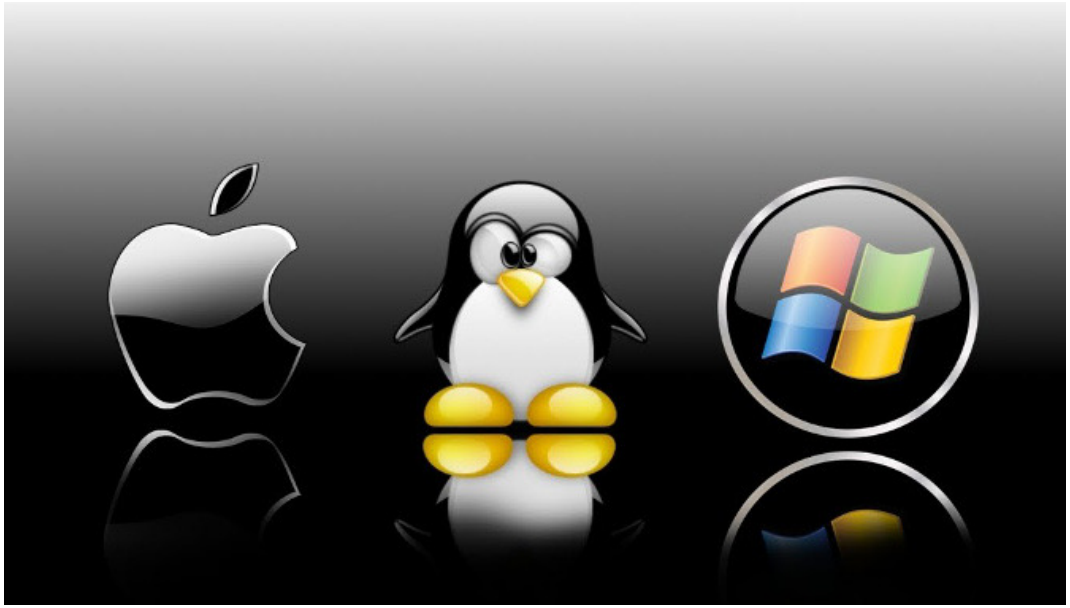
SISTEMI OPERATIVI (M-Z)



C.d.L. in Informatica

(laurea triennale)

AA 2022-2023

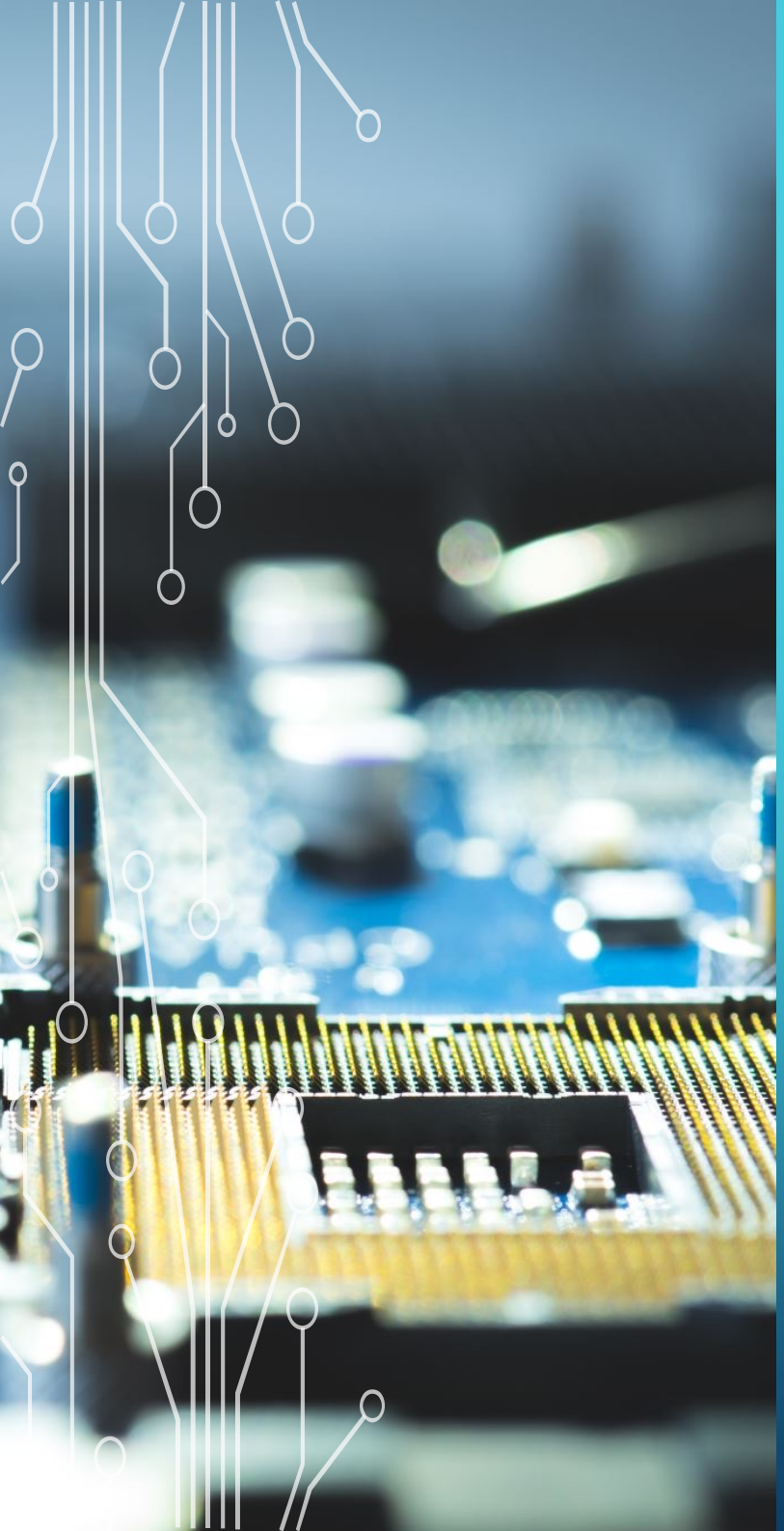
Prof. Mario F. Pavone

Dipartimento di Matematica & Informatica

Università degli Studi di Catania

mario.pavone@unict.it

http://www.dmi.unict.it/mpavone/

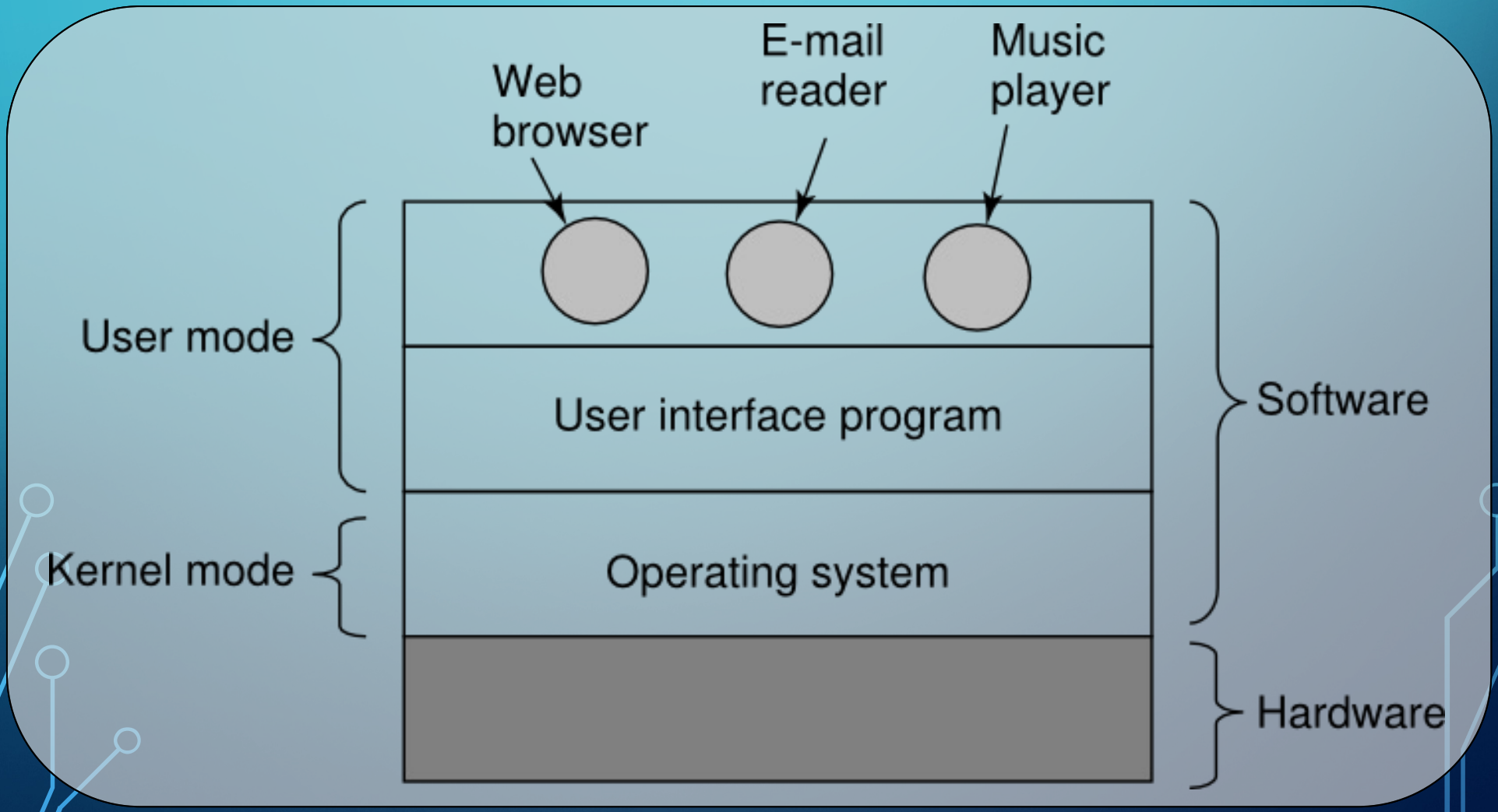


COS'È UN SISTEMA OPERATIVO?

- Un **moderno calcolatore** è tipicamente formato da:
  - uno o più processori;
  - memoria centrale;
  - dischi;
  - stampanti e altre periferiche di I/O.
- I dettagli di basso livello sono **molto complessi.**
- Gestire tutte queste componenti richiede uno strato intermedio software: il **Sistema Operativo.**

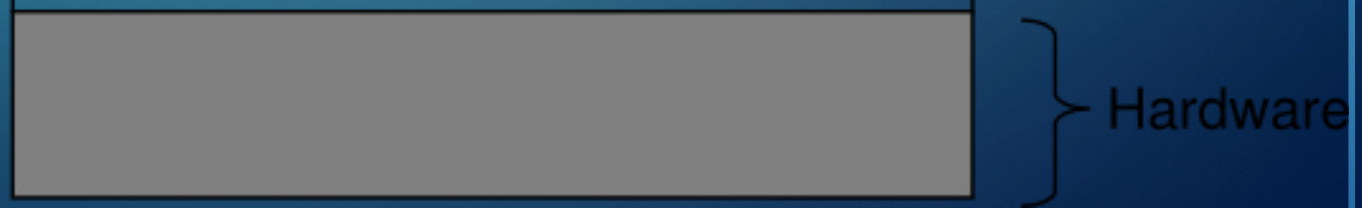
Cos'è un Sistema Operativo?

- Doppia modalità supportate dall'hardware:
  - **modalità kernel** (o supervisor);
  - **modalità utente**.

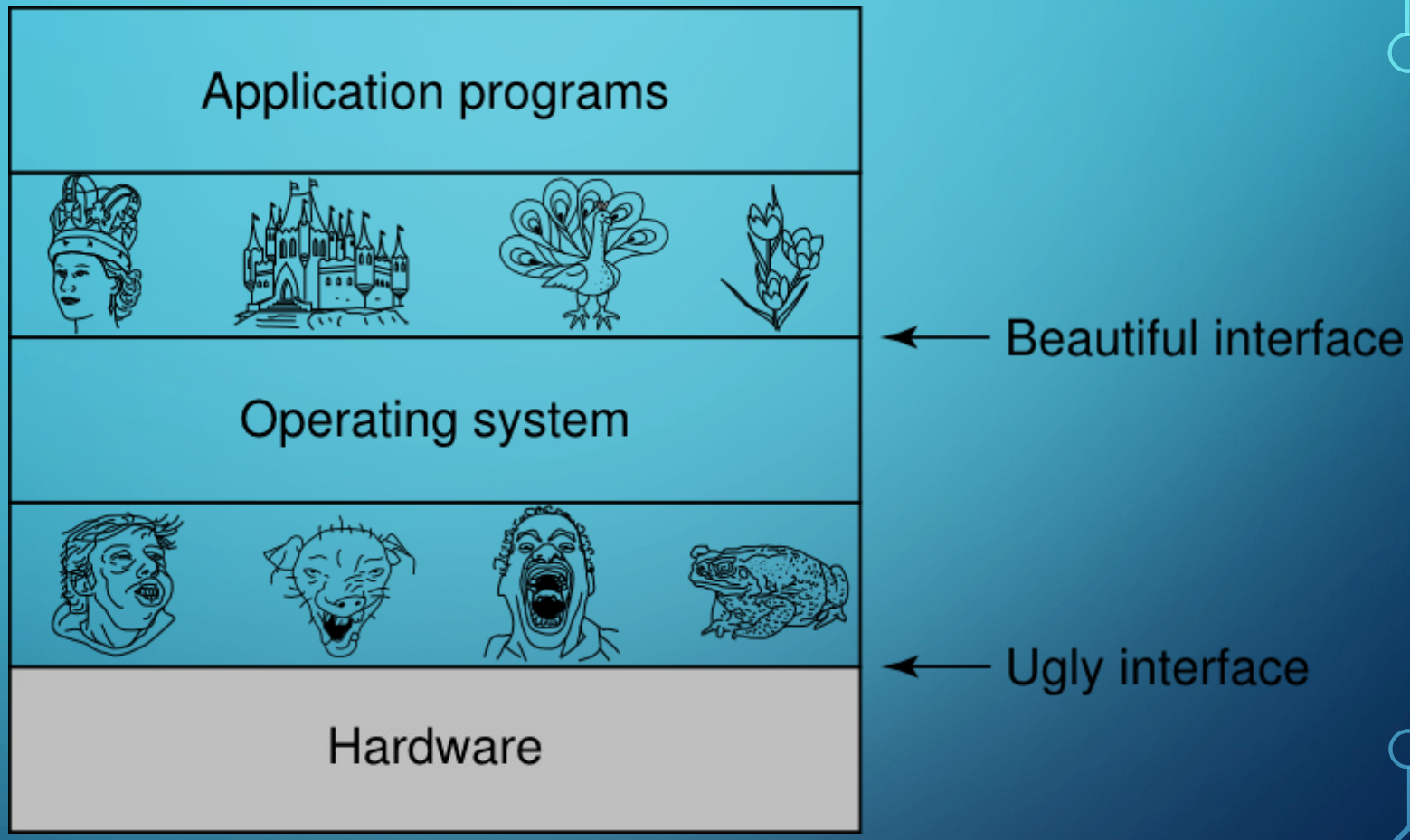


Cos'è un Sistema Operativo?

- **Eccezioni**: utility in UM che si interfacciano con il SO (**cambio password**) o funzionalità del kernel implementate in UM (**filesystem**).
- Tutto ciò che viene eseguito in modalità Kernel è parte del SO

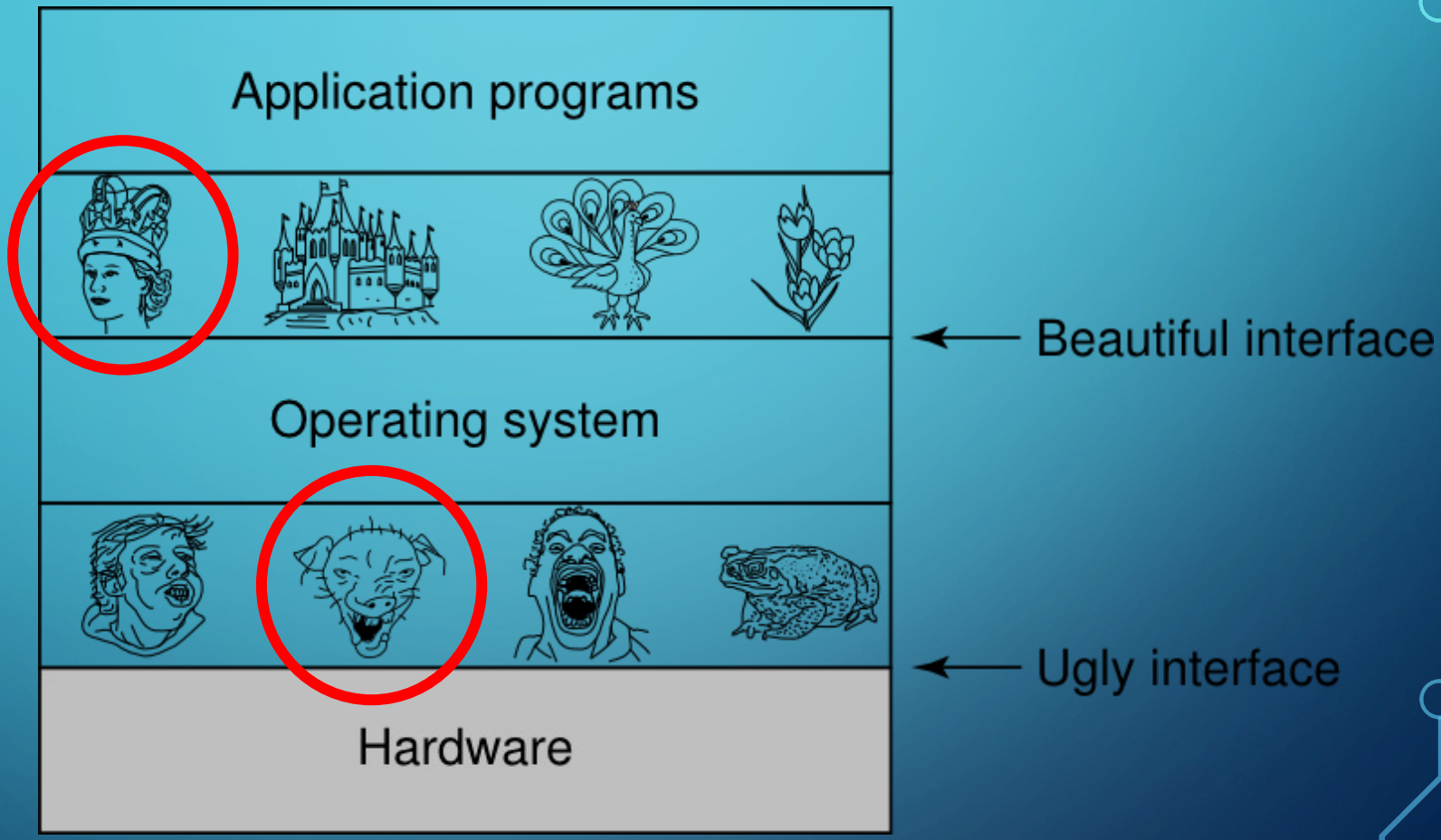


Il sistema operativo come macchina estesa

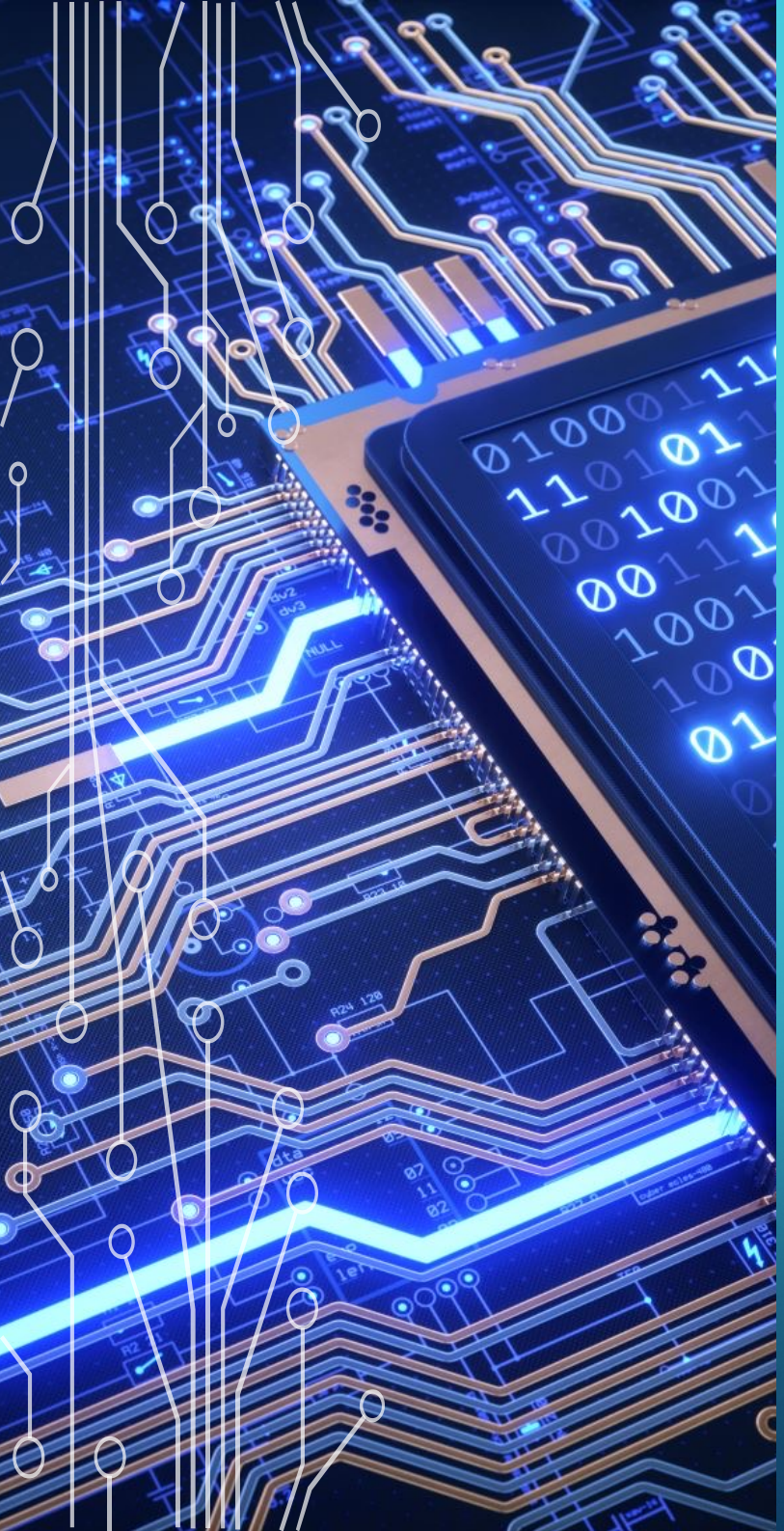


- concetto di **astrazione**

Il sistema operativo come macchina estesa

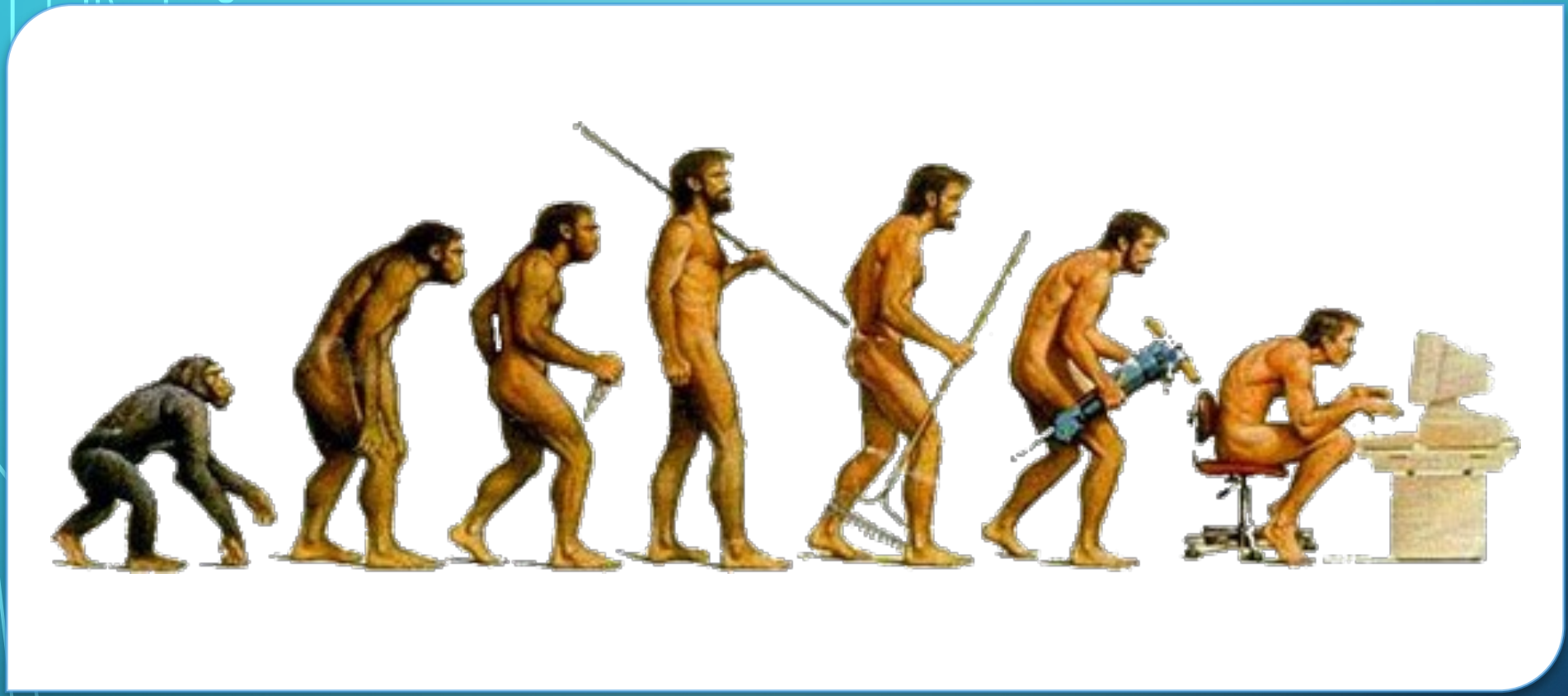


I Sistemi Operativi trasformano ciò che è brutto in bello!!!

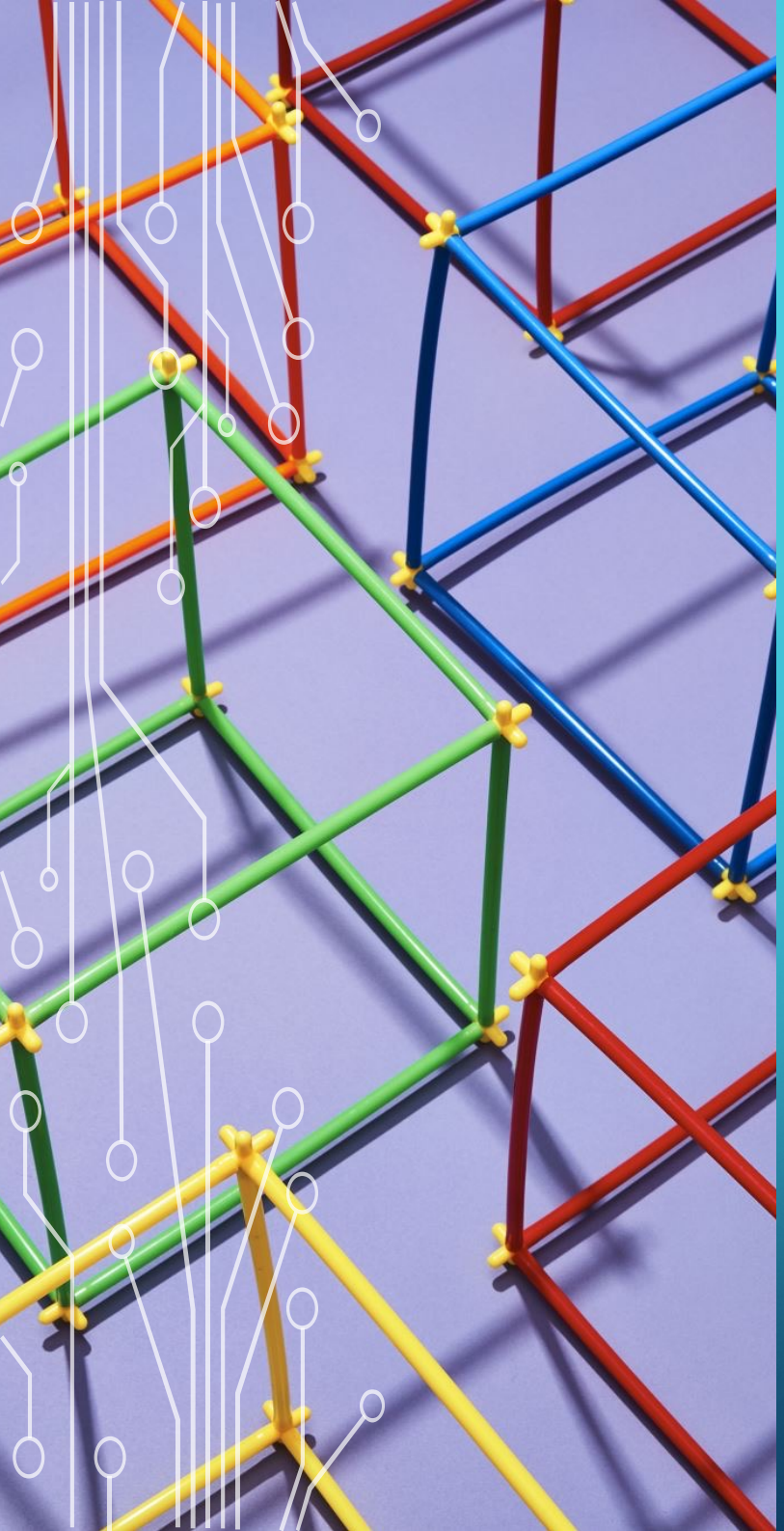


IL SISTEMA OPERATIVO COME GESTORE DELLE RISORSE

- Da un moderno sistema operativo ci aspettiamo che gestisca:
  - **più programmi** in esecuzione;
  - **più utenti.**
- Necessita **allocazione ordinata e controllata** di risorse quali: processori, memoria, unità di I/O,...
- Non solo hardware: file, database,...
- **Multiplexing** (condivisione risorse):
  - **nel tempo**: CPU, stampante,...
  - **nello spazio**: memoria centrale, disco,...



L'EVOLUZIONE DEI SISTEMI OPERATIVI



L'EVOLUZIONE DEI SISTEMI OPERATIVI

1° GENERAZIONE (1945-55): TUBI A VUOTO (ELETTRONICHE), RELÉ; SCHEDE A SPINOTTI E A PERFORAZIONE; LINGUAGGIO MACCHINA; NESSUN SISTEMA OPERATIVO. (SEMPLICI CALCOLI MATEMATICI)

2° GENERAZIONE (1955-65): TRANSISTOR, MAINFRAME, JOB SU SCHEDE, FORTRAN, ASSEMBLY, BATCH, NASTRI MAGNETICI, STAMPANTI. (CALCOLI SCIENTIFICI -> EQUAZIONI DIFFERENZIALI PARZIALI)

3° GENERAZIONE (1965-80): IC (INTEGRATED CIRCUITS) E MULTIPROGRAMMAZIONE; IBM SYSTEM/360; SOLO UNIVERSITÀ E GROSSE AZIENDE, SISTEMI OPERATIVI MOLTO COMPLESSI SCRITTI IN ASSEMBLY, DISCHI MAGNETICI, TIMESHARING, MULTICS (POCO SUCCESSO MA APRÌ UNA STRADA), UNIX (SCRITTO PER PDP-7) A CODICE DISPONIBILE, DUE VERSIONI SYSTEM V (AT&T) E BSD (BERKLEY), STANDARD POSIX, MINIX, LINUX.

L'EVOLUZIONE DEI SISTEMI OPERATIVI

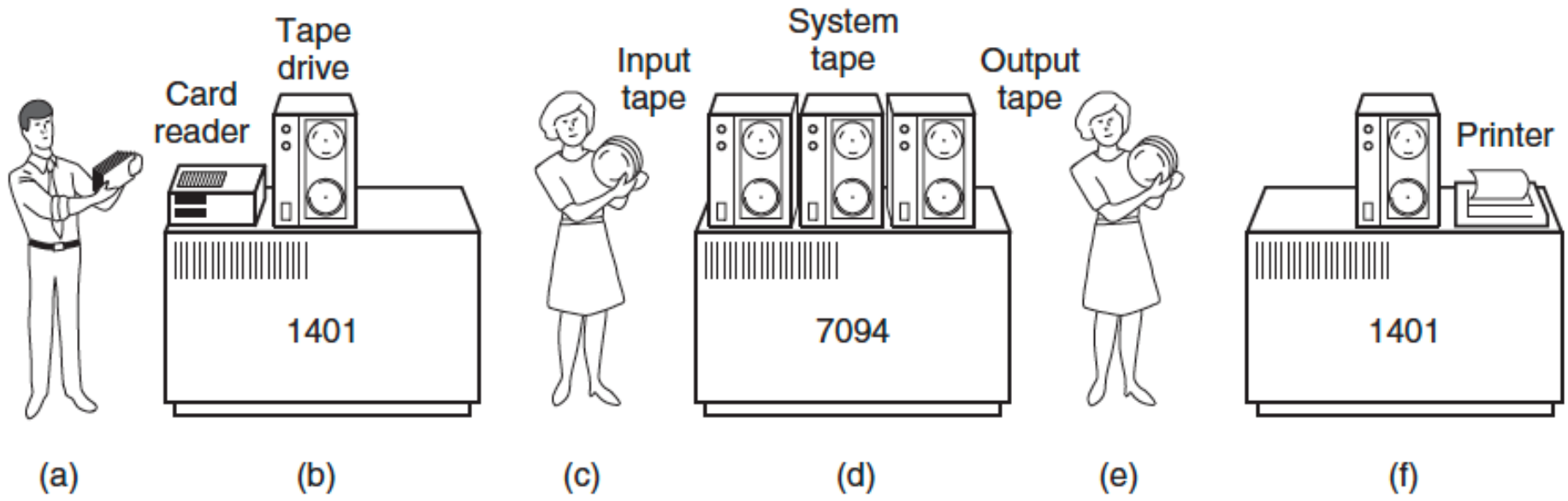
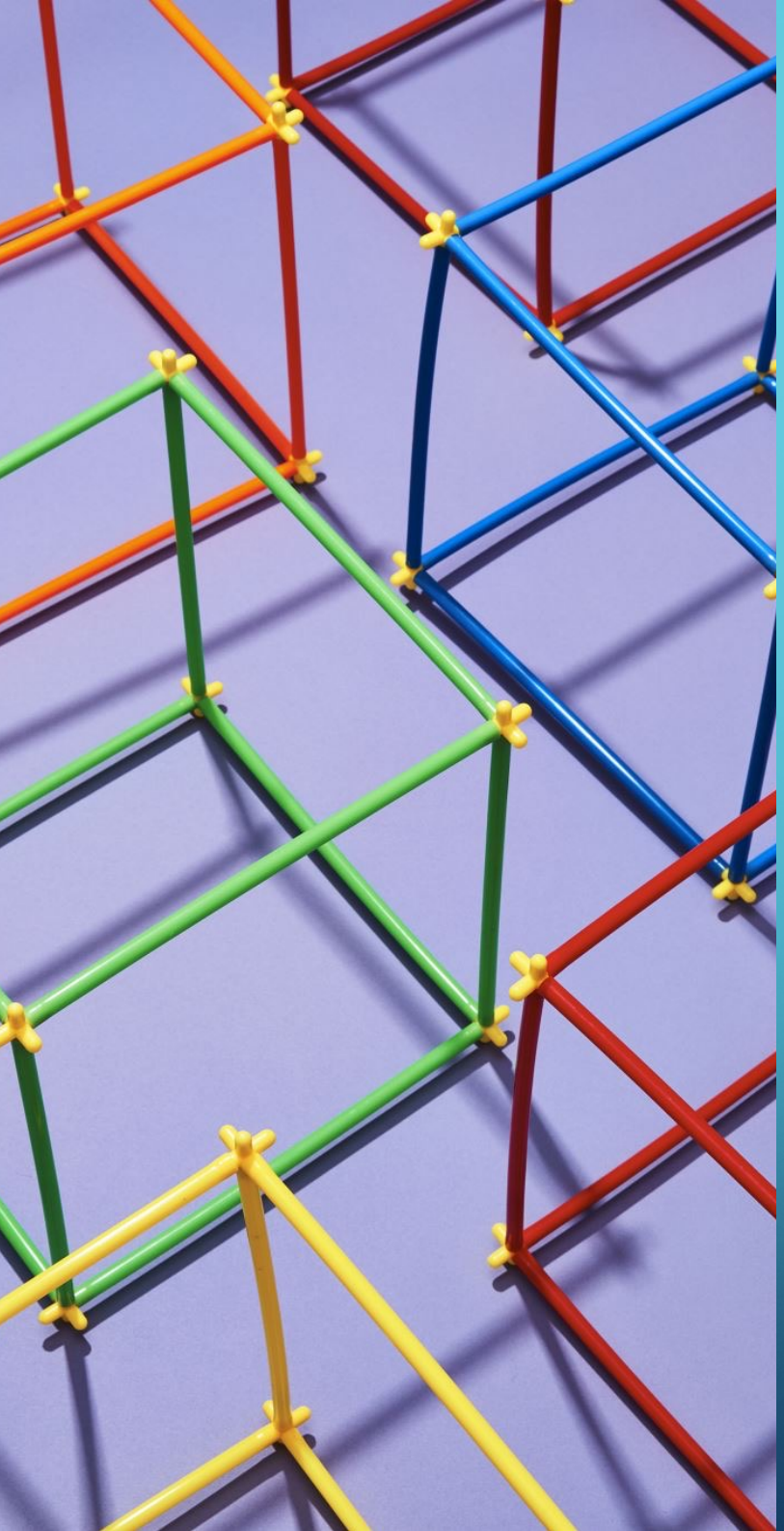


Figure 1-3. An early batch system. (a) Programmers bring cards to 1401. (b) 1401 reads batch of jobs onto tape. (c) Operator carries input tape to 7094. (d) 7094 does computing. (e) Operator carries output tape to 1401. (f) 1401 prints output.

Sistema Batch

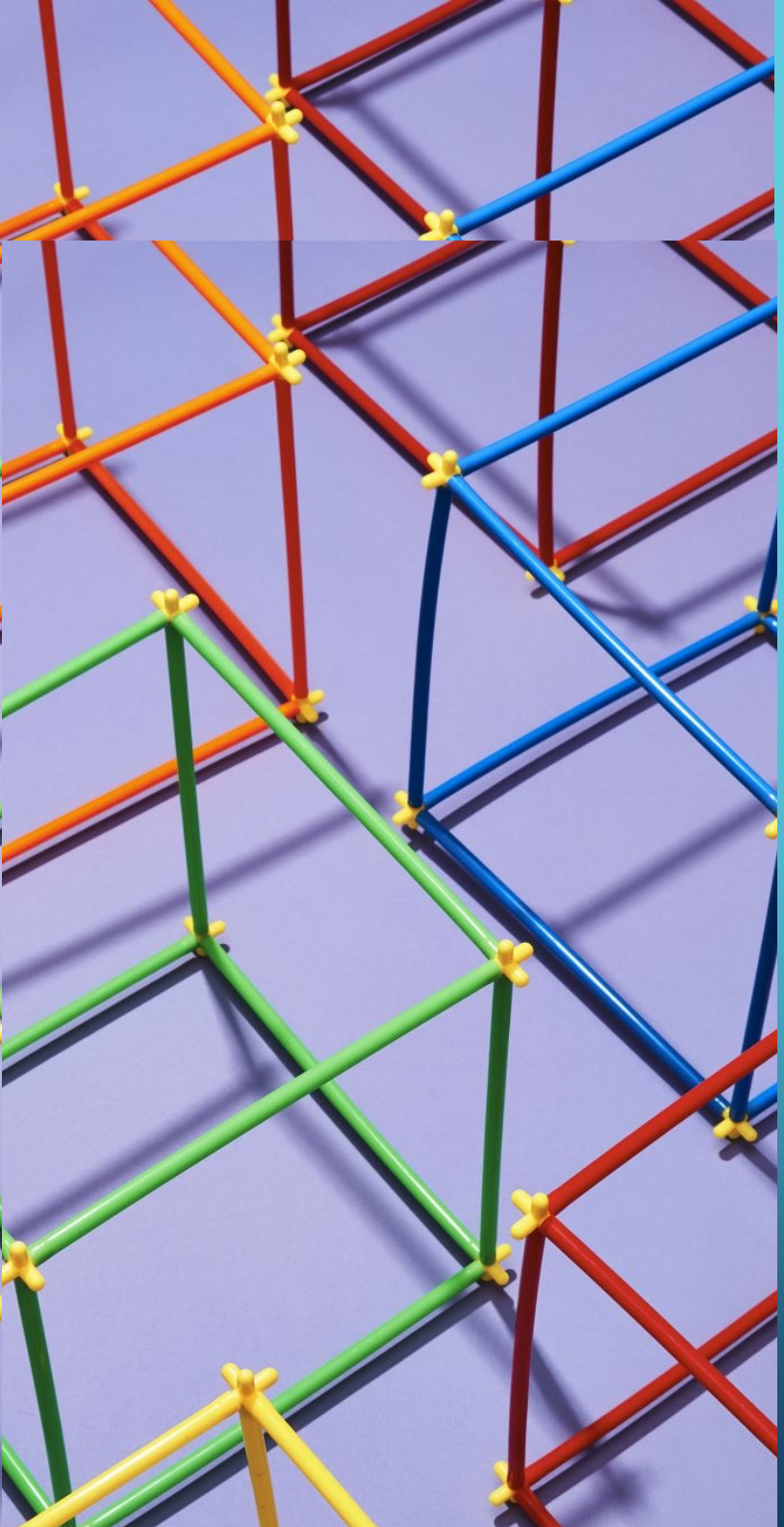


L'EVOLUZIONE DEI SISTEMI OPERATIVI

1° GENERAZIONE (1945-55): TUBI A VUOTO (ELETTRONICHE), RELÉ; SCHEDE A SPINOTTI E A PERFORAZIONE; LINGUAGGIO MACCHINA; NESSUN SISTEMA OPERATIVO. (SEMPLICI CALCOLI MATEMATICI)

2° GENERAZIONE (1955-65): TRANSISTOR, MAINFRAME, JOB SU SCHEDE, FORTRAN, ASSEMBLY, BATCH, NASTRI MAGNETICI, STAMPANTI. (CALCOLI SCIENTIFICI -> EQUAZIONI DIFFERENZIALI PARZIALI)

3° GENERAZIONE (1965-80): IC (INTEGRATED CIRCUITS) E MULTIPROGRAMMAZIONE; IBM SYSTEM/360; SOLO UNIVERSITÀ E GROSSE AZIENDE, SISTEMI OPERATIVI MOLTO COMPLESSI SCRITTI IN ASSEMBLY, DISCHI MAGNETICI, TIMESHARING, MULTICS (POCO SUCCESSO MA APRÌ UNA STRADA), UNIX (SCRITTO PER PDP-7) A CODICE DISPONIBILE, DUE VERSIONI SYSTEM V (AT&T) E BSD (BERKLEY), STANDARD POSIX, MINIX, LINUX.



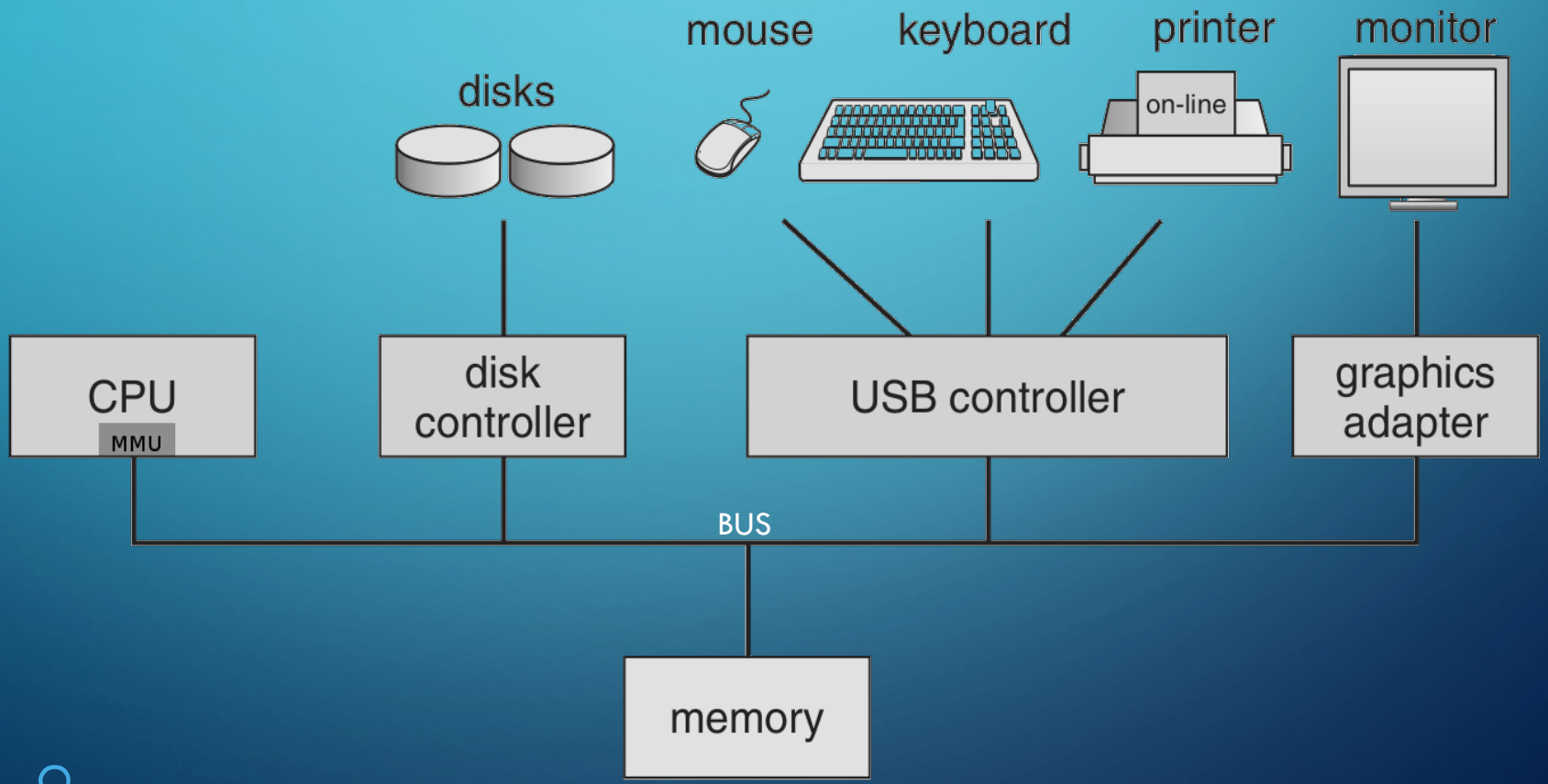
L'EVOLUZIONE DEI SISTEMI OPERATIVI

- 4° generazione (1980-oggi): i Personal Computer (PC), circuiti LSI (Large Scale Integration), Intel 8080 (1974), CP/M (Digital Research), PC IBM, Gates, MS-DOS, 8086, 286, 386, 4086, GUI (Xerox), Jobs, Apple, Windows 3.1, 95, 98 (16-bit), NT, NT4, 2000, XP (32-bit), Vista, Seven, UNIX oggi, FreeBSD, Mac OSX, Gnome/KDE,Android, iOS.

- 5° generazione (1990-oggi): Mobile Computers, PDA, Symbian OS, iOS, Android, Windows Mobile

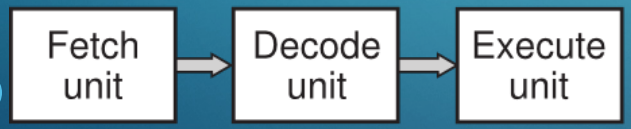
Uno sguardo all'hardware

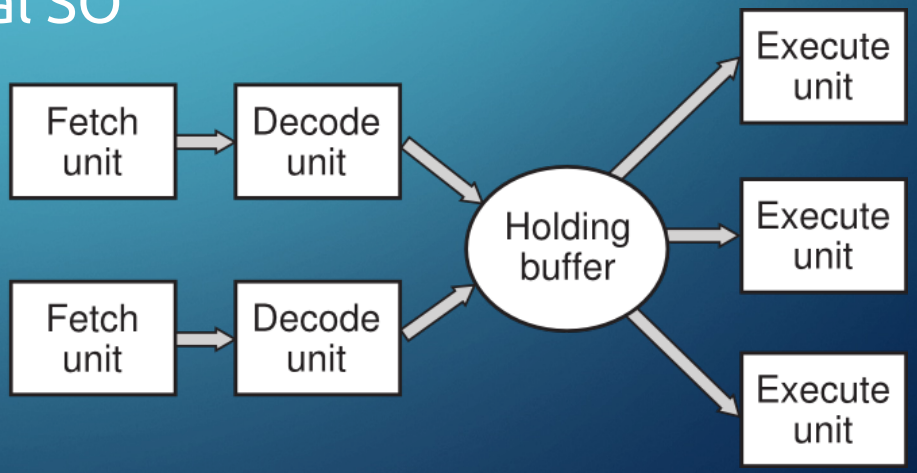
- Architettura (semplificata) di un calcolatore



Il Processore

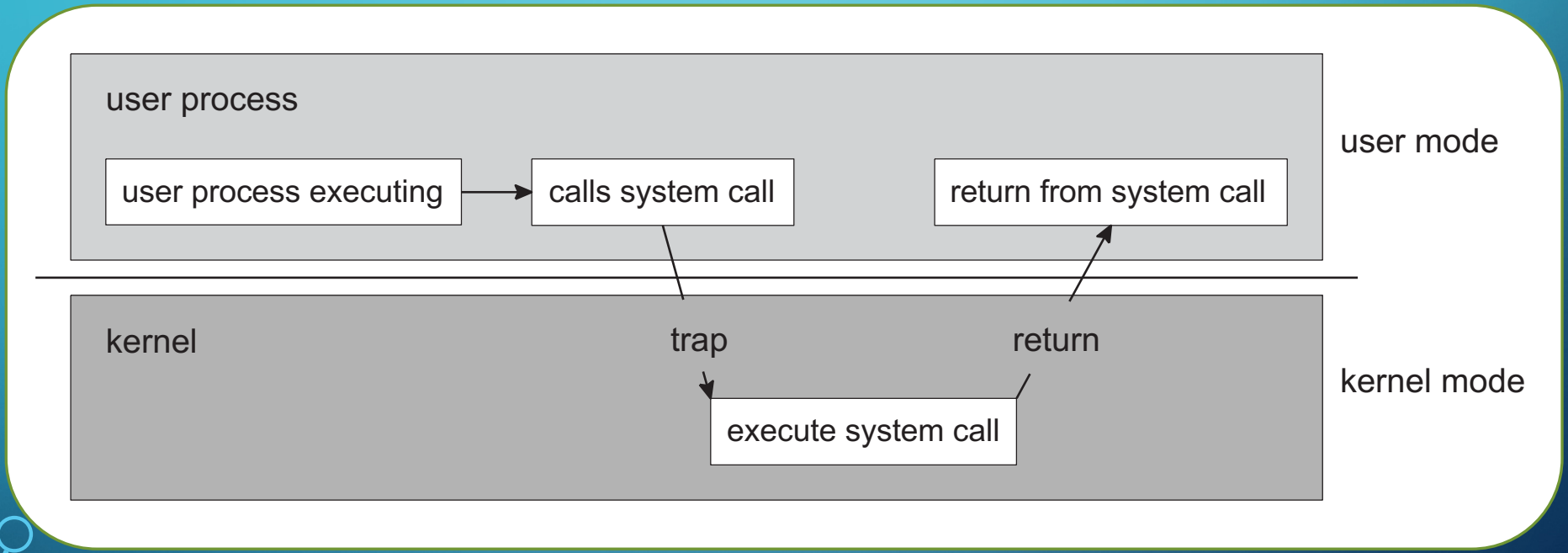
- **Ciclo di base**: prelevamento (fetch), decodifica, esecuzione
- Registri particolari:
  - **Program Counter** (PC);
  - **Stack Pointer** (SP);
  - **Program Status Word** (PSW).
- Progettazioni avanzate: **pipeline**, **cpu superscalare**
  - non del tutto trasparenti al SO





Modalità di esecuzione

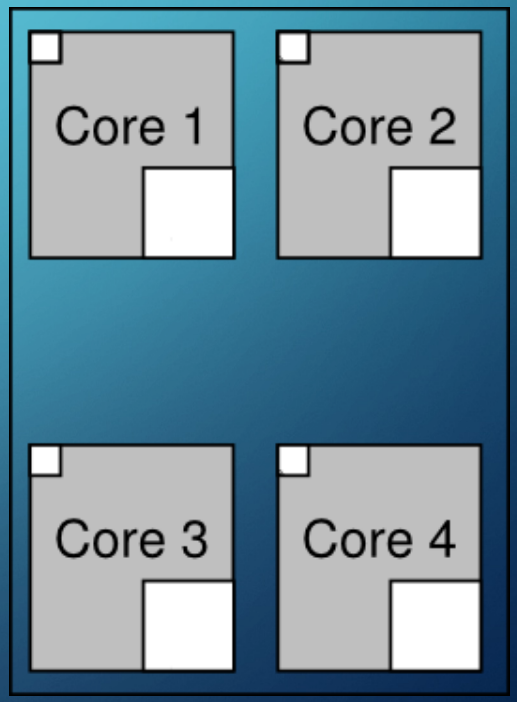
- Doppia modalità di esecuzione;
- **chiamate di sistema** (TRAP);



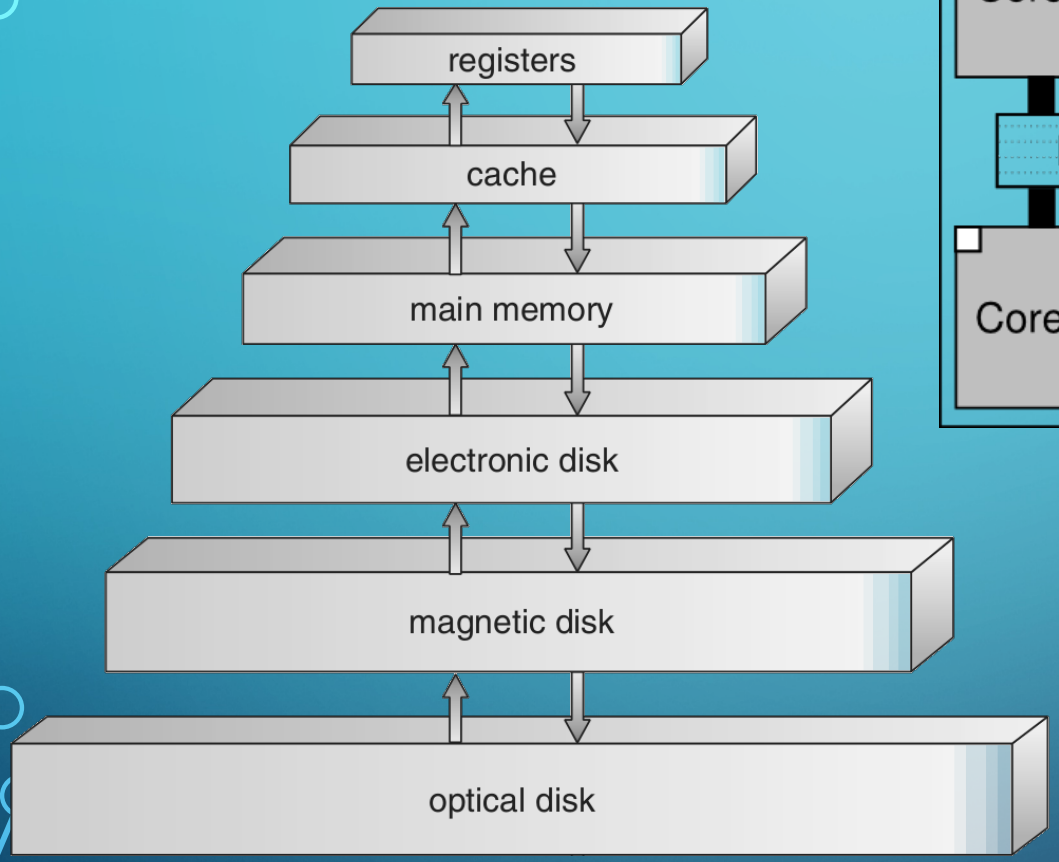
- **interrupt** hardware.

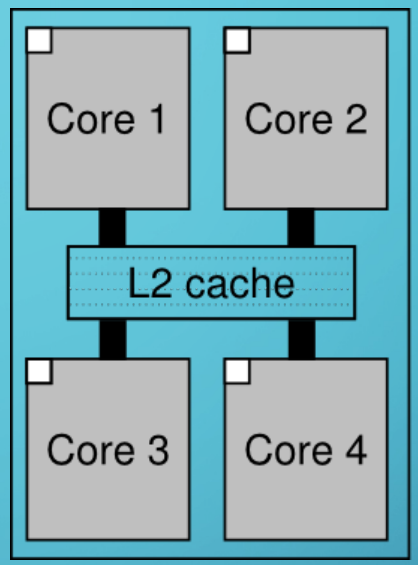
Più processori

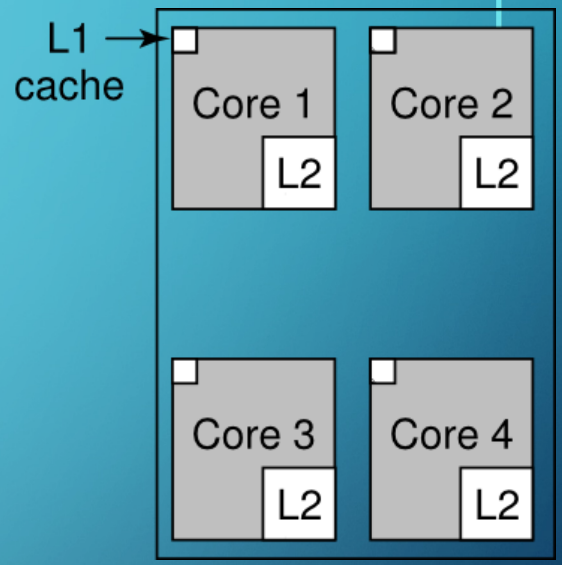
- **Multithreading** (o hyperthreading):
  - tiene all'interno della CPU lo stato di due thread;
  - non c'è una esecuzione parallela vera e propria;
  - il S.O. deve tenerne conto.
- **Multiprocessori**, vantaggi:
  - **throughput**;
  - **economia di scala**;
  - **affidabilità**;
- **Multicore**;
- **GPU**.

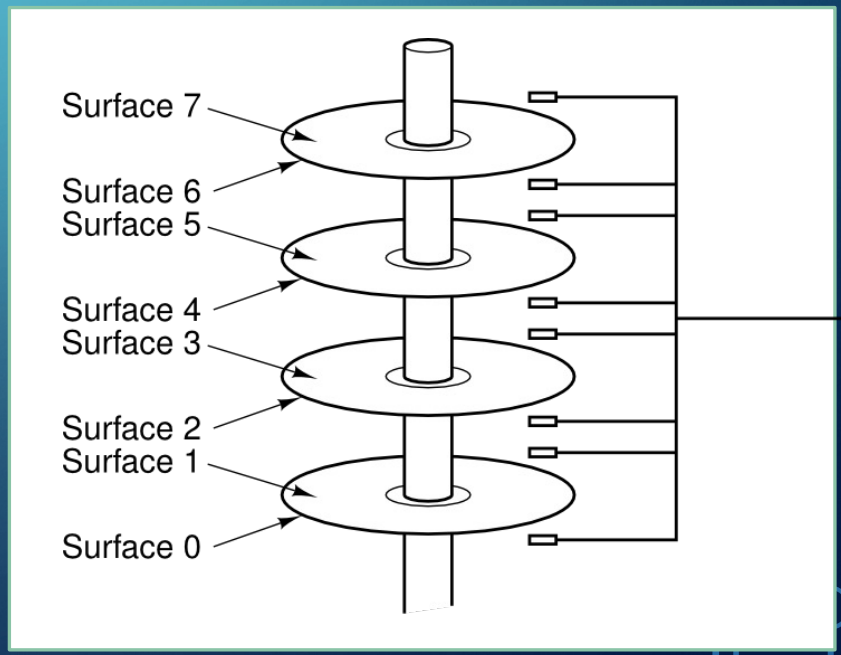


Memorie



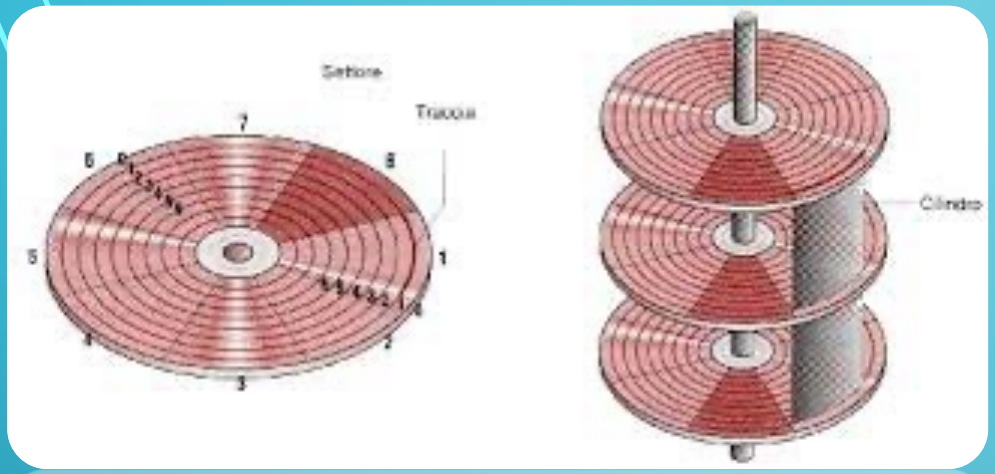


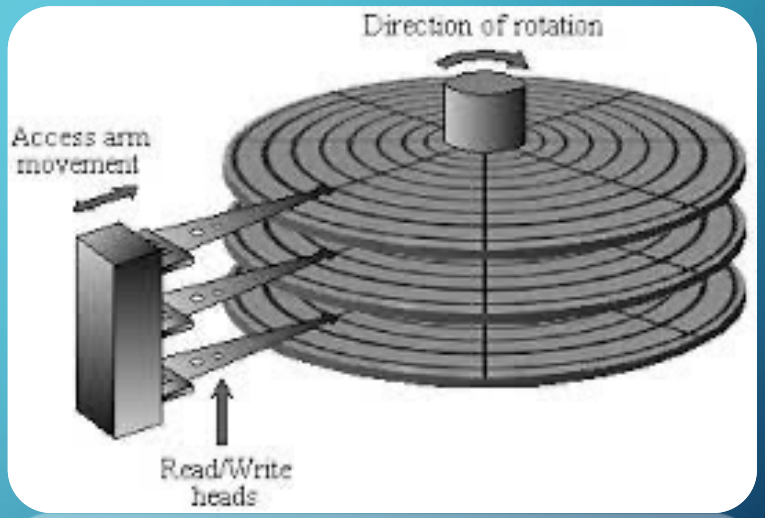


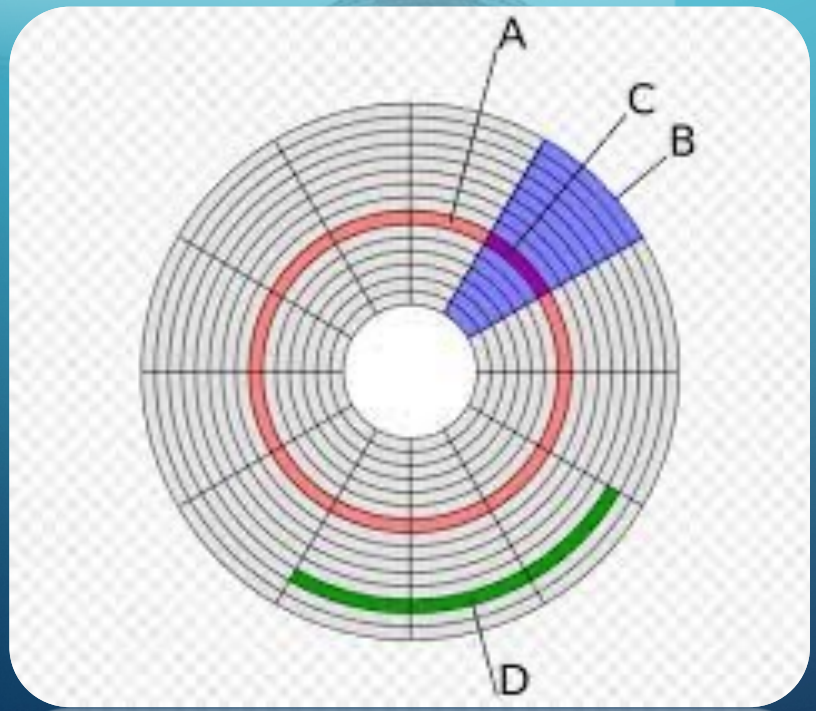


- Unità di misura: kB/MB/GB *vs.* KiB/MiB/GiB

Disco Magnetico







DISPOSITIVI DI I/O

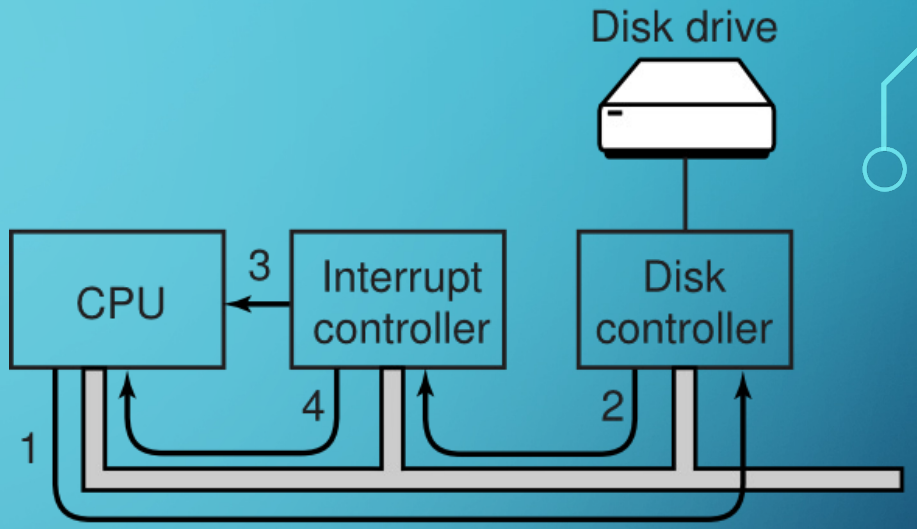
SO interagisce spesso con Dispositivi I/O

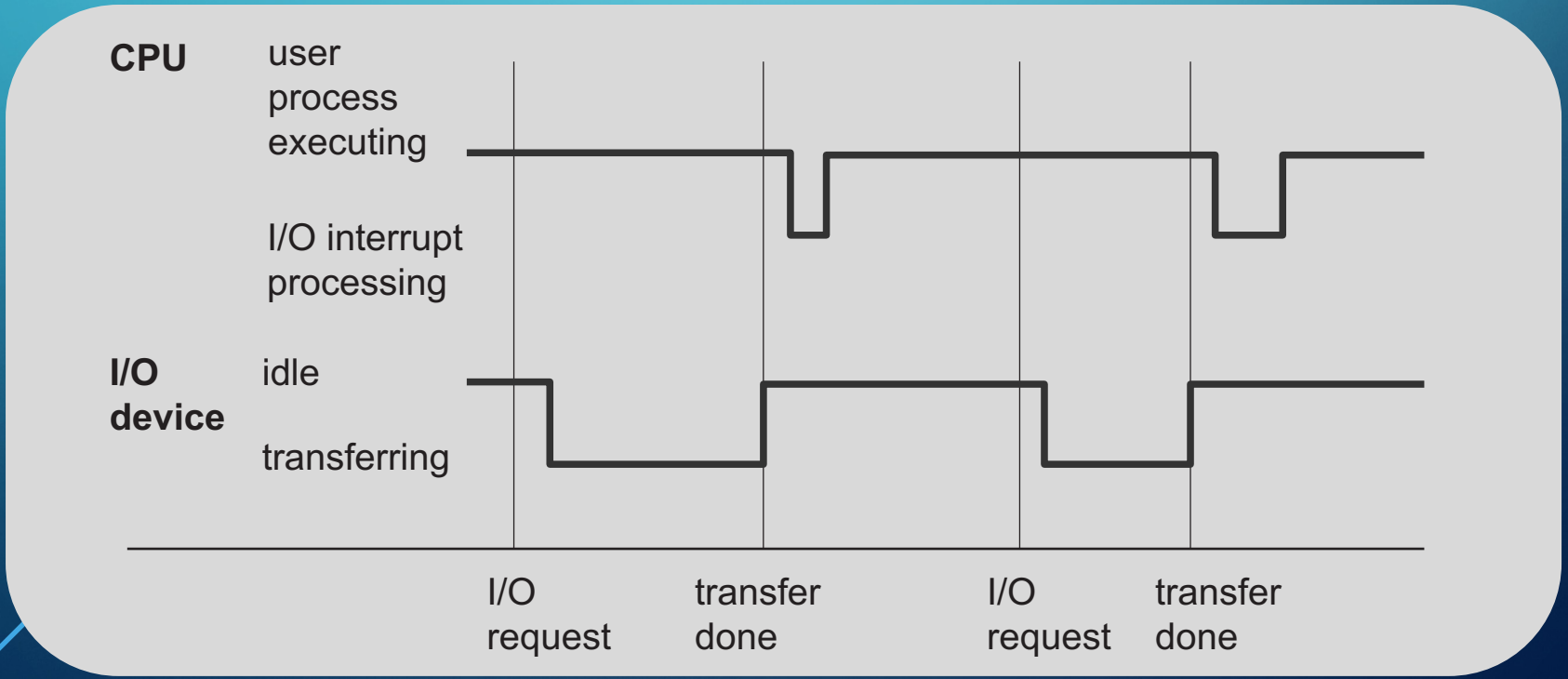
Costituito da due componenti:

- **Controller**: chip o insieme di chip che controlla il dispositivo
  - accetta comandi dal SO
- più semplice da usare per il SO;
- **Dispositivo** stesso: interfaccia elementare ma complicata da pilotare.
- esempio: dischi SATA.

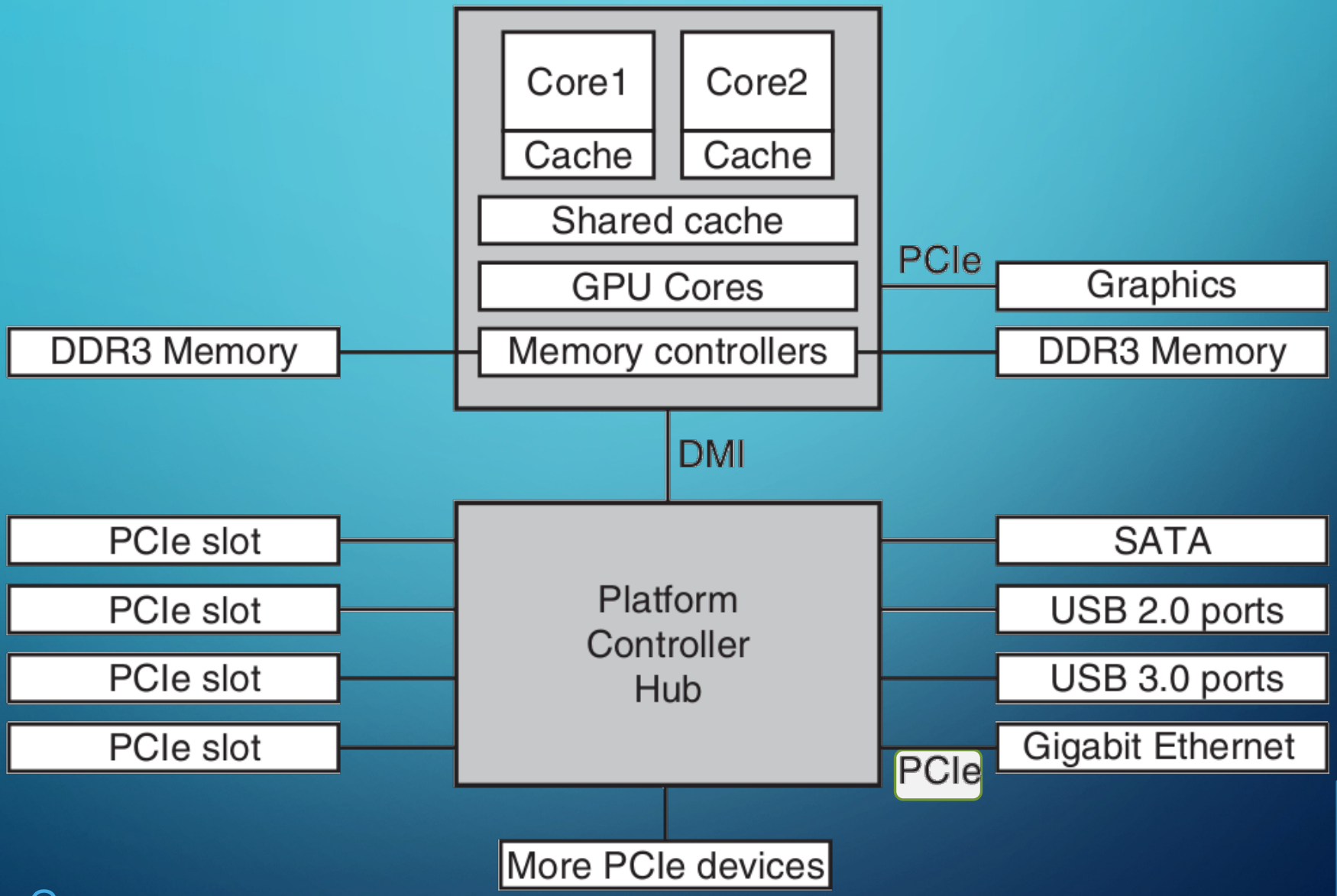
Dispositivi di I/O

- Modalità di I/O:
  - **busy waiting**;
  - con programmazione di **interrupt**;
  - con uso del **DMA**.





Bus


Core1
Core2
Cache
Cache
Shared cache
GPU Cores
Memory controllers
PCIe
Graphics
DDR3 Memory
DDR3 Memory
DMI
PCIe slot
PCIe slot
PCIe slot
PCIe slot
Platform Controller Hub
SATA
USB 2.0 ports
USB 3.0 ports
Gigabit Ethernet
PCIe
More PCIe devices

LO ZOO DEI SISTEMI OPERATIVI

- Sistemi operativi per mainframe/server
- Sistemi operativi per personal computer
- Sistemi operativi per multiprocessore
- Sistemi operativi per palmari/smart-phone
- Sistemi operativi per sistemi integrati (embedded)
- Sistemi operativi real-time

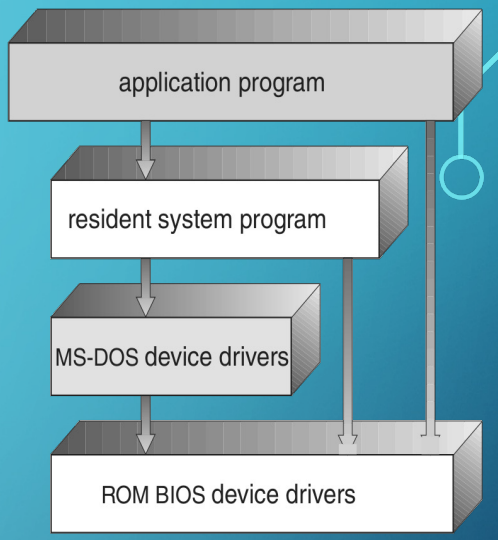
STRUTTURA DI UN SISTEMA OPERATIVO

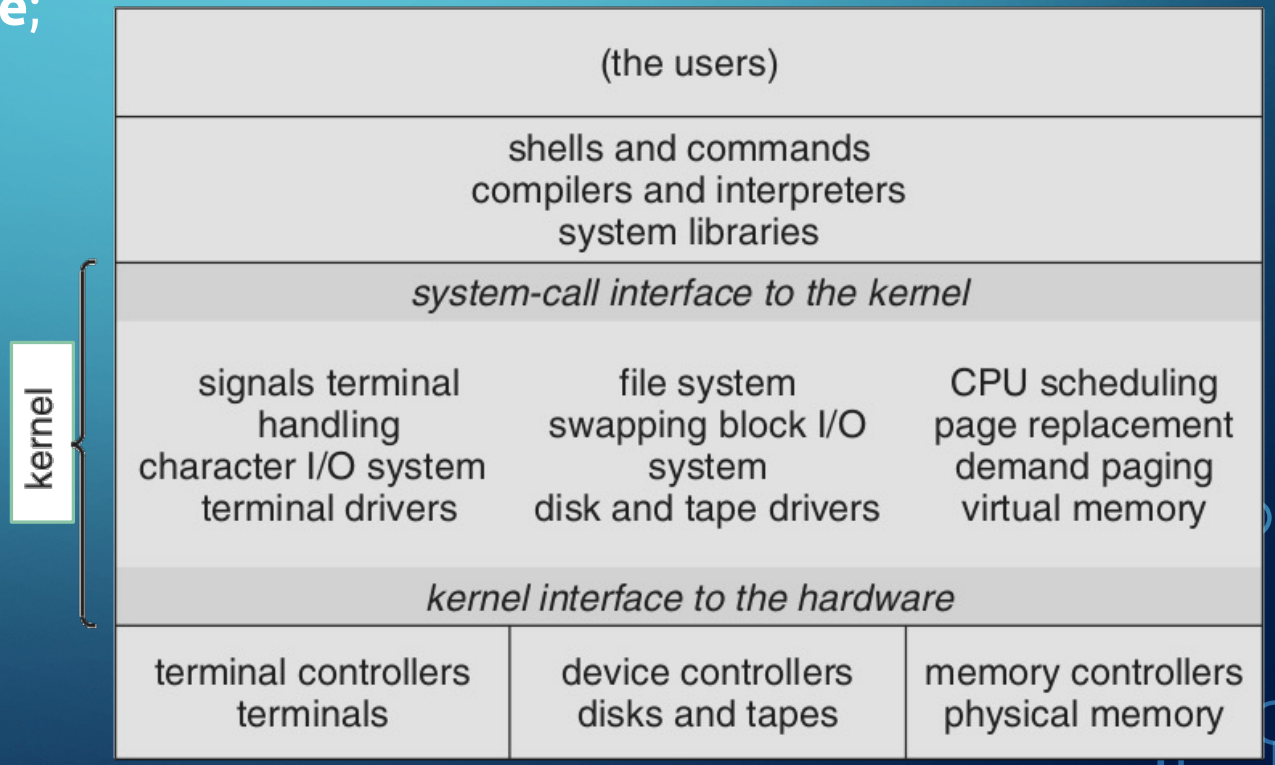
Alcune **possibili strutture** per un SO:

- Monolitici
- A livelli (o a strati)
- Microkernel
- A Moduli
- Macchine virtuali

Struttura monolitica

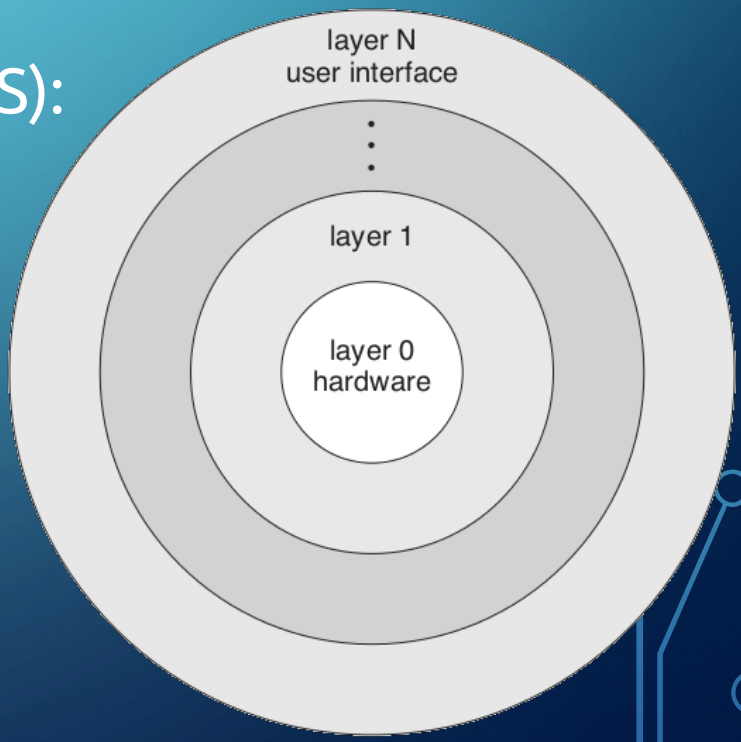
- **Monolitici:**
  - **nessun supporto** hardware;
    - problemi...;
    - esempi: MS-DOS, UNIX;
  - arrivò il supporto hardware alla **modalità kernel/utente**;
    - **unico kernel** con tutto dentro;
    - ogni componente può richiamare tutti gli altri
    - poco gestibile nel tempo.



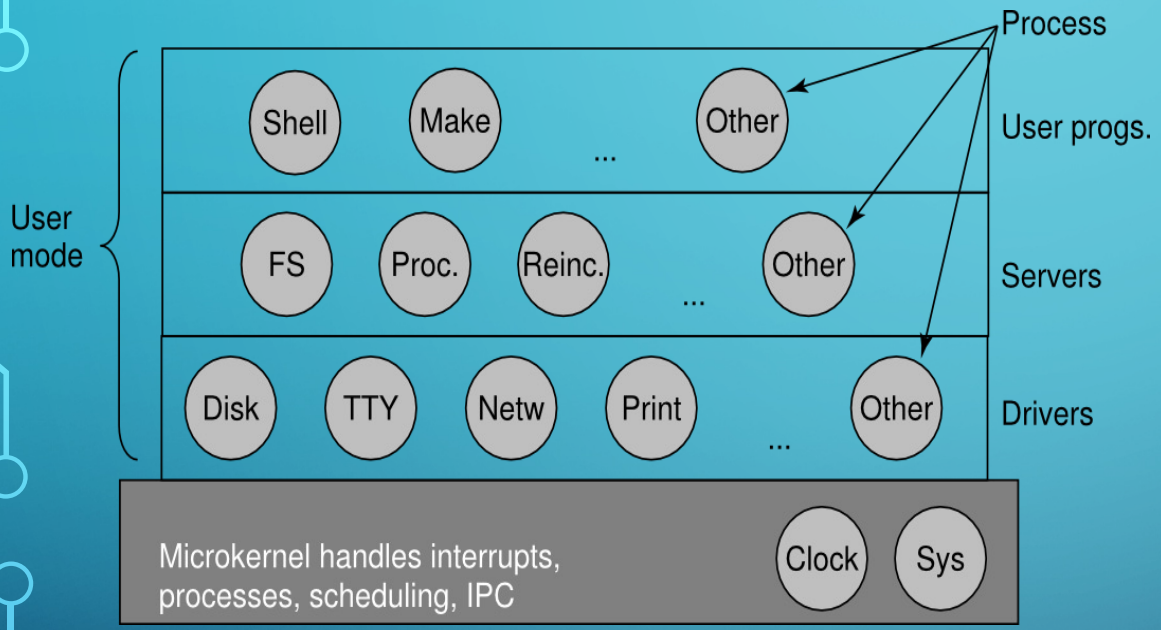


Struttura a livelli (o a strati)

- Si utilizza una **gerarchia di livelli**;
- ogni livello implementa delle **funzionalità** impiegando quelle fornite da quello inferiore;
- **migliore progettazione**, più semplice da sviluppare e controllare (incapsulamento tipo OOP); suddivisione a livello progettuale;
- variante ad **anelli concentrici** (MULTICS):
  - **separazione forzata** dall'hardware;
- **problemi di prestazioni** dovuti alle chiamate nidificate e al relativo overhead.



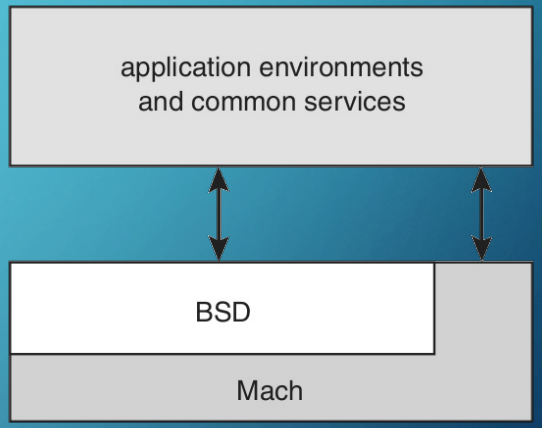
MICROKERNEL

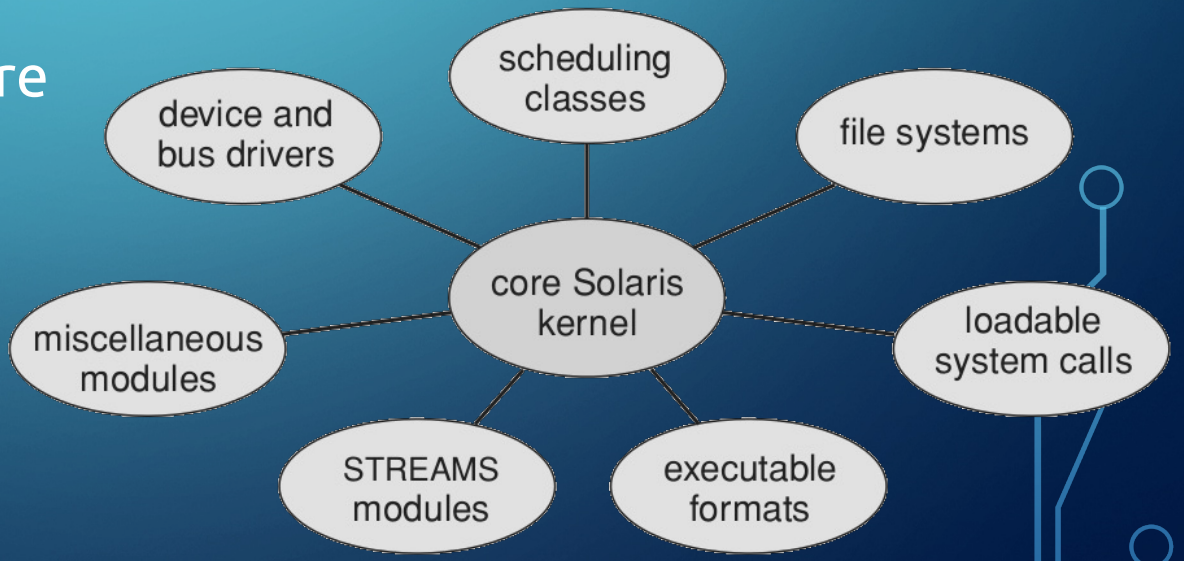


- Uso di un microkernel minimale che si occupa dello scheduling, memoria e IPC;
- tutto il resto gestito da **moduli** (livello utente): filesystem, driver di dispositivi;
- comunicazione attraverso **messaggi**;
- miglior design (componenti piccoli) e migliore stabilità;
- esempi: MINIX 3, Mach, QNX, Mac OS X (Darwin), Windows NT

Struttura a Moduli

- Idea della **programmazione OO** applicata al kernel;
- **moduli** che implementano un qualche aspetto specifico;
  - filesystem, driver,...
- **kernel principale** a funzionalità ridotte;
- moduli **caricabili dinamicamente**;
- design pulito (ad oggetti);
- **efficiente**:
  - ogni modulo può invocare qualunque altro modulo direttamente;
  - niente messaggi;
- esempi: Solaris, Linux, Mac OS X (ibrido).





Macchine virtuali

- L'estremizzazione del concetto di astrazione porta alla **virtualizzazione**;
- **Perchè?** Uso di più SO, VPS, isolamento dei servizi,...
- **Simulazione**, **paravirtualizzazione**.
- Viene tutto gestito dallo **Hypervisor**:
  - **Hypervisor di tipo 1**:
    - gira direttamente sull'hardware;
    - esempi: VMware ESX/ESXi, Microsoft Hyper-V hypervisor;
  - **Hypervisor di tipo 2**:
    - è un processo in un SO Host
    - esempi: VMware Workstation, VirtualBox.
- **Supporto hardware** per efficienza.
- **Java Virtual Machine**.