Modulo di Sistemi Operativi per il corso di Master
RISS: Ricerca e Innovazione nelle Scienze della Salute
Unisa, 17-26 Luglio 2012



Sistemi Operativi

Architettura di un sistema operativo e dipendenze dalle tipologie di hardware

Docente: Paolo D'Arco
E-mail : paodar@dia.unisa.it, pdarco@gmail.com

http://www.dia.unisa.it/professori/paodar/

Avete conoscenze di base di …

- Matematica
- Linguaggi di Programmazione
- Architettura degli Elaboratori
- Algoritmi e Strutture Dati
- Sistemi Operativi
- …

Perché studiare i sistemi operativi?

- Perché ci aiuta a capire che il SO svolge un ruolo chiave in un sistema di calcolo
  - rende la nostra vita "migliore"
  - realizza "magicamente" ciò che vogliamo
  - ci dà "potere"
- Perché ci aiuta a capire come un sistema di calcolo lavora realmente, e.g.,
  - quante procedure vengono attivate quando digitiamo una sequenza di caratteri sulla tastiera
  - cosa succede quando un nostro programma effettua una divisione per zero

Per avere maggiore coscienza del funzionamento della tecnologia che usiamo quotidianamente …

- Pervasive computers
  - Wearable computers
  - Communication devices
  - Entertainment equipment
  - Computerized vehicle
- OS are specialized
  - Embedded OS
  - Specially configured general-purpose OS





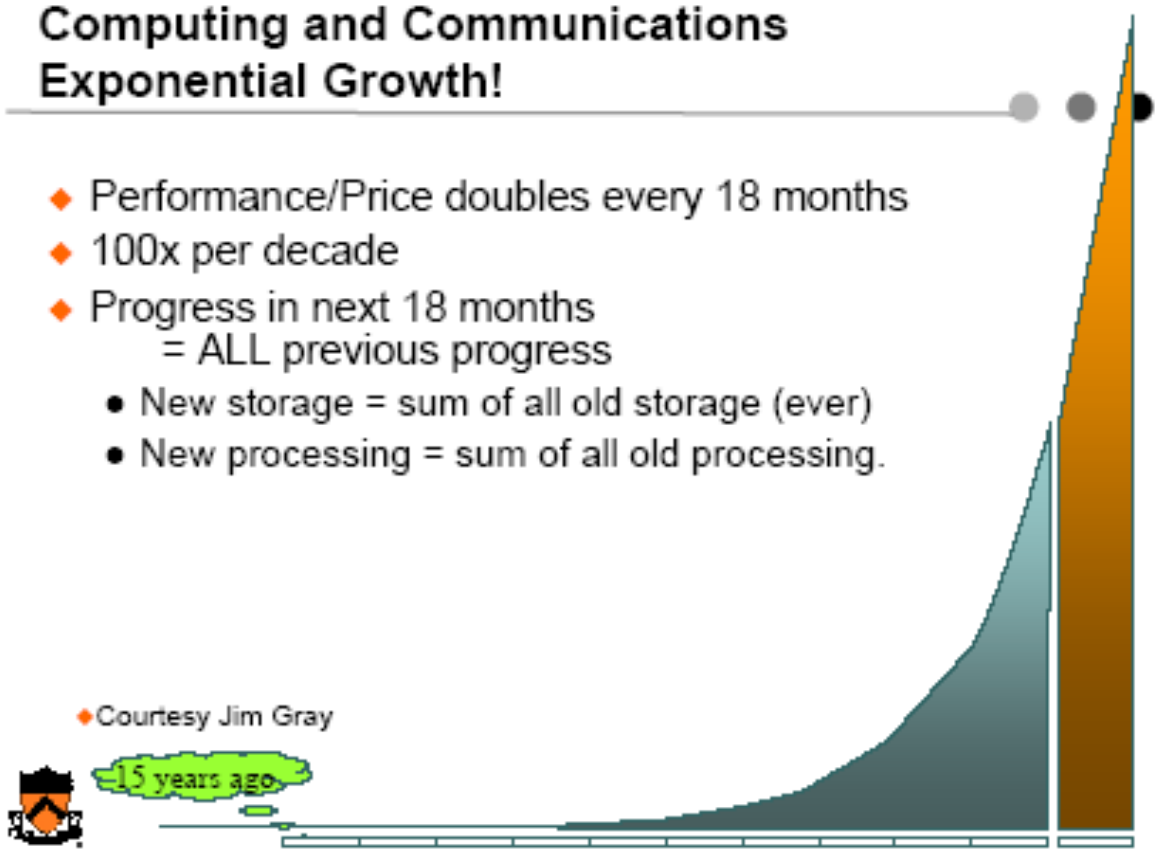


… in un mondo che cambia velocemente

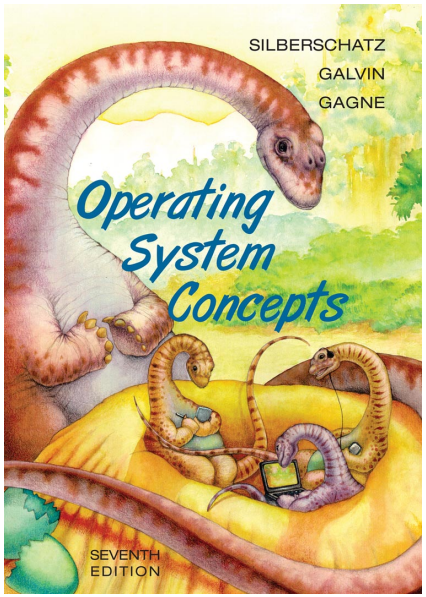
A Typical Academic Computer (1986 vs. 2007)

	1986	2007	Ratio
CPU clock	4Mhz	4×3Ghz	3000x
$/machine	$60k	$600	1/100x
DRAM	1MB	2GB	2000x
Disk	50MB	0.5-1TB	10K-20Kx
Network BW	10Mbits/sec	1GBits/sec	100x
Address bits	32	64	2x
Users/machine	10s	< 1	>10x
$/Performance	$60k	$600/3000	1/200,000x

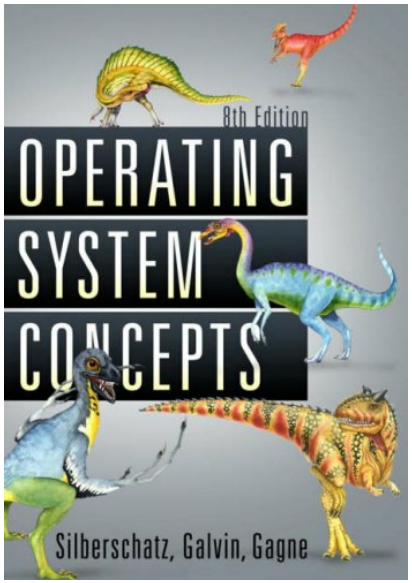
… in un mondo che cambia velocemente



Maggior parte dei contenuti tratti da …







Che cos'è un sistema operativo?

È un **programma** che opera da **intermediario** tra l'utente e l'hardware del computer

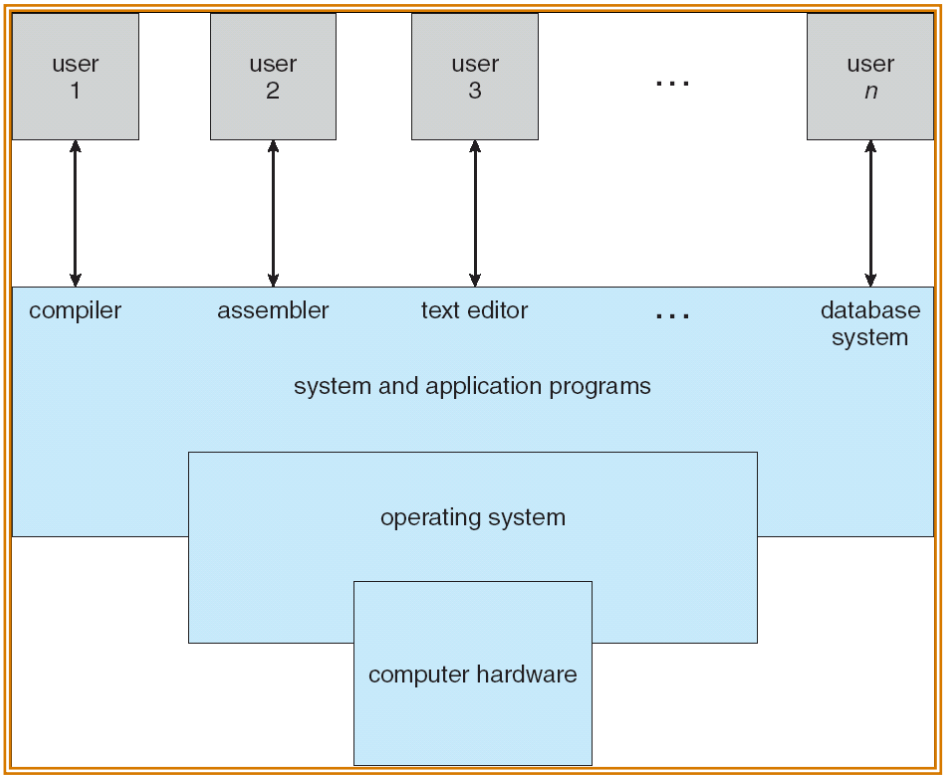


Assicura che il computer operi **correttamente** e che le risorse siano usate **efficientemente**

Esegue i programmi degli utenti e **facilita** i loro compiti offrendo un ambiente d'uso **conveniente**

Elementi 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, dispositi, altri computer



Ruolo del Sistema Operativo



PC. Il sistema operativo è progettato principalmente per facilitare l'uso del computer.



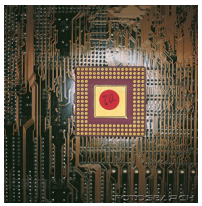
Mainframe e Minicomputer. Occorre massimizzare l'uso delle risorse.



Workstation. Compromesso ottimale tra uso risorse individuali e risorse condivise.



Palmari e simili. Progettati per l'uso individuale con attenzione alle prestazioni 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

- **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)

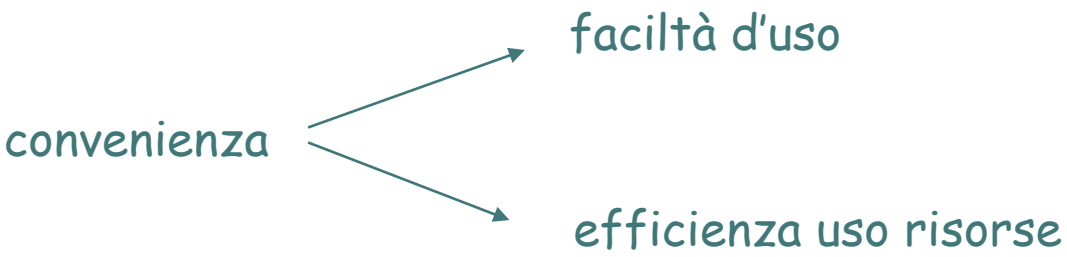
Definizione di Sistema Operativo

- 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" - **kernel del SO**. Tutto il resto è o un programma di sistema o un programma applicativo.

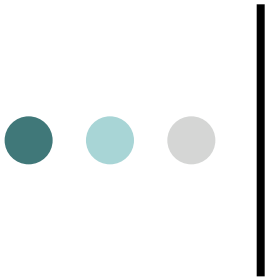
Microsoft
- 1998: Browser Explorer - Dip. della Giustizia USA
- 2007: Windows Media Player - UE

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

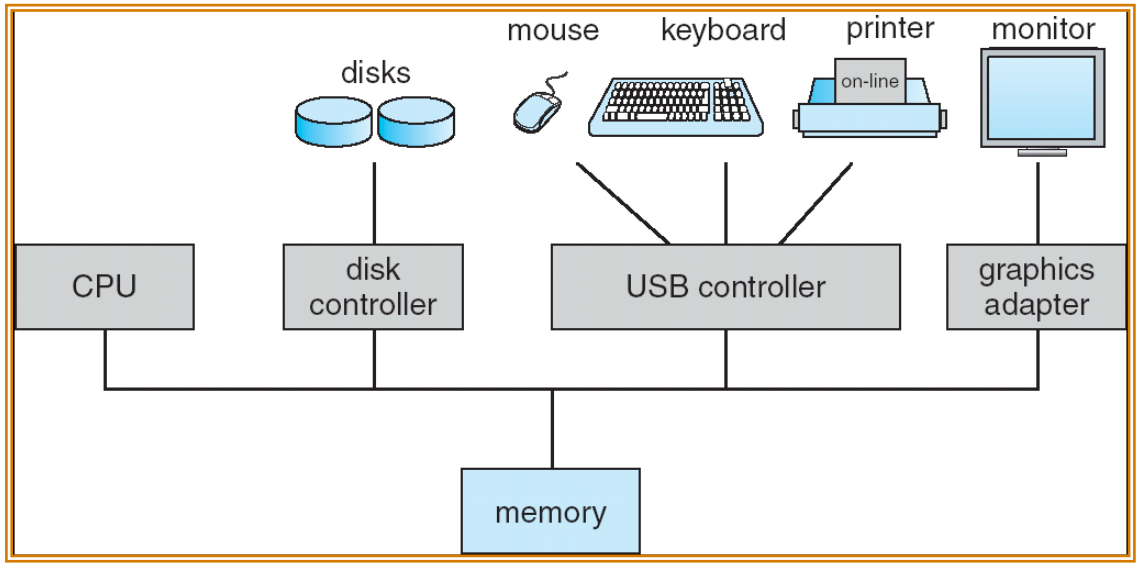


Organizzazione di un elaboratore

Componenti e meccanismi

Struttura di un elaboratore

- 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
- La CPU trasferisce dati dalla/alla memoria in/da buffer locali
- I/O avviene tra i dispositivi e i buffer locali dei controller
- I controller dei dispositivi informano la CPU che hanno finito il proprio lavoro generando un **interrupt**.

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 aspetta che si verifichino **eventi** o richieste degli utenti da eseguire

Interrupt driven

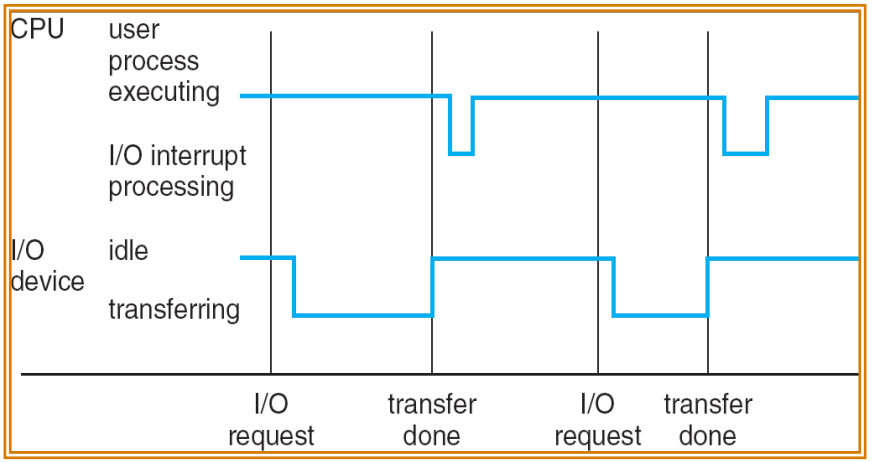
Gli eventi sono segnalati da **interrupt** o da eccezioni (**trap**).

- 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).

Meccanismo degli Interrupt

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.**

Vettore degli eventi del Pentium

- 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.

vector number	description
0	divide error
1	debug exception
2	null interrupt
3	breakpoint
4	INTO-detected overflow
5	bound range exception
6	invalid opcode
7	device not available
8	double fault
9	coprocessor segment overrun (reserved)
10	invalid task state segment
11	segment not present
12	stack fault
13	general protection
14	page fault
15	(Intel reserved, do not use)
16	floating-point error
17	alignment check
18	machine check
19–31	(Intel reserved, do not use)
32–255	maskable interrupts

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.

Memoria Centrale
Random Access Memory



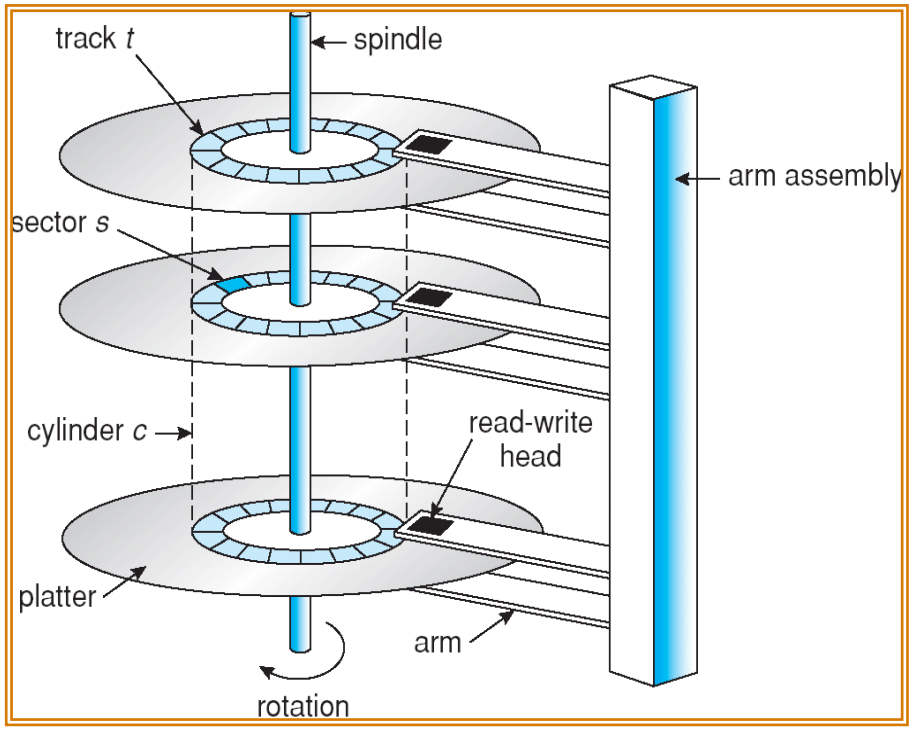
Struttura della memoria secondaria

Memoria secondaria. Disco magnetico

Un disco è composto da piatti

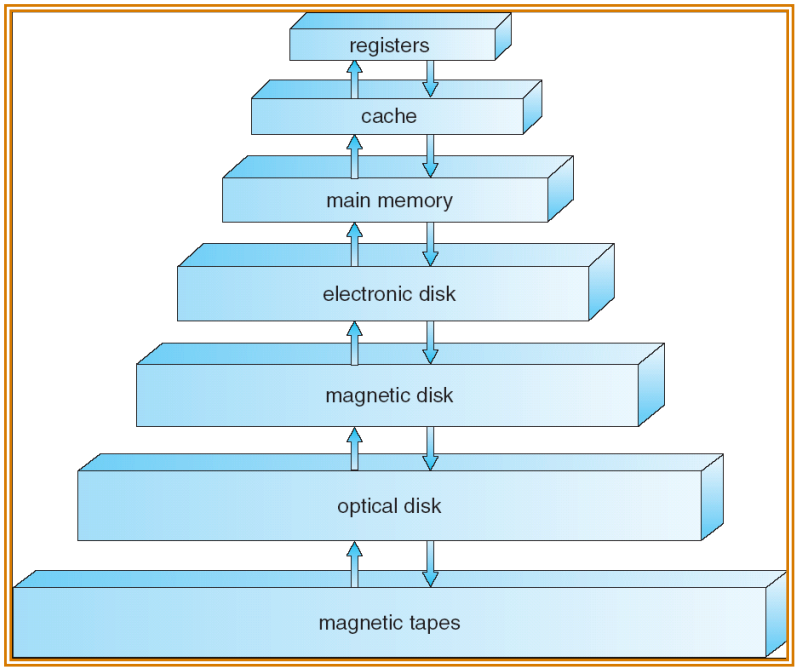
Una testina di lettura - scrittura sfiora ogni piatto

Ogni piatto è diviso in **tracce**. Ogni traccia in **settori**. L'insieme delle tracce che si trova sotto un braccio forma un **cilindro**.



Gerarchia di memorizzazione

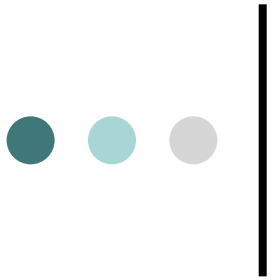
- I sistemi di memorizzazione sono organizzati **gerarchicamente**.
  - Velocità
  - Costo
  - Volatilità



Unità di misura

- **Kb** Kilobit = 1024 bit
- **Mb** Megabit = 1024^2 bit
- **Gb** Gigabit = 1024^3 bit
- **KB** Kilobyte = 1024 byte
- **MB** Megabyte = 1024^2 byte
- **GB** Gigabyte = 1024^3 byte
- **TB** Terabyte = 1024^4 byte

- **ms** millisecondo = 10^{-3} sec
- **µs** microsecondo = 10^{-6} sec
- **ns** nanosecondo = 10^{-9} sec

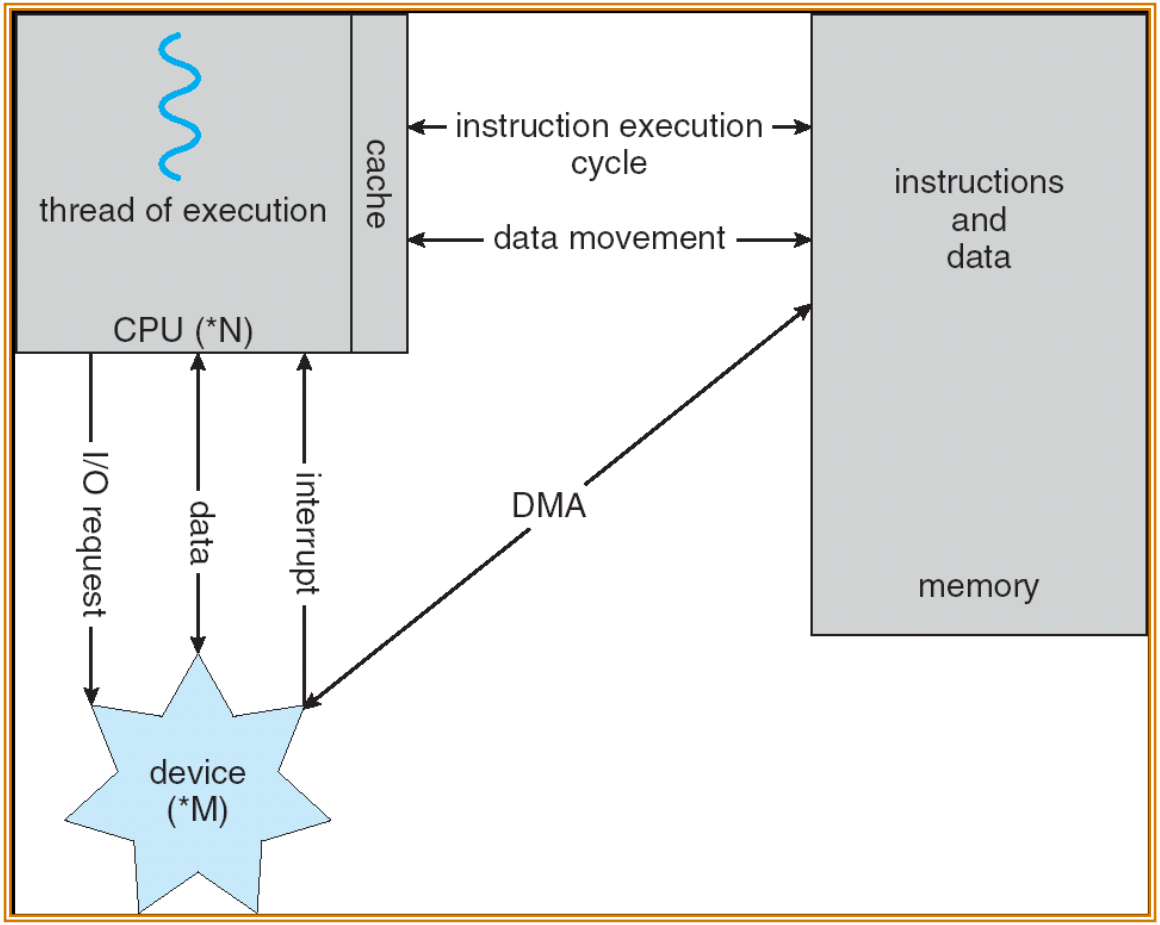


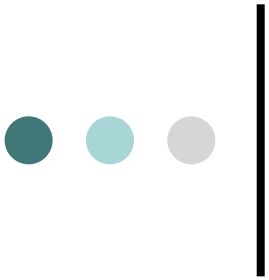
DMA

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.

DMA


thread of execution
CPU (*N)
cache
instruction execution cycle
data movement
instructions and data
memory
I/O request
data
interrupt
DMA
device (*M)



Architetture

Architetture a singolo processore

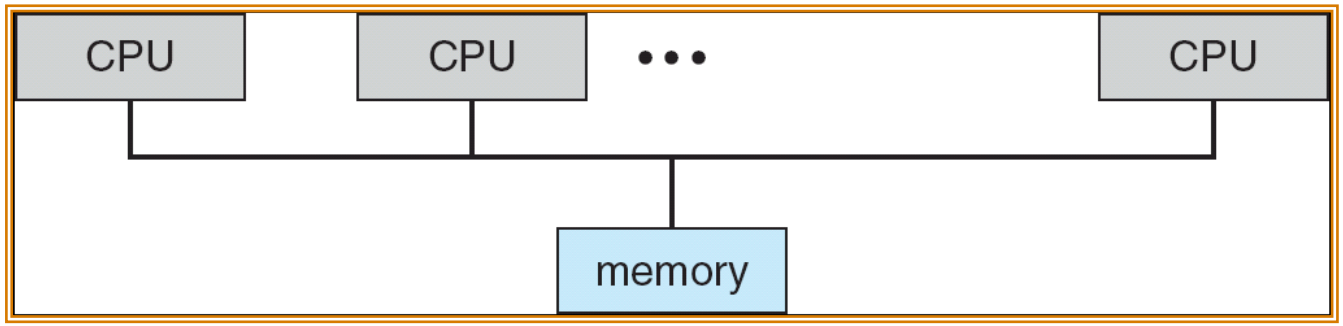
Tali sistemi sono dotati di un singolo processore che esegue un set di istruzioni general-purpose

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).

A volte la CPU principale comunica con questi processori. Altre volte, essi sono totalmente autonomi.



Architetture multiprocessore



Sistemi multiprocessore

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
- Economia di scala
- Aumento affidabilità

Esistono due tipi di sistema multiprocessore:

• **sistema multiprocessore asimmetrico** – un processore principale (master) organizza e gestisce il lavoro per gli altri (slave)

• **sistema multiprocessore simmetrico** (SMP) – ogni processore esegue una copia del sistema operativo e, tali copie, comunicano tra loro

CPU Multi-core e Server Blade

Un microprocessore **multi-core** combina due o più processori indipendenti su un singolo supporto, spesso un singolo circuito integrato



- 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

Trend: Multiple "Cores" per Processor

- Multicore or Manycore transition
  - Intel and AMD have released 4-core CPUs
  - SUN's Niagara processor has 8-cores
  - Azul packed 24-cores onto the same chip
  - Intel has a TFlop-chip with 80 cores
- Accelerated need for software support
  - OS support for manycores
  - Parallel programming of applications



Cluster



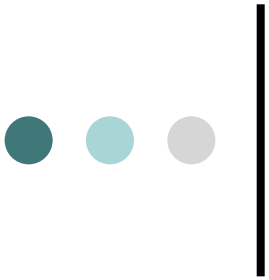
Sistemi Cluster

I sistemi cluster 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.

Cluster simmetrico: le macchine eseguono le applicazioni e si controllano a vicenda

La tecnologia dei cluster sta evolvendo rapidamente (e.g., cluster paralleli) 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.



Struttura di un sistema operativo

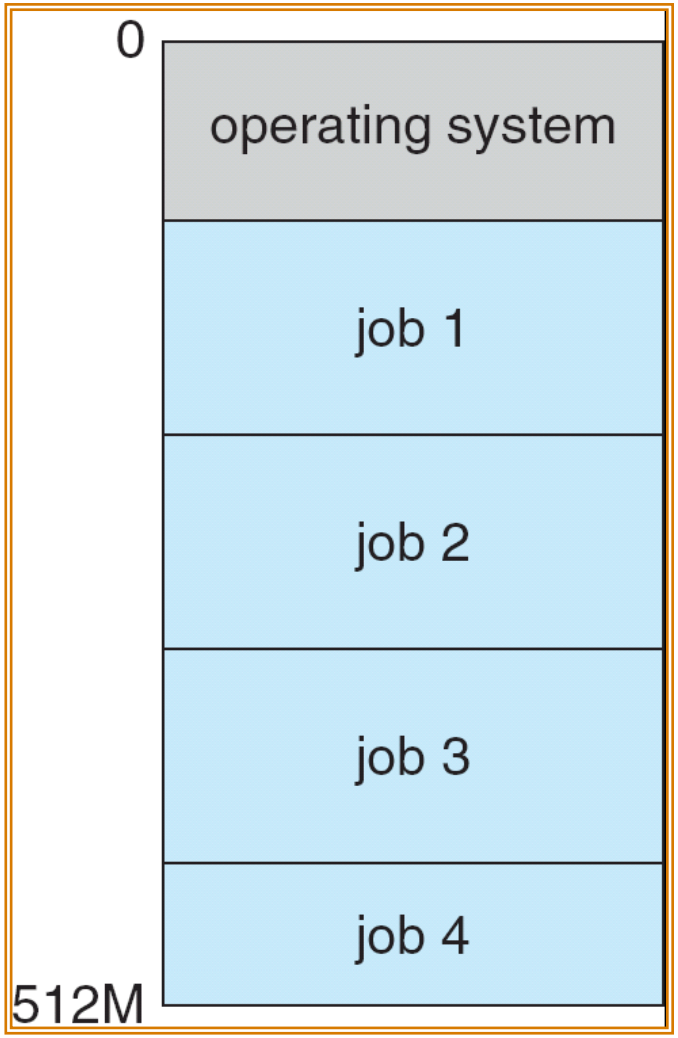
Idee di base e componenti principali

Concetti chiave

Terminologia: job – processo - task

- **Multiprogrammazione** - necessaria per efficienza
  - Un solo job 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

Immagine della memoria in un sistema multiprogrammato



Concetti chiave

- **Timesharing (multitasking)** - estensione logica della multi-programmazione: 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
  - 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

Supporto hardware

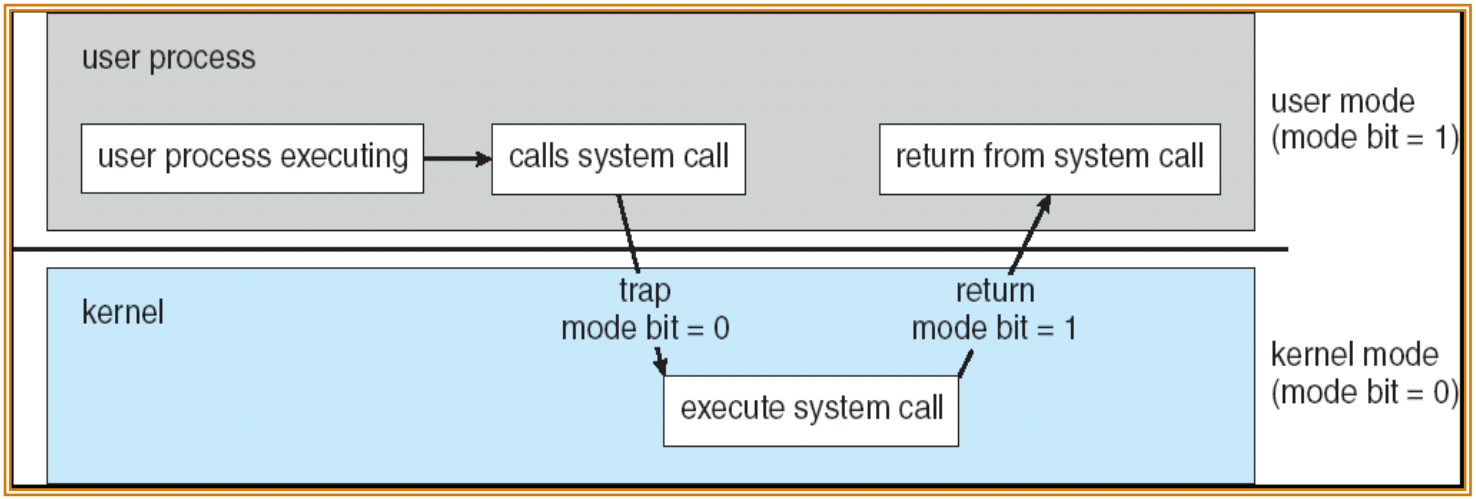
- L'hardware genera interrupt
- Errori software o richieste generano **eccezioni (trap)**
- Alcuni processi potrebbero ciclare all'infinito, o potrebbero tentare di modificare codice e dati di altri processi o del SO

- 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

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



Timer

- 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

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 richiedere 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 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 utilizzo CPU e 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 driver (i.e., disk driver, tape driver)
    - 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
  - 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 volatili

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

- Principio importante, implementato a vari livelli in un computer (in hardware, nel SO, e nel software)
- L'informazione in uso viene copiata da memorie più lente a memorie più veloci temporaneamente
- Si cerca prima all'interno della memoria veloce (cache) per determinare se l'informazione è disponibile
  - Se c'è, l'informazione viene prelevata direttamente dalla cache
  - Altrimenti, viene copiata nella cache e prelevata da lì
- Le cache sono solitamente piccole. Problemi importanti sono
  - La **gestione** della Cache
  - Nel **progetto**, la scelta della dimensione della cache e la politica di sostituzione degli elementi

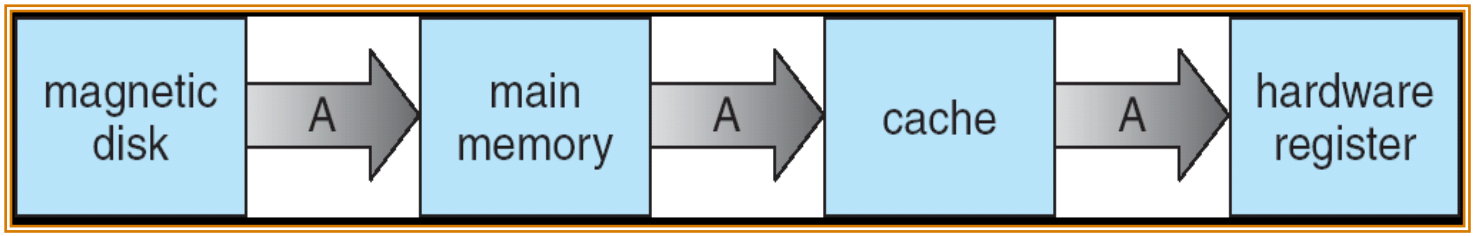
Performance

- Lo spostamento di dati tra i livelli della gerarchia può essere implicito o esplicito

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

Memorie in una workstation

Coerenza/Consistenza



- In ambienti **multitask**, se parecchi processi provano ad accedere all'intero A, occorre assicurarsi che ognuno di essi 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

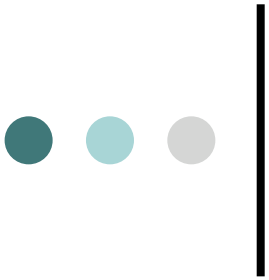
- Uno degli scopi del SO è di nascondere le peculiarità dei dispositivi hardware all'utente

- Il sottosistema di I/O è responsabile di:
  - **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).
  - **Interfaccia generale** per i driver dei dispositivi
  - **Driver** per specifici dispositivi hardware

Protezione e sicurezza

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

- 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



Particolari tipologie di sistemi

Sistemi Distribuiti

- **Un sistema distribuito** è una collezione di processori debolmente accoppiati interconnessi tramite una **rete** di comunicazione

- I processori sono chiamati in vari modi: **nodi, computer, macchine, host**
  - **sito** indica la locazione di un processore

- Vantaggi dei sistemi distribuiti sono
  - Condivisione delle risorse
    - Accesso a file memorizzati in siti remoti
    - Elaborazione informazioni presenti in database distribuiti
    - Uso di hardware specializzato di un sito remoto
  - Speedup delle computazioni
  - Condivisione del carico (**load sharing**)
  - Affidabilità
  - Comunicazioni

Sistemi Distribuiti

Una rete è una struttura di comunicazione tra due o più sistemi. I sistemi distribuiti dipendono dalla connessione di rete. Il protocollo TCP/IP è il protocollo di rete più usato per le comunicazioni.

Le reti di computer vengono classificate in base alla **distanza** tra i rispettivi nodi:

LAN (Local Area Network) – dipartimento, casa, palazzo

WAN (Wide Area Network) – città o nazioni

MAN (Metropolitan Area Network) – quartiere all'interno di una città

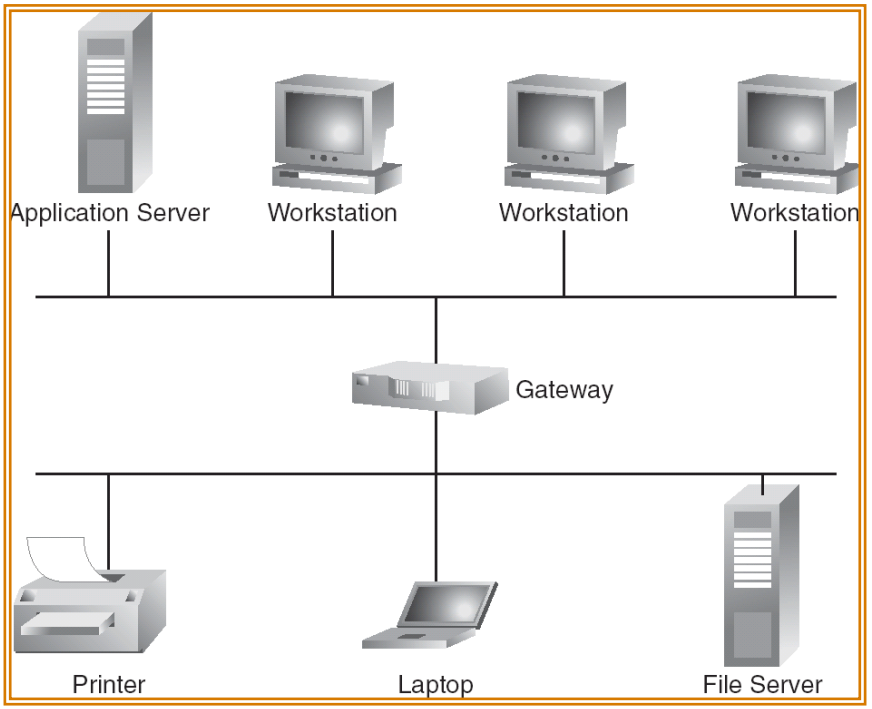
Struttura di rete

Reti locali (LAN)

Le velocità di comunicazione variano da 1 Mb a 10 Gb

Comuni sono:

10BaseT Ethernet	10Mb
100BaseT Ethernet	100Mb



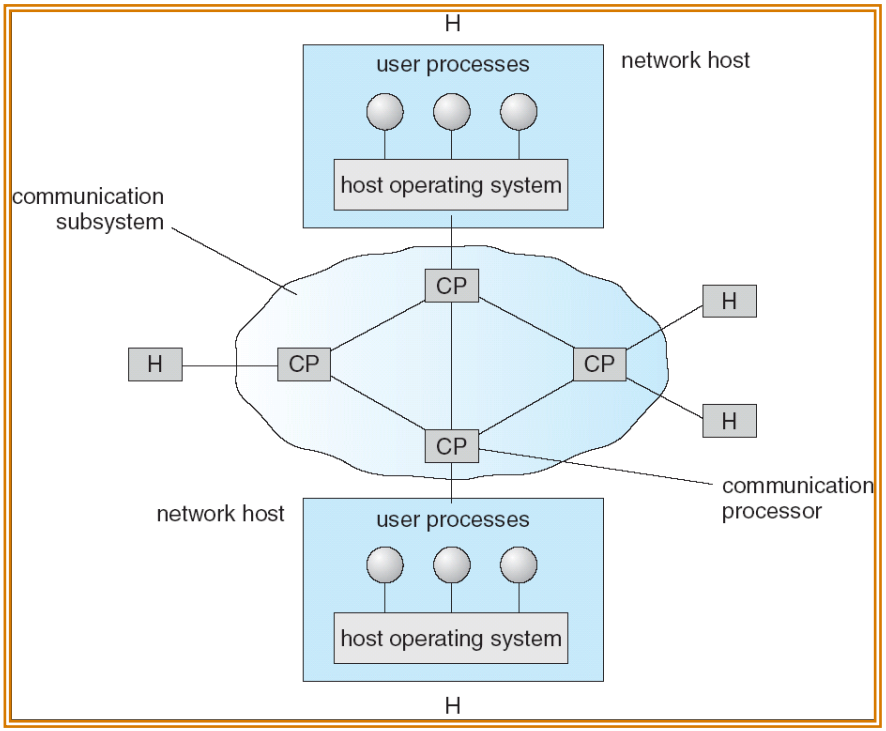
Struttura di rete

Reti geografiche (WAN)

Le velocità di comunicazione variano da 56Kb a diversi Mb

Le connessioni sono controllate da CP, communication processor

Internet: router, linee T1. Linee telefoniche, modem, ADSL.



Sistemi operativi di rete

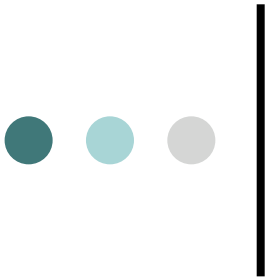
- Esistono due tipi di sistema: sistemi operativi di rete e sistemi operativi distribuiti

- In un sistema operativo di rete gli utenti **sono a conoscenza** della presenza di altre macchine. Pertanto, l'accesso alle risorse delle altre macchine viene fatto esplicitamente. Esempi sono:

  - Logging remoto (telnet, ssh)
  - Desktop Remoto (Microsoft Windows)
  - Trasferimento di file da macchine remote alla macchina locale (ftp)

Sistemi operativi distribuiti

- Gli utenti **non si accorgono** della presenza di più macchine
  - L'accesso alle risorse remote è simile all'accesso alle risorse locali
- **Migrazione dei dati** – trasferimento dei dati attraverso il trasferimento dell'intero file o soltanto delle porzioni necessarie
- **Migrazione delle computazioni** – trasferimento delle computazioni, piuttosto che dei dati, attraverso il sistema



Sistemi real time

Un sistema in tempo reale deve soddisfare **vincoli di tempo** ben definiti.

Hard real-time: è indispensabile completare una computazione o un trasferimento dati entro un tempo rigidamente prefissato.

Spesso sono impiegati per supportare il controllo in un'applicazione dedicata o in un sistema dedicato. Molte delle caratteristiche dei SO per sistemi general-purpose non sono presenti nei real-time.

Soft real-time: i vincoli non sono stringenti. In questi sistemi, compiti critici hanno priorità più elevata rispetto ad altri processi (e.g., applicazioni multimediali).

Sistemi real time

Sistemi embedded. Si trovano ovunque: nei sistemi di automazione industriale, nei VCR, nei forni a microonde

"L'accesso al web permetterà al proprietario di casa di impartire l'ordine di accendere il riscaldamento prima di arrivare a casa. Il frigorifero potrà chiamare direttamente il droghiere se si accorge che manca il latte."

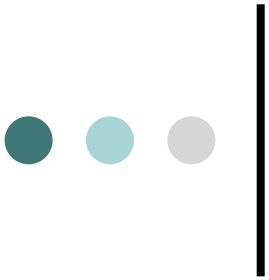
Elaborazione multimediale. I recenti sviluppi tecnologici tendono a introdurre dati multimediali in numerose applicazioni informatiche e di intrattenimento (e.g., file audio in formato MP3, filmati, videoconferenza, streaming)

Sistemi palmari

I sistemi palmari comprendono i **personal digital assistant (PDA)** quali Palm, Pocket-PC, e telefoni cellulari.

Problema fondamentale nel disegno di un SO: **dimensioni e caratteristiche limitate** dei dispositivi (e.g., memorie da 128KB a 512KB, processori lenti, schermo di 20 cm quadrati, batterie piccole)





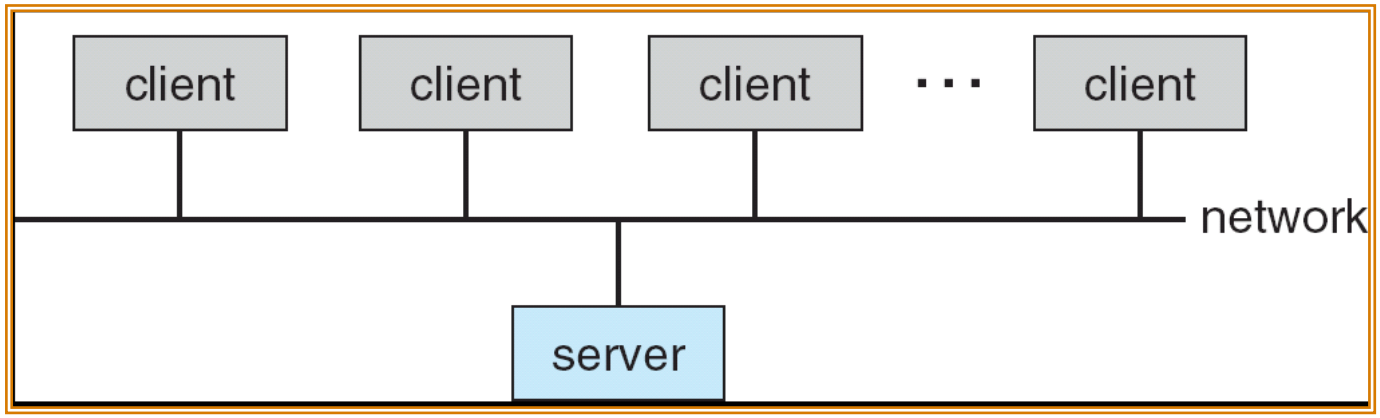
Ambienti di Elaborazione

Ambienti di Elaborazione

- **Computazione tradizionale**
  - Ufficio
    - PC connessi alla rete, terminali attaccati a mainframe o minicomputer che fornivano esecuzione batch e timesharing
    - Oggi ci sono portali che rendono possibile accessi ai sistemi della stessa rete e a sistemi remoti
  - Home network
    - Erano sistemi singoli, usavano i modem per la connessione
    - Oggi sono protetti da firewall e sono connessi in rete tramite linee veloci

Ambienti di Elaborazione

- **Computazioni Client-Server**
  - Sistemi **server** rispondono alle richieste generate da **client**
    - **Computing-server**: fornisce un'interfaccia alla quale un client può inviare richiesta per un servizio (i.e. database)
    - **File-server**: fornisce un'interfaccia ai client per memorizzare o recuperare file



Ambienti di Elaborazione

- Sistemi **Peer-to-Peer** (P2P, in breve)
- **P2P non distingue server e client**
  - Al contrario, ogni nodo può agire come server, client o sia come server che come client
  - I nodi debbono far parte della rete P2P
    - Ogni nodo si registra presso un registro centrale della rete, oppure
    - Invia in broadcast richieste di servizio e risponde alle richieste che riceve attraverso un protocollo di scoperta
  - Esempi: **Napster, Gnutella, …**







Ambienti di Elaborazione

- Il Web è diventato onnipresente
- I PC sono i dispositivi d'accesso prevalente
- Molti dispositivi che non erano precedentemente in rete ora offrono accessi mediante filo o wireless
- Nuovi dispositivi simili a server, quali i **load balancer**, per gestire il traffico web, hanno preso piede
- L'uso di sistemi operativi come Windows 95, client-side, è stato soppiantato dall'uso di Linux e Windows XP, Vista, Windows 7, che possono funzionare sia come client che come server

Sistemi Operativi Open-Source

- Sistemi operativi in codice sorgente piuttosto che in formato binario (closed-source)
- In opposizione al copy protection e al movimento per il Digital Rights Management (DRM)
- Nasce con la Free Software Foundation (FSF), che ha introdotto la “copyleft” GNU Public License (GPL)
- Esempi includono GNU/Linux, BSD UNIX (incluso il kernel Mac OS X) e Sun Solaris