

# Sistemi operativi

---

Un sistema di elaborazione dati può essere così rappresentato:



- **dispositivi fisici:** (CHIP, alimentatori, memorie, ...)
- software primitivo che controlla i dispositivi del livello inferiore mediante codice di **microprogramma** (generalmente su **ROM**)
- **linguaggio macchina:** istruzioni elementari per muovere dati, eseguire calcoli e comparare valori; su questo livello i dispositivi I/O sono controllati da valori caricati in speciali registri (registri d'interfaccia)
- **sistema operativo vero e proprio**, che nasconde la complessità dei livelli inferiori e fornisce al programmatore un insieme di istruzioni di alto livello (**KERNEL MODE**)
- **software di sistema:** interprete dei comandi (shell), web browser, editor, interfaccia grafica (**USER MODE**)
- **programmi applicativi** (Word, Excel, Eclipse, ... )

## Concetti introduttivi

Un **moderno sistema** di calcolo comprende:

- una (ma generalmente più) **CPU**
- una memoria centrale (**core memory**)
- **cache**
- **clock**
- **terminali**
- **dischi**
- interfacce di **rete**
- input/output **testuale**
- input/output **multimediale**
- dispositivi di interfaccia ad **altri sistemi**

**Software** che gestisca tutte le parti del sistema e che fornisca un'opportuna interfaccia all'utente:

- Il **Software**:
  - **programmi di sistema** (o di base) che rendono facilmente disponibile all'utente le potenzialità offerte dalla macchina (hardware). Una parte consistente del software di base è costituito dal **Sistema Operativo**
  - **programmi applicativi**
- Il **S.O.** può essere considerato come una interfaccia tra hardware e utente
  - Il S.O. deve rendere la macchina trasparente all'utente
  - Quindi saper utilizzare una macchina significa realmente conoscere e saper utilizzare il suo S.O.
  - Macchine di architettura molto diversa a livello hardware possono utilizzare lo stesso S.O. esso svolge le stesse funzioni e si presenta all'utente nello stesso modo

## Classificazione dei S. O.

L'evoluzione delle architetture dei calcolatori segna anche l'evoluzione dei sistemi operativi.

In ordine cronologico:

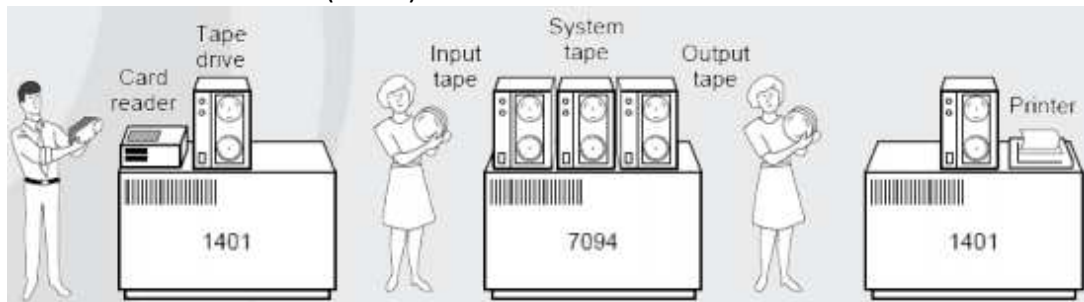
1. **Dedicati**
2. A lotti (**batch**)
3. **Multiprogrammazione**
4. **Interattivi** (Time-Sharing)
5. Quarta generazione (**Personal computer**)
6. Quinta generazione (**Mobile**)

## Prima generazione 1945-1955

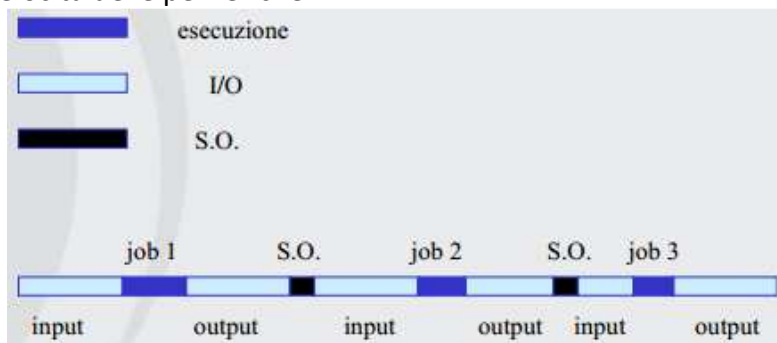
- I calcolatori erano basati sulle valvole,
- praticamente non vi era S.O.
- Sistemi dedicati quasi esclusivamente per il calcolo numerico/scientifico
  - Howard Aiken
  - John von Neumann
  - Presper Eckert
  - William Manchley
  - Konrad Zuse
- Caratteristiche della 1° generazione
  - un unico staff per
    - progetto
    - costruzione
    - programmazione
    - esecuzione
    - manutenzione
  - tutti i programmi erano scritti in **linguaggi macchina** (non esisteva l'assembler)
  - i linguaggi di **alto livello** di programmazione erano **sconosciuti**
  - i **sistemi operativi erano assenti** (controlli fisici d'integrità delle valvole)
  - negli ultimi anni, si introdussero i primi sistemi a **schede perforate**

## Seconda generazione: 1955-1965

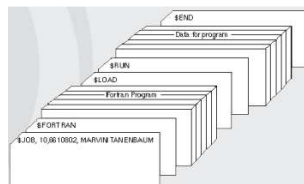
- S.O. **batch** (gestione a lotti)
  - è determinata dall'introduzione di importanti innovazioni tecnologiche: **transistor**
  - i calcolatori potevano essere prodotti e **venduti** a clienti
  - nascono figure professionali distinte nel processo di **produzione di un programma (JOB)**
  - **costi di milioni di dollari** (solo Università e grandi industrie)
- un programmatore **scrive il programma** su carta (Fortran, Assembler), e **perfora le schede**;
- le riceve un operatore che **le inserisce in una coda dei programmi**, le immette nel calcolatore e, se necessario, carica l'opportuno compilatore;
- l'operatore consegna poi al programmatore l'output finale
- **problema**: enorme spreco di tempo
  - Caricamento nastro compilatore, esecuzione compilatore, scaricamento compilatore, caricamento assembler, esecuzione assembler, scaricamento assembler, caricamento oggetto, esecuzione programma ..
- **soluzione**: sistemi a lotti (batch)



- Nati per sfruttare meglio la velocità crescente delle macchine: eliminano i tempi morti tra programmi successivi di utenti diversi automatizzando le operazioni manuali
- **Un insieme di lavori (jobs) viene accorpato in un lotto (batch)** tramite un calcolatore ausiliario e trasferito su una unità di ingresso veloce (nastro)
- Ogni **lavoro** viene caricato da un operatore **ed eseguito in sequenza senza interruzione** fino al termine
- L'output viene scritto su un secondo nastro invece di essere stampato (stampa off line)
- La **CPU viene ancora sottoutilizzata** perché, durante le operazioni di I/O deve adeguarsi alla bassa velocità delle periferiche

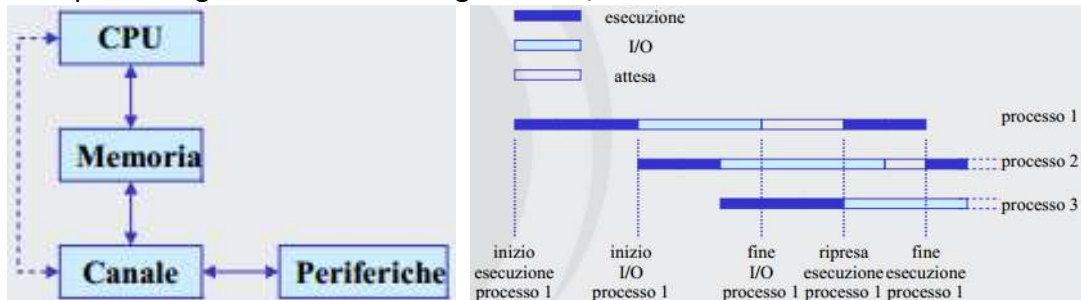


- Si noti che servono due sistemi distinti.



### Terza generazione: 1965-1980

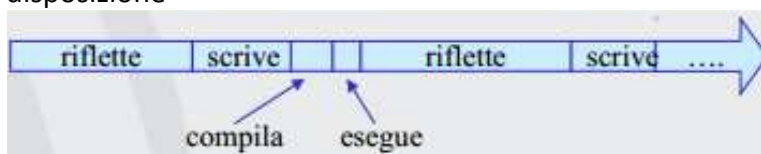
- Circuiti integrati
- Multiprogrammazione
  - Con la multiprogrammazione (multitasking) più di un programma viene caricato in memoria contemporaneamente. In genere si parla di **processo**: sequenza di istruzioni eseguite dalla CPU per portare a termine un programma
  - **Le operazioni di I/O di un programma sono sovrapposte temporalmente all'esecuzione delle istruzioni di un altro programma**
  - ciò è possibile grazie al canale che gestisce l'I/O



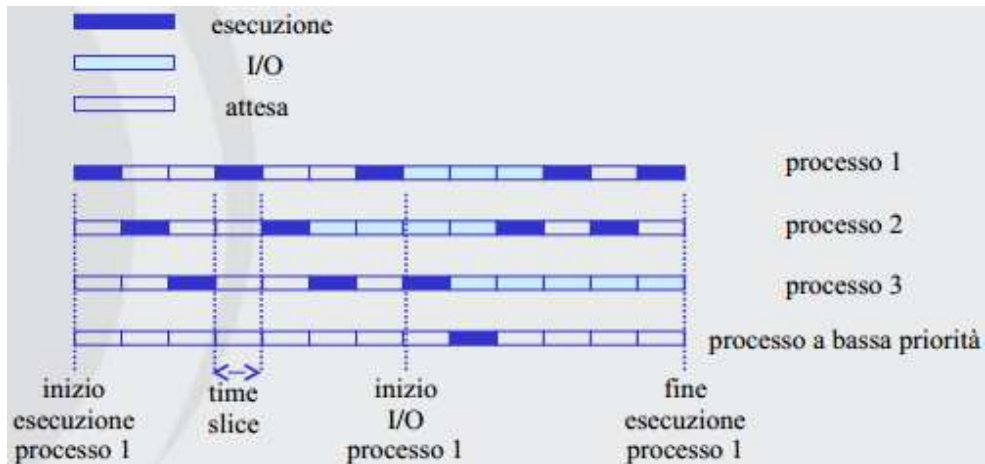
- Gestione dei processi
  - Il **multitasking** non è in grado di distinguere fra programmi che richiedono un uso frequente delle periferiche, da quelli che richiedono l'utilizzo per lunghi tempi della CPU (es: programmi di elaborazione matematica). Se uno di questi programmi ottiene la **CPU non la rilascia spontaneamente**, bloccando gli altri programmi.



- S.O. Time Sharing
  - Nei S.O. interattivi (Time sharing) **il tempo di utilizzo della CPU viene suddiviso dal sistema operativo in fette** (time slice di durata 50-500 ms)
  - Ogni processo in memoria riceve a turno l'uso della CPU per una unità di tempo
  - **Al termine del time slice il controllo torna comunque al S.O. che decide a chi affidare la CPU**
  - Sono S.O. usati soprattutto per lo sviluppo di software in **ambiente multiutente**
  - L'utente lavora al terminale e ha l'impressione di avere la macchina a sua completa disposizione



- L'**overhead** per la gestione della CPU può diventare significativo con molti processi attivi oppure con time slice molto piccolo
- Dopo i primi sistemi CTSS (MIT) (Compatible Time-Sharing System) sono derivati gli attuali sistemi UNIX (Ken Thompson)



- Batch e Time Sharing

	Batch	Time Sharing
Obiettivo principale	Massimizza l'uso del processore	Minimizza il tempo di risposta
Sorgente dei comandi al Sistema Operativo	Job control language	Comandi da terminale

#### Quarta generazione 1980

- Sono basati sulla tecnologia VLSI (**Very Large Scale Integration**)
- Due sistemi operativi hanno dominato la scena dei Personal Computer e delle Workstation
  - **MS-DOS** (Microsoft) con il derivato Windows
    - L'utilizzo della CPU non è più critico (1 solo utente)
    - Non vi è concorrenza (primi esemplari)
    - Non vi sono meccanismi di protezione (vedi virus)
  - **UNIX** (Bell Labs)

#### Quinta generazione 1990

- Sistemi mobile
- Personal Digital Assistant (PDA)
- Telefoni cellulari
- Problemi:
  - Memoria limitata
  - Processori lenti
  - Schermo piccolo

- Reti
- La crescita di reti di PC e di Web Site ha permesso lo sviluppo di
  - **Network Operating Systems**
    - l'utente "vede" più calcolatori, può accedere a macchine remote e copiare file; ogni macchina ha il suo sistema operativo locale
  - **Distributed Operating Systems**
    - appare all'utente come un tradizionale sistema monoprocesso anche
    - se è composto da più processori; l'esecuzione di programmi può essere
    - a carico di macchine diverse (anche in parallelo)
    - Due casi: gli N processori non condividono o condividono clock e/o
    - memoria
- Sistemi operativi **Real Time**
  - Sistemi operativi al servizio di una specifica applicazione che ha dei **vincoli precisi nei tempi di risposta**
  - Il S.O. deve garantire un tempo massimo entro il quale mandare in esecuzione un programma a seguito di un evento
    - Gestione di strumentazione
    - Controllo di processo
    - Gestione di allarme
    - Sistemi transazionali (banche, prenotazioni)
  - In generale si ha un sistema real-time quando il tempo di risposta dalla richiesta di esecuzione di un processo al completamento della stessa è sempre minore del tempo prefissato