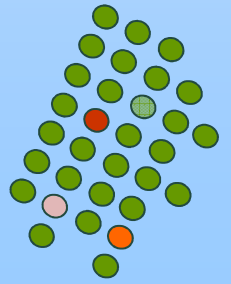
Corso di
Sistemi Operativi

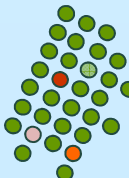


Informazioni

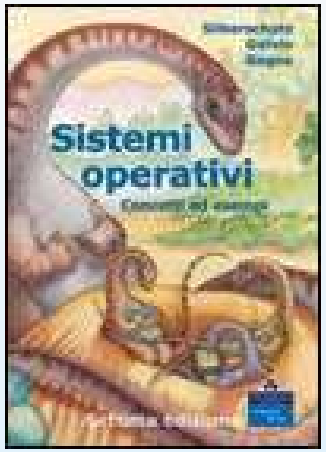
- Docente: Prof.ssa Adele Rescigno
  - E-mail: rescigno@dia.unisa.it
  - http://www.dia.unisa.it/professori/rescigno/SO/lo.htm
    - Programma dettagliato
    - Slide delle lezioni
    - Informazioni sul corso
    - …. e tutto quanto sarà necessario

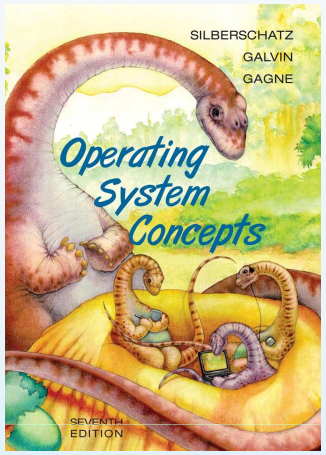
- Orario delle lezioni
  - Martedì 9:00 - 11:00, aula P/6 Mercoledì 11:00 – 13:00, aula P/6
  - Giovedì 11:00 -14:00, Lab. Turing

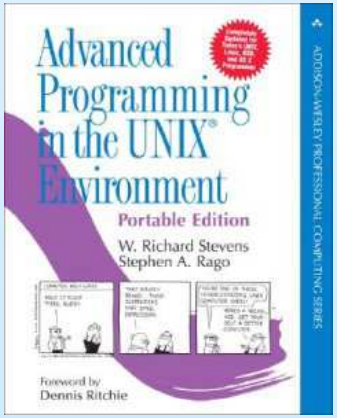
- Ricevimento
  - Martedì 11:00 - 13:00 Giovedì 10:00 - 11:00
    - Uff. 59 (4° piano, stecca 7)

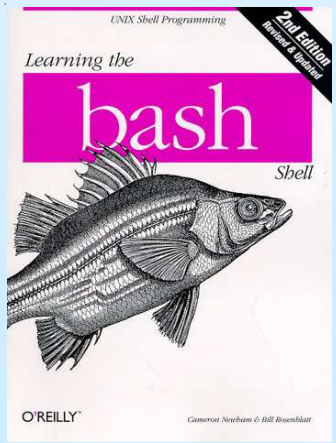


Libri di testo









Ulteriori informazioni

- Linux Documentation Project:
  - Linux User's Guide

- M. Sobell, "A practical Guide to Linux“
  Addison Wesley

- *Guida dell'utente di Linux*,
  L. Greenfield, http://www.pluto.it/files/ildp/guide/GuidaUtente/index.html
  (in italiano)

- *Bash Reference Manual*,
  Free Software Foundation
  http://www.gnu.org/software/bash/manual/bash.html

- *Appunti di Linux - Bash*,
  http://appunti.linux.it/a22.htm (in italiano)

Installazione linux

- Giovedì 29 febbraio
- Laboratorio Turing (secondo piano nel corridoio adiacente ex segreteria studenti)
- Portare una pen drive USB da 8 GiBytes

Sistemi Operativi

Capitolo 1 -- Silberschatz

Concetti introduttivi

- Che cos'è un sistema operativo
- Organizzazione di un sistema di calcolo
- Architettura degli elaboratori
- Struttura del sistema operativo
- Attività del sistema operativo
- Gestione dei processi
- Gestione della memoria
- Gestione della memoria di massa
- Protezione e sicurezza

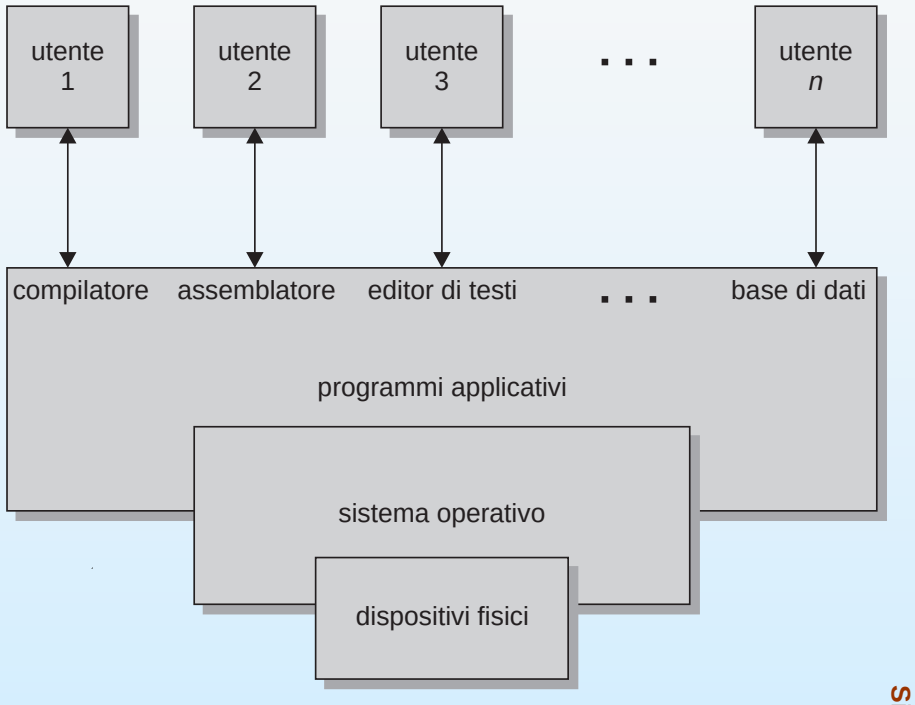
Che cos’è un sistema operativo?

Che cos’è un sistema operativo?

- È un programma che opera da intermediario tra l’utente e l’hardware del computer
- Assicura che il computer operi correttamente e che le risorse siano usate efficientemente
- Esegue i programmi degli utenti e facilita i loro compiti offrendo un ambiente d’uso conveniente

Componenti di un sistema di calcolo

- **Hardware** – fornisce risorse computazionali di base
  - CPU, memoria, I/O device
- **Sistema Operativo**
  - Controlla e coordina l'uso dell'hardware tra applicazioni e utenti
- **Programmi** – definiscono i modi attraverso i quali le risorse del sistema vengono usate per risolvere problemi computazionali degli utenti
  - word processor, compiler, web browser, database,video game
- **Utenti**
  - Persone, dispositivi, altri computer



Ruolo del Sistema Operativo

- **PC**. Il sistema operativo è progettato principalmente per facilitare l'uso del computer.
- **Mainframe e Minicomputer**. Occorre massimizzare l'uso delle risorse.
- **Workstation**. Compromesso ottimale tra uso risorse individuali e risorse condivise.
- **Palmari e simili**. Progettati per l'uso individuale prestando attenzione al consumo della batteria
- **Sistemi Embedded**. Concepiti per funzionare senza l'intervento dell'utente

Visione del sistema

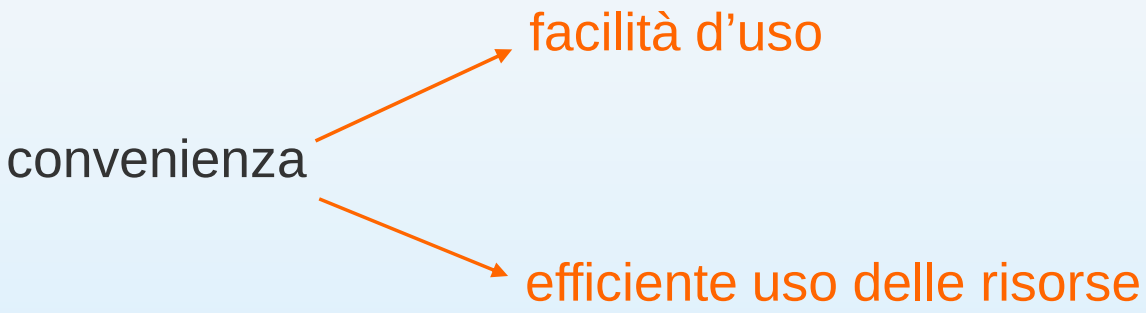
- Il **S**istema **O**perativo (**SO** in breve) è il programma più intimamente connesso con l'hardware. Quindi, è:
  - **allocatore di risorse:** di fronte a richieste conflittuali, decide come assegnare equamente ed efficientemente le risorse ai programmi (e.g. tempo di CPU, spazio di memoria)
  - **programma di controllo:** garantisce l'esecuzione dei programmi senza errori e usi impropri del computer
  - **esecutore di funzioni comuni:** esegue funzioni di utilità generale comuni ai diversi programmi (e.g. routine di I/O)

Definizione di Sistema Operativo

- Non esiste una definizione universalmente accettata
- “Tutto ciò che il venditore ti invia quando ordini un sistema operativo” - è una buona approssimazione ma varia ampiamente
- “Il programma che è sempre in esecuzione sul computer” è il **kernel.** Tutto il resto è o un programma di sistema o un programma applicativo.

Sistema Operativo: cos’è e cosa fa?

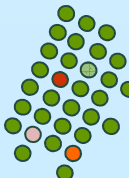
- I sistemi operativi esistono perché forniscono agli utenti uno strumento conveniente per l’uso di un sistema di calcolo



- Gran parte della teoria dei SO si è concentrata sull’efficienza. Inoltre, hardware e SO si sono influenzati vicendevolmente

Organizzazione di un elaboratore

Componenti e Meccanismi



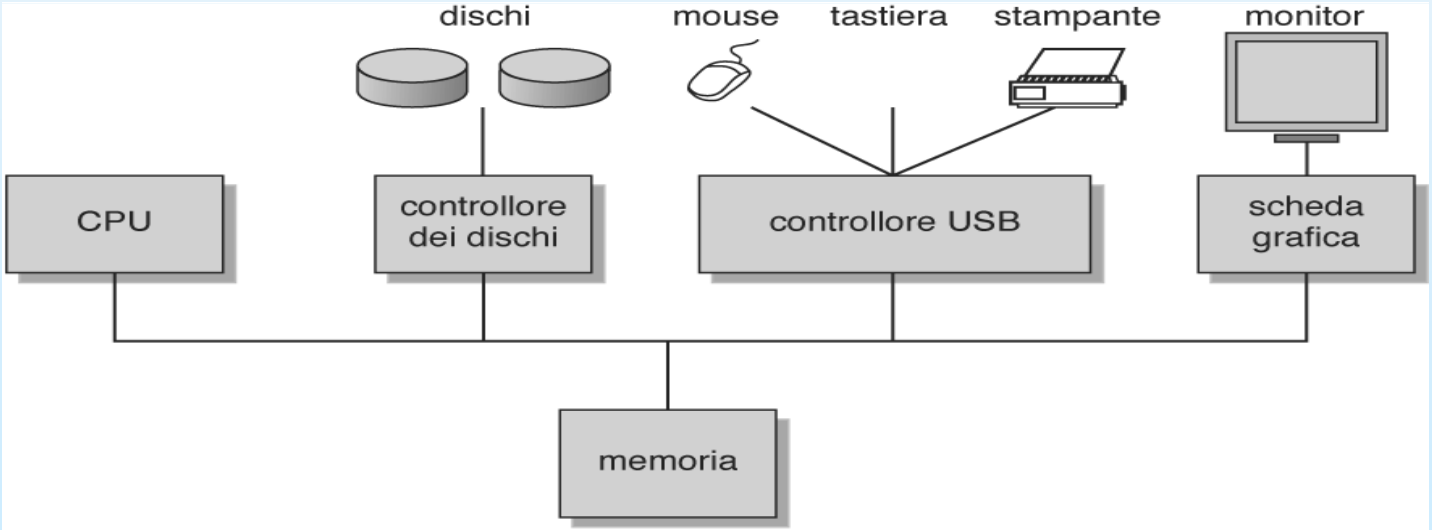
Bootstrap

- **Un programma di bootstrap** viene caricato quando il computer viene accesso o viene riavviato
  - Tipicamente è memorizzato in una ROM o in una EEPROM (**firmware)**
  - Inizializza tutte le funzioni principali del sistema, dai registri della CPU ai controller della memoria
  - Carica il kernel del sistema operativo e comincia l'esecuzione

Il kernel del sistema operativo avvia l'esecuzione del primo processo (**init**) ed aspetta che si verifichino **eventi** o richieste degli utenti da eseguire

Moderno sistema di calcolo

- Modalità di funzionamento
  - Le CPU e controller di dispositivi sono connessi ad un bus comune che fornisce accesso alla memoria condivisa
  - Le CPU e i controller dei dispositivi competono per ottenere cicli di accesso alla memoria



Struttura di un elaboratore

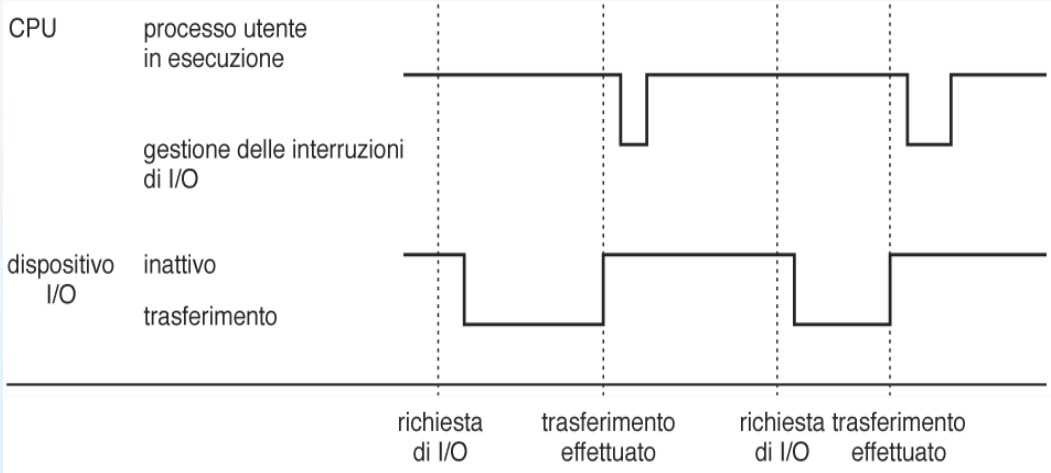
- I dispositivi di I/O e la CPU lavorano concorrentemente.
  - Ciascun controller gestisce un particolare tipo di dispositivo
  - Ogni controller possiede un buffer locale
  - I/O avviene tra i dispositivi e i buffer locali dei controller
  - La CPU trasferisce dati dalla/alla memoria in/da buffer locali
  - I controller dei dispositivi informano la CPU che hanno finito il proprio lavoro generando un *interrupt.*

Interrupt driven

- Gli eventi sono segnalati da **interrupt** o da eccezioni (**trap).**
  - Per ogni tipo di interrupt, segmenti separati del codice del SO (**routine di gestione dell'interrupt**) determinano le azioni da intraprendere per gestire l'evento.
  - Una **trap** è un interrupt generato dal software, causato o da un errore durante la computazione o da una richiesta specifica dell'utente (chiamata di sistema).
  - Gli interrupt sono solitamente **disabilitati** mentre si sta eseguendo una routine di gestione di un altro interrupt, al fine di prevenire la perdita di interrupt

Meccanismo delle interruzioni

Quando la CPU riceve un interrupt, **sospende** ciò che sta facendo e **comincia ad eseguire** codice a partire da una locazione fissa, che contiene l'indirizzo di partenza della routine di interrupt.



- L'architettura dell'interrupt trasferisce il controllo alla routine di gestione attraverso il **vettore degli interrupt**, che contiene gli indirizzi di tutte le routine di servizio.
- L'architettura dell'interrupt **salva** l'indirizzo dell'istruzione interrotta e lo stato del processore (i.e., registri) in un'area di memoria chiamata **stack.**

Due metodi per l'I/O

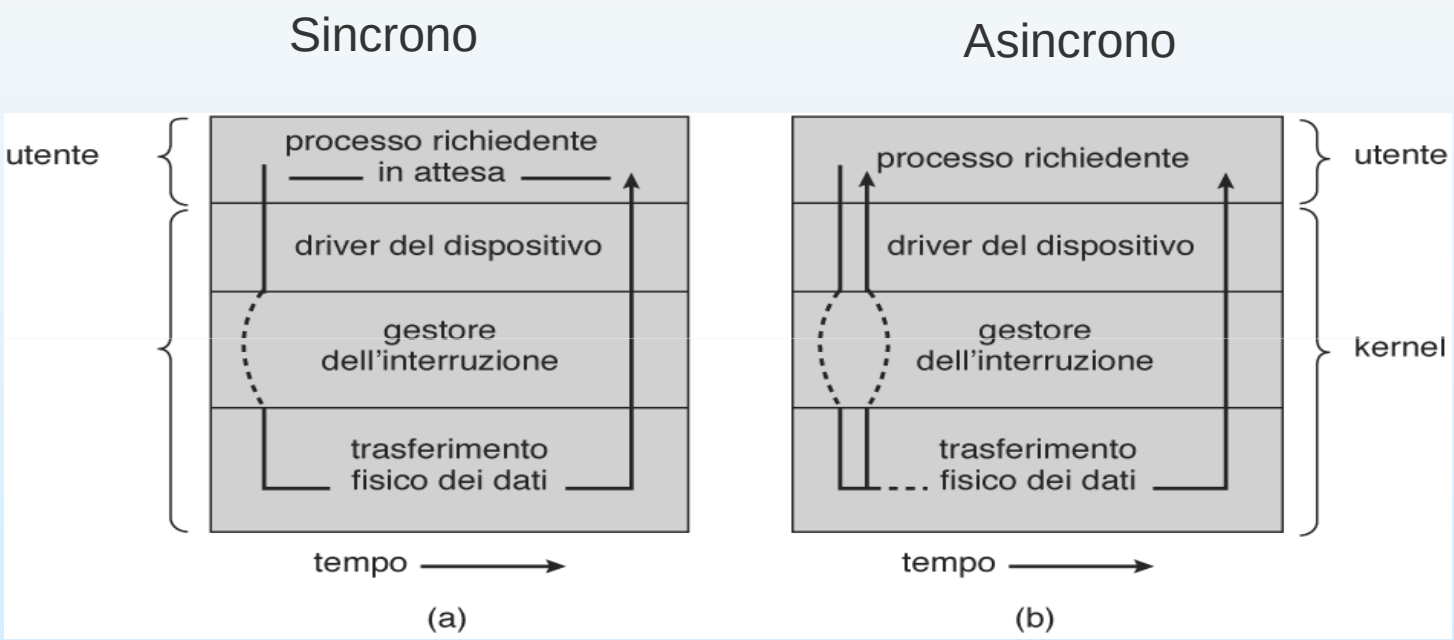
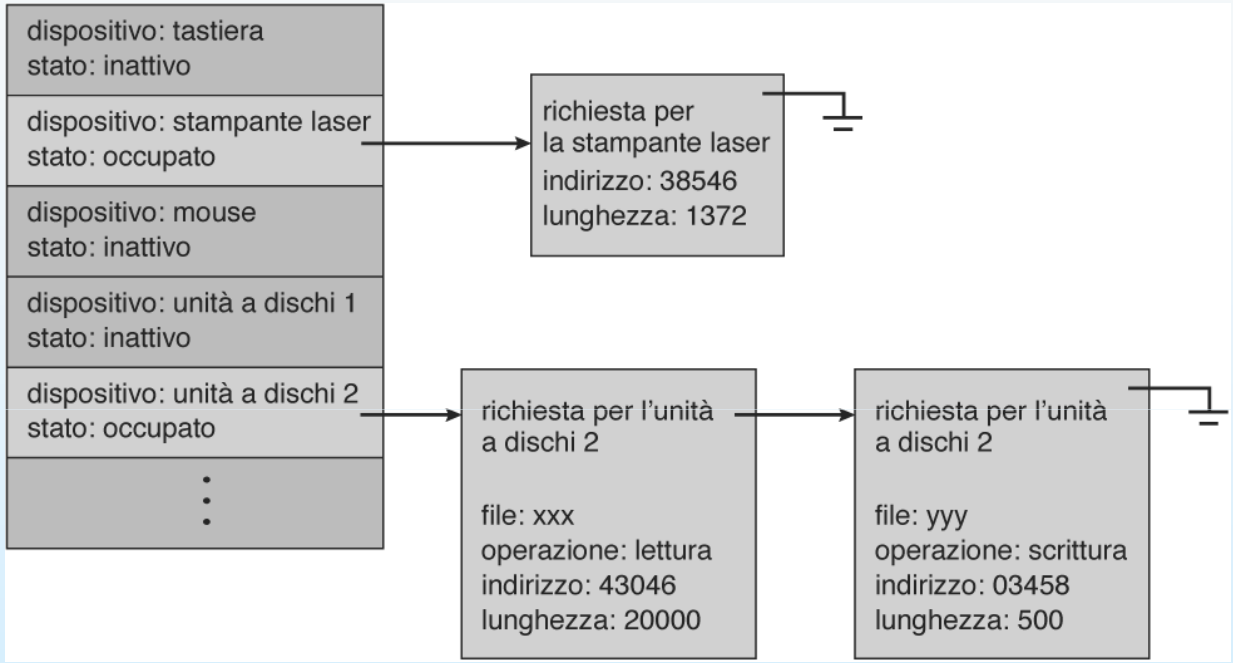


Tabella dello stato dei dispositivi

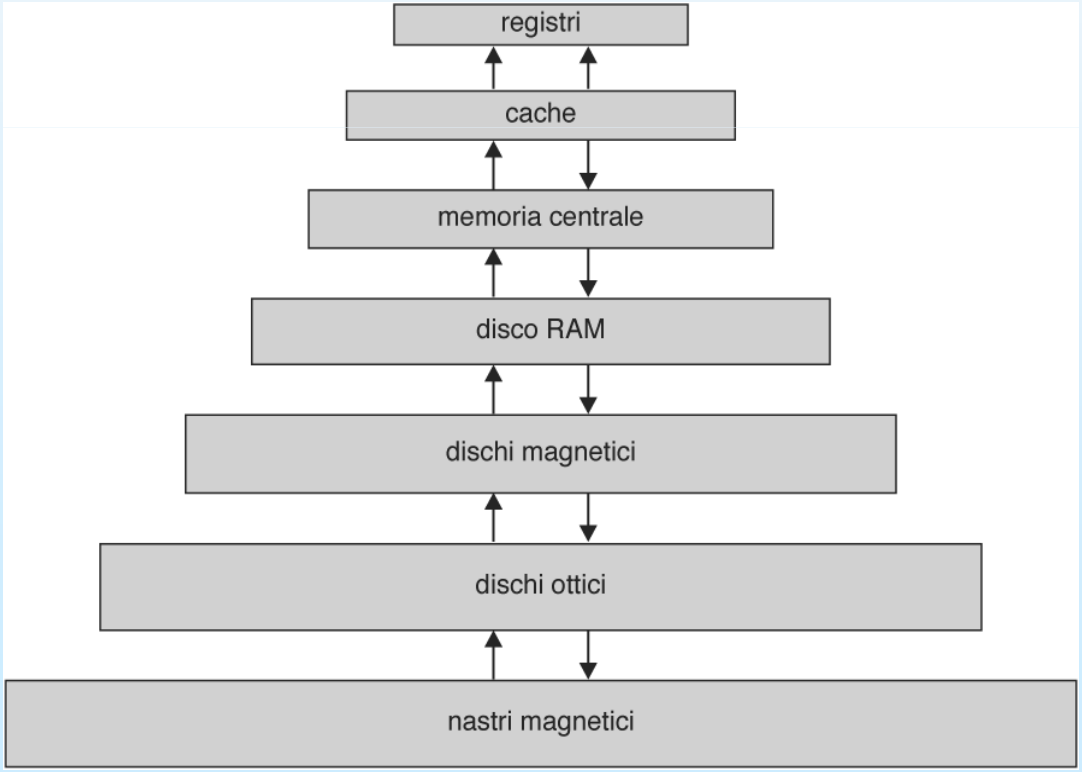


Dispositivi di memoria

- **Memoria centrale** – è la sola grande memoria a cui la CPU può accedere.
- **Memoria secondaria** – estensione della memoria centrale che fornisce grande capacità di memorizzazione non volatile.
- **Dischi magnetici** – piatti rigidi di metallo o vetro coperti da materiale magnetico

Gerarchia dei dispositivi di memoria

- I sistemi di memorizzazione sono organizzati gerarchicamente.
  - Velocità
  - Costo
  - Volatilità

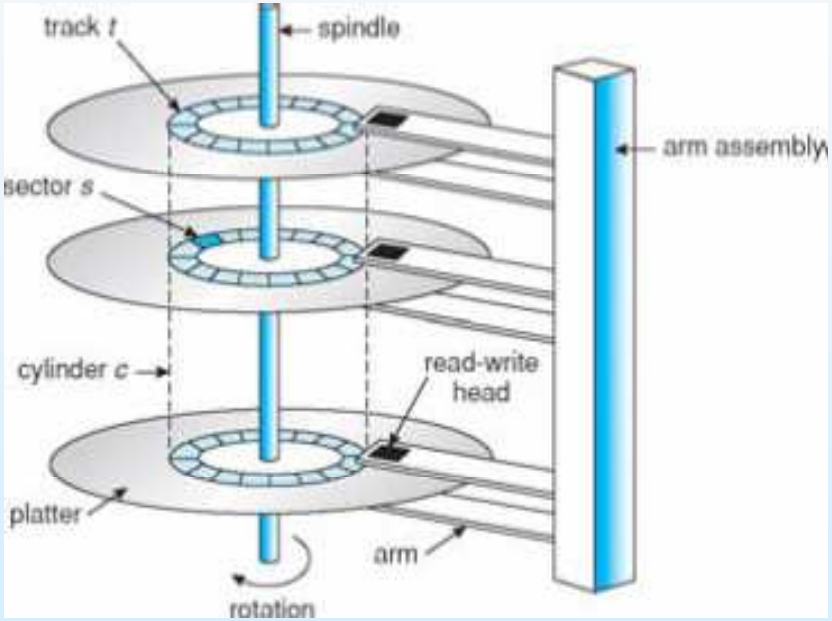


Struttura della memoria centrale

- La memoria centrale (RAM) è una **sequenza di parole** a cui il processore può accedere direttamente attraverso il bus.
- Ogni parola ha un proprio **indirizzo**.
- L'interazione avviene tramite istruzioni **load** e **store**.
- La memoria contiene istruzioni e dati, ma l'unica cosa che vede è un **flusso** di indirizzi.
- È volatile.

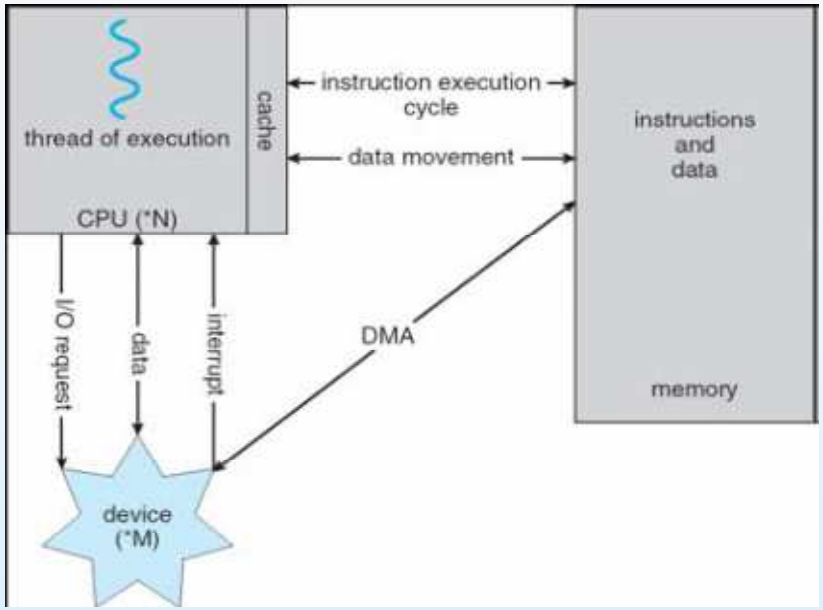
Struttura della memoria secondaria

- Disco magnetico
  - Un disco è composto da piatti
  - Un testina di lettura/scrittura sfiora ogni piatto
  - Ogni piatto è diviso in **tracce**.
  - Ogni traccia in **settori**.
  - L'insieme delle tracce che si trova sotto un braccio forma un **cilindro.**



Accesso diretto alla memoria (DMA)

- Usato per dispositivi di I/O (nastri, dischi, reti di comunicazione) capaci di trasmettere informazione ad **alte velocità**, prossime a quella della memoria.
  - Il controller del dispositivo trasferisce **blocchi** di dati dal buffer direttamente alla memoria **senza l'intervento della CPU**.
  - Viene generato soltanto un interrupt per blocco, piuttosto che un interrupt per ogni byte.



Architetture

Architetture a singolo processore

- Tali sistemi sono dotati di un singolo processore che esegue un set di istruzioni general-purpose

- Spesso usano anche processori special purpose, quali processori di I/O, che muovono velocemente dati tra le componenti (e.g., disk-controller, processori associati alle tastiere).
  - I processori special purpose eseguono un insieme ristretto di istruzioni e non processi utenti
  - il microprocessore associato al controller del disco deve implementare la coda di richieste e l'algoritmo di scheduling

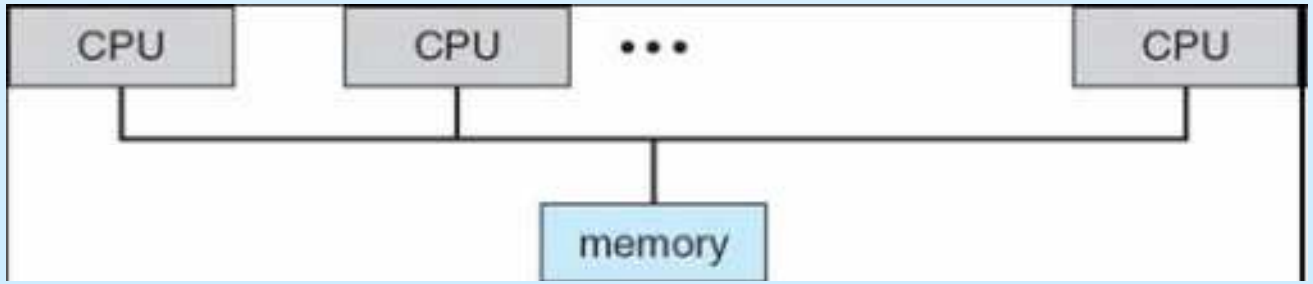
- A volte la CPU principale comunica con questi processori. Altre volte, essi sono totalmente autonomi.
  - In tutti i casi la CPU è unica e quindi si tratta di un sistema a monoprocessore

Architetture a multi-processore

- Questi sistemi, anche detti paralleli o con processori strettamente accoppiati, posseggono più processori che **condividono il bus del computer, il clock, la memoria e le periferiche**.
  - Maggiore quantità di elaborazione effettuata
    - è possibile svolgere un lavoro maggiore in meno tempo
    - con n unità la velocità non aumenta di n volte
  - Economia di scala
    - c'è risparmio perché i dispositivi periferici, gli alimentatori elettrici ed altro possono essere condivisi
    - dovendo operare sullo stesso insieme di dati è inutile duplicare
  - Aumento affidabilità
    - un guasto di alcuni processori non blocca l'intero sistema
    - si rallenta perché sulle unità rimanenti si spalma il lavoro delle unità guaste, ma non si ferma

Architetture a multi-processore

- Esistono due tipi di sistema multiprocessore:
  - **sistema multiprocessore asimmetrico** – un processore principale (master) organizza e gestisce il lavoro per gli altri (slave) che svolgono compiti specifici
  - **sistema multiprocessore simmetrico** (SMP) – ogni processore esegue una copia del sistema operativo e, tali copie, comunicano tra loro
    - Ogni processore può compiere tutte le operazioni
    - Tutti i processori su un piano di parità



CPU Multi-core

- Un microprocessore **multi-core** combina due o più processori indipendenti su un singolo supporto, spesso un singolo circuito integrato
  - nel corso del 2005 sono arrivati i primi chip dual core sul mercato : questo perché in pratica, si è giunti ad un momento in cui aumentare ulteriormente la frequenza di clock di una CPU (che fino a quel momento erano state single core) è diventato molto oneroso e complicato, per via dei consumi che hanno superato abbondantemente i 100 W e il conseguente problema di raffreddamento dei circuiti.
  - La soluzione che è sembrata più ovvia ai progettisti è stata quella di puntare tutto sul parallelismo in modo da poter aumentare il numero di operazioni eseguibili in un unico ciclo di clock.

Server Blade

- Un **server blade** (a lama) è essenzialmente un alloggiamento per schede madri, ciascuna contenente uno o più processori, memoria centrale, e connessioni di rete, che condividono il sistema di alimentazione e di raffreddamento dell'intera infrastruttura e le memorie di massa
  - tali server sono costituiti da svariati sistemi multiprocessore indipendenti

Sistemi Cluster

- I sistemi cluster hanno il compito di svolgere attività d'elaborazione comuni.

- Mettono assieme due o più computer che **condividono la memoria di massa e sono collegati tramite cavi veloci**. Cluster è solitamente sinonimo di alta affidabilità.
  - **Cluster asimmetrico**: una macchina si trova in stato di attesa a caldo (hot-standby mode) mentre l'altra esegue le applicazioni desiderate.
    - se la seconda presenta problemi, la prima se ne accorge e la sostituisce diventando server attivo
  - **Cluster simmetrico**: le macchine eseguono le applicazioni e si controllano a vicenda.

Sistemi Cluster

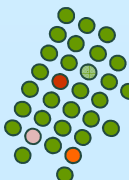
- La tecnologia dei cluster sta evolvendo rapidamente (e.g., cluster paralleli o sistemi connessi attraverso WAN) ed è strettamente legata allo sviluppo delle **SAN (storage area network)** che permettono a molti sistemi di accedere ad un gruppo di dischi direttamente connessi alla rete.
  - Questo comporta il controllo degli accessi e la gestione della mutua esclusione

Struttura di un Sistema Operativo

Componenti principali

Concetti chiave

- Un **processo** è un programma in esecuzione. É l'unità di lavoro nel sistema. Un programma è una **entità passiva**, un processo è **un'entità attiva**.
  - **Terminologia: job – processo - task**
- **Multiprogrammazione** - necessaria per efficienza
  - Un solo utente non può tenere CPU e dispositivi I/O occupati per tutto il tempo
  - Esegue più job e la CPU è sempre impegnata con uno di essi
  - Un sottoinsieme dei job si trova in memoria centrale
  - Un job viene selezionato (**job scheduling)** ed eseguito
  - Quando attende (e.g., operazione di I/O), il SO esegue un altro job



Configurazione della memoria per un sistema con multiprogrammazione

Concetti chiave

- **Timesharing (multitasking)** - estensione logica della multiprogrammazione: la CPU commuta tra i job così frequentemente che gli utenti possono interagire con ciascun job mentre è in esecuzione, realizzando una computazione **interattiva**
  - **Tempo di Risposta** < 1 secondo
  - Ciascun utente ha almeno un programma in esecuzione in memoria
    - **processo**
  - Se diversi processi sono pronti per essere eseguiti
    - **CPU scheduling**
  - Se i processi non entrano in memoria, lo **swapping** li sposta dentro e fuori per eseguirli
  - **La memoria virtuale** permette l'esecuzione di processi che non sono completamente in memoria

Operazioni di un Sistema Operativo

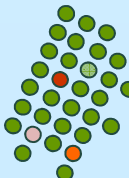
Operazioni di un Sistema Operativo

- Il sistema operativo giace di norma in uno stato di quiete in attesa che accada qualcosa
- Gli eventi sono segnalati da interrupt
  - L'hardware genera interrupt
  - Errori software o richieste generano **exception** o **trap**

Supporto hardware

- Alcuni processi potrebbero ciclare all'infinito, o potrebbero tentare di modificare codice e dati di altri processi o del SO: supporti hardware vengono in aiuto.

- Il **dual-mode** permette al SO di proteggere se stesso e le altre componenti del sistema
  - **user mode** e **kernel mode**
  - **mode bit** fornito dall' hardware
    - Permette di distinguere quando il sistema sta eseguendo codice utente da quando sta eseguendo codice kernel
    - Alcune istruzioni vengono definite **privilegiate**, e sono eseguibili soltanto in modalità kernel



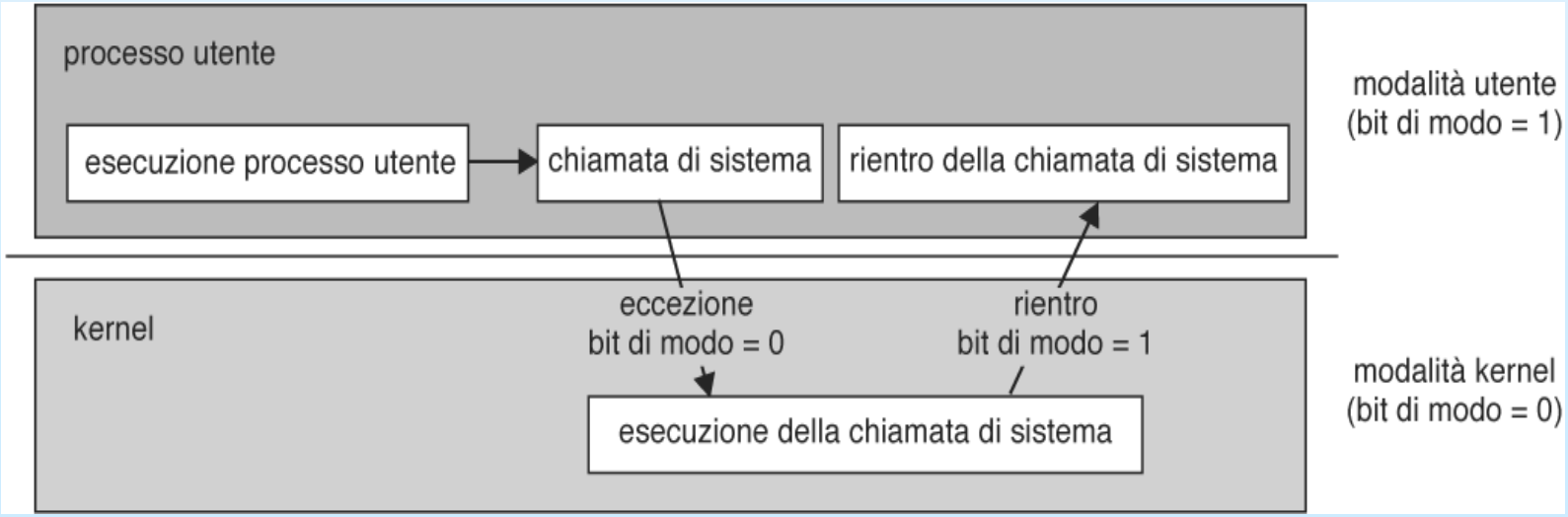
Supporto hardware

- Un **Timer** previene loop infiniti / rilascio risorse dai processi
  - Permette di settare l'invio di un interrupt al termine di uno specifico periodo di tempo
  - Il SO decrementa un contatore. A zero genera l'interrupt
  - Viene settato prima di schedulare i processi per riacquisire il controllo e terminare programmi che eccedono nel tempo

System call – Chiamata di sistema

Le chiamate di sistema sono gli strumenti con cui un programma
utente chiede al sistema di svolgere per lui azioni ad esso riservate

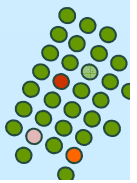
Il SO, nell'esecuzione di una system call, passa in kernel mode. Alla fine della call, ritorna in user mode



Gestione dei processi

Gestione dei processi

- Un processo è un programma in esecuzione. É l'unità di lavoro nel sistema. Un programma è una **entità passiva**, un processo è **un'entità attiva**.
  - Un processo necessita di risorse per svolgere il proprio compito
    - CPU, memoria, I/O, file, dati d'inizializzazione
  - La terminazione di un processo richiede il recupero delle risorse
  - Un processo a singolo flusso d'esecuzione (single thread) ha un **program counter** che specifica la locazione della prossima istruzione da eseguire
    - Un processo esegue istruzioni sequenzialmente, una dopo l'altra.
  - Un processo con più flussi d'esecuzione (multi thread) ha un **program counter per ogni flusso d'esecuzione**



Gestione dei processi

- Tipicamente un sistema ha processi, utente e del SO, eseguiti concorrentemente su una o più CPU
  - La concorrenza viene realizzata commutando le CPU tra i diversi processi / thread
- Il SO è responsabile delle seguenti attività in relazione alla gestione dei processi:
  - **Creazione e cancellazione** sia di processi utenti che del SO
  - **Sospensione** e riavvio di processi
  - Fornire meccanismi per la **sincronizzazione** dei processi
  - Fornire meccanismi per la **comunicazione** dei processi
  - Fornire meccanismi per la **gestione dello stallo** (deadlock)

Gestione della memoria

Gestione della memoria centrale

- Tutti i dati debbono essere in memoria prima e dopo l'elaborazione
  - Tutte le istruzioni debbono essere in memoria per essere eseguite
  - La gestione della memoria determina il contenuto della memoria,
    - ottimizzazione dell'utilizzo CPU e delle risposte agli utenti
  - In relazione alla gestione della memoria, il SO deve
    - **Tener traccia** di quali parti della memoria sono correntemente usate e da chi
    - **Decidere** quali processi (o parti di) e dati **muovere** dentro e fuori dalla memoria
    - **Allocare e deallocare** lo spazio di memoria secondo le esigenze

Gestione dei file

- Il SO fornisce una **visione logica uniforme** della memorizzazione delle informazioni
  - Astrae proprietà fisiche dispositivi in unità logica – **file**
  - Ciascun supporto è controllato da un device (i.e., disk drive, tape drive)
    - Varie proprietà: velocità di accesso, capacità, velocità di trasferimento, metodo d'accesso (sequenziale o random)
- La gestione del File System richiede
  - **Organizzazione** dei file in directory
  - **Controllo d'accesso** per stabilire chi può accedere a cosa

Gestione dei file

- Attività che il SO deve supportare
  - **Creazione e cancellazione** di file e directory
  - **Primitive** per manipolare file e directory
  - **Mappatura** dei file sul disco
  - **Backup** dei file su memorie non volatile

Gestione della memoria di massa

- Il calcolatore usa la memoria secondaria a sostegno della memoria centrale
- I programmi restano sul disco fino al momento del caricamento in memoria e molti di essi si servono del disco come sorgente e destinazione delle loro computazioni (compilatori, editor, …)
- Il sistema operativo è responsabile di
  - gestione spazio libero
  - assegnazione spazio
  - scheduling del disco

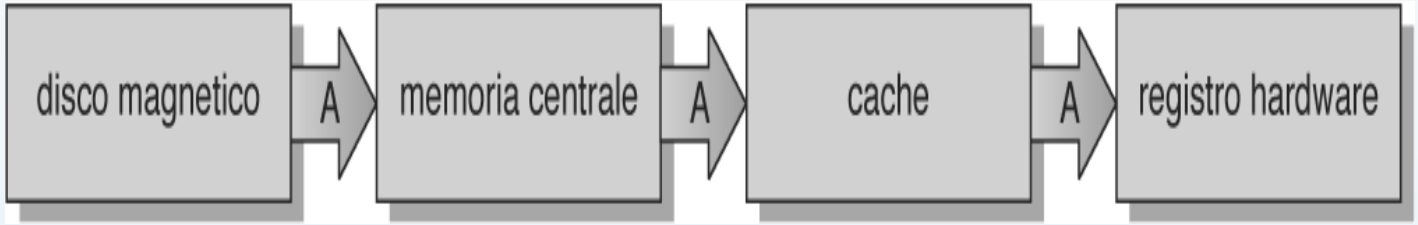
Caching

- E’ un principio importante, implementato nel computer a molti livelli (in hardware, sistema operativo, software)
  - Cache gestite dall’architettura del sistema: cache per la memorizzazione della prossima istruzione da eseguire.
  - I registri indice rappresentano per la memoria centrale una cache ad alta velocità
  - Il compilatore ha algoritmi di assegnazione e aggiornamento della cache
- L’informazione che si sta usando viene copiata temporaneamente da un mezzo di memorizzazione più lento ad uno più veloce (cache)

Caching

- La cache è in primo mezzo di memorizzazione che viene controllato quando si ricerca un dato
  - Se il dato è lì, viene usato direttamente (veloce)
  - Se non lo è, il dato è copiato nella cache ed usato
- Le cache sono solitamente piccole. Problemi:
  - come regolare la taglia della cache?
  - quando effettuare il ripristino?

Coerenza/Consistenza



- In ambienti **multitask**, se parecchi processi provano ad accedere all'intero A, occorre assicurarsi che ognuno di esso ottenga il valore più aggiornato
- In ambienti **multiprocessore** l'hardware deve fornire la coerenza delle cache, in modo tale che tutte le CPU possano accedere al valore aggiornato nelle cache locali
- In ambienti **distributi** la situazione diventa ancora più complessa
  - Possono esistere diverse copie di un dato

Gestione dell’ I/O

- Uno degli scopi del SO è di nascondere le peculiarità dei dispositivi hardware all’utente
- Il sottosistema di I/O è responsabile per
  - **Gestione della memoria**
    - **buffering** (memorizzazione temporanea di dati durante i trasferimenti),
    - **caching** (spostamento di dati in memorie più veloci),
    - **spooling** (sovrapposizione dell’output di un job con l’input di altri in caso di gestione di più processi).
  - **Interfaccia generale** per i driver dei dispositivi
  - **Driver** per specifici dispositivi hardware

Protezione e sicurezza

Protezione e sicurezza

- **Protezione** – ogni meccanismo definito dal SO per controllare l'accesso di processi o utenti a risorse
- **Sicurezza** – difese del SO contro attacchi interni ed esterni
  - e.g., denial-of-service, worm, virus.

Protezione e sicurezza

- I SO generalmente distinguono gli utenti per stabilire chi può fare cosa
  - Identità utenti (**user ID**, security ID) includono nomi e identificativi numerici
  - Gli user ID sono poi associati ai file e ai processi di quell'utente per il controllo dell'accesso
  - Identificativi di gruppo (**group ID**) permettono la definizione e il controllo di gruppi di utenti e sono associati a processi e file
  - **La modifica dei privilegi** permette all'utente il cambio temporaneo del proprio user ID per aver più permessi