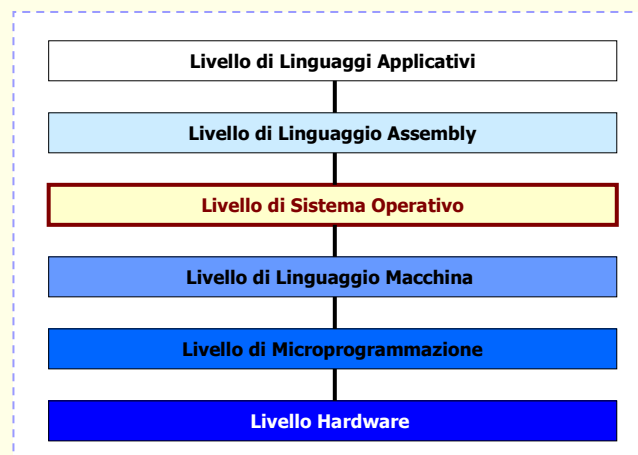


# Sistemi Operativi

## Livello di Sistema Operativo



## Livello di Sistema Operativo

---

- **Operating System Machine (OSM)**
  - ❖ Livello che implementa il sistema operativo
  - ❖ Livello che aggiunge nuove istruzioni e caratteristiche di più alto livello rispetto a quelle fornite dal livello ISA
- **Instruction Set di OSM**
  - ❖ L' insieme delle istruzioni di OSM comprende le istruzioni del livello ISA e nuove istruzioni aggiunte dal Sistema Operativo, dette "**chiamate di sistema (system call)**"
    - Le **istruzioni del livello ISA** vengono eseguite direttamente dal livello sottostante
    - Le **istruzioni di chiamata** (istruzioni che servono a richiedere un servizio al S. O.) sono interpretate dal sistema operativo

## Sistema Operativo

---

- I. **Introduzione**
  - ❖ **Funzionalità dei sistemi operativi**
  - ❖ **Evoluzione dei sistemi operativi**

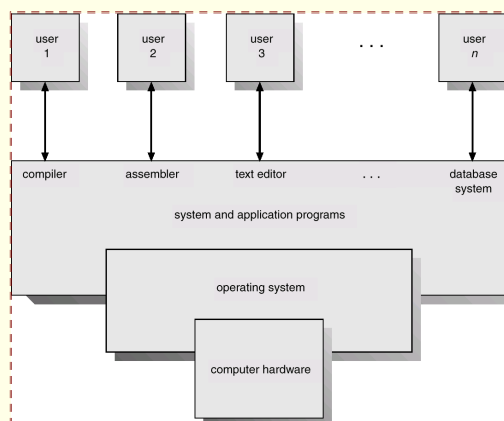
# Sistema Operativo

## □ Sistema Operativo (S. O.)

- ❖ Insieme di programmi che operano sull' hardware di un sistema di elaborazione con l'obiettivo di
  - semplificare lo sviluppo di programmi
    - rendere indipendente il software applicativo dall'hardware (trasparenza)
  - gestire le risorse hardware
    - usare le risorse HW (es. I/O) tramite chiamate di sistema
    - evitare conflitti di accesso alle risorse
    - definire criteri con cui assegnare una risorsa a fronte di più richieste contemporanee
  - garantire la protezione, sicurezza e tolleranza ai guasti
    - garantire la riservatezza dei dati
    - proteggere i dati da accessi non autorizzati
    - gestire eventi anomali

# Sistema Operativo

**Il S.O. coordina e controlla l'uso dei dispositivi da parte dei programmi di applicazione per gli utenti**



# Sistema Operativo

## □ **Funzionalità dei S.O.**

- ❖ Insieme di strumenti software che realizzano due funzioni fondamentali apparentemente non correlate
  1. nascondere al programmatore i dettagli dell'hardware sottostante e presentare una **interfaccia** amichevole all'utente (**Macchina Astratta**)
  2. controllare l'utilizzo di tutte le risorse dell'elaboratore (**Gestore delle risorse**)
- ❖ Obiettivi dei S. O.
  - Efficienza
    - Obiettivo è il funzionamento efficiente del sistema di elaborazione (sistemi mainframe multiuser)
  - Semplicità
    - Obiettivo è la facilità d'uso da parte dell'utente (personal computer)

# Sistema Operativo

## □ **S.O. come Macchina Astratta**

- ❖ Il software virtualizza le caratteristiche dell' hardware sottostante, offrendo la visione di una macchina astratta (virtuale) più potente e più semplice da utilizzare di quella fisicamente disponibile
- ❖ Funzioni di servizio tipicamente fornite del S.O. per svolgere il ruolo di **interfaccia**
  - Creazione ed esecuzione di applicazioni
    - S.O. offre servizi per la stesura dei programmi (editor e debugger) e per il caricamento
  - Accesso ai dispositivi di I/O
    - S.O. maschera i dettagli di basso livello del controllo delle periferiche, come la gestione dei segnali per trasferimento dati ...
  - Archiviazione di dati e programmi
    - S.O. fornisce una organizzazione logica dei dati sotto forma di cartelle e file per particolari dispositivi di memorizzazione come i dischi rigidi
  - Controllo di accesso alle risorse
  - Gestione dei malfunzionamenti

# Sistema Operativo

## □ S.O. come Gestore delle Risorse

- ❖ Un sistema di elaborazione è un insieme di **risorse** finalizzate a trasferire dati, memorizzarli ed elaborarli
- ❖ La gestione di una risorsa comprende
  1. tener traccia dello stato della risorsa
  2. stabilire la politica che determina a quale processo assegnare la risorsa, quando e per quanto tempo
  3. allocare e deallocare la risorsa
- ❖ **Risorsa**
  - Oggetto usato da un **processo** e che condiziona il suo stato di avanzamento
    - Risorse: memoria, CPU, dispositivi di I/O, programmi, dati
- ❖ **Processo = Programma in esecuzione (E, S)**
  - E: Codice eseguibile del programma e dati
  - S: Stato del processo
    - insieme dei valori contenuti in memoria e nei registri della CPU durante l'esecuzione del programma

**Programma: oggetto statico e invariante nel tempo**  
**Processo: oggetto dinamico, che evolve nel tempo**

# Evoluzione dei S.O.

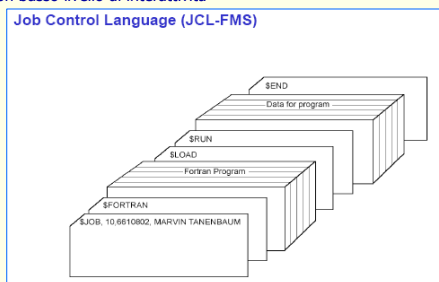
## □ Assenza di S.O.

- ❖ Programmi in codice macchina caricati attraverso il dispositivo di input (es. lettore schede) ed avviati dall'utente
  - Meccanismo manuale di prenotazione della macchina al fine di determinare un ordine di caricamento ed esecuzione dei vari programmi (**job**)
    - Spreco di tempo macchina ogni volta che un programma terminava in anticipo rispetto al tempo prenotato
    - Tempo di preparazione dell'ambiente di supporto all'esecuzione di un programma spesso elevato (compilazione del codice, montaggio di nastri, installazione di funzioni di libreria)

## Evoluzione dei S.O.

### □ Elaborazione a lotti (batch) (Anni 50-60)

- ❖ Impiego di un programma detto **"monitor"** (antenato dei S.O.) in grado di automatizzare l'avviamento e l'esecuzione sequenziale di un lotto di programmi
  - Eliminazione dell'accesso diretto alla macchina da parte dell'utente
  - Mancanza di interattività e tempo di latenza elevato
- ❖ Uso di un apposito linguaggio, detto **Job Control Language (JCL)**, per la descrizione delle operazioni che il monitor doveva compiere
  - Ottimizzazione dello sfruttamento delle risorse con accurata distribuzione del tempo di calcolo tra i vari programmi da eseguire
  - Applicazioni con basso livello di interattività

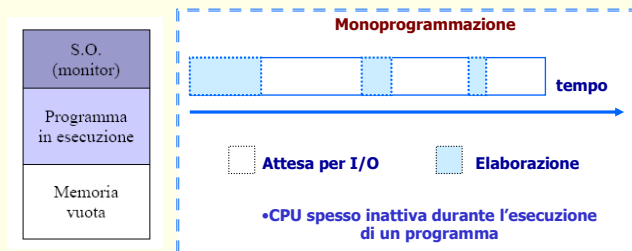


## Evoluzione dei S.O.

### □ Sistemi Monoprogrammati

#### ❖ Monoprogrammazione

- In memoria centrale risiede, oltre al sistema operativo, un solo programma
  - MS-DOS

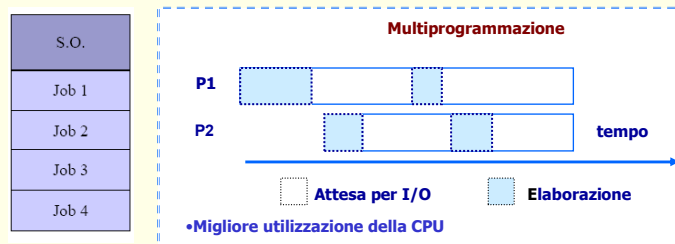


## Evoluzione dei S.O.

### ❑ Sistemi Multiprogrammati (Anni 60)

#### ❖ Multiprogrammazione o Multitasking

- In memoria centrale risiedono più programmi utente
- Il S.O. deve scegliere quali job devono essere caricati in memoria centrale e quale tra essi deve andare in esecuzione
  - Uso del processore durante i periodi di I/O di un job per eseguire altri job
  - Gestione della memoria



## Evoluzione dei S.O.

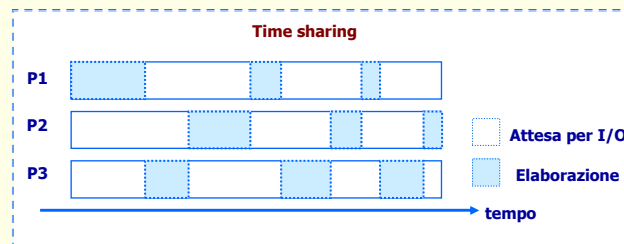
- ❖ La multiprogrammazione consente una **elaborazione batch efficiente**
  - La commutazione fra i programmi residenti in memoria può avvenire in modo efficiente, essendo il tempo di accesso alla memoria centrale inferiore a quello delle memorie di massa
  - Il processore non viene lasciato inattivo (**idle**) durante operazioni di I/O molto lunghe
  - Migliora il throughput, numero di job eseguiti nell'unità di tempo
- ❖ La multiprogrammazione non consente modalità di esecuzione interattiva efficienti

## Evoluzione dei S.O.

### □ Sistemi a partizione di tempo (time-sharing) (anni 70)

#### ❖ Time-sharing

- Suddivisione del tempo di CPU in unità elementari, dette **quanti**, da assegnare secondo opportune politiche, a tutti i programmi che definiscono il carico elaborativo del sistema
  - Allo scadere del quanto, il programma corrente viene interrotto e l'esecuzione passa ad un altro programma, anche in assenza di operazioni di I/O
- La durata del quanto incide sia sulle prestazioni del sistema che sulla efficacia del parallelismo di esecuzione dei programmi
- La partizione di tempo consente la **elaborazione interattiva**
  - Più utenti hanno accesso simultaneo al calcolatore



## Evoluzione dei S.O.

### □ Sistemi Operativi Monoutente

- ❖ Sistemi multitasking (capacità di eseguire più processi contemporaneamente) per desktop
- ❖ Adottano tecnologie sviluppate per S.O. più complessi: poiché l'utilizzo della CPU non ha un ruolo determinante, i criteri progettuali mirano alla comodità e prontezza d'uso per l'utente
  - Microsoft Windows, Linux, Macintosh OS

### □ Sistemi Operativi Multiutente

- ❖ Sistemi a partizione del tempo, più complessi dei sistemi multitasking
  - UNIX

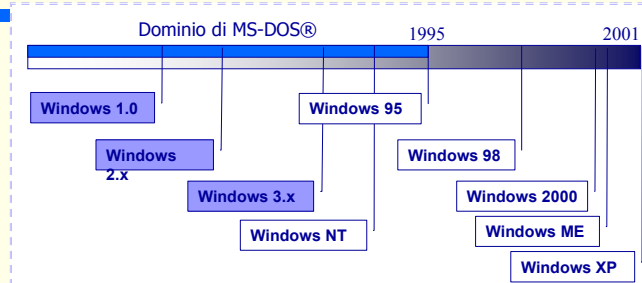
### □ Sistemi Operativi di Rete

- ❖ S.O. che fornisce la possibilità di comunicazione e di condivisione di file tra computer collegati in rete
  - Ogni computer opera indipendentemente dagli altri

### □ Sistemi Operativi Distribuiti

- ❖ S.O., condiviso da una rete di calcolatori, che effettua un controllo su scala globale delle risorse di rete

## Evoluzione dei S.O.



### ❖ Windows

- Multitasking
  - Processori Intel dall'80386 in poi, alcune versioni anche su processori SPARC e Alpha
- User-friendly
  - Interfaccia grafica evoluta, cartelle e risorse
- Plug-and-play
  - Connessione facile di dispositivi esterni
- Facilità d'uso
  - Utilizzabile sia da utenti naive che da utenti esperti
- Elevato numero di applicazioni software
  - La maggior parte del software è scritto per Windows