

# Ripasso Sistemi Operativi

Matteo Franchini

17 agosto 2023

## 1. Introduzione

## 2. Evoluzione dei sistemi operativi

## 1. Introduzione

## 2. Evoluzione dei sistemi operativi

# Che cos'è un sistema operativo?

## Introduzione

Un programma che funge da intermediario tra l'utente di un computer e l'hardware del computer stesso

### **Obbiettivi del sistema operativo:**

- Eseguire i programmi dell'utente e facilitarne la risoluzione dei problemi
- Rendere il sistema informatico comodo da usare
- Utilizzare l'hardware del computer in modo efficiente

# Definizione di SO

## Introduzione

Il **sistema operativo** è un allocatore di risorse:

- gestisce tutte le risorse
- decide tra richieste in conflitto per un uso efficiente ed equo delle risorse

Il **sistema operativo** è un **programma di controllo**

- controlla l'esecuzione dei programmi per prevenire gli errori e l'uso improprio del computer

Non c'è una definizione universalmente accettata

- con il termine sistema operativo si intende quell'insieme di programmi che provvedono alla gestione Hw e Sw di un sistema di calcolo
- *"Tutto ciò che viene fornito quando si ordina un SO"*

Una definizione alternativa (**Tanenbaum**):

*un sistema operativo è un programma che controlla le risorse di un calcolatore e fornisce ai suoi utenti un'interfaccia o macchina virtuale più agevole da utilizzare della macchina "nuda"*

*L'unico programma che è sempre in esecuzione sul computer è il **kernel (nucleo)** del SO.*

Il resto è

- un programma di sistema (fornito con il SO di cui costituisce una parte)
- un programma applicativo

Può essere visto come:

1. **Allocatore di risorse Hw e Sw:**

- tempo di CPU, spazio di memoria, dispositivo di I/O, compilatori
- Le risorse devono essere assegnate a programmi specifici secondo determinate politiche

2. **Programma di controllo:** controlla l'esecuzione dei programmi per prevenire errori ed usi impropri del calcolatore

Obbiettivi principali del SO:

- rendere più **semplice** l'uso di un sistema di elaborazione
- rendere più **efficiente** l'uso delle risorse del sistem adi elaborazione

Il SO è costituito dall'insieme dei programmi (sw o fw) che **rendono praticamente utilizzabile** l'elaboratore agli utenti cercando contemporaneamente di **ottimizzarne le prestazioni**.

- **Visione top-down**: il sistema operativo come una macchina estesa (astrazione)
- **Visione bottom-up**: il sistema operativo come un gestore di risorse (fornisce protezione, risoluzione dei conflitti)



Vi è un **programma di bootstrap** che normalmente è memorizzato in una **memoria non volatile**, inizializza e verifica il corretto funzionamento dei componenti Hw del sistema. **Carica il kernel del SO e inizia l'esecuzione.**

Nei PC più vecchi c'era il **BIOS**, ora invece è stato sostituito con il **UEFI**.

# Proprietà fondamentali di un SO

## Introduzione

- **Affidabilità**
- **Efficienza**
- **Sicurezza**



# Funzionamento di un elaboratore

## Introduzione

- I dispositivi di I/O e la CPU possono operare simultaneamente
- Ogni controller di dispositivo è responsabile di un particolare dispositivo
- Ogni controllore di dispositivo ha un buffer locale
- La CPU sposta i dati da/alla memoria principale a/dai buffer locali
- L'I/O è dal dispositivo al buffer locale del controllore
- Il controllore del dispositivo informa la CPU di aver terminato la sua operazione causando un interrupt

# Funzioni comuni degli interrupt

## Introduzione

La gestione degli interrupt deve salvare l'indirizzo dell'istruzione interrotta

### Eccezione

Una **trap** o eccezione è un'interruzione generata dal Sw e causata da un errore o da una richiesta al SO da parte di un programma

Un sistema operativo è guidato dagli interrupt

# Gestione degli interrupt

## Introduzione

Il sistema operativo preserva lo stato della CPU memorizzando i registri e il contatore del programma

# Struttura dell'I/O

## Introduzione

1. Dopo l'avvio dell'I/O, il controllo ritorna al programma utente solo al completamento dell'I/O
  - Un'istruzione *wait* mette a riposo la CPU fino al prossimo interrupt
  - Ciclo di attesa
  - È in sospenso al massimo una richiesta di I/O alla volta, nessuna elaborazione simultanea di I/O
2. Dopo l'avvio dell'I/O, il controllo torna al programma utente senza attendere il completamento dell'I/O
  - **System call**: richiesta al SO per consentire al programma di attendere il completamento dell'I/O
  - La tabella di stato dei dispositivi contiene una voce in cui è indicato l'**indirizzo** e lo **stato**
  - Il SO **indicizza la tabella dei dispositivi di I/O** per determinare lo stato del dispositivo e modificare la voce della tabella per includere interrupt

# Funzioni specifiche di gestione

## Introduzione

### Gestione della memoria centrale

- caricare in memoria programmi e dati
- evitare interferenze fra programmi diversi
- assegnare la memoria in base a criteri di efficienza
- minimizzare i trasferimenti tra memoria centrale e memoria di massa



## Gestione della memoria secondaria

- consentire l'accesso all'informazione in base alla sua organizzazione logica anziché fisica
- controllare i diritti di accesso ai file da parte degli utenti
- consentire creazione, modifica, cancellazione dei file

## Gestione dei dispositivi periferici

- mascherare al programmatore la complessità delle operazioni di I/O
- effettuare controlli sul corretto funzionamento delle operazioni
- risolvere conflitti nell'utilizzo di una stessa periferica da parte di più programmi

## Gestione dei processi

- decidere quale programma userà il processore (**scheduling**) in base a criteri di corretto funzionamento e di efficienza
- verificare che i programmi rilascino il processore entro il tempo stabilito

# Funzioni di un SO

## Introduzione

- definizione e gestione dell'interfaccia utente
- gestione dei job
- gestione delle risorse di sistema
- ausili per la messa a punto dei programmi
- ausili per la gestione dei dati - file system

# Struttura della memoria

## Introduzione

### Memoria principale

solo supporti di memoria di grandi dimensioni a cui la CPU può accedere direttamente

### Memoria secondaria

estensione della memoria principale che **fornisce un'ampia capacità di memorizzazione non volatile**

- Dischi rigidi
- A stato solido

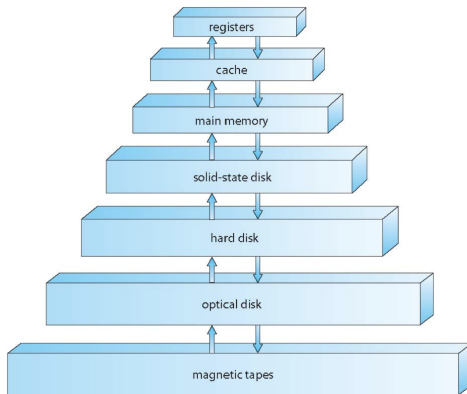
# Gerarchia di memoria

## Introduzione

**Sistemi di storage organizzati in una gerarchia.**

**Catching:** copiare le informazioni in un sistema di archiviazione più veloce

**Driver del dispositivo:** per ogni controller di dispositivo per gestire l'I/O



Le informazioni in uso vengono **copiate temporaneamente da una memoria più lenta a una più veloce**

La memoria più veloce (**cache**) viene controllata per prima per determinare se **le informazioni sono lì**, in caso contrario, i dati vengono copiati nella cache e utilizzati lì.

La **cache è più piccola della memoria di cui si fa il caching**, la gestione della cache è un importante problema di progettazione.

# Struttura di accesso diretto alla memoria

## Introduzione

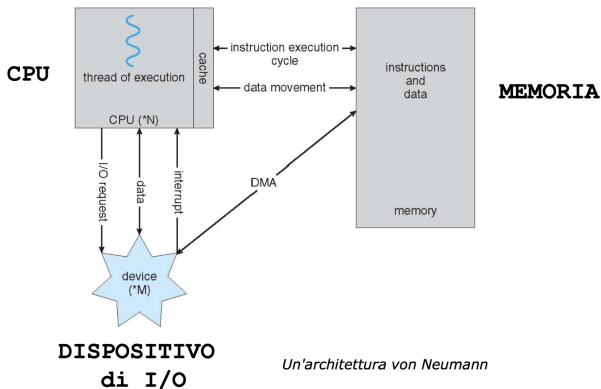
### DMA (Direct Memory Access)

Utilizzata per dispositivi di I/O ad alta velocità in grado di trasmettere informazioni a velocità prossime a quelle della memoria.

Il controller del dispositivo trasferisce blocchi di dati dalla memoria tampone direttamente alla memoria principale senza l'intervento della CPU

# Funzionamento PC moderno

## Introduzione





# Architettura del sistema informatico

## Introduzione

L'uso e l'importanza dei sistemi multiprocessore è in crescita.

I vantaggi includono:

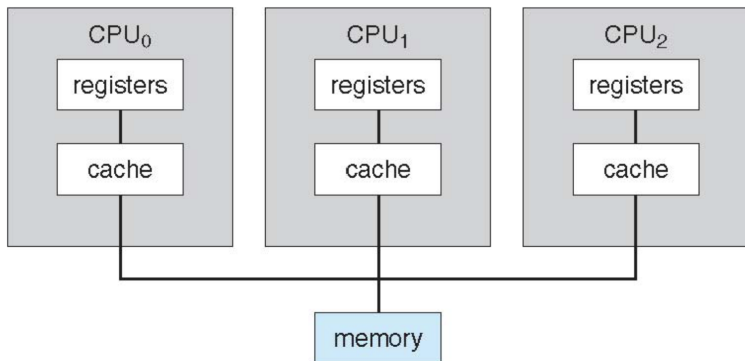
1. **Aumento del throughput**
2. **Economia di scala**
3. **Maggiore affidabilità: *graceful degradation o tolleranza ai guasti***

Ci sono **due tipi di multiprocesso**:

1. **Multiprocesso asimmetrico**: a ogni processore
2. **Multiprocesso simmetrico**: ogni processore esegue tutti i compiti

# Architettura del multiprocessore simmetrico

## Introduzione



# Sistemi cluster

## Introduzione

Sono come i sistemi multiprocessore, ma con **più sistemi che lavorano insieme**, di solito condividono lo **storage** tramite una **rete SAN (Storage Area Network)**.

### Tipi di cluster

- **Clustering asimmetrico**: prevede una macchina in modalità **hot-standby**
- **Clustering simmetrico**: prevede più nodi che eseguono applicazioni, monitorandosi a vicenda

Alcuni cluster sono destinati al calcolo ad alte prestazioni (HPC), altri hanno un **gestore di lock distribuito (DLM)** per evitare operazioni in conflitto.

### Multiprogrammazione (sistema batch)

- Un singolo utente non può tenere occupati CPU e dispositivi di I/O in ogni momento
- La **multiprogrammazione organizza i lavori** (codice e dati) in modo che la CPU ne abbia sempre uno da eseguire
- Un sottoinsieme di lavori totali nel sistema viene tenuto in memoria
- Quando deve aspettare, il sistema operativo **passa a un altro lavoro**

## Multitasking

Il **timesharing** è un'estensione logica in cui la **CPU** passa ai lavori con una **frequenza** tale da consentire agli utenti di interagire con ciascun lavoro mentre è in esecuzione

- Il tempo di risposta deve essere  $< 1$  secondo
- Ogni utente ha almeno un programma in esecuzione
- Se diversi lavori sono pronti per essere eseguiti contemporaneamente abbiamo la **schedulazione della CPU**
- La memoria virtuale consente l'esecuzione di processi non completamente in memoria

# Operazioni del sistema operativo

## Introduzione

### Guidato dalle interruzioni (Hw e Sw):

- Interruzione hardware da parte di uno dei dispositivi
- Interruzione software (eccezione o trap): errore, richiesta di servizio, altri problemi

Il funzionamento in **doppia modalità** permette al sistema operativo di proteggere se stesso e gli altri componenti del sistema: **modalità utente** e **modalità kernel**.

Alcune istruzioni designate come *privilegiate* sono eseguibili solo in **modalità kernel**.

# Transizione dalla modalità utente a quella kernel

## Introduzione

C'è un **timer per prevenire il loop infinito**.

Il timer è impostato per interrompere il computer dopo un certo periodo di tempo e conserva un **contatore che viene decrementato dall'orologio fisico**.

Quando il **contatore è zero viene generato un interrupt**.

### Processo

Un processo è un **programma in esecuzione**. È un'unità di lavoro all'interno del sistema. Il programma è un'entità **passiva**, il processo è un'entità **attiva**.

Un **processo a singolo thread** ha un contatore di programma che specifica la posizione della prossima istruzione da eseguire.

Un **processo multi-thread** ha un contatore di programma per ogni thread.



# Attività di gestione dei processi

## Introduzione

Il sistema operativo è responsabile delle seguenti attività relative alla gestione dei processi:

- Creazione e cancellazione di processi utente e di sistema
- Sospendere e riprendere i processi
- Meccanismi di sincronizzazione dei processi
- Meccanismi di comunicazione tra i processi
- Meccanismi per la gestione dei deadlock

# Gestione della memoria di massa

## Introduzione

Generalmente i dischi sono utilizzati per memorizzare dati che non stanno nella memoria principale o dati che devono essere tenuti per un periodo di tempo "lungo".

Attività del sistema operativo

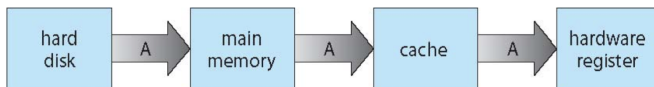
- Gestione dello spazio libero
- Allocazione dello spazio di archiviazione
- Pianificazione dei dischi

# Migrazione del dato "A" da un disco ad un registro

## Introduzione

Gli ambienti multitasking devono fare attenzione a utilizzare il valore più recente, indipendentemente dalla sua posizione nella gerarchia di memorizzazione.

Gli ambienti multiprocessore devono garantire la coerenza della cache in Hw, in modo che tutte le CPU abbiano il valore più recente nella loro cache



# Sottosistema I/O

## Introduzione

Uno degli scopi del sistema operativo è quello di **nascondere all'utente le peculiarità dei dispositivi Hw.**

Il sottosistema I/O è responsabile di:

- Gestione della memoria dell'I/O comprende il buffering, il caching in una memoria più veloce, lo spooling
- Interfaccia generale dispositivo-driver
- Driver per dispositivi Hw specifici

# Protezione e sicurezza

## Introduzione

### Protezione

Qualsiasi meccanismo di controllo dell'accesso dei processi o degli utenti alle risorse definite dal SO

### Sicurezza

Difesa del sistema da attacchi interni ed esterni

I sistemi in genere distinguono innanzitutto fra gli utenti, per determinare chi può fare cosa:

- Le identità degli utenti (**ID utente**)
- L'ID utente viene poi associato a tutti i file e processi di quell'utente per determinare il controllo degli accessi
- L'identificativo di gruppo (**ID di gruppo**) consente di definire un insieme di utenti e di gestire i controlli, quindi anche di associarlo a ciascun processo, file o altro
- L'**escalation dei privilegi** consente all'utente di passare ad un ID effettivo con maggiori diritti

I **portali** forniscono accesso web ai sistemi interni. I **computer in rete** sono come terminali web.

I computer mobili si collegano tramite **reti wireless**

# Ambienti informatici - Mobile

## Introduzione

Permette nuovi tipi di applicazioni come la realtà aumentata.

Utilizza reti wireless IEEE 802.11 o reti dati cellulari per la connettività



Collezione di **sistemi separati**, eventualmente eterogenei, collegati in rete tra loro.

La **rete** è un percorso di comunicazione, il **TCP/IP** è il più comune:

- Rete Locale (LAN)
- Rete geografica (WAN)
- Rete metropolitana (MAN)
- Rete personale (PAN)

# Ambienti informatici - Virtualizzazione

## Introduzione

### Emulazione

Utilizzata quando il tipo di CPU di origine è diverso da quello di destinazione.

Questo metodo è generalmente più lento.

Quando il linguaggio del computer non viene compilato in codice nativo siamo di fronte all'**interpretazione**

### Virtualizzazione

Sistema operativo compilato in modo nativo per la CPU, che esegue sistemi operativi **guest** anch'essi compilati in codice nativo

# Ambienti di elaborazione - Cloud Computing

## Introduzione

Fornisce elaborazione, storage e persino applicazioni come servizio attraverso una rete.

Estensione logica della virtualizzazione perché utilizza la virtualizzazione come base per le sue funzionalità.

Alcuni tipi:

- **Cloud pubblico:** disponibile via internet a chiunque sia disposto a pagare
- **Cloud privato:** gestito da un'azienda per uso proprio
- **Cloud ibrido:** include componenti di cloud pubblico e privato
- **SaaS (Software as a Service):** uno o più applicazioni disponibili via internet
- **PaaS (Platform as a Service):** stack di software pronto per l'uso di applicazioni via internet
- **IaaS (Infrastructure as a Service):** server o storage disponibili via internet

# Aree di applicazione di un SO

## Introduzione

- **Sistemi di tipo generale**
- **Sistemi in tempo reale**
  - applicazioni per il controllo di processo e di apparati fisici
  - applicazioni interattive, interrogazione di basi di dati, query web

# Ambienti di elaborazione - Sistemi embedded in tempo reale

## Introduzione

I sistemi **embedded in tempo reale** sono la forma più diffusa di computer.  
Il sistema operativo in tempo reale ha vincoli temporali fissi ben definiti

- L'elaborazione deve essere eseguita entro i vincoli
- Operazione corretta solo se i vincoli sono rispettati

# Sistemi operativi open source

## Introduzione

Sistemi operativi resi disponibili in formato di codice sorgente piuttosto che solo binari closed-source.

Avviato dalla **Free Software Foundation (FSF)**, che ha una licenza pubblica GNU (GPL).

Esempi: GNU/LINUX

### Sistemi aperti

- Realizzati e mantenuti da comunità di sviluppatori volontari
- Movimento per il software "open source"
- Le applicazioni beneficiano della piena conoscenza del SO, il SO evolve nel tempo e migliora grazie alla sua trasparenza
- Esempio canonico: Linux
- Interfaccia utente e **look and feel** sono modificabili

- **Utenti finali del sistema:** per essi il sistema operativo è trasparente
- **Programmatori applicativi:** utilizzano i servizi del SO per la realizzazione e l'esecuzione dei loro programmi
- **Programmatori di sistema:** aggiornano e modificano i programmi del SO per adeguarli a nuove necessità del sistema o degli utenti applicativi
- **Operatori:** controllano il funzionamento e rispondono alle richieste di intervento da parte del sistema
- **Amministratore del sistema:** stabilisce le politiche di gestione del sistema e ne cura l'osservanza



### Sistemi proprietari

- Progettati da costruttori al fine di sfruttare in modo ottimale le risorse di ciascun tipo di macchina
- Programmi utente e applicazioni si interfacciano al SO in modo diverso tra le diverse famiglie di sistemi
- IBM: OS/360 370, VM, MVS

### Sistemi standard

- Progettati da aziende software o da grandi utenti per consentire lo sviluppo di applicazioni portabili su sistemi diversi
- Interfaccia di programmazione con cui le applicazioni interagiscono con il SO rimane costante nelle diverse versioni
- Esempi: UNIX, MS-DOS

# Proprietà fondamentali di un SO

## Introduzione

- **Architettura:** com'è organizzato?
- **Condivisione:** quali risorse vengono condivise?
- **Efficienza:** come massimizzare l'utilizzo delle risorse disponibili?
- **Affidabilità/tolleranza ai guasti**
- **Estensibilità**
- **Protezione e sicurezza**
- **Conformità a standard**

1. Introduzione

2. Evoluzione dei sistemi operativi

# Stadi evolutivi e modalità d'uso dei sistemi

## Evoluzione dei sistemi operativi

- Sistema isolato ('50-'60): batch
- Sistema centralizzato ('60-'70): remote job entry, teleprocessing, time sharing
- Sistemi decentralizzati ('70-): minicalcolatore, controllo real-time, data logging
- Sistemi distribuiti ('80-): multiprocessori, reti locali

# Client-Server

## Evoluzione dei sistemi operativi

I dispositivi **client** sono utilizzati dagli utenti, molti sistemi sono **server** che rispondono alle richieste di servizio dei client offrendo: servizi di calcolo o gestione (DBMS), servizi di gestione dei file (NFS), altri servizi

# Sistema di elaborazione distribuito

## Evoluzione dei sistemi operativi

È costituito da nodi tra loro collegati in ciascuno dei quali sono presenti capacità di: **elaborazione, memorizzazione, comunicazione.**

### Vantaggi:

- Tolleranza al guasto: un guasto non provoca l'arresto del sistema, ma solo una riduzione delle prestazioni
- Prestazioni: l'elaborazione di norma è effettuata nel posto stesso di utilizzazione, si ha un miglioramento delle prestazioni
- Condivisione: la capacità di elaborazione, i programmi ed i dati esistenti nell'intero sistema sono patrimonio comune di tutti gli utenti

# Evoluzione dei SO

## Evoluzione dei sistemi operativi

- **Primi calcolatori:** privi di SO, problemi di complessità di operazioni, inefficienza
- **Prima generazione ('50-'60):** virtualizzazione dell'I/O, librerie di controllo dei device, problemi di debug, nasce il **sistema operativo** come stratificazione successiva di funzioni volte ad aumentare l'efficienza e la semplicità d'uso della macchina
- **Seconda generazione ('60-'65):** indipendenza tra programmi e dispositivi usati, parallelizzazione degli utenti tramite **multiprogrammazione e time sharing**
- **Terza generazione ('65-'75):** SO unico per una famiglia di elaboratori, risorse virtuali, sistemi multifunzione

- **Quarta generazione ('75-'85):** sistemi a macchine virtuali, sistemi multiprocessore e distribuiti
- **Quinta generazione ('85-'95):** elaboratori personali, reti locali, avvio di internet
- **Sesta generazione ('95-'05):** elaborazione distribuita, peer to peer, servizi online



# Tecniche di gestione di un sistema di calcolo

## Evoluzione dei sistemi operativi

### Monoprogrammazione

Gestisce in **modo sequenziale** nel tempo i diversi programmi. L'inizio di un programma avviene solamente dopo il completamento del programma precedente.

**Tutte le risorse sono dedicate ad un solo programma.**

Bassa utilizzazione delle risorse

$$uso\ CPU = \frac{T_p}{T_t}$$

$T_p$  = tempo dedicato dalla CPU all'esecuzione del programma

$T_t$  = tempo totale di permanenza nel sistema del programma

## Sistema mutliprogrammato

Carica in memoria e gestisce **simultaneamente più programmi indipendenti**, nel senso che ciascuno di essi può iniziare o proseguire l'elaborazione prima che un altro sia terminato.  
Le risorse risultano meglio utilizzate in quanto **riduono i tempi morti**