

# ONE PIECE





# 3 - ENIES LOBBY

Enies Lobby

Nico Robin

Ohara

Imperatori

Gen. - Feb. - Mar.

Introduzione

Storia

Architetture

Processi





# NICO ROBIN

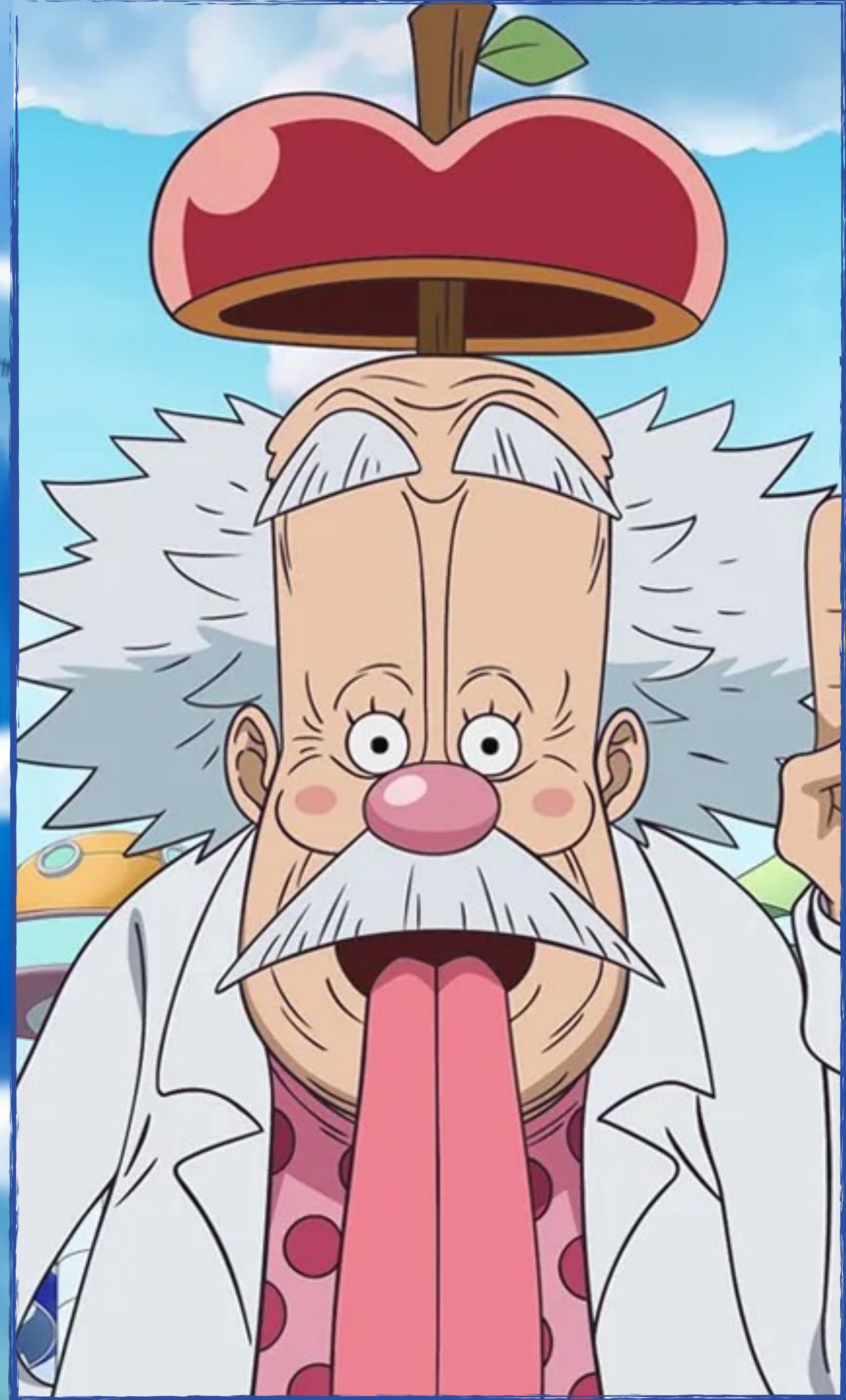




# MEMORIA

*"L'astrazione è uno degli strumenti più potenti dell'informatica."*

(Edsger W. Dijkstra)





# Storia dei Sistemi Operativi

Sistemi Operativi



# Storia dei Sistemi Operativi

## Argomenti del capitolo

- Introduzione
- Generazione 1 - Tubi a vuoto
- Generazione 2 - Transistor
- Generazione 3 - Circuiti integrati
- Generazione 4 - Circuiti integrati LSI/VLSI (Personal Computer)



# Introduzione

Storia dei Sistemi Operativi



# Storia dei Sistemi Operativi

## Introduzione

### **Evoluzione dei SO**

- Influenzata dai progressi dell'hardware.
- Ha guidato il progresso dell'hardware (gestione degli interrupt, memoria virtuale, protezione della memoria).

Quattro generazioni di computer:

- Generazione 1, 1945-1955, tubi a vuoto.
- Generazione 2, 1955-1965, transistor.
- Generazione 3, 1965-1980, circuiti integrati.
- Generazione 4, dal 1980, PC.



# Storia dei Sistemi Operativi

## Introduzione

Conoscere la storia per capire il funzionamento

Non è argomento del Corso, però certe scelte/caratteristiche/funzionalità degli attuali SO:

- sono state proposte già negli anni '60 (Esempi: interrupt, semafori, .....)
- sono più facili da capire se si ha chiaro il contesto storico nel quale sono stati introdotti
- sono più facili da capire se si ha chiaro quanto accadeva prima che fossero introdotte
- sono più facili da capire se si ha chiaro il perché siano state introdotte.



# Generazione 1 - Tubi a vuoto

Storia dei Sistemi Operativi



# Storia dei Sistemi Operativi

## Generazione 1

### **Tubi a vuoto (1945-1955)**

#### **Uso**

I calcolatori occupano interi piani di edifici. Chi progetta, costruisce, programma e manutiene le macchine sono la stessa persona. Usate per calcoli scientifici.

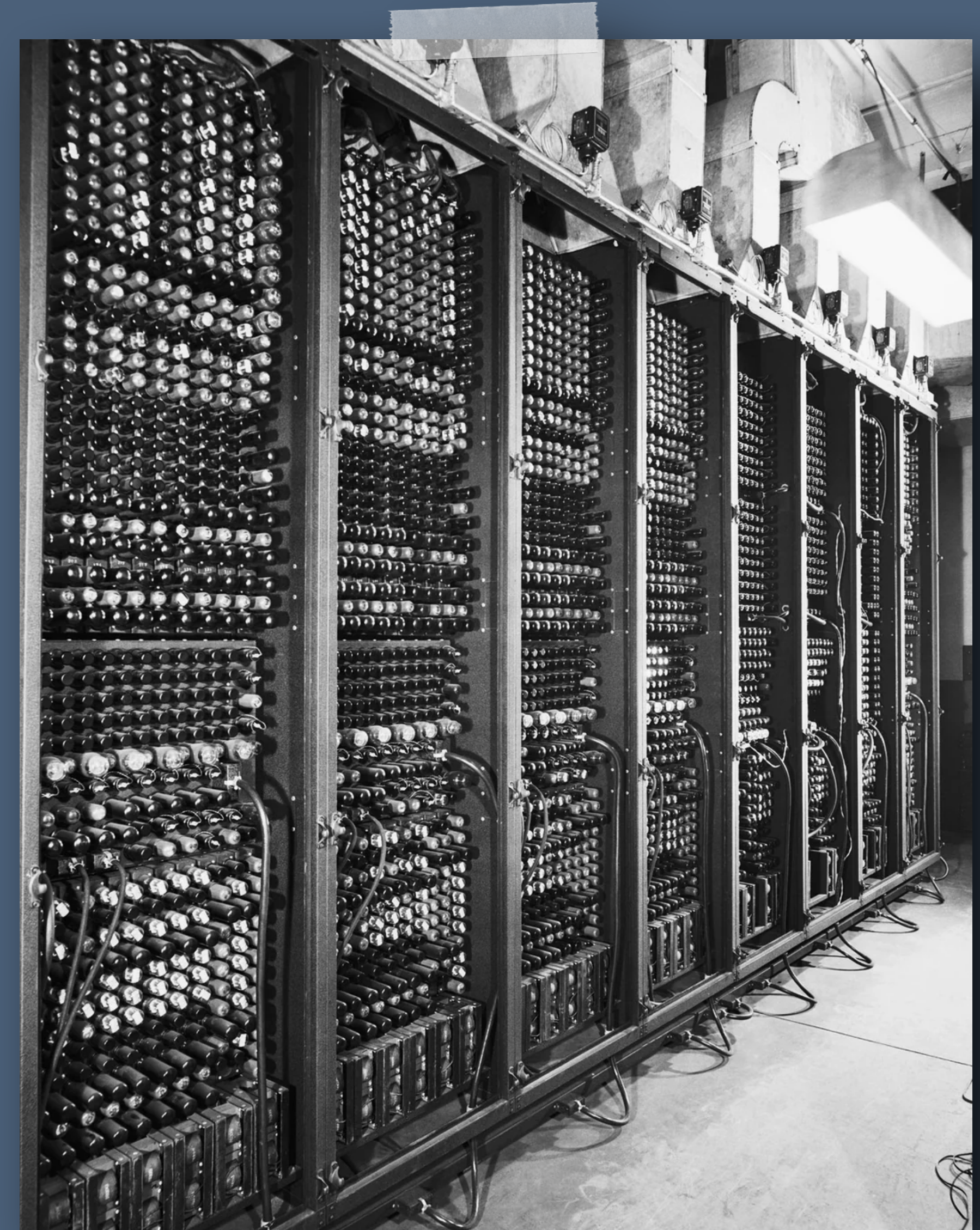
#### **Hardware**

Tubi a vuoto e tavole di commutazione.

#### **Programmazione**

Cablano i circuiti o con un linguaggio macchina (assembly o linguaggi ad alto livello).

ENIAC





# BUG

Cablano i circuiti o con un linguaggio macchina (assembly o linguaggi ad alto livello).

Correzione:

Nella prima generazione non esisteva l'assembly: si programmava cablando i circuiti o in linguaggio macchina puro.





# Storia dei Sistemi Operativi

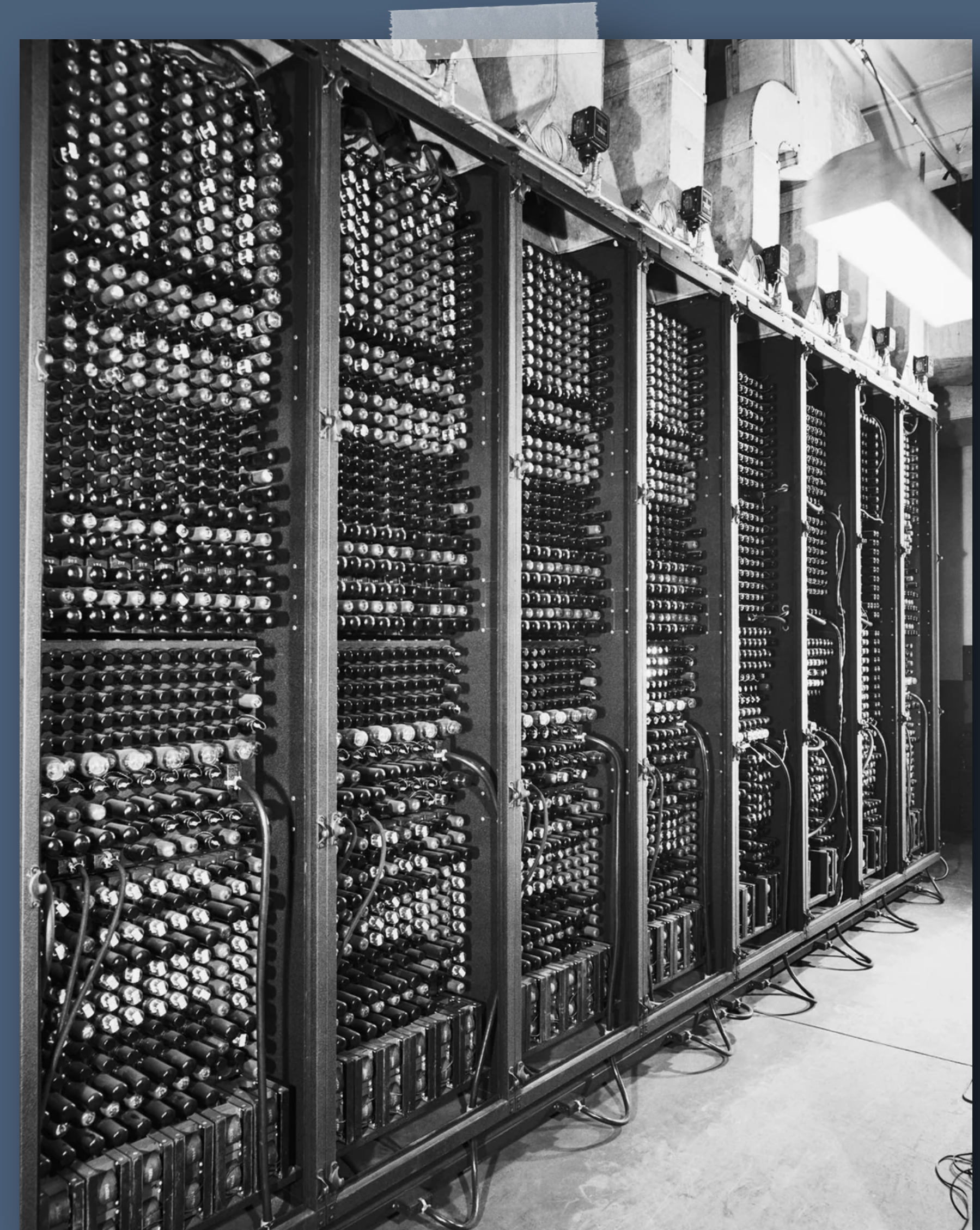
## Generazione 1

### **Sistema Operativo**

Erano già presenti semplici sistemi operativi che gestivano l'esecuzione dei programmi.

Non c'è nessuna macchina virtuale sopra la macchina fisica:

- gira un solo programma per volta
- il programma ha a disposizione tutte le risorse fisiche
- il programma deve crearsi le risorse logiche
- il programma deve controllare tutti i dispositivi fisici senza intermediari





# BUG

Erano già presenti semplici sistemi operativi che gestivano l'esecuzione dei programmi.

Correzione:

Non esiste alcun sistema operativo: il programma controlla direttamente hardware e risorse.





# Generazione 2 - Transistor

Storia dei Sistemi Operativi



# Storia dei Sistemi Operativi

## Generazione 2

### **Transistor (1955-1965)**

#### **Uso**

Macchine prodotte in serie e vendute, anche se in pochi esemplari.

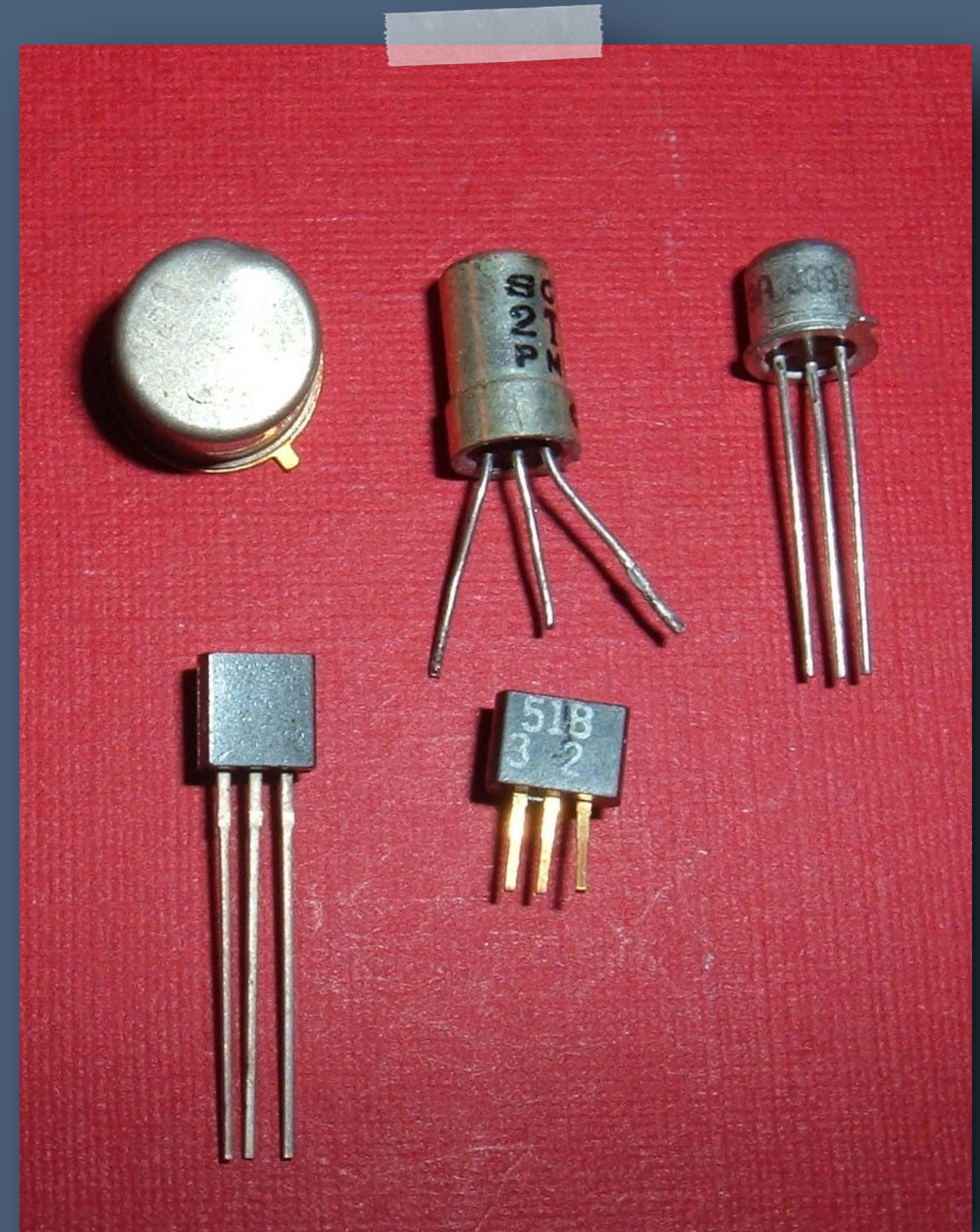
Distinzione tra progettista, costruttore, operatore, programmatore-utente, manutentore.

Ancora impiegate prevalentemente per calcoli scientifici, e l'utente è il programmatore.

#### **Hardware**

Transistor. I calcolatori occupano stanze.

Transistor





# Storia dei Sistemi Operativi

## Generazione 2

### **Programmazione**

Fortran (compiler) o assembly (assembler), con programmi codificati su schede perforate.

### **Sistema Operativo**

Dopo un periodo di assenza, vengono introdotti i Sistemi Batch, detti anche monitor residenti o batch monitor (IB-SYS, Fortran Monitor System).

IBM 7090 - IB SYS





# Storia dei Sistemi Operativi

## Generazione 2

### **Dispositivi**

Lettori di schede e stampanti. Si interagisce con la macchina caricando le schede a mano nel lettore di schede e prelevando le stampe.





# Storia dei Sistemi Operativi

## Generazione 2

### **Esecuzione (prima dei SO)**

- Incisione del programma sorgente sulle schede perforate.
- Caricamento programma in memoria con lettore di schede. Se in Fortran/assembly, caricato anche il compiler/assembler.
- Se in Fortran o assembly, esecuzione del compiler o assembler per ottenere l'eseguibile.
- Più job potevano risiedere contemporaneamente in memoria.
- Esecuzione del programma eseguibile.
- Raccolta dell'output dalla stampante.





# BUG

Più job potevano risiedere contemporaneamente in memoria.

Correzione:

Nei batch system classici un solo job alla volta risiede in memoria (User Area), insieme al SO (System Area).





# Storia dei Sistemi Operativi

## Generazione 2

### **Sistema Batch**

Il programma su schede è detto job.

Vengono successivamente introdotti i nastri magnetici.

- Alcune schede con indicazioni al SO (carica il compiler, carica l'eseguibile, esegui, ...)
- Un batch di job viene caricato su un nastro magnetico usando una macchina dedicata (economica).
- Il SO esegue i job del batch uno alla volta, caricandoli dal nastro. I risultati vengono inseriti in un altro nastro. Il SO sfrutta anche un nastro di sistema, che contiene per esempio il compiler Fortran e l'assembler.
- Su un'altra macchina dedicata (economica) i risultati dei job del batch contenuti nel nastro vengono stampati su carta.



# Storia dei Sistemi Operativi

## Generazione 2

### Esempio in Fortran

Job Control Language per SO Fortran Monitor System:

\$JOB - inizio del job

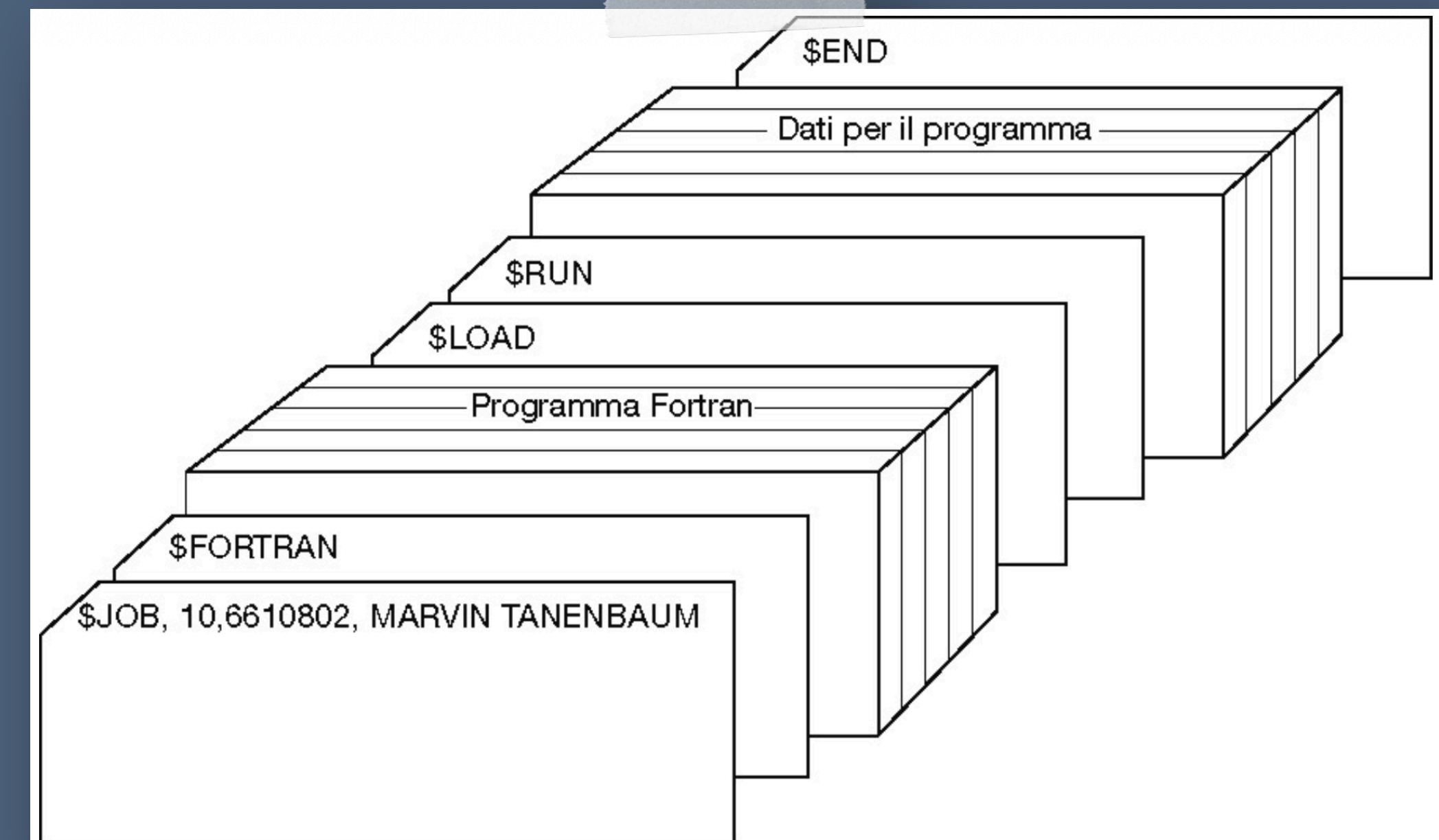
\$FORTRAN - carica il compiler

\$LOAD - carica l'eseguibile

\$RUN - esegui sui dati seguenti

\$END - fine del job

Tanenbaum - Fig. 1.4





# Storia dei Sistemi Operativi

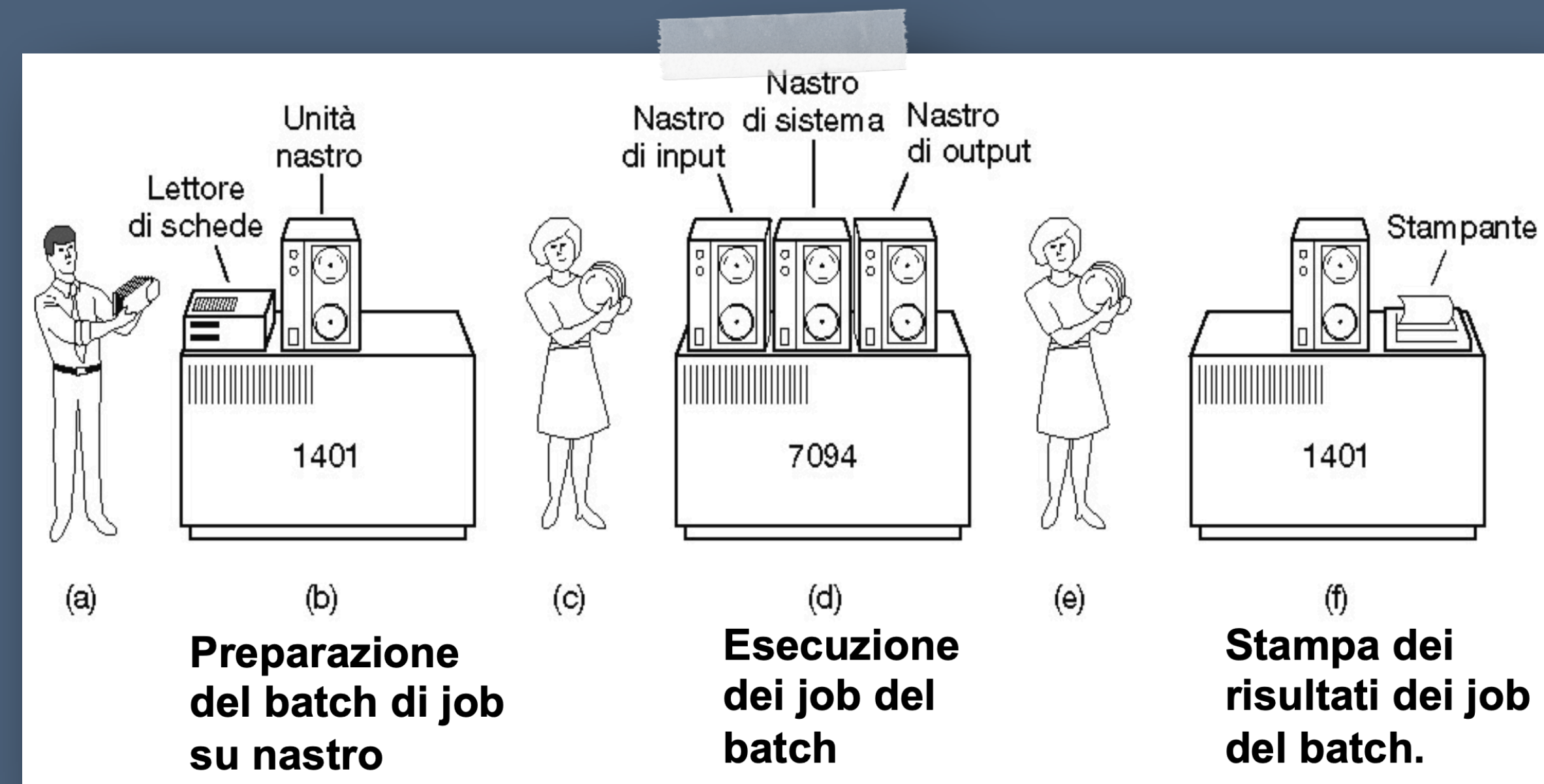
## Generazione 2

### IBM1401 (1959)

Macchina "economica".

### IBM 7094 (1961)

Macchina "potente" su cui gira il SO batch IB-SYS.

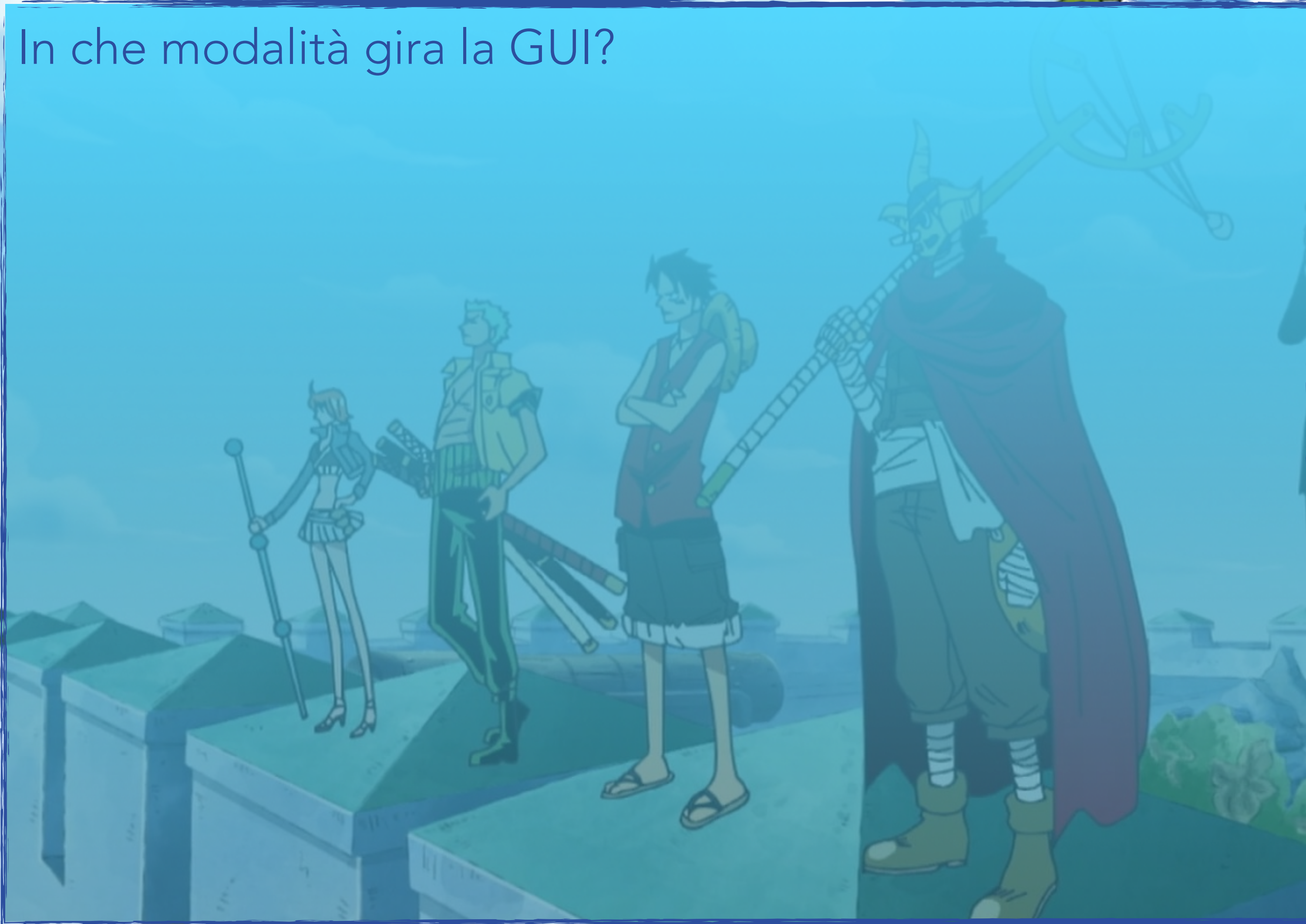


Tanenbaum - Fig. 1.3



# ANSIA

In che modalità gira la GUI?





# ANSIA

In che modalità gira la GUI?

Risposta:  
In user mode





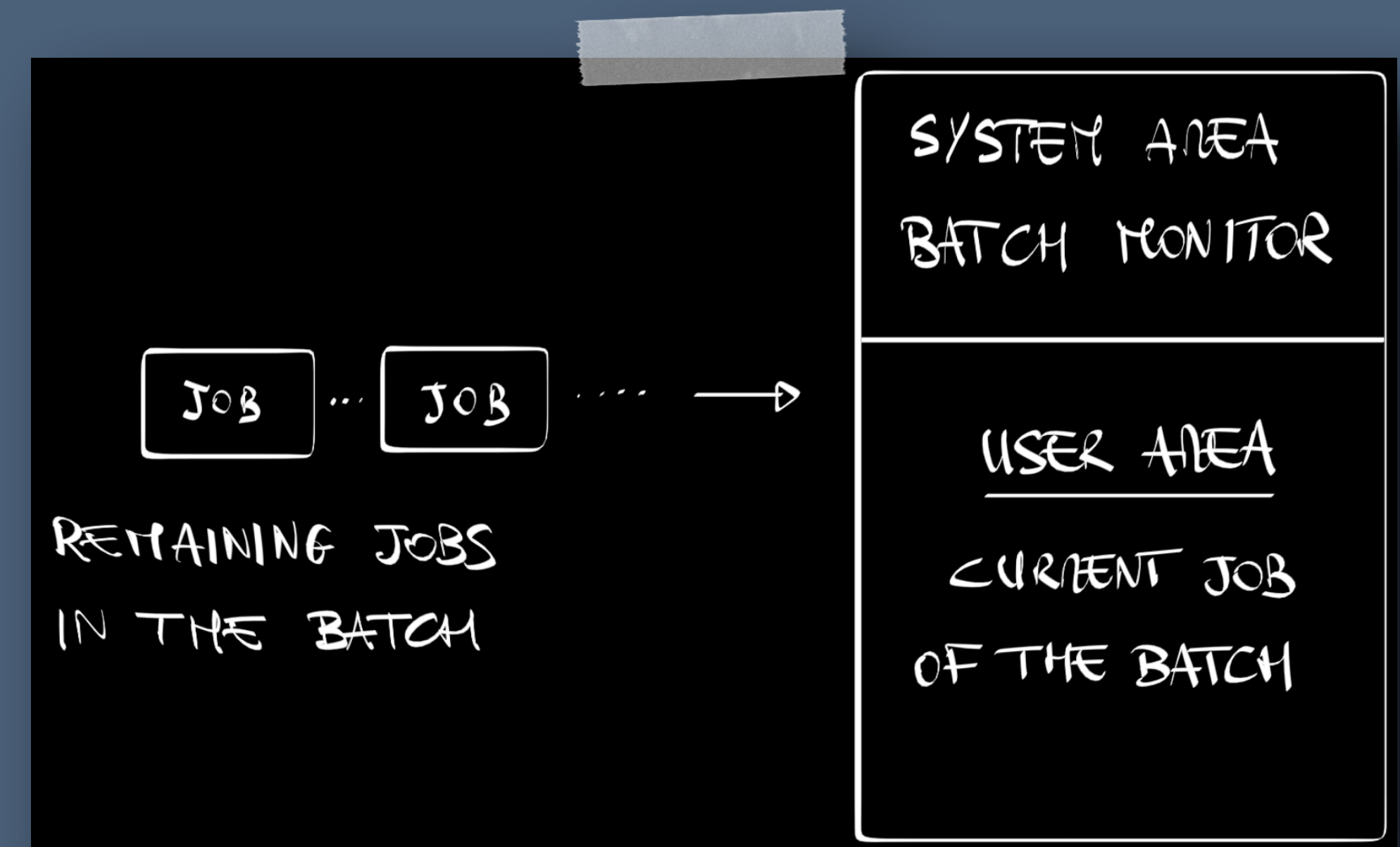
# Storia dei Sistemi Operativi

## Generazione 2

### Schema di esecuzione

Una zona di memoria, **System Area**, è riservata al SO.  
Un'altra zona di memoria, **User Area**, ospita un solo job alla volta, cioè il job in esecuzione.

Batch: insieme di job anche di utenti diversi.





# Storia dei Sistemi Operativi

## Generazione 2

### **Effetti dei SO Batch**

- Diminuzione tempi CPU inutilizzata: non caricati a mano ma dal nastro.
- Per il singolo utente, il tempo di turn-around, cioè il tempo intercorso tra la sottomissione del job e la restituzione dei risultati, può però dilatarsi: i batch vengono preparati quando si ha un numero sufficiente di job.
- Si privilegia l'uso efficiente della CPU (costosa per il periodo storico), a danno della user convenience.



# Storia dei Sistemi Operativi

## Generazione 2

### **Funzioni dei SO Batch**

Scheduling: implicito quando si forma il batch.

Gestione memoria: semplice in quanto, al momento del booting, una parte viene riservata al SO e l'altra al job corrente.

Protezione: anche se due job non possono coesistere in memoria, un minimo di protezione è necessaria. Si pensi al caso in cui un job richieda 10 schede di dati e ne vengano fornite solo 6: va evitato che le altre 4 vengano "rubate" al job seguente. Va evitato che un job acceda alla system area.

Interazione con l'utente: l'utente interagisce direttamente con il sistema durante l'esecuzione del programma.



# BUG

l'utente interagisce direttamente con il sistema durante l'esecuzione del programma.

Correzione:

Nei sistemi batch non c'è interazione diretta: l'utente consegna le schede e ritira l'output stampato.





# ESERCIZIO

Confronta Generazione 1 e Generazione 2 compilando una tabella con:

- presenza/assenza del SO
- modalità di programmazione
- dispositivi di input/output
- ruolo dell'utente





# ESERCIZIO

Confronta Generazione 1 e Generazione 2 compilando una tabella con:

- presenza/assenza del SO
- modalità di programmazione
- dispositivi di input/output
- ruolo dell'utente

	Generazione 1	Generazione 2
SO	NO	Batch
Programmazione	Cablata	Batch
In/Out	Cablati	Batch
Utente	Tutto	Batch





# MEMORIA

*"La storia dell'informatica è la storia del tentativo di usare al meglio risorse estremamente costose."*

(Andrew S. Tanenbaum)





# COMPITI

Spiega in 15 righe massimo:

Perché i sistemi batch sono stati un enorme passo avanti rispetto alla prima generazione, pur peggiorando l'esperienza del singolo utente.

Descrivi il flusso di esecuzione di un job batch, dal momento in cui l'utente prepara le schede fino alla stampa dell'output.

Rispondi: quale problema fondamentale dei sistemi batch porterà alla nascita dei sistemi time-sharing?

