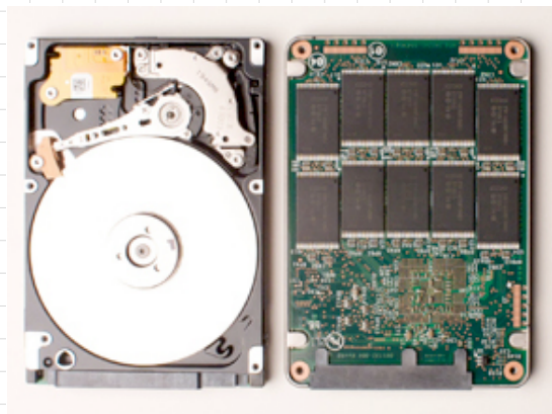
I dischi magnetici sono composti da più **piatti rotanti** suddivisi in **tracce** e **settori**. Nei settori, che hanno un loro indirizzo, sono memorizzati i dati che vengono letti e scritti da **testine di lettura e scrittura**.

I dati memorizzati sono organizzati sotto forma di **file**.

Un file (archivio) è una **sequenza di byte**. Se il file occupa più settori, la loro sequenza è mantenuta specificando gli indirizzi dei settori che lo costituiscono.

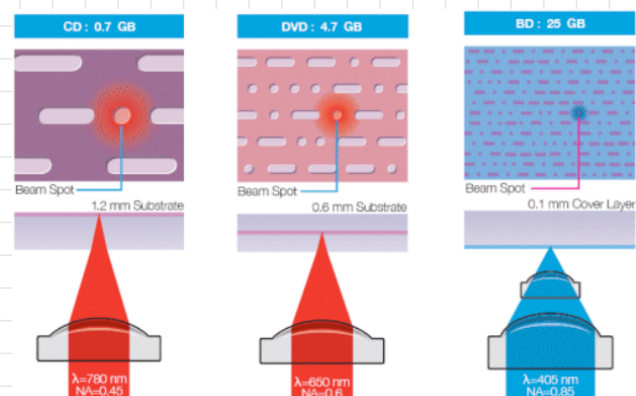


CD, DVD, BD, SSD



Confronto tra un disco magnetico e un *disco a stato solido*. Quest'ultimo è costituito da memorie flash (memoria elettronica non-volatile che può essere cancellata e riscritta).

Confronto tra le densità di tracce per CD, DVD e BD.



Il File System

Per **File System** si intende il modo in cui i file sono gestiti, immagazzinati e organizzati su un dispositivo di archiviazione, come un disco o un DVD.

Esistono diversi file system, creati per diversi sistemi operativi, per diverse unità di memorizzazione e per diversi usi.

La capacità di memorizzare programmi e dati è una caratteristica fondamentale di un calcolatore.

Il file è un meccanismo per strutturare dati che permette di aggregare informazioni elementari in strutture più complesse.

Il File

Il file permette di aggregare e di archiviare in un unico oggetto informazioni che devono essere mantenute insieme, ad esempio istruzioni di un programma o frasi che compongono un testo.

A prescindere dal modo in cui l'informazione viene organizzata, un **file** è comunque formato da una **sequenza finita di byte**.

Un file è **identificato da un nome**, a cui ci si riferisce spesso con il termine inglese **filename**.

I nomi comprendono lettere, numeri e altri caratteri come lo spazio.

Es.: **prova1.txt**

Il File System

- Fornisce un meccanismo per creare e identificare i file, associando lo spazio fisico sulle memorie secondarie a un nome.
- L'effettiva struttura fisica del supporto è trasparente all'utente
- L'occupazione dello spazio libero e di quello ricavato da cancellazioni è gestito automaticamente
- Fornisce le funzioni di lettura e modifica dei dati
- Realizza meccanismi di protezione dei dati che permettono di stabilire chi e come può accedere alle informazioni memorizzate sui supporti
- Si utilizza una struttura gerarchica per rappresentare il contenuto di un disco o altra unità di memorizzazione permanente ad accesso diretto.

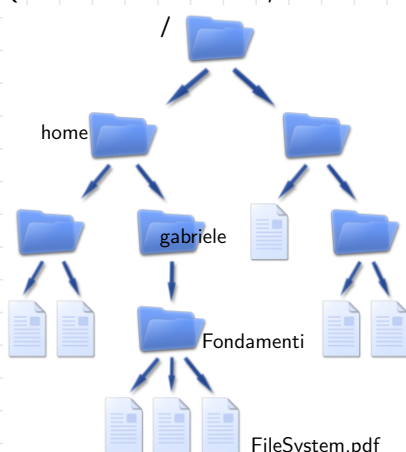
Gerarchia di file e directory

I file sono raggruppati in **cartelle** (**directory**).

Una directory può contenere file e altre directory, formando così una **gerarchia ad albero**.

Ad ogni directory è associato un nome.

In molti SO esiste una directory che contiene tutte le altre detta root (indicata con "/" in Linux)



Il **nome completo** di un file è dato dalla sequenza (**path**) dei nomi delle directory in cui il file è contenuto a partire dalla radice e dal nome del file stesso.

Es. (in Linux):

`/home/gabriele/Fondamenti/FileSystem.pdf`