

# Sistemi Operativi

## Introduzione

41

## Cosa è un Sistema Operativo?

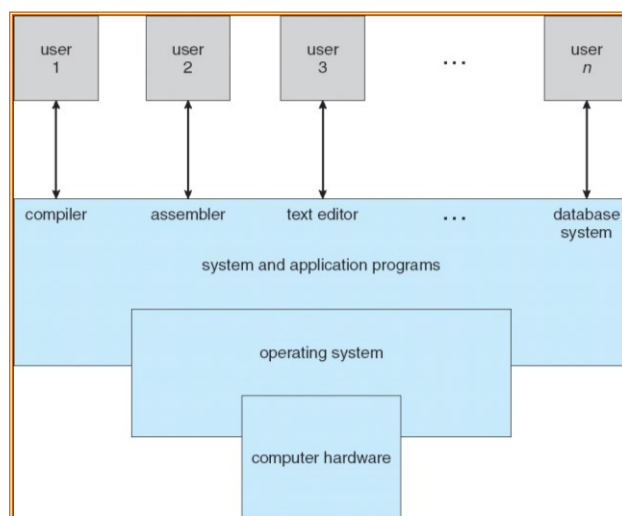
- Un programma che agisce come intermediario tra un utente e l'hardware dell'elaboratore.
- Obiettivi dei Sistemi Operativi:
  - Eseguire i programmi degli utenti e facilitare la risoluzione dei problemi degli utenti.
  - Rendere conveniente l'uso dei calcolatori.
  - Usare l'hardware del calcolatore in modo efficiente.

42

## Struttura di un Calcolatore

- Un Calcolatore può essere diviso in 4 componenti
  - **Hardware** – fornisce componenti di base
    - CPU, memoria, dispositivi I/O
  - **Sistema Operativo**
    - Controlla e coordina l'uso dell'hardware tra le varie applicazioni e gli utenti
  - **Programmi Applicativi** – usano le risorse del sistema per risolvere i problemi computazionali degli utenti
    - Es. Word processors, compilatori, web browsers, database systems, video games
  - **Utenti**
    - Persone, macchine, altri calcolatori

## Quattro componenti di un Calcolatore



## Definizione di Sistema Operativo

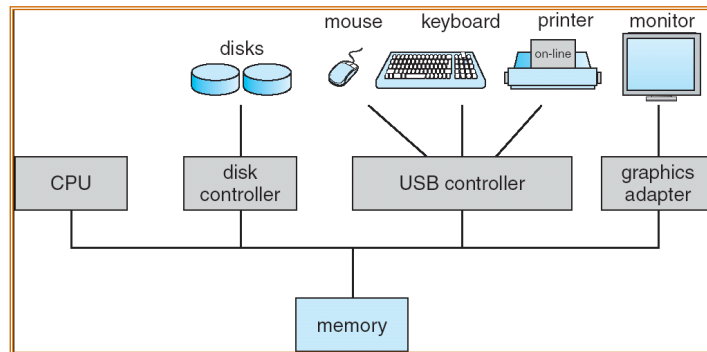
- Il Sistema Operativo è un **allocatore di risorse**
  - Gestisce tutte le risorse del calcolatore
  - Decide tra richieste in conflitto per un uso efficiente e giusto delle risorse disponibili.
- Il Sistema Operativo è un **programma di controllo**
  - Controlla l'esecuzione dei programmi per prevenire errori ed un uso improprio del calcolatore
- Non esiste una definizione universalmente accettata
- "Il programma sempre in esecuzione sul calcolatore" è il **kernel (nucleo) del SO**.  
Ogni altra cosa o è un **programma di sistema** (venduto con il sistema operativo) o è un **programma applicativo**

## Avvio del calcolatore

- Il **programma di bootstrap** è caricato all'accensione o al reboot
  - Tipicamente memorizzato in ROM o EEPROM, noto come **firmware**
  - Inizializza e controlla tutti gli aspetti del sistema
  - Carica il kernel del sistema operativo e inizia la sua esecuzione
- In caso di fallimento il sistema non è utilizzabile
- Es. bootloop dei telefoni cellulari

## Organizzazione di un Calcolatore

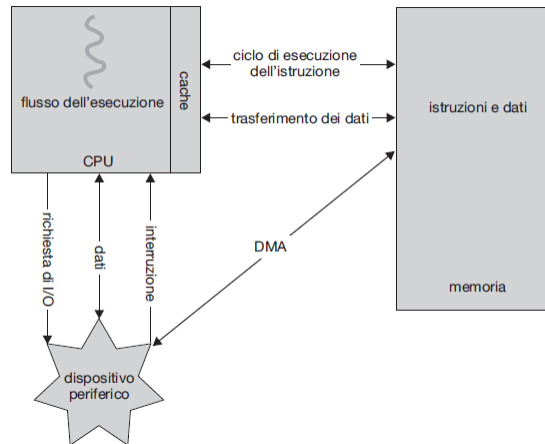
- Nel Calcolatore:
  - Una o più CPU e controllori di dispositivi connessi attraverso un bus comune che fornisce accesso a una memoria condivisa
  - Esecuzione concorrente delle CPU e dispositivi che competono per l'uso della memoria (memory cycles)



## Calcolatore

- Dispositivi I/O e la CPU possono operare in modo concorrente.
- Ogni controllore di dispositivo si occupa di un tipo di dispositivo (es. controller del disco).
- Ogni controllore di dispositivo ha un suo buffer locale.
- CPU sposta dati da/a memoria a/da buffer locali
- I/O è dal dispositivo al buffer locale del controllore.
- Il controllore di dispositivi informa la CPU che ha finito la sua operazione causando una interruzione (*interrupt*).

## Calcolatore



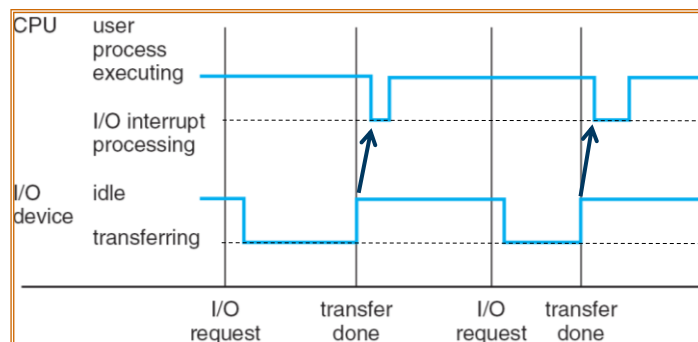
## Funzioni comuni delle Interruzioni

- L'interruzione trasferisce il controllo alla routine di gestione della interruzione attraverso il **vettore delle interruzioni** che contiene gli indirizzi di tutte le routine di servizio.
- L'interruzione deve salvare l'indirizzo dell'istruzione interrotta.
- Le interruzioni vengono *disabilitate* quando una interruzione viene gestita per impedire la *perdita di interruzioni*.
- Una *trap* e' una interruzione generata dal software causata o da un errore o da una richiesta dell'utente.
- **I sistemi operativi sono guidati dalle interruzioni** (*interrupt driven*).

## Gestione delle Interruzioni

- Il Sistema Operativo preserva lo stato della CPU memorizzando il contenuto dei registri e del program counter.
- Determina che interruzione e' avvenuta:
  - *polling*
  - *vectored* interrupt system
- Segmenti di codice separati determinano che azioni intraprendere per ogni tipo di interrupt

## Timeline delle Interruzioni



## Architettura degli elaboratori

### ■ Sistemi monoprocesso

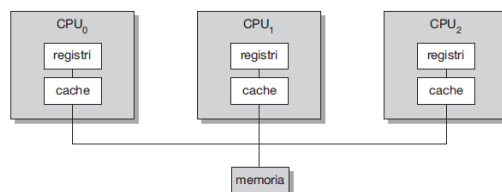
- Possiedono una sola CPU per l'esecuzione dei programmi utente
- Possono avere altri processori che svolgono compiti specifici es. controller del disco (ma non sono gestiti dal SO)

### ■ Sistemi multiprocessore

- Più di una unità di elaborazione strettamente connesse, condividono il bus, il clock e dispositivi di memorizzazione e periferici.
- Vantaggi:
  - ▶ **Maggiore produttività**
  - ▶ **Economia di scala**, condivisione periferiche
  - ▶ **Incremento della affidabilità**, un guasto non blocca il sistema, lo rallenta (fault-tolerant)

## Sistemi multiprocessore

- Di due tipi:
  - **Multielaborazione asimmetrica (AMP)**, in cui ogni unità ha un compito specifico e una unità principale coordina e assegna il lavoro alle altre unità (soluzione poco frequente)
  - **Multielaborazione simmetrica (SMP)**, in cui ogni processore è abilitato ad eseguire tutte le funzioni del sistema (soluzione più frequente)
- Tendenza recente di avere più **core** in uno stesso circuito integrato (risparmio energia)



## Struttura della memoria

- **Memoria primaria** - la sola unità di memorizzazione a cui la CPU può accedere direttamente.
- **Memoria secondaria** - estensione della memoria primaria che fornisce una grande capacità di memorizzazione non volatile.
- **Dischi magnetici** - piatti di metallo o vetro coperti da materiale magnetizzabile
  - La superficie del disco è divisa in *tracce* che sono suddivise in *settori*.
  - Il controller del disco determina l'interazione tra il dispositivo e il computer.

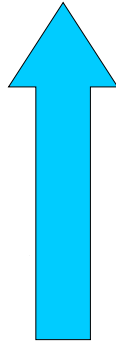
## Gerarchia della Memoria

- Sistemi di memorizzazione organizzati in una gerarchia, secondo tre aspetti
  - Velocità
  - Costo
  - Volatilità
- *Caching* - copiare informazioni in un sistema di memorizzazione più veloce; la memoria primaria può essere vista come *cache* per la memoria secondaria.

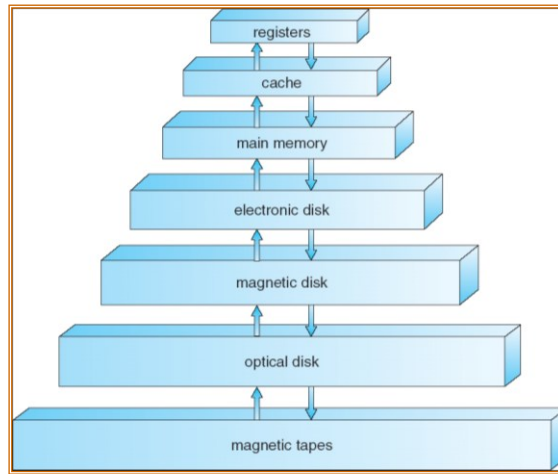


## Gerarchia dispositivi memorizzazione

+veloci, +costosi, volatili



-veloci, -costosi,  
permanenti



57

## Caching

- Principio importante, utilizzato a vari livelli in un calcolatore (nell'hardware, sistema operativo, software)
- Informazione in uso viene copiata temporaneamente da dispositivo di memorizzazione lento a uno più veloce.
- Memoria veloce (cache) controllata prima per determinare se l'informazione è presente
  - Se lo è, l'informazione è usata direttamente dalla cache (veloce)
  - Altrimenti, dati copiati nella cache e usati da lì
- La cache è più piccola della memoria dietro la cache
  - Gestione della cache, importante problema di progettazione
  - Dimensione della cache e politica di rimpiazzamento

58

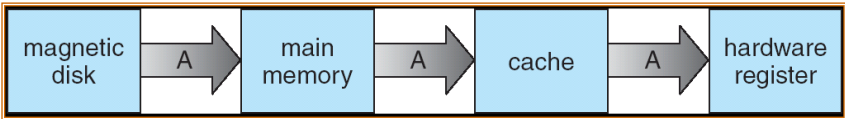
## Performance a vari livelli di memorizzazione

- Movimenti di dati tra livelli della gerarchia possono essere espliciti (es. caricamento dati da disco) o impliciti (es. cache processore)

Level	1	2	3	4
Name	registers	cache	main memory	disk storage
Typical size	< 1 KB	> 16 MB	> 16 GB	> 100 GB
Implementation technology	custom memory with multiple ports, CMOS	on-chip or off-chip CMOS SRAM	CMOS DRAM	magnetic disk
Access time (ns)	0.25 – 0.5	0.5 – 25	80 – 250	5,000,000
Bandwidth (MB/sec)	20,000 – 100,000	5000 – 10,000	1000 – 5000	20 – 150
Managed by	compiler	hardware	operating system	operating system
Backed by	cache	main memory	disk	CD or tape

## Migrazione di un intero A dal disco al registro

- In ambienti **multitasking** si deve stare attenti ad usare il valore più recente, indipendentemente di dove è memorizzato nella gerarchia delle memorie



- Sistemi **multiprocessore** devono fornire coerenza della cache in hardware in modo tale che tutte le CPU abbiano lo stesso valore nella loro cache.
- In ambienti **distribuiti** la situazione è ancora più complessa
  - Possono esistere molte copie del dato in calcolatori diversi

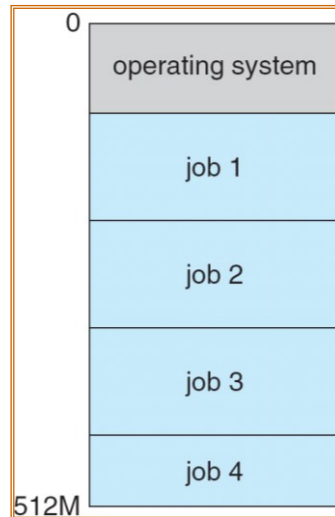
## Struttura di un Sistema Operativo

- **Multiprogrammazione** necessaria per efficienza
  - Un singolo utente non può tenere la CPU e i dispositivi I/O sempre occupati
  - La multiprogrammazione gestisce lavori (jobs) (codice e dati) in modo che la CPU abbia sempre qualcosa da fare...
  - Un sottoinsieme dei lavori presenti sul sistema è tenuta in memoria
  - Un lavoro viene selezionato ed eseguito attraverso il **job scheduling**
  - Quando deve aspettare (per I/O per esempio), OS commuta su un altro job.

## Struttura di un Sistema Operativo

- **Timesharing (multitasking)** la CPU cambia job così frequentemente che l'utente può interagire con il job mentre è in esecuzione, dando vita all' **interactive computing**
  - **Tempo di risposta** dovrebbe essere  $< 1$  secondo
  - Ogni utente ha almeno un programma in esecuzione in memoria  $\Rightarrow$  **processo**
  - Se molti processi sono pronti all'esecuzione allo stesso tempo  $\Rightarrow$  **CPU scheduling**
  - Se i processi non stanno tutti in memoria, lo **swapping** li sposta dentro/fuori dalla memoria per essere eseguiti
  - La **memoria virtuale** permette l'esecuzione di processi anche non completamente in memoria

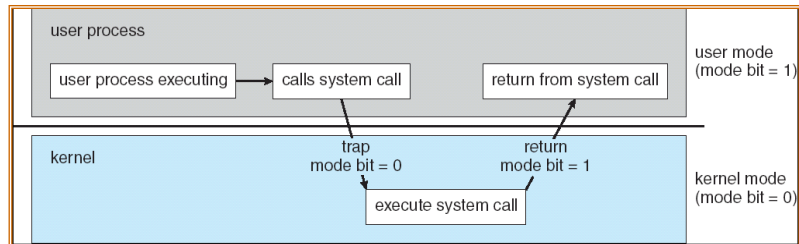
## Schema della memoria in un sistema multiprogrammato



## Sistema Operativo

- Interruzioni guidate dall'hardware
- Un errore software o una richiesta crea una **eccezione** o una **trap**
  - Divisione per zero, richiesta di un servizio del SO
- Altri problemi dei processi includono: i loop infiniti, processi che si modificano a vicenda o modificano il sistema operativo
- **Dual-mode** permette al SO di proteggersi e proteggere altri componenti del sistema
  - **User mode** e **kernel mode**
  - **Bit di modo** fornito dall'hardware
    - Permette di distinguere quando il sistema sta eseguendo codice utente o codice del kernel
    - Alcune istruzioni designate come **privilegiate** sono eseguibili solo in modo kernel
    - Le chiamate al sistema cambiano il bit di modo a modo kernel e al termine della chiamata ritorna al modo utente

## Transizione da modo utente a modo kernel



## Timer

- Il SO usa un timer per prevenire i loop infiniti / processi che si impossessano delle risorse del calcolatore senza rilasciarle
  - Il timer hardware genera un interrupt ad intervalli regolari
  - Il Sistema Operativo decrementa un contatore
  - Quando il contatore raggiunge lo zero genera un interrupt
  - Il contatore viene impostato dal sistema operativo prima di schedare un processo, per riottenere il controllo e terminare il processo se occupa più tempo del previsto.

## Gestione Processi

- Un processo è un programma in esecuzione. E' una unità di lavoro nel sistema di calcolo. Un Programma è una **entità passiva**, un processo è una **entità attiva**.
- Un processo ha bisogno di risorse per svolgere il suo compito
  - CPU, memoria, I/O, files
  - Dati iniziali
- La terminazione di un processo implica la riappropriazione da parte del SO di ogni risorsa riusabile
- Un *processo single-thread* ha un **program counter** che specifica la locazione della prossima istruzione da eseguire
  - Il processo esegue le istruzioni sequenzialmente, una alla volta, fino al completamento
- Un *processo multi-thread* ha un program counter per thread
- Tipicamente il sistema ha molti processi in esecuzione su una o più CPU, alcuni processi sono degli utenti altri del sistema operativo
  - Concorrenza ottenuta facendo multiplexing della/e CPU tra i processi / threads

## Attività della Gestione dei Processi

Il Sistema Operativo è responsabile delle seguenti attività per la gestione dei processi:

- Creare e cancellare processi utente o del sistema operativo
- Sospendere e riprendere processi
- Fornire meccanismi per la sincronizzazione dei processi
- Fornire meccanismi per la comunicazione tra processi
- Fornire meccanismi per la gestione dello stallo (deadlock)

## Gestione della Memoria

- Tutti i dati sono in memoria prima e dopo l'esecuzione
- Tutte le istruzioni devono essere in memoria per poter essere eseguite
- La gestione della memoria determina cosa è in memoria e quando
  - Ottimizza l'utilizzo della CPU e la risposta del sistema
- Attività di gestione della memoria
  - Tiene traccia di quali parti della memoria sono utilizzate e da chi
  - Decide quali processi (o sue parti) e dati muovere in/out dalla memoria primaria
  - Alloca e disalloca spazi di memoria in base alle necessità

## Gestione Memoria Secondaria

- Il Sistema Operativo fornisce una vista uniforme e logica della memoria secondaria
  - Astrae proprietà fisiche (dischi, tracce, settori) in unità logiche di memorizzazione, i **file**
  - Ogni medium è controllato da un dispositivo (i.e., disk drive, tape drive)
    - Si differenziano per velocità di accesso, capacità, velocità di trasferimento, metodo di accesso (sequenziale o diretto)
- **Gestione del File-System**
  - Files solitamente organizzati in cartelle/directories
  - Controllo degli accessi per determinare chi può usare cosa
  - Attività del SO includono
    - Creare e cancellare files e directories
    - Primitive per manipolare files e directories
    - Mappare files in memoria primaria
    - Backup files su supporti di memorizzazione stabili (non volatili)

## Gestione Memorie di massa

- Solitamente i dischi sono usati per memorizzare dati che non entrano in memoria primaria o dati che devono essere memorizzati per lungo tempo.
- Una gestione oculata è fondamentale
- L'intera velocità del calcolatore è basata sul sottosistema di gestione dei dischi e sui suoi algoritmi di gestione
- Attività del SO
  - Gestione dello spazio libero
  - Allocazione
  - Scheduling del disco

## Sottosistema I/O

- Uno degli scopi dei Sistemi Operativi è di nascondere all'utente le peculiarità dei dispositivi hardware
- Il sottosistema I/O è responsabile di:
  - Gestione della memoria per I/O per:
    - **buffering** (memorizzare i dati temporaneamente mentre vengono trasferiti),
    - **caching** (memorizzare dati in memorie veloci per aumentare la performance),
    - **spooling** (memorizzazione dati verso device lenti, I/O asincrono)
  - Interfaccia di uso generale tra device e driver
  - Drivers per dispositivi speciali



## Protezione e Sicurezza

- **Protezione** - ogni meccanismo per controllare l'accesso dei processi/utenti a risorse definite dal Sistema Operativo
- **Sicurezza** - difesa del sistema contro attacchi interni ed esterni
  - Tipo: denial-of-service, worms, viruses, furto di identità, furto di servizi
- I sistemi generalmente distinguono gli utenti per determinare chi può fare cosa
  - Identità degli utenti - User identities (**user IDs**, security IDs) includono nome e un numero associato, uno per ogni utente
  - User ID è associato a tutti i file e processi dell'utente per realizzare il controllo degli accessi
  - Group identifier (**group ID**) permette di identificare un gruppo di utenti e può essere usato per controllare accesso risorse a livello di gruppo, associato a ogni processo/file
  - **Privilege escalation** permette in alcuni casi all'utente di cambiare il proprio id per avere più diritti

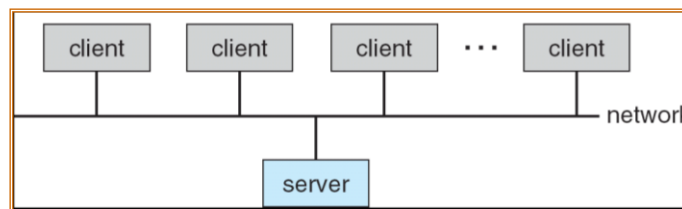
## Ambienti di Elaborazione

- **Personal Computer**
  - In ufficio
    - PCs connessi a una rete locale, terminali connessi a un mainframe o minicomputers forniscono accessi batch e in timesharing
    - Portali permettono a sistemi in rete e remoti l'accesso alle stesse risorse
  - A casa
    - Era un singolo sistema
    - Ora più sistemi in rete dietro firewall

## Ambienti di Elaborazione (Cont.)

- Client-Server Computing

- Terminali stupidi soppiantati dai PC
- Molti sistemi sono **servers**, che rispondono a richieste fatte dai **clients**
  - **Compute-server** fornisce una interfaccia ai client per richiedere servizi (es. database)
  - **File-server** fornisce una interfaccia ai client per memorizzare e accedere ai file



## Peer-to-Peer Computing

- Un altro modello di sistema distribuito
- P2P non distingue tra client e server
  - Tutti i nodi sono considerati peers (pari)
  - Ogni nodo può agire come client, server o entrambi
  - Un nodo deve connettersi a una rete P2P
    - Registra il suo servizio su un servizio centrale, o
    - Fa richieste in broadcast per un servizio e risponde a richieste di servizio usando un *discovery protocol*
  - Esempi sono *BitTorrent*, *Gnutella*, ...

## Altri ambienti di elaborazione

- Mobile computing
  - Risorse «limitate»
  - Basso consumo energetico
  - Sensori GPS, movimento
  - Sempre connessi
- Cloud computing
  - Cloud pubblico/privato/ibrido
  - Si basa su virtualizzazione
- Sistemi embedded (IoT)
  - Risorse limitate
  - Connessi