

Storia dei sistemi operativi

I sistemi operativi hanno sempre avuto uno stretto rapporto con l'architettura dei computer so cui sono eseguiti.

Possiamo infatti dividere la storia dei sistemi operativi e delle architetture in diverse generazioni:

Prima generazione (1945-55): valvole termoioniche

I progressi nella costruzione di computer digitali, dopo [Babbage](#) furono minimi fino alla [II Guerra Mondiale](#) e nacque il primo computer digitale. Nello stesso periodo vennero costruiti anche:

- **Z3** fatto di relè elettromeccanici.
- **Colossus**
- **Mark 1**
- **ENIAC**

Tutti molto primitivi, impiegavano secondi per svolgere le operazione più semplici.

In questo periodo singoli gruppi di persone progettavano, costruivano, azionavano e mantenevano le macchine che erano programmate esclusivamente in linguaggio macchina o cablando fili. Non esistevano linguaggi moderni e sistemi operativi.

Nei primi anni '50 ci fu un miglioramento con l'introduzione delle schede perforate, i programmi erano scritti su schede e leggerli invece di usare schede di collegamento.

Seconda generazione (1955-65): transistor e sistemi batch

Ci fu un cambiamento radicali a metà degli anni '50 con l'avvento dei transistor, i computer stavano diventando sempre più affidabili e per la prima volta la separazione fra progettisti, operatori, programmatori, ecc.. diventò netta.

 **Scrivere un programma su questi Mainframe:**

Per eseguire un **job** il programmatore doveva:

1. Scrivere un programma su carta.

2. Perforarlo sulle schede.
3. Portare il mazzo di schede in sala input.
4. Passarlo agli operatori in attesa che l'output fosse pronto.
5. L'operatore poi stampava l'output.

Per cercare di diminuire il tempo di questo processo vennero adottati dei **sistemi batch**.

Funzionamento di un sistema batch:

1. Raccogliere una cassetta piena di job.
2. Le schede venivano lette e scritte sul nastro magnetico.
3. I computer tramite un programma leggevano il nastro ed eseguivano le istruzioni.
4. Una volta finito si aveva un nastro di output.
5. L'output veniva portato in sala stampa.

La struttura di un job era composta da schede:

Scheda \$JOB: Specificava il tempo massimo di esecuzione, il numero utente e il nome del programmatore.

Scheda \$FORTRAN: Indica al Sistema Operativo il compilatore, FORTRAN.

Scheda \$LOAD: Indica di caricare il programma appena compilato.

Scheda \$RUN: Indica di eseguire il programma con i dati.

Scheda \$END: Sanciva la fine di un job.

I grandi computer di seconda generazione erano usati per risolvere calcoli matematici, erano programmati in **FORTRAN** o **Assembly** e i tipici sistemi operativi erano **FMS**, **IBSYS** o il **SO di IBM**.

Terza generazione (1965-80): IC e multiprogrammazione

Nei primi anni '60 abbiamo due tipi di computer principalmente:

- *word-oriented*: Usati per calcoli numerici e scientifici.
- *character-oriented*: Usati in banche.

Era costoso però mantenere due tipi diversi di computer nella produzione e cominciavano a riscontrarsi anche i primi **problemi con la retrocompatibilità**.

IBM volle lavorare su entrambi i problemi contemporaneamente con **IBM system/360** che permetteva sia lo svolgimento di qualsiasi calcolo e le istruzioni erano compatibili tra tutti i modelli IBM.

Non solo furono i primi ad adottare la tecnologia **IC (Circuiti Integrati)** che migliorava il rapporto qualità/prezzo, quindi venne a crearsi una **famiglia di computer**.

OS/360 era un disastro.

Perché doveva essere un SO in grado di lavorare con macchine piccole (che sostituevano le 1410) con macchine grandi (che sostituevano le 7094), doveva funzionare con computer con poche periferiche, ma anche con molte e doveva operare in diversi indirizzi.

Risultato: Un **SO estremamente complesso e pieno di errori** che venne man mano ripulito con il corso degli anni.

Ogni versione correggeva alcuni errori, ma ne aggiungeva altri.

Vennero sviluppate anche altre tecnologie tra cui la **Multiprogrammazione**.

Semplicemente viene **partizionata la memoria in tanti job**, quando un job è messo in attesa per il completamento dell' I/O, un **altro poteva usufruire della CPU**.

Venne aggiunta un'altra caratteristica lo **Spooling**, cioè la capacità di leggere job sul disco appena portati in sala macchine.

Un altro problema:


I tempi tra inserimento in coda di un job e la ricezione di output erano **lunghi**

La risposta fu' una variante della multiprogrammazione, il **Timesharing** semplicemente **la CPU viene assegnata a quegli utenti che stanno svolgendo delle operazioni**, mentre in background, quando non è impiegata, smaltisce job batch.

Il primo sistema di Timesharing era il **CTSS (Compatible Timesharing System)**.

Uno dei primi SO che introduceva il concetto di Timesharing e che lo voleva portare a livelli superiori era **Multics** il quale fu' di ispirazione per **UNIX**.

Il creatore, **Ken Thompson**, aveva come idea una versione **monoutente** di Multics. Fu' un successo tale che cominciarono a nascere molte altre versioni e il caos era inevitabile. Fortunatamente la IEEE sviluppo uno standard chiamato **POSIX** ancora oggi funzionante e molti SO UNIX seguono questo standard.

 **Tanenbaum si basò su POSIX per creare MINIX che ne era essenzialmente una versione didattica, il desiderio di una versione di produzione di MINIX portò uno studente, Linus Torvalds, a scrivere LINUX.**

Quarta generazione (1980 a oggi): Personal computer

Con lo sviluppo dei circuiti **LSI** (Large Scale Integration) i **chip** divennero sempre più veloci e di piccole dimensioni.


Rimaneva sempre il problema di trovare un SO che potesse essere utilizzato da tutte le persone.

E in questo periodo **Apple e Microsoft** introdussero i primi SO basati su **GUI**, infatti:

- Viene inventato **Apple Lisa**: Primo computer con GUI;
- Viene sviluppato **Windows** un ambiente grafico per **MS-DOS**.

Da allora, però, LINUX rimase sempre in testa alle classifica di SO usati, specialmente:

- Server
- Mobile: Con **Android**.

 **Anche MINIX viene continuato ad essere usato, infatti MINIX3 è stato adottato da Intel per il suo motore di gestione ed è ora presente in molti computer, server e laptop.**