



# Argomenti trattati

## 1 Introduzione ai SO

## 2 Sistemi operativi

- Funzioni e obiettivi
- Storia ed evoluzione

- Sfrutta le risorse hardware di un sistema computerizzato
  - uno o più processori
  - memoria primaria (RAM)
  - memoria secondaria (dischi)
  - dispositivi di input/output
  
- Fornisce un insieme di servizi agli utenti
  - in particolare: offre un ambiente di esecuzione **facilitato** alle applicazioni utente

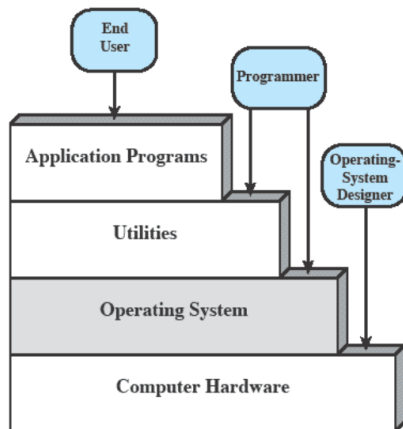
# Sistemi Operativi: Esempi

- Windows
  - ultima versione: Windows 10
- macOS
  - ultima versione: macOS Mojave
- Linux Ubuntu
  - ultima versione: Ubuntu 19.04 (Disco Dingo)
- Windows Phone, iOS, Android

## Sistemi operativi

# Funzioni e obiettivi

# Strati e Utenti



# Servizi Offerti da un SO

- Sviluppo di programmi
  - compilatori, editor e debugger
  - system calls
  - visione semplificata della memoria RAM
  - alcuni servizi non fanno parte del *core* del SO, ma sono forniti con il SO
- Esecuzioni di programmi
  - app(licazioni)
  - servizi
  - anche più applicazioni e servizi contemporaneamente
  - il SO gestisce la temporizzazione delle varie operazioni

# Servizi Offerti da un SO

- Accesso ai dispositivi di input/output
  - il SO nasconde al programmatore i dettagli dei singoli dispositivi di I/O
  - nel caso dei dispositivi di memoria di massa, tramite filesystem
- Accesso controllato ai files
  - il SO gestisce i diversi formati dei (oltre ai diversi dispositivi)
  - in caso di utenti multipli, serve un meccanismo di protezione per l'accesso ai file
- Accesso al sistema
  - il SO controlla l'accesso a tutto il sistema o alle singole risorse
  - serve protezione delle risorse e dei date risoluzione dei conflitti

# Servizi Offerti da un SO

- Rilevamento e risposta agli errori
  - errori di hardware interno ed esterno
  - errori software
  - richiesta di un applicativo non soddisfacibile
  - il SO sceglie la soluzione più appropriata (terminare programma, ripetere operazione, segnalare il tipo di errore)
- Accounting (chi fa cosa)
  - collezione di statistiche dell'uso del sistema
  - monitoraggio delle performance
  - usato per capire cosa occorre migliorare
  - usato per far pagare in base all'uso del sistema

# Sistema Operativo

- Quindi un sistema operativo:
  - è un **programma** che controlla l'esecuzione dei programmi applicativi
  - fornisce un'**interfaccia** tra le applicazioni e l'hardware
  - fornisce un'**interfaccia** tra utente e computer
    - l'utente finale non vuole (e non deve) preoccuparsi dei dettagli hardware e vede il computer come un insieme di applicazioni

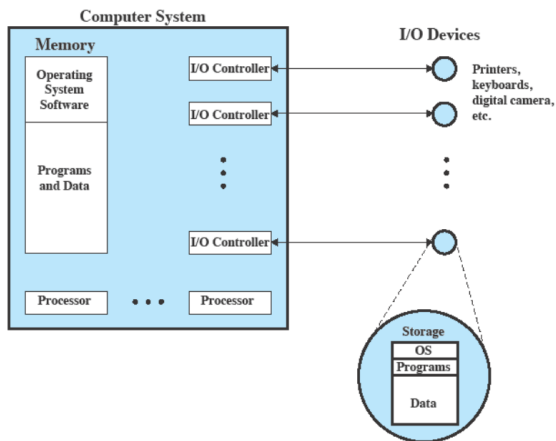


# Sistema Operativo

- Un computer è un insieme di risorse per elaborazione, memorizzazione e trasferimento di informazioni
- Il SO è **responsabile della gestione delle risorse**
  - funziona allo stesso modo del software normale: è un programma in esecuzione
    - tuttavia, lo fa con privilegi più alti
  - concede il controllo del processore ad altri programmi
    - e controlla l'accesso alle altre risorse (RAM, I/O)

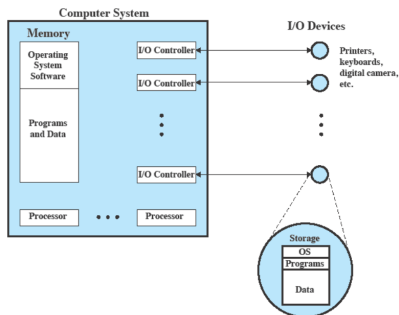
# Sistema Operativo

- SO come **responsabile della gestione delle risorse**



# Il Kernel

- Il **Kernel** o **nucleo** è:
  - la parte di sistema operativo che si trova sempre in memoria principale
  - contiene le funzioni più usate



Sistemi operativi

# Storia ed evoluzione

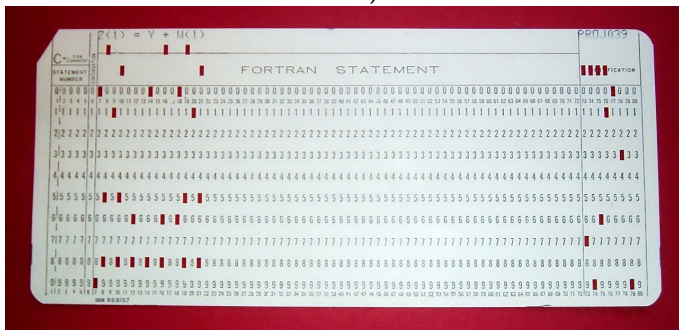
# Storia dei SO

- I sistemi operativi **primitivi** (**anni Quaranta**) erano molto diversi da quelli attuali
- La loro evoluzione è dovuta principalmente a:
  - aggiornamento dell'hardware o nuovo tipo hardware
  - nuovi servizi
  - correzione di errori
- <http://www.computerhistory.org/timeline/>



## Storia dei SO

- Computazione seriale (**anni Quaranta**)
  - già all'inizio l'input viene parzialmente semplificato con dispositivi per leggere schede perforate (esistenti già da 2 secoli)





# Monitor e Caratteristiche Hardware

- Protezione della memoria
  - non permette che la zona di memoria contenente il monitor venga modificata
- Timer
  - impedisce che un job monopolizzi l'intero sistema
- Istruzioni privilegiate
  - alcune istruzioni macchina possono essere eseguite solo dal monitor
- Interruzioni
  - i primi modelli di computer non le avevano



# Sistemi Batch: Sottoutilizzazione

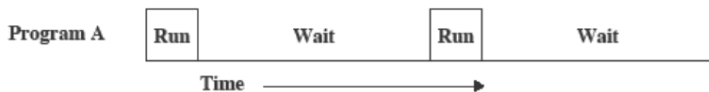
Il 96% del tempo è sprecato ad aspettare i dispositivi di I/O

Read one record from file	$15 \mu s$
Execute 100 instructions	$1 \mu s$
Write one record to file	<u><math>15 \mu s</math></u>
TOTAL	$31 \mu s$

$$\text{Percent CPU Utilization} = \frac{1}{31} = 0.032 = 3.2\%$$

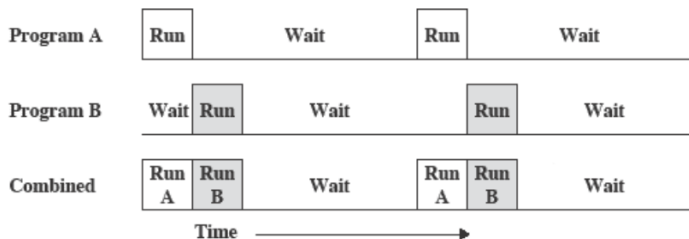
# Programmazione Singola

Il processore deve aspettare che le istruzioni di I/O siano completate prima di procedere

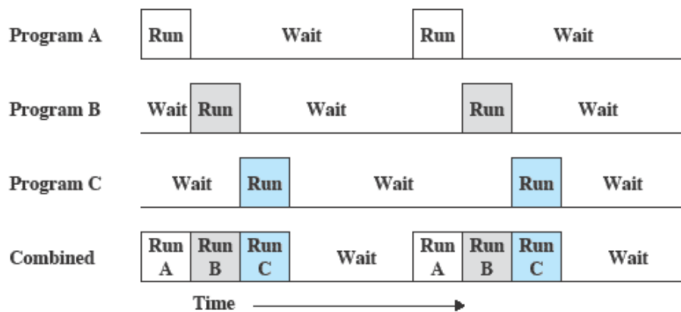


# Multiprogrammazione

L'approccio **multiprogrammazione** o **multitasking** si basa sul fatto che se un job deve aspettare che si completi dell'I/O, allora il processore può passare ad un altro job



# Multiprogrammazione



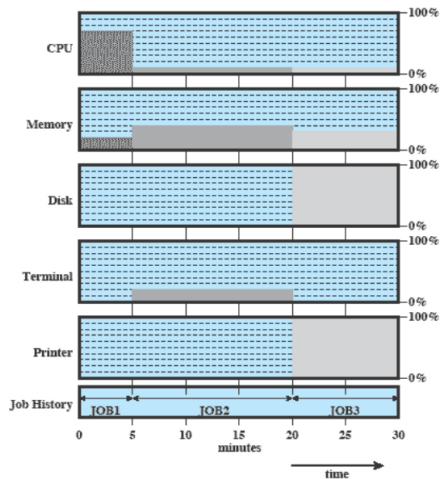
(c) Multiprogramming with three programs

# Esempio

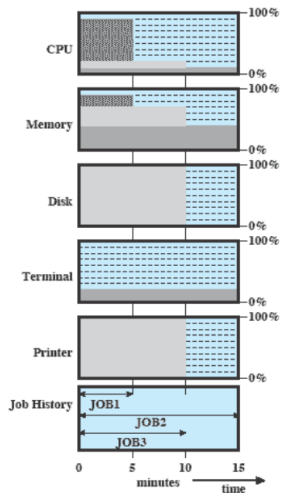
- Consideriamo un computer con: 250 Mbyte di memoria disponibile, disco, terminale, stampante
- Consideriamo tre programmi con le caratteristiche in tabella, che in ambiente batch vengono eseguiti in sequenza

	JOB1	JOB2	JOB3
<b>Type of job</b>	Heavy compute	Heavy I/O	Heavy I/O
<b>Duration</b>	5 min	15 min	10 min
<b>Memory required</b>	50 M	100 M	75 M
<b>Need disk?</b>	No	No	Yes
<b>Need terminal?</b>	No	Yes	No
<b>Need printer?</b>	No	No	Yes

## Istogrammi di Utilizzo



(a) Uniprogramming



(b) Multiprogramming

# Uso del Processore

- Prime 4 righe: dagli istogrammi la media per uniprogramming e per multiprogramming (*i valori non tornano...*)
- Elapsed time: tempo per vedere completati tutti e 3 i job
- Throughput:  $\frac{\text{numero job completati}}{\text{ore}}$
- Mean response time: media dei tempi di completamento (nel caso di uniprogramming: 5, 20, 30  $\rightarrow$  18.3)

	Uniprogramming	Multiprogramming
<b>Processor use</b>	20%	40%
<b>Memory use</b>	33%	67%
<b>Disk use</b>	33%	67%
<b>Printer use</b>	33%	67%
<b>Elapsed time</b>	30 min	15 min
<b>Throughput</b>	6 jobs/hr	12 jobs/hr
<b>Mean response time</b>	18 min	10 min



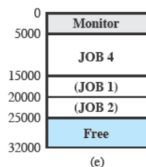
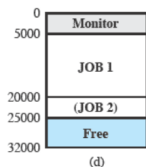
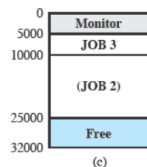
# Batch vs. Time Sharing

	<b>Batch</b>	<b>Time Sharing</b>
<b>Scopo principale</b>	Massimizzare l'uso del processore	Minimizzare il tempo di risposta
<b>Provenienza delle direttive al SO</b>	Comandi del job control language, sottomessi con il job stesso	Comandi dati da terminale



## CTSS: Compatible Time-Sharing System

## Esempio con 4 job



## Riassumendo

- **anni Quaranta** - Computazione seriale
- **anni Cinquanta/Sessanta** - Semplice sistema non interattivo o *batch*
  - multiprogrammazione
- **dagli anni Settanta** - Time-Sharing: sistemi a condivisione di tempo
  - i job sono tipicamente interattivi