



SAPIENZA
UNIVERSITÀ DI ROMA

**Corso di laurea in Ingegneria
dell'Informazione
Indirizzo Informatica**

Reti e sistemi operativi

Introduzione ai sistemi operativi

Contatti

- Docente: Alberto Pretto
 - pretto@dis.uniroma1.it
- Ricevimento
 - Appuntamento tramite e-mail
- Slide
 - Pubblicate sulla pagina moodle del corso

Programma di massima

- Cos'è un sistema operativo. Le sue funzioni. La sua storia.
- Interrupt e chiamate di sistema
- Processi e thread
- Algoritmi di scheduling
- Programmazione concorrente: sincronizzazione dei processi e deadlock
- Gestione della memoria principale: memoria virtuale e paginazione
- Gestione del file system
- Casi di studio basati su S.O. Linux: la shell ed il filesystem, i comandi Unix fondamentali, il multithreading, ...

Testi di riferimento

- Silberschatz, Galvin, Gagne: Sistemi Operativi. Concetti ed Esempi. Pearson Ottava edizione (2009)
- Slide e dispense del corso.
- Per consultazione:
 - Andrew S. Tanenbaum: I moderni sistemi operativi. Pearson Ottava edizione (2009) 3/Ed.

Valutazione

- **Esame scritto:**
 - Prova scritta parte di reti
 - Prova scritta parte di sistemi operativi

Introduzione ai sistemi operativi, cenni storici.

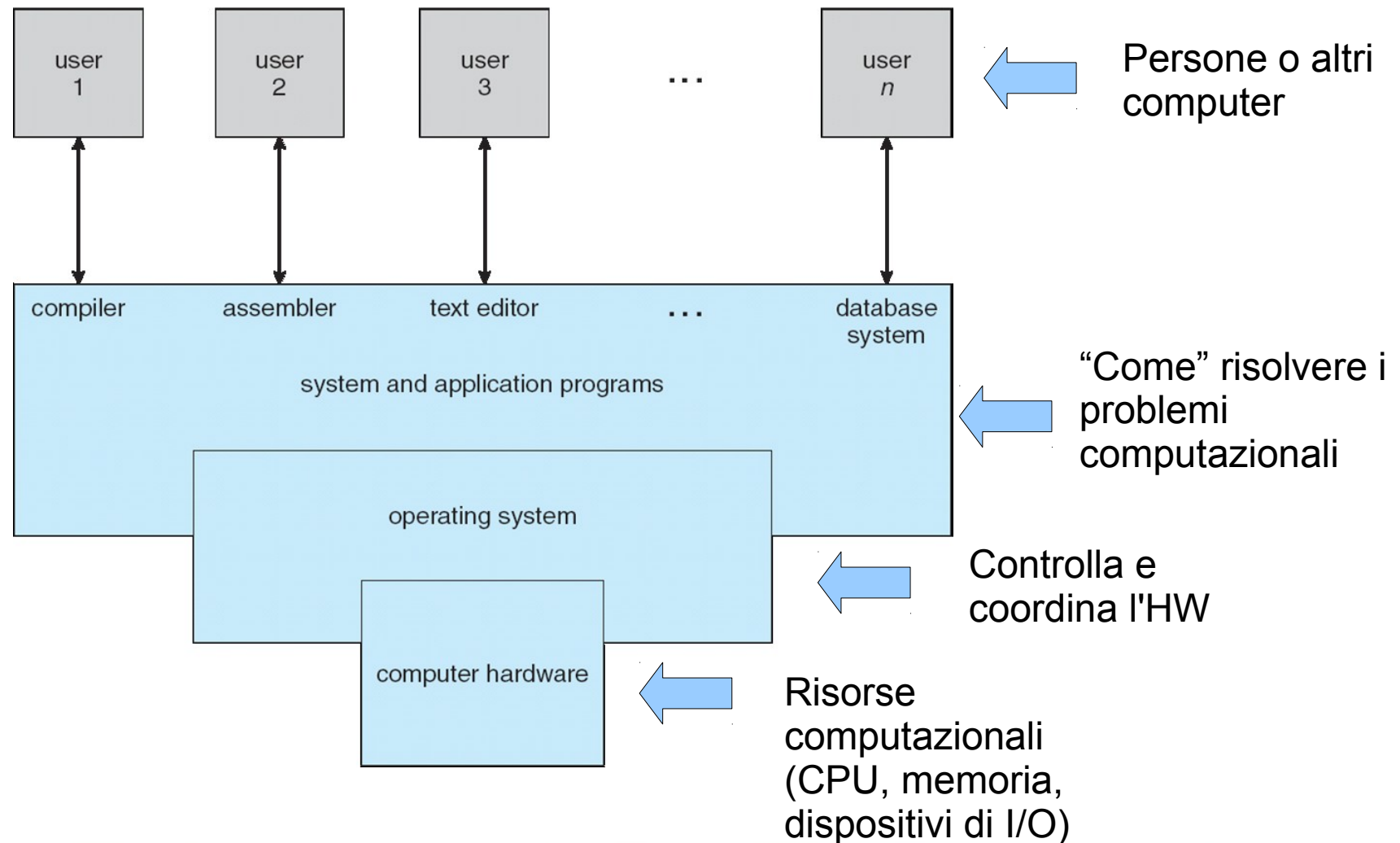
Struttura e funzioni di un SO

- Un sistema operativo è un **programma** che controlla l'esecuzione dei programmi applicativi e funge da interfaccia tra l'utente di un computer e l'**hardware**.
- Un sistema operativo:
 - **Alloca risorse**, evitando conflitti tra i vari processi che le utilizzano
 - **Controlla il flusso di esecuzione** dei programmi, gestendo le situazioni di errore

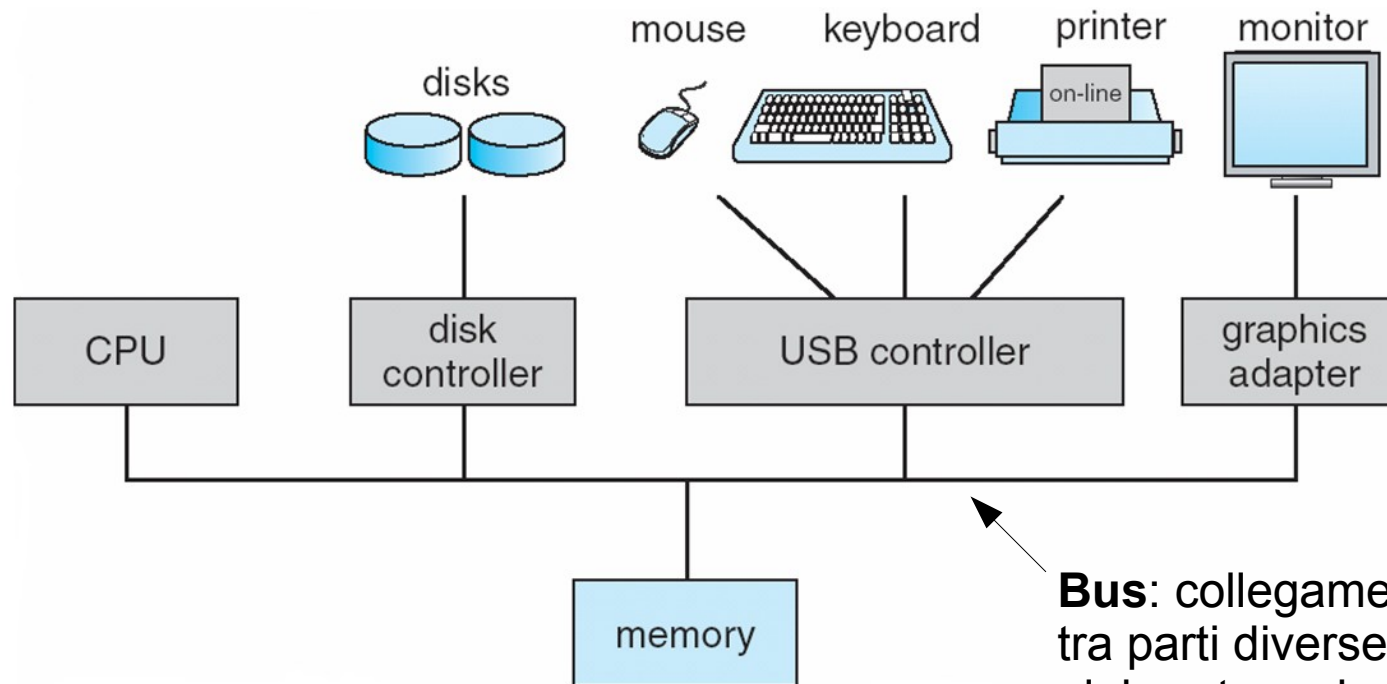
Obiettivi di un S.O.

- Eseguire in maniera opportuna i programmi
- **Facilitare l'uso del computer**
- **Far sì che le risorse del computer siano usate in modo efficiente**

Le quattro componenti di un sistema di elaborazione



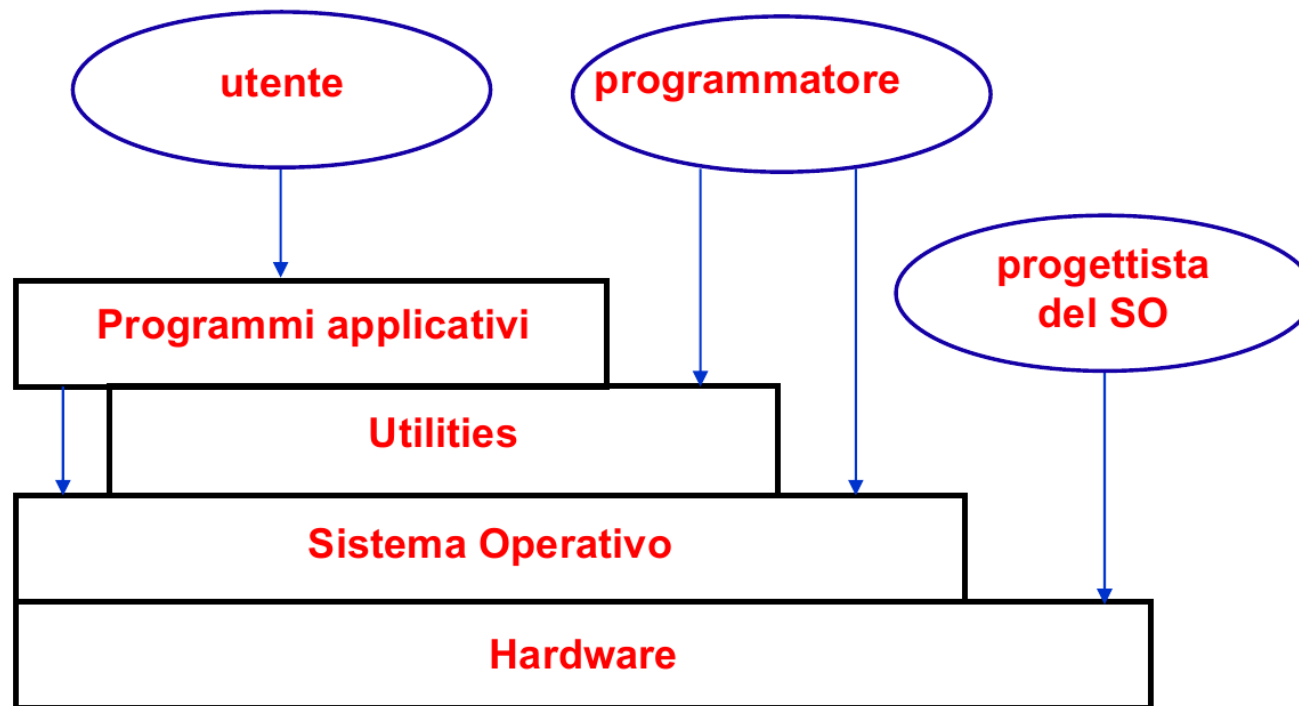
Organizzazione di base di un sistema di elaborazione



Bus: collegamento elettrico tra parti diverse di un elaboratore che consente il trasferimento di informazione. Un bus è composto da una o più linee che consentono il trasferimento di più bit in parallelo (ogni linea consente₁₀ di trasferire un bit)..

Facilitare l'uso del computer

- il SO fornisce un'interfaccia tra:
 - Computer e utente
 - Computer e programmatore



Servizi forniti da un SO

- Per facilitare l'uso del computer il SO fornisce diversi servizi:
 - **Esecuzione di programmi**
 - **Accesso ai dispositivi di I/O**
 - **Accesso protetto** alle risorse condivise.
 - **Gestione degli errori** e dei malfunzionamenti.
 - **Sviluppo di software** - editor, compilatori, debugger, etc.
 - **Gestione di statistiche** e raccolta dati per monitorare le prestazioni.

Usare le risorse in modo efficiente

- Le risorse che il SO cerca di usare in modo efficiente sono:
 - Il processore
 - Multitasking → più programmi in esecuzione
 - Scheduler → pianifica e determina le esecuzioni
 - La memoria
 - Paging → suddivisione della memoria in porzioni di dimensioni fisse
 - Memoria virtuale → facilita la gestione della memoria per i programmi e simula una quantità maggiore di RAM
 - I dispositivi di I/O
 - Gestione dei file → organizzazione, diritti di accesso
 - I/O buffering → ottimizzare le operazioni di I/O
 - Assegnazione dei dispositivi → accesso **concorrente**

Evoluzione dei SO

- **Elaborazione seriale (1945-1955)**
 - Un programma alla volta, niente SO, dispositivi rudimentali, codice binario
- **Elaborazione batch (1955-1965)**
 - Un programma di seguito all'altro, monitor, JCL (Job Control Language), protezione della memoria
- **Sistemi batch multiprogrammati (1965-1980)**
 - Più di un programma in memoria, scheduler, maggiore efficienza
 - Hardware più sofisticato (interrupt, DMA, MMU)
 - Gestione dei **processi**
- **Time-sharing**
 - Job **interattivi**, esecuzione a turno dei job, virtualizzazione del processore

Evoluzione dei SO (1945-1955)

- **Elaborazione seriale (1945-1955)**
 - Un programma alla volta
 - L'utente andava direttamente sulla macchina a caricare il suo software
 - Niente SO
 - Interruttori, dispositivi di input rudimentali (lettori di schede perforate, lettori di banda perforata, ...)
 - Programmi scritti in codice binario
 - Errori indicati da lampadine
 - Completamento normale → output su stampante
 - Prenotazione su fogli cartacei
 - Tempi di preparazione molto lunghi (caricamento del programma, montaggio e smontaggio di nastri magnetici, banda perforata, pacchi di schede perforate, ...).

Evoluzione dei SO (1955-1965)

- **Elaborazione batch (1955-1965)**
 - Un programma di seguito all'altro, senza soste.
 - Il **batch monitor** (programma residente in memoria) carica ed esegue uno alla volta i programmi (con i dati relativi), leggendoli da un dispositivo di input (lettore di schede o unità nastro), stampando di volta in volta i risultati.
 - L'utente non accede alla macchina, ma consegna il pacco di schede (programma e dati) ad un operatore che li inserisce, uno di seguito all'altro, nel dispositivo di input.
 - Schede di controllo contenenti istruzioni per il monitor (**JCL - Job Control Language**) sono inserite prima del programma (job), tra programma e dati e alla fine dei dati.

Evoluzione dei SO (1965-1980)

- **Sistemi batch multiprogrammati**
 - Più di un programma in memoria (**multitasking**)
 - Quando il programma in esecuzione chiede al SO di eseguire un'operazione di I/O (lenta), il SO avvia l'operazione e, nell'attesa che sia completata, cede il controllo ad un altro programma (che sia pronto).
 - Il SO comprende uno **scheduler**.
 - Il funzionamento è ancora di tipo batch (l'utente consegna il pacco di schede e ripassa più tardi a ritirare le sue schede e il pacco con le stampe).
 - La produttività è più alta, perché il processore è usato in modo più efficiente.

Sistemi batch multiprogrammati

- Caratteristiche aggiuntive che devono essere presenti nell'hardware:
 - Gestione dell'I/O tramite **interruzioni**
 - Gestione dell'I/O tramite **DMA**
 - Gestione della memoria (**MMU**)
 - Lo scheduler può gestire:
 - Programmi indipendenti
 - Attività (task) di un medesimo programma che possono procedere in parallelo (**multitasking**)
 - Lo scheduler di un SO multitasking gestisce:
 - **Processi** (programmi indipendenti o task cooperanti).
- Processo** → entità attiva che rappresenta l'esecuzione del programma.

Sistemi time-sharing

- Le modalità di funzionamento non sono più di tipo “batch”
- Gli utenti usano un terminale per accedere al sistema in cui vengono eseguiti **job interattivi**
- Adatti ad applicazioni con transazioni interattive
- il SO, tramite lo scheduler, esegue a turno ciascun programma per un breve quanto temporale (time-slicing):
 - Sfrutta la lentezza dei tempi di reazione dell'utente
 - I tempi di risposta percepiti dagli utenti sono paragonabili a quelli di un sistema dedicato

Obiettivi dei SO time-sharing

- Un SO time-sharing è un SO multiprogrammato con obiettivi diversi:
 - SO multiprogrammato:
 - rendere massima l'**utilizzo del processore**
 - SO time-sharing:
 - Rendere **minimi i tempi di risposta**.
- e con diverse modalità di interazione:
 - SO multiprogrammato → tramite comandi **JCL** forniti con il job
 - SO time-sharing → tramite **terminale**

Evoluzione dei SO (1980 - 1995)

- **Personal Computer**
 - Inizialmente SO monoutente e monotask (MS-DOS) per processori di potenza limitata (Intel 8086, 80186, 80286)
- Successivamente SO multitasking e multiutente (Windows, Unix, Macintosh) con processori più potenti (Intel 80386, Intel 80486, Intel Pentium, AMD 64 bit, Intel Core iX, ...).

SO per personal computer

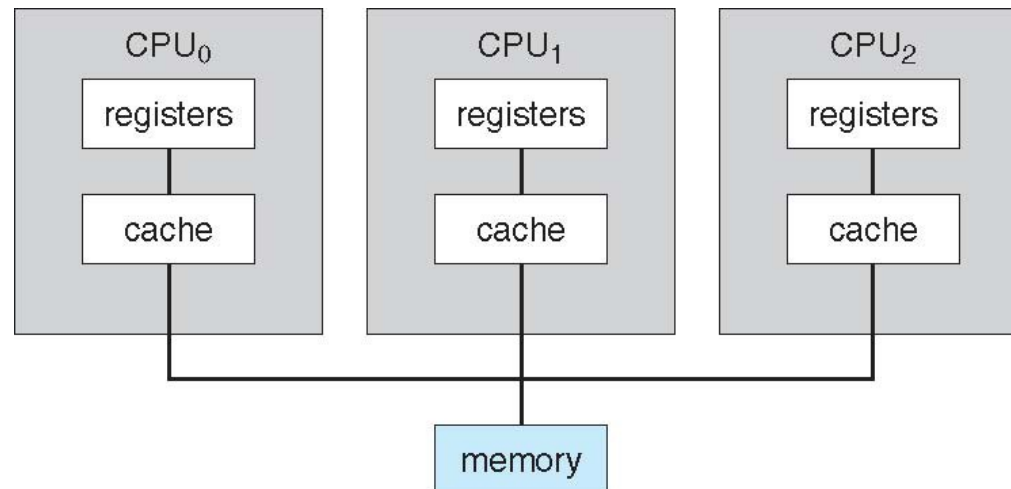
- **Obiettivi diversi dai SO per mainframe:**
 - SO per mainframe: massimizzare l'uso delle risorse (processore e dispositivi)
 - SO per PC: massimizzare la facilità d'uso e la prontezza di risposta
- **Caratteristiche comuni ai SO per mainframe e per PC:**
 - Scheduler
 - Protezione della memoria
 - Gestione della concorrenza
- **Altre caratteristiche:**
 - Protezione dei file
 - Protezione dai virus

Evoluzione dei SO (1995 - ...)

- Sistemi paralleli → aumentare le prestazioni
 - Sistemi multiprocessore (simmetrici, asimmetrici)
 - Sistemi distribuiti (cluster)
- Sistemi real-time → fornire il risultato corretto entro un tempo prefissato
 - Controllo di processo
 - Sistemi embedded
 - Hard e soft real-time

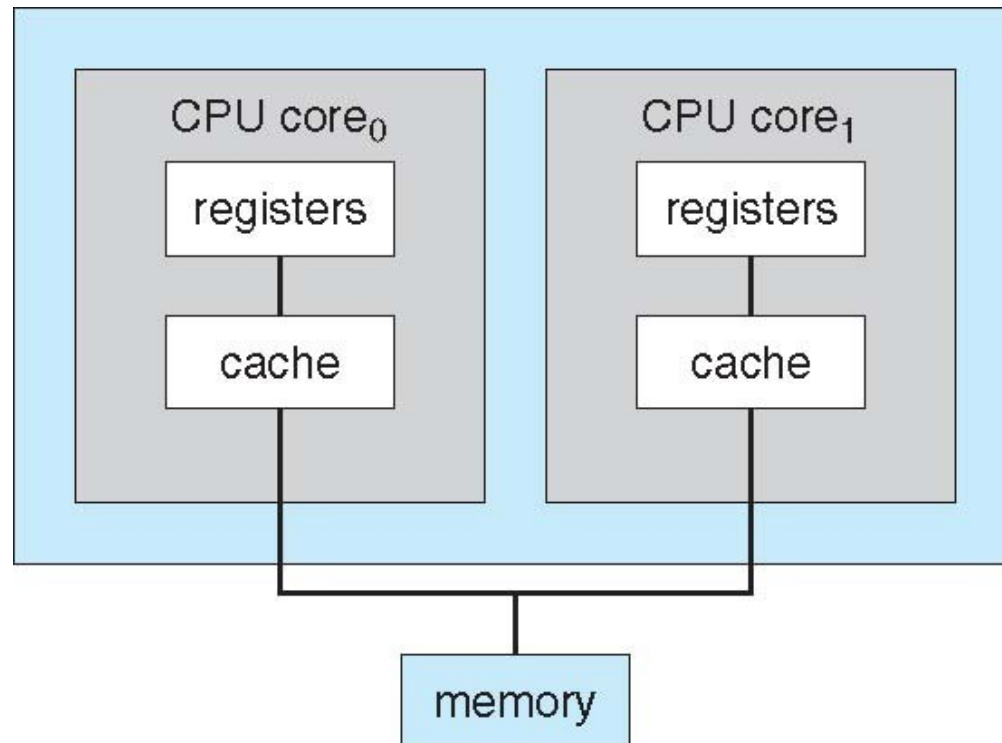
Sistemi multiprocessore

- Vantaggi:
 - Prestazioni migliori
 - Scalabilità
 - Affidabilità maggiore (**graceful degradation** or **fault tolerance**)



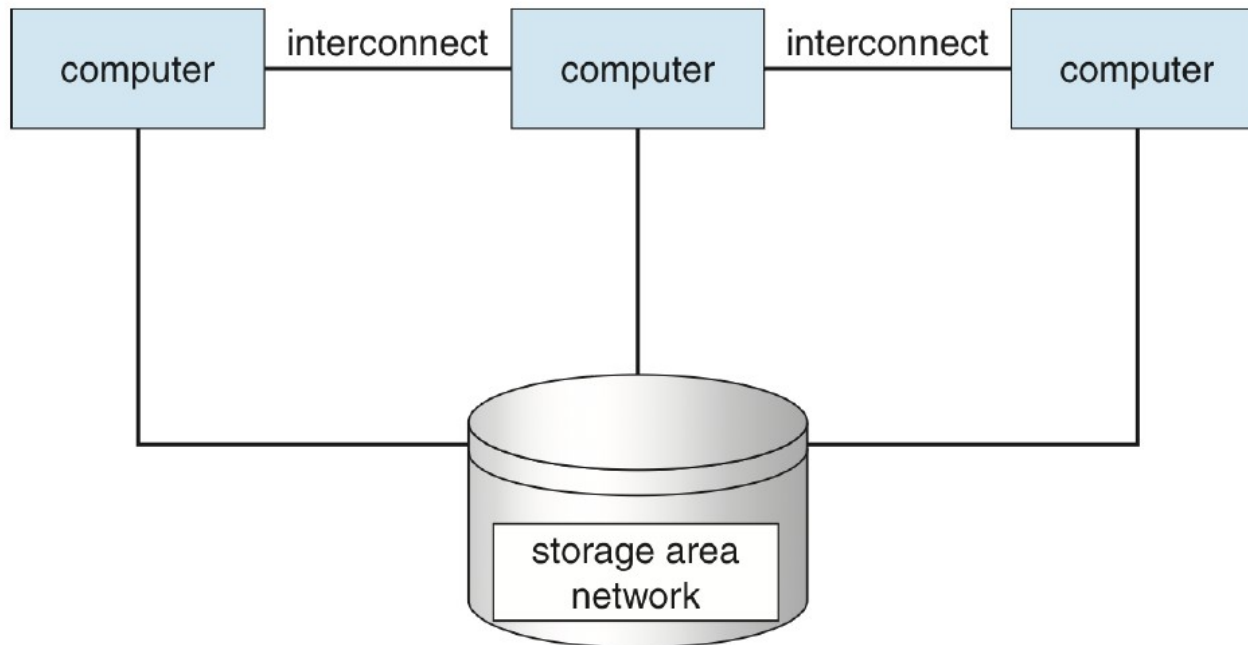
Sistemi multicore

- Più CPU nello stesso package (chip)



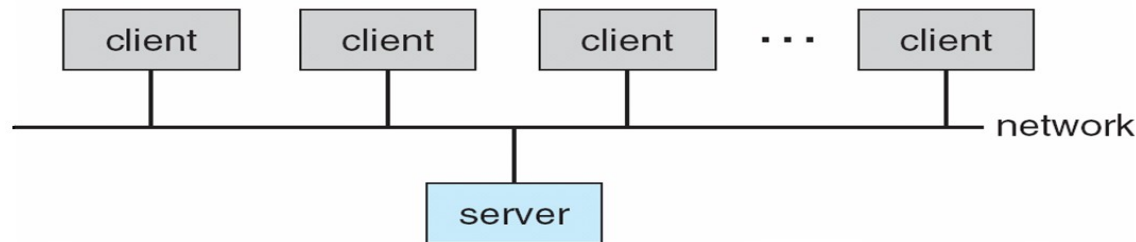
Sistemi cluster

- Più computer interconnessi fra di loro (es. attraverso interfaccia Ethernet)



Client-Server e P2P Computing

- Client-server: ruoli ben definiti, un server, molti client, es.:
 - Compute server (database)
 - File server



- Peer-to-Peer: non vi è distinzione tra i computer connessi
 - EMule

Componenti di un SO

- Gestore dei processi
- Gestore della memoria
- Gestore dei sistemi di I/O
- Gestore della memoria secondaria
- Gestore dei file
- Gestore delle comunicazioni di rete
- Gestore della sicurezza
- Interprete dei comandi

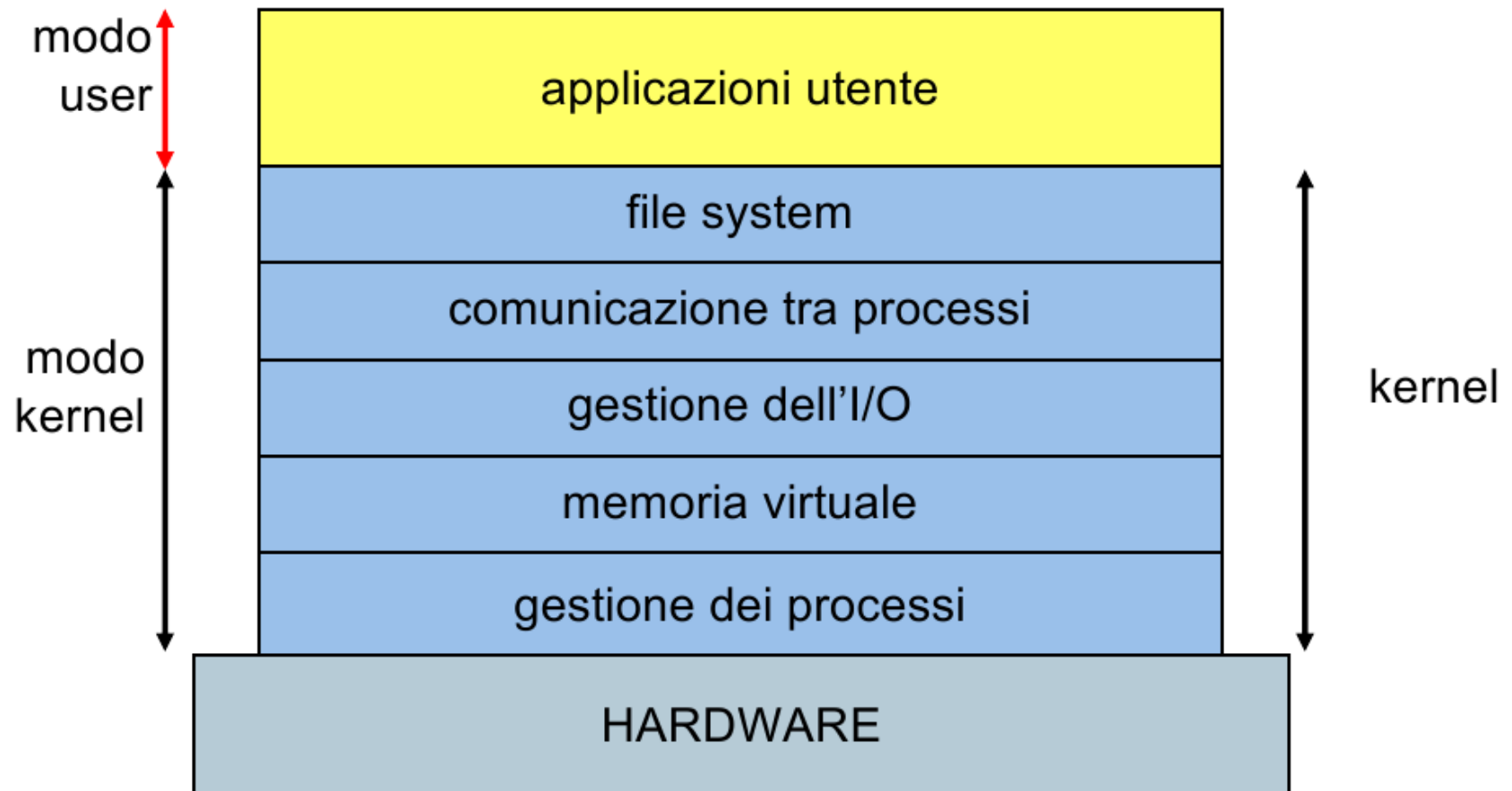
Servizi di un SO

- Richiesti al SO da:
 - **Utenti**, tramite l'interprete dei comandi (CLI o GUI)
 - **Programmi**, tramite **system call**
- Esecuzione di programmi
- Operazioni di I/O
- Accesso al file system
- Comunicazioni tra processi (shared memory, pipes, ...)
- Gestione delle risorse
- Protezione interna e sicurezza verso l'esterno
- Gestione delle condizioni anomale
- Gestione delle statistiche

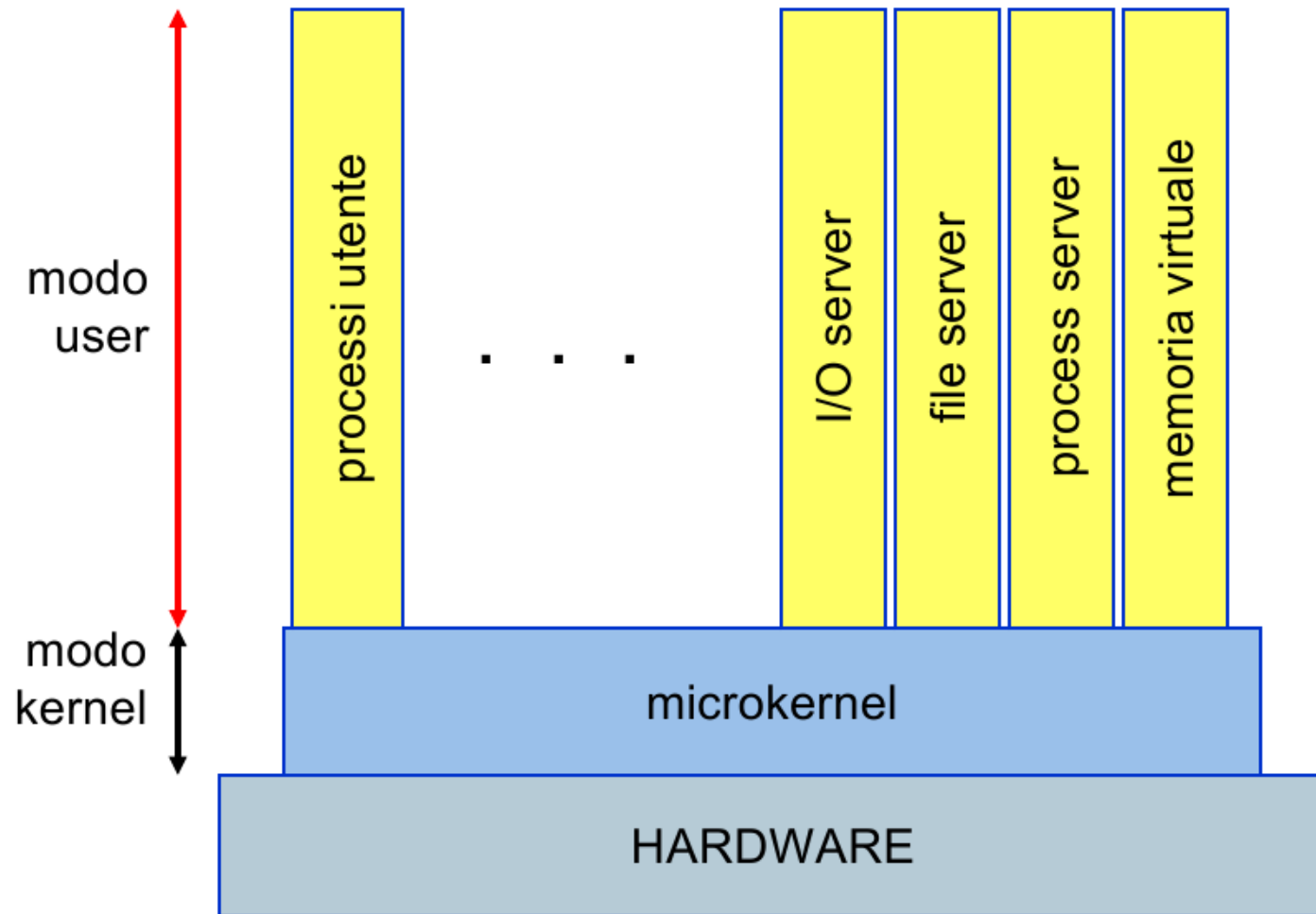
Strutture dei SO

- A strati o livelli (layered):
 - Ciascun livello fornisce funzionalità al livello superiore, usando (e nascondendo) quelle del livello inferiore
 - Il livello più basso del SO è il kernel (nucleo)
 - Non è facile definire l'ordinamento delle varie funzioni
 - Può introdurre inefficienza
- Microkernel:
 - Solo le funzioni essenziali fornite dal kernel
 - Le altre funzioni sono affidate a processi (eseguiti in modo supervisore o in modo utente) trattati dal microkernel alla stregua degli altri processi

SO con architettura a strati



SO con architettura microkernel



Architettura di un SO UNIX

