

Università di Roma Tor Vergata  
Corso di Laurea triennale in Informatica  
**Sistemi operativi e reti**

A.A. 2022-2023

Pietro Frasca

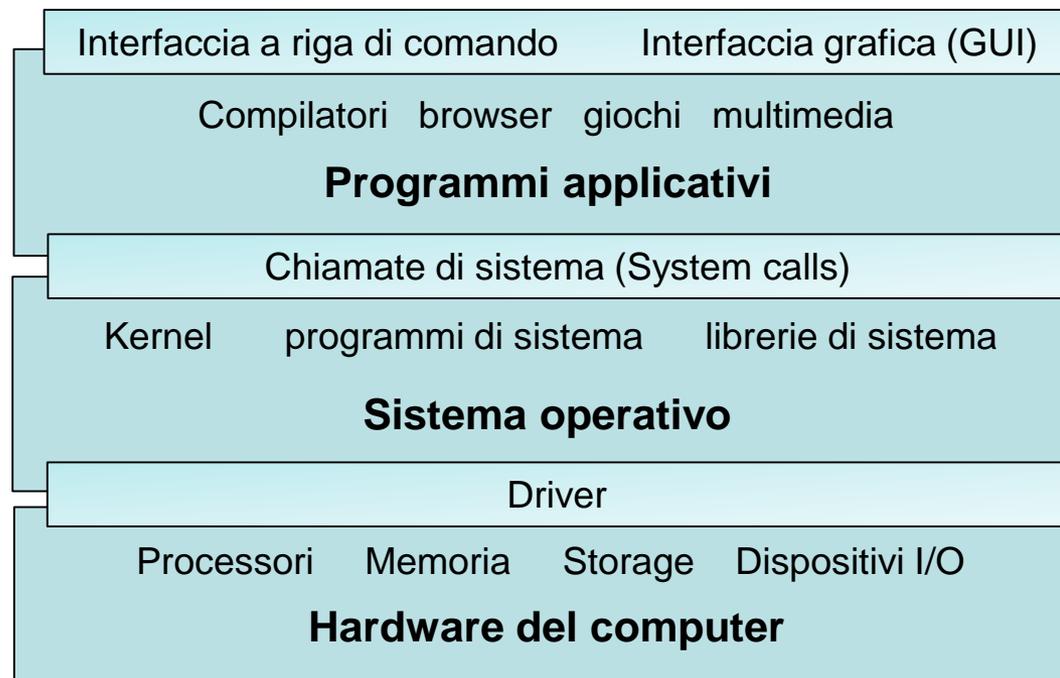
# Lezione 1

Martedì 4-10-2022

# Introduzione

- Un sistema operativo è un software che gestisce l'hardware di un computer. Esso fornisce anche un supporto software ai programmi applicativi e un'interfaccia di comunicazione tra l'utente e l'hardware del computer.
- Nel tempo, sono state progettate e realizzate varie tipologie di sistemi operativi. I sistemi operativi per mainframe sono progettati principalmente per ottimizzare l'utilizzo dell'hardware. I sistemi time sharing sono nati per consentire l'uso del calcolatore a più utenti contemporaneamente. I sistemi operativi per personal computer sono sistemi d'uso generale in grado di eseguire applicazioni di vario tipo come applicazioni di rete, programmi di automazione dell'ufficio, CAD, giochi e molto altro ancora. I sistemi operativi per dispositivi portatili forniscono un ambiente in cui un utente può facilmente usarli ed eseguire programmi, soprattutto applicazioni per la rete Internet.
- Poiché un sistema operativo è un software molto complesso e di grandi dimensioni, è realizzato in moltissimi moduli.

- Ciascuno di questi moduli implementa una parte del sistema che svolge compiti ben precisi.
- Prima di descrivere il funzionamento dei sistemi operativi in dettaglio, è necessario introdurre varie nozioni sull'architettura dei computer e sulle operazioni principali svolte da un sistema operativo.
- Un sistema di elaborazione può essere diviso a grandi linee in tre parti: *l'hardware, il sistema operativo e i programmi applicativi.*

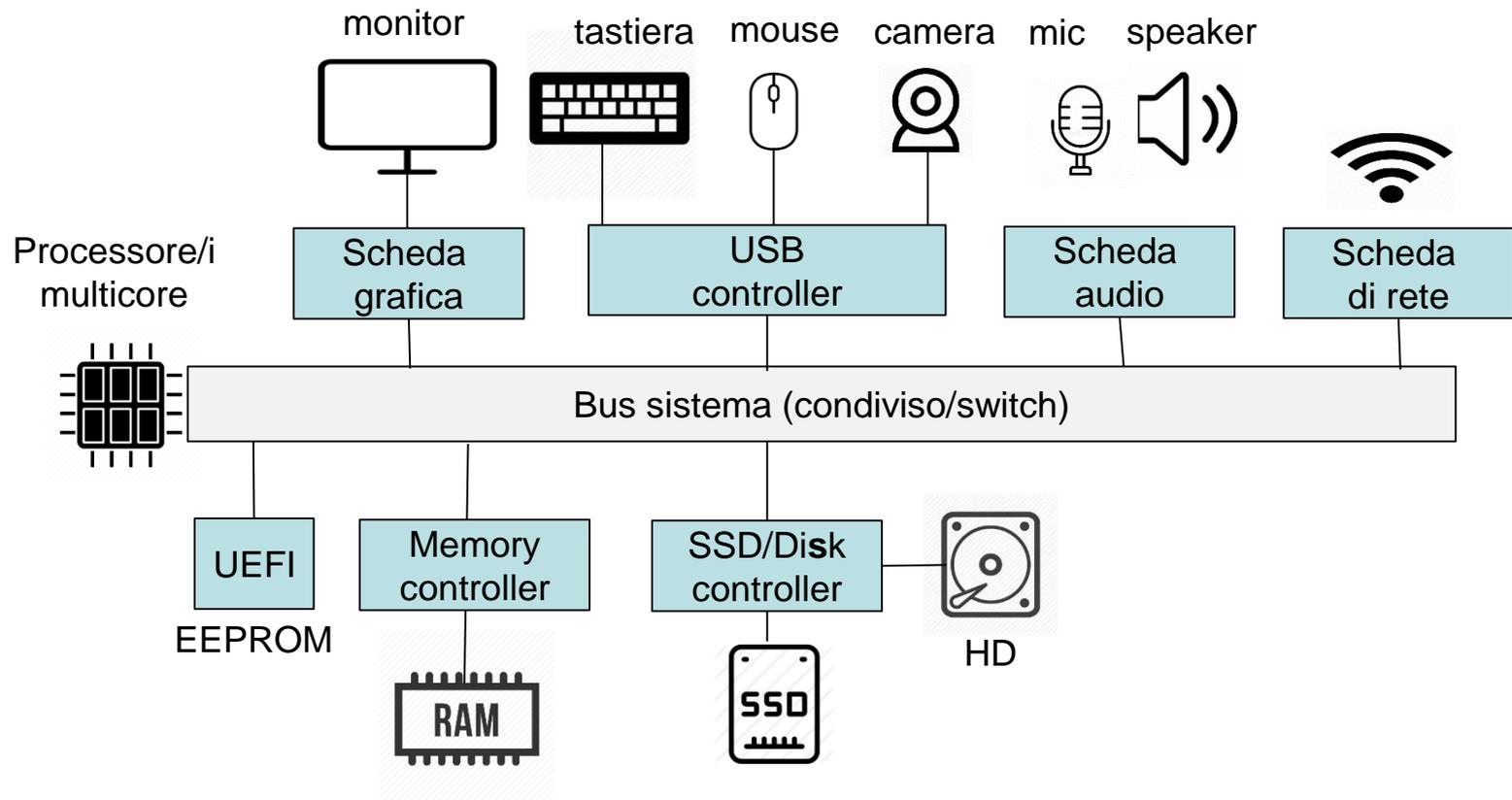


- L'hardware, costituito da processori, memoria, e dispositivi d'ingresso/uscita (I/O), fornisce le risorse di calcolo di base al sistema. I programmi applicativi come word processor, browser, CAD, compilatori sono le risorse software utilizzate dagli utenti.
- Il sistema operativo gestisce e controlla l'hardware coordinando il suo utilizzo tra le varie applicazioni degli utenti.
- L'utente interagisce con il computer in base al tipo d'interfaccia fornita dal sistema operativo. Le interfacce più *user-friendly* e sofisticate si trovano nei sistemi operativi per personal computer. Le interfacce più complicate, dette a *riga di comando*, si usano prevalentemente nei server e sono disponibili anche nella maggior parte di sistemi operativi.
- Recentemente si sono ampiamente diffusi vari tipi di dispositivi mobili, come tablet e smartphone i quali, generalmente, sono collegati alla rete Internet mediante tecnologie wireless, soprattutto con la wi-fi.

- Un computer ha molte risorse hardware come processori, memoria principale e secondaria, dispositivi di I/O, che sono necessarie per l'esecuzione di un programma. Il sistema operativo funge da gestore di tali risorse. Di fronte a numerose e generalmente concorrenti richieste di risorse, il sistema operativo deve decidere come assegnarle ai programmi in modo che il calcolatore possa funzionare in modo corretto ed efficiente.
- A questo punto, è chiaro che il sistema operativo svolge molte funzioni anche se queste variano in numero e complessità secondo il tipo e modello di computer.
- Tuttavia, non è facile dare una definizione precisa di sistema operativo, o meglio stabilire quali componenti software debbano appartenere ad un sistema operativo. Una definizione spesso diffusa è che il sistema operativo è il software funzionante in ogni momento sul computer, che di solito è chiamato **kernel**. Comunque, oltre al kernel, ci sono numerosi programmi di sistema che svolgono compiti più o meno fondamentali per il funzionamento di un computer.

# Funzionamento di un computer

- Un moderno computer di uso generale è costituito da uno o più processori e da un numero di controller di dispositivi connessi mediante bus di sistema comune che forniscono l'accesso alla memoria condivisa.



- Un controller è un modulo elettronico programmabile. Ogni controller gestisce un particolare tipo di dispositivo, ad esempio, scheda grafica, unità SSD o disco, adattatore di rete (ad esempio scheda ethernet o wi-fi), periferiche audio e molti altri tipi di dispositivo.
- I processori e i controller dei dispositivi sono in grado di eseguire istruzioni in parallelo, competendo tra loro per l'accesso alla memoria principale. Pertanto, per garantire un ordinato accesso alla memoria condivisa, nell'hardware del computer è presente un controller che sincronizza l'accesso alla memoria.
- Quando si accende un computer, il processore inizia a eseguire istruzioni a partire da un indirizzo predefinito. A tale indirizzo è mappato un chip di memoria EEPROM (Electrically Erasable Programmable Read-Only Memory), una memoria programmabile cancellabile elettricamente che può essere riscritta più volte pur mantenendo i dati in modo permanente. In questo chip, appartenente all'hardware del computer è memorizzato il firmware del calcolatore. Tale firmware, noto con il termine UEFI (Unified Extensible Firmware Interface) che recentemente ha sostituito il BIOS (Basic Input Output System), è costituito da un insieme di

routine che consentono di inizializzare e testare le componenti hardware del computer. Inoltre, nel firmware è memorizzato l'indirizzo del blocco del disco (o altro dispositivo di memoria secondaria) nel quale è memorizzato un piccolo programma che consente di caricare il kernel del sistema operativo. Una volta in esecuzione, il kernel legge i file di configurazione del sistema e avvia un certo numero di programmi di sistema. Tali programmi prendono il nome di *servizi* (nei sistemi Windows) o *daemon* (nei sistemi Unix-like) di sistema.

- Terminata questa fase, il sistema attende un evento. Il verificarsi di un evento di solito è segnalato da un ***interrupt*** che può essere di tipo hardware o software.
- L'hardware può attivare un'interruzione inviando un segnale al processore, di solito mediante le linee di controllo del bus di sistema.
- Il software può attivare un interrupt eseguendo un'operazione speciale detta ***chiamata di sistema***.
- Quando riceve un interrupt, il processore sospende il flusso d'istruzioni che stava eseguendo e continua l'esecuzione da un indirizzo di memoria prefissato.

- Questa posizione fissa contiene l'indirizzo di memoria iniziale in cui è memorizzata la routine di servizio per il particolare interrupt ricevuto.
- La CPU esegue la routine di servizio dell'interrupt e al termine, riprende la sequenza d'istruzioni in precedenza interrotta.
- La tecnica dell'interrupt è una caratteristica importante di un'architettura di un computer. Ogni architettura ha il proprio funzionamento d'interrupt, ma molte funzioni sono comuni. In ogni caso, gli interrupt devono essere gestiti rapidamente.
- Generalmente, poiché il numero degli interrupt è limitato, si usa una tabella di puntatori a routine d'interrupt. Gli elementi della tabella, detta **vettore delle interruzioni**, contengono gli indirizzi iniziali delle routine di servizio d'interruzione per i vari dispositivi.
- Ogni dispositivo è identificato da un numero intero, che indicizza la tabella quando avviene una richiesta d'interruzione, per fornire l'indirizzo della routine di servizio d'interruzione per il dispositivo.
- La funzione d'interrupt deve anche salvare l'indirizzo dell'istruzione interrotta.

- E' inoltre necessario salvare lo stato attuale del processore e quindi ripristinare tale stato prima di tornare al flusso di esecuzione interrotto.
- Al termine della routine di servizio dell'interrupt, il flusso interrotto riprende come se l'interruzione non fosse avvenuta.

## **Struttura della memoria**

- Il processore legge le istruzioni dalla memoria principale, pertanto è necessario che i programmi siano caricati in memoria prima di essere eseguiti.
- I computer generalmente eseguono la maggior parte dei programmi prelevando le istruzioni dalla memoria principale.
- Tutti i tipi di memoria sono gestiti come un vettore di byte in cui è possibile eseguire operazioni di scrittura e lettura. Ogni byte ha un proprio indirizzo. L'istruzione di lettura trasferisce un byte o una parola (gruppo di byte) da una particolare locazione della memoria principale a uno dei registri interni della CPU, mentre l'istruzione di scrittura, viceversa, copia il contenuto di uno dei registri della CPU a una locazione specifica di memoria principale.

- Un tipico ciclo di esecuzione di un'istruzione, eseguita su un sistema con una semplice architettura, prima preleva un'istruzione dalla memoria e la memorizza nel registro istruzioni. L'istruzione è poi decodificata ed eventualmente può richiedere il trasferimento di alcuni operandi dalla memoria e copiati in qualche registro interno della CPU. Dopo l'esecuzione dell'istruzione sugli operandi, il risultato può essere memorizzato in qualche locazione della memoria principale.
- Per migliorare le prestazioni, molti processori attuali hanno un architettura che consente l'esecuzione di più istruzioni simultaneamente. Ad esempio, un processore può avere unità separate di *prelievo (fetch)*, *decodifica*, ed *esecuzione*, in modo che quando esegue l'istruzione  $n$ , allo stesso tempo potrebbe anche decodificare l'istruzione  $n + 1$  e recuperare l'istruzione  $n + 2$ .
- La memoria principale è di solito troppo piccola perché contenga tutti i programmi e i dati necessari in modo permanente.
- La memoria principale è un dispositivo di memorizzazione volatile che perde il suo contenuto quando non è alimentata elettricamente.

- Così, la maggior parte di computer sono dotati di dispositivi di memorizzazione secondaria. Il requisito principale per la memoria secondaria è che sia in grado di contenere grandi quantità di dati in modo permanente. I dispositivi più comuni di memoria secondaria sono i dischi magnetici e gli SSD (Solid State Drive). Altri dispositivi di memorizzazione sono le memorie cache, DVD, CD-ROM, nastri magnetici.
- Le principali differenze tra i vari tipi di memoria sono in velocità di accesso, dimensioni, volatilità e costo.
- Le cache sono usate per migliorare le prestazioni in cui esiste una grande disparità in tempo di accesso e velocità di trasferimento tra due componenti.

## Struttura di I/O

- Una grande parte del sistema operativo è dedicata alla gestione dei dispositivi di I/O.
- Un computer *general-purpose* è composto da uno o più processori e da più controller di dispositivi che sono connessi attraverso un bus comune.

- Ogni controller si occupa di uno specifico tipo di dispositivo.
- Un controller di dispositivo è dotato di un buffer locale di memorizzazione e di una serie di registri.
- Un controller di dispositivo è responsabile di trasferire i dati tra i dispositivi periferici che esso controlla e il suo buffer locale.
- In genere, nei sistemi operativi è presente un **driver** per ogni controller di dispositivo. Un driver è un modulo software che fornisce al resto del sistema operativo un'interfaccia per la gestione del dispositivo. Per avviare un'operazione di I/O, il driver carica dati (codici operativi) in specifici registri del controller del dispositivo. Il controller del dispositivo, a sua volta, esamina il contenuto di questi registri per determinare l'azione da fare, ad esempio, leggere un byte da un mouse. Il controller avvia il trasferimento dei dati dal dispositivo al buffer locale. Una volta che il trasferimento dei dati è completo, il controller del dispositivo informa il driver di periferica tramite interrupt che ha terminato il suo compito. Il driver restituisce poi il controllo al sistema operativo, restituendo i dati o un puntatore ai dati in caso di un'operazione di lettura. Per altre operazioni, il driver di periferica restituisce informazioni di stato.

- Questa tecnica di trasferimento dati basata su interruzione I/O è idonea per piccole quantità di dati. Tuttavia, produce molto overhead quando si usa per trasferire grandi blocchi di dati tra memoria e alcuni tipi di periferiche come ad esempio dischi o schede di rete.
- Per risolvere questo problema, è utilizzato l'accesso diretto alla memoria (DMA). Dopo aver impostato buffer, puntatori, e contatori per il dispositivo di I/O, il controller del dispositivo trasferisce direttamente un intero blocco di dati tra il proprio buffer e la memoria, senza alcun intervento da parte della CPU. Un solo interrupt è generato per ogni blocco, per informare il driver di periferica che l'operazione è stata completata, piuttosto che un interrupt per ogni byte generato per periferiche a bassa velocità. Mentre il controller di periferica esegue queste operazioni, la CPU è disponibile per svolgere altri compiti.
- Alcuni sistemi con architetture hardware più sofisticate utilizzano switch anziché il bus condiviso. In questi sistemi, più componenti possono comunicare tra loro contemporaneamente, invece di competere per l'uso del bus condiviso. In questo caso, il DMA è ancora più efficace.