

Sistemi Operativi (M-Z)

C.d.L. in Informatica
(laurea triennale)

A.A. 2020-2021

Prof. Mario F. Pavone

Dipartimento di Matematica e Informatica
Università degli Studi di Catania

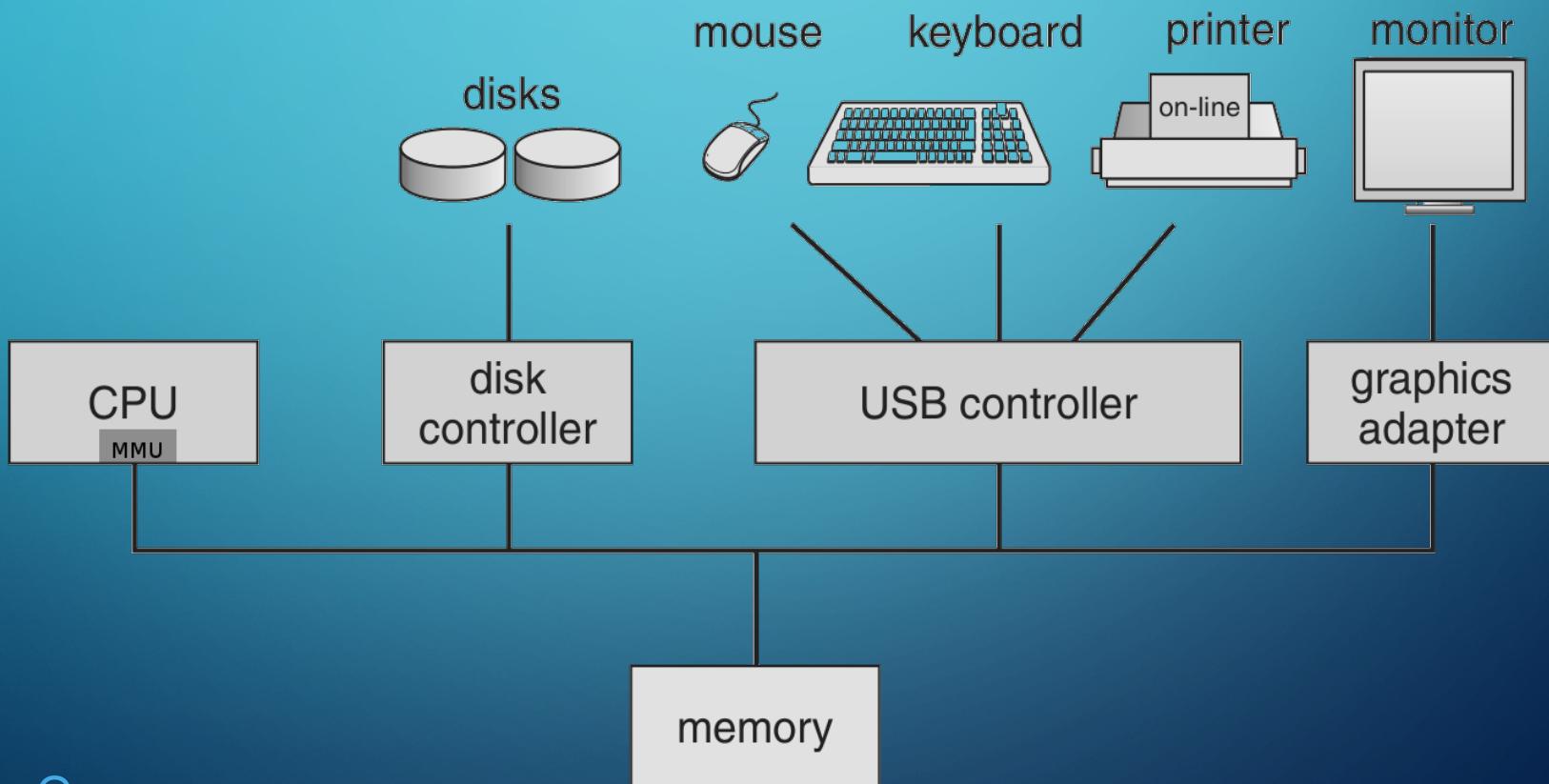
mario.pavone@unict.it

mpavone@dmi.unict.it

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

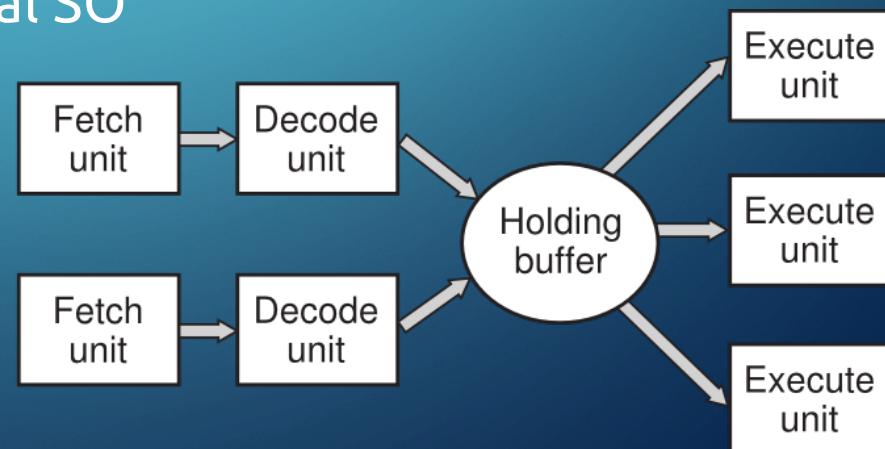
Uno sguardo all'hardware

Architettura (semplificata) di un calcolatore



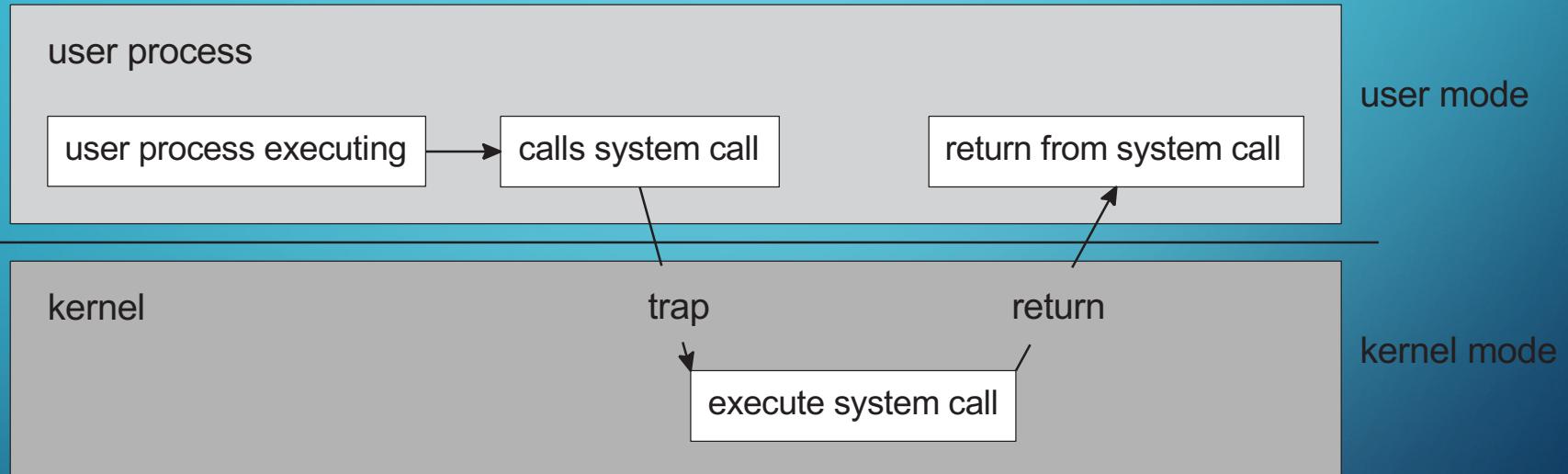
Il Processore

- 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):

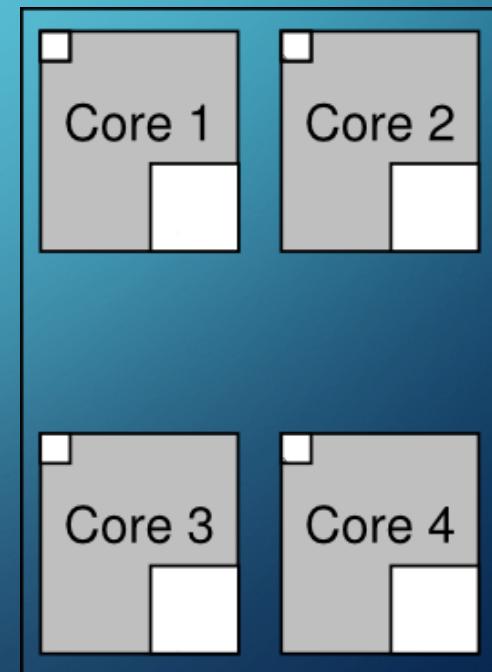
- tiene all'interno della CPU lo stato di due thread;
- non c'è una esecuzione parallela vera e propria;
- il S.O. deve tenerne conto.

- **Multiprocessori**, vantaggi:

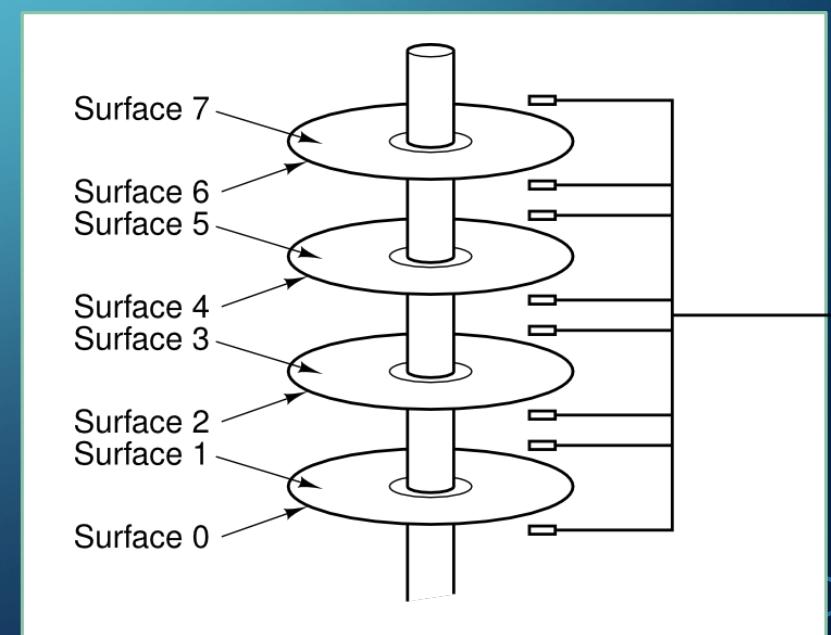
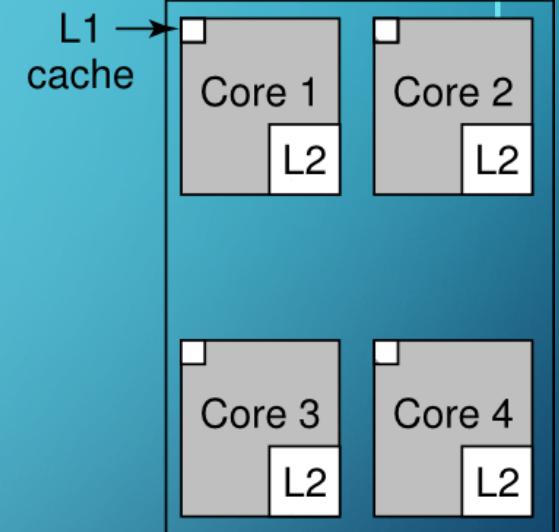
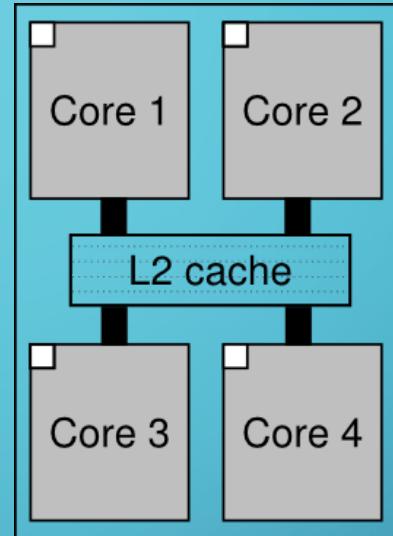
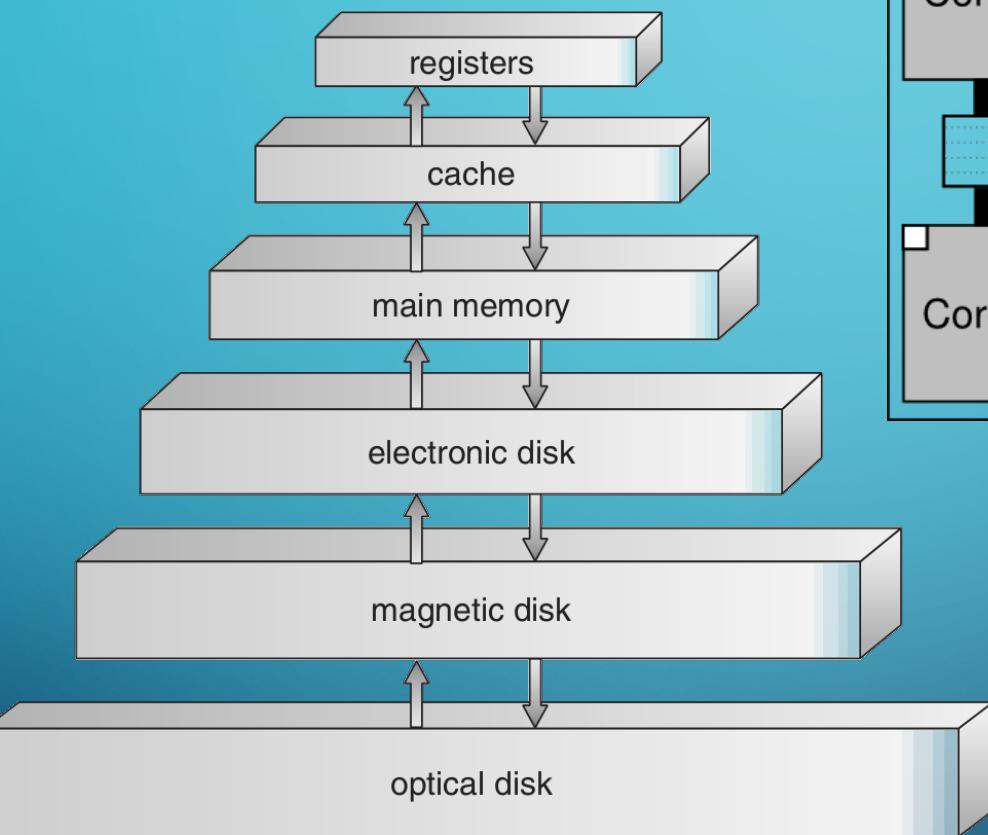
- throughput;
- economia di scala;
- affidabilità;

- **Multicore**;

- **GPU**.



Memorie



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

DISPOSITIVI DI I/O

SO interagisce spesso con Dispositivi I/O

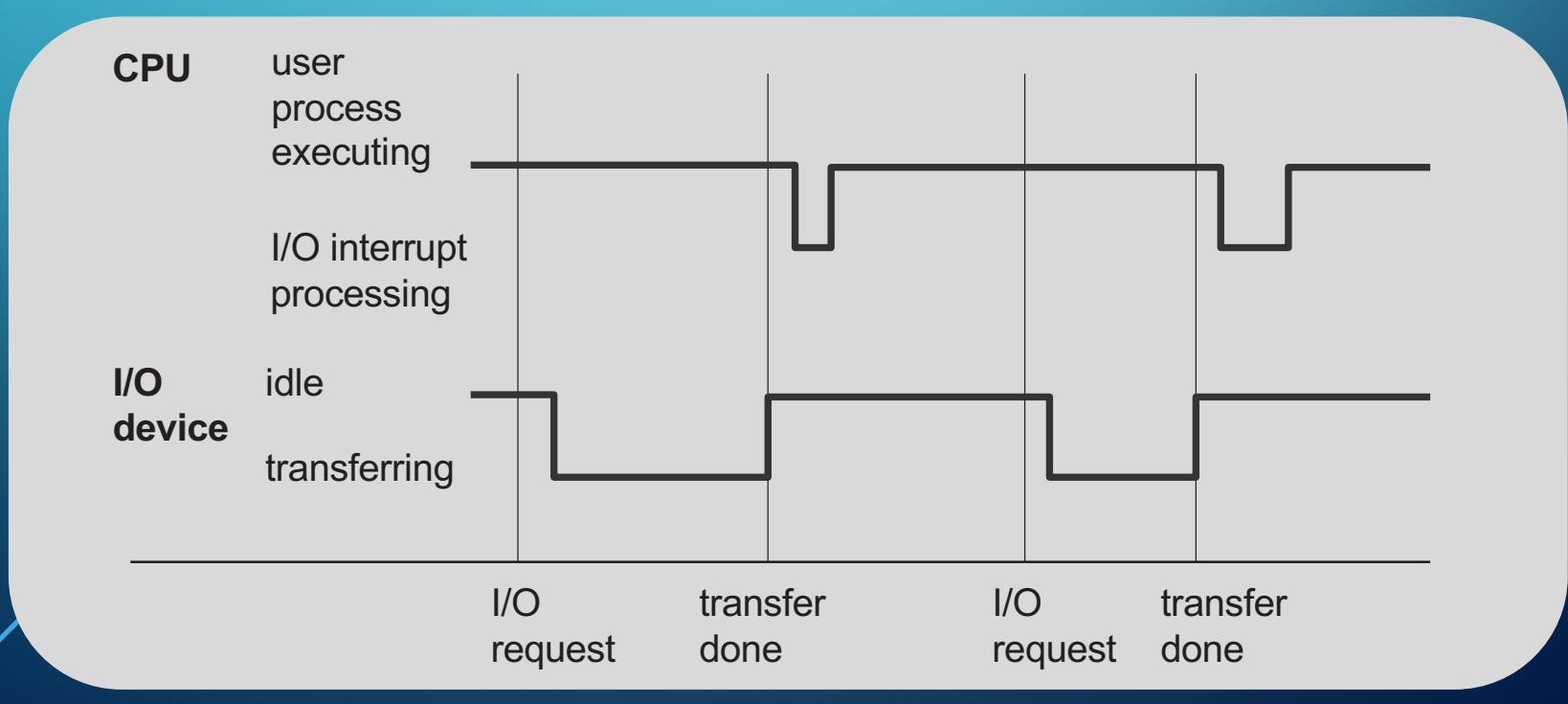
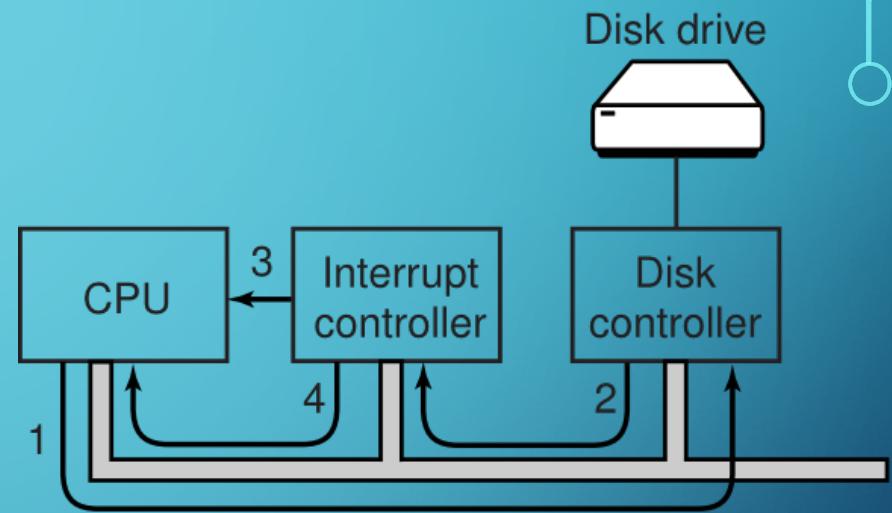
Costituito da due componenti:

- **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.

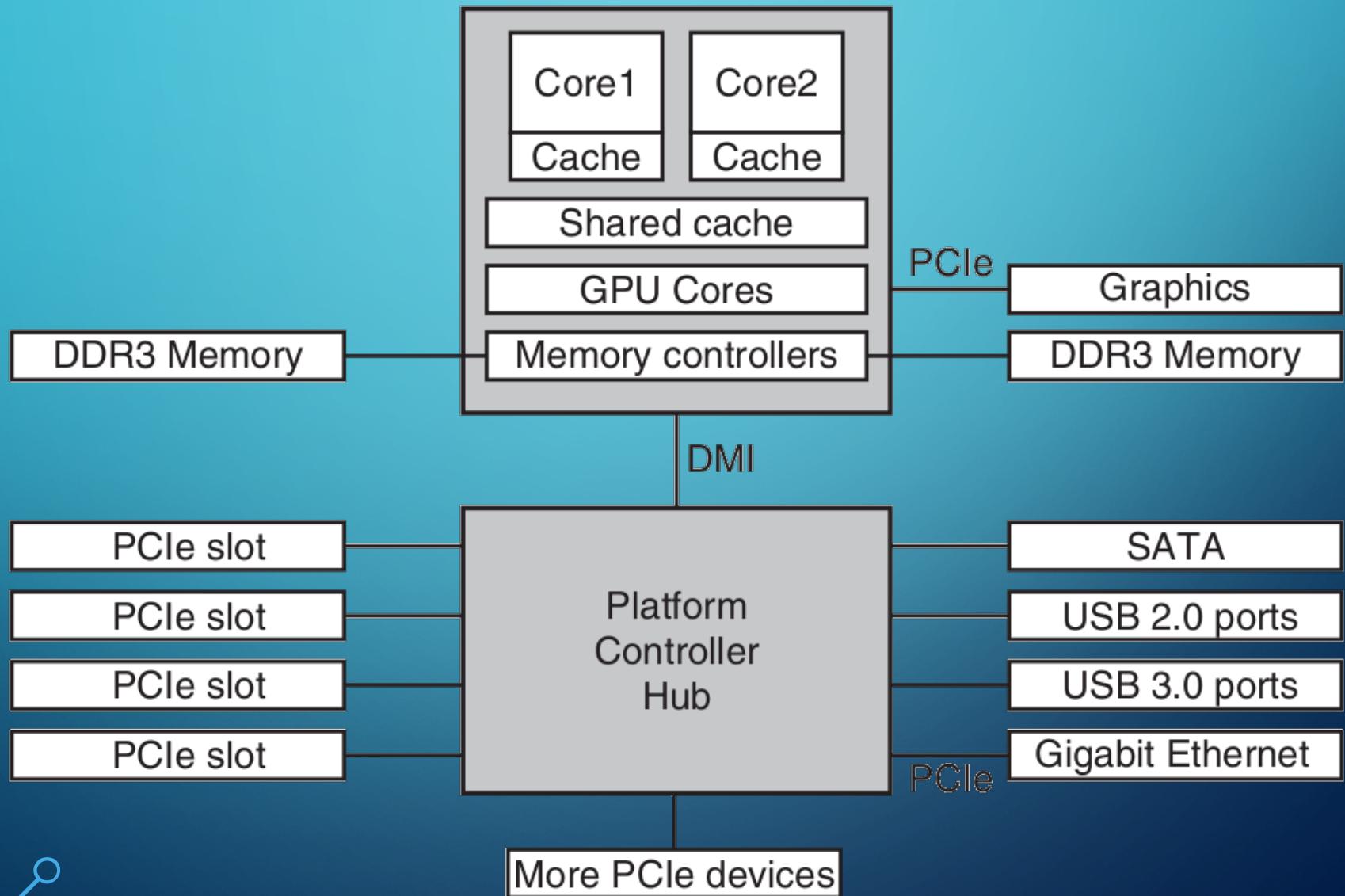
Dispositivi di I/O

Modalità di I/O:

- **busy waiting;**
- con programmazione di **interrupt**;
- con uso del **DMA**.



Bus





LO ZOO DEI SISTEMI OPERATIVI

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

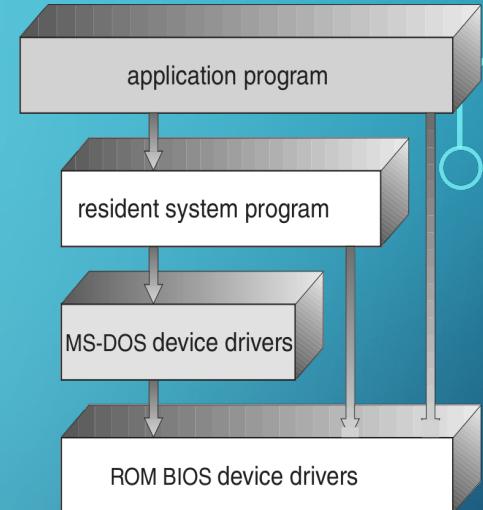
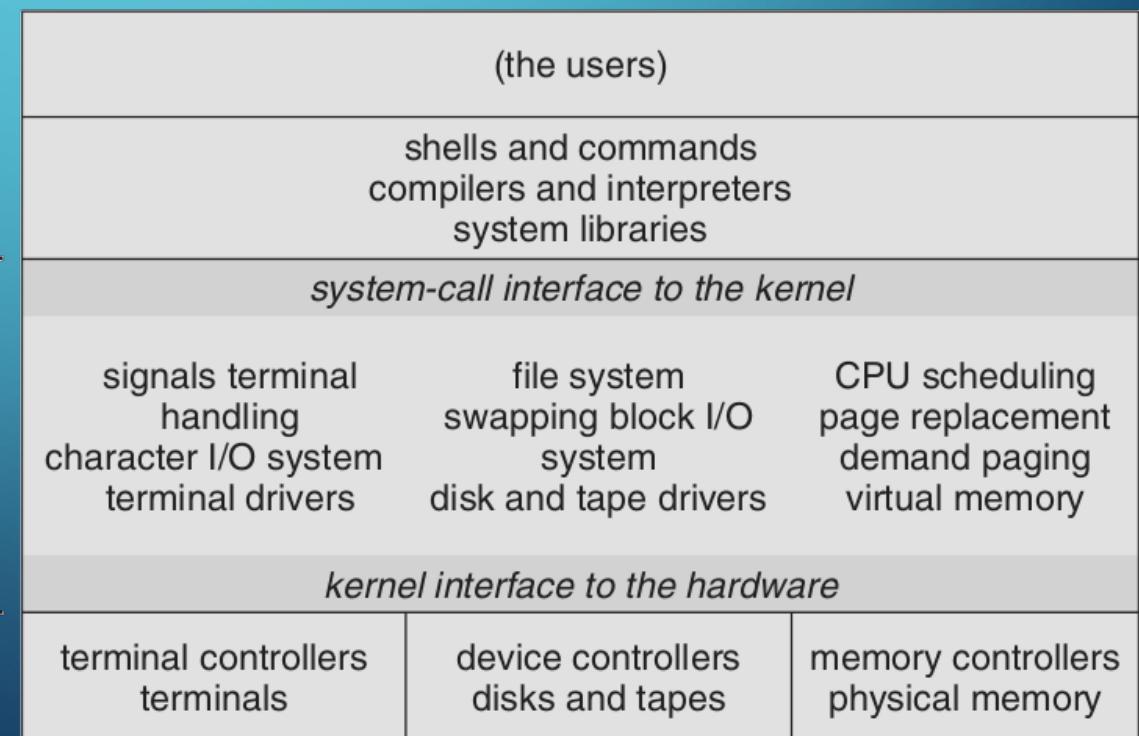
STRUTTURA DI UN SISTEMA OPERATIVO

- Alcune possibili strutture per un SO:
 - Monolitici
 - A livelli (o a strati)
 - Microkernel
 - A Moduli
 - Macchine virtuali
- categorie con intersezioni (sistemi ibridi);
- tassonomia non per forza completa o condivisa.

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)

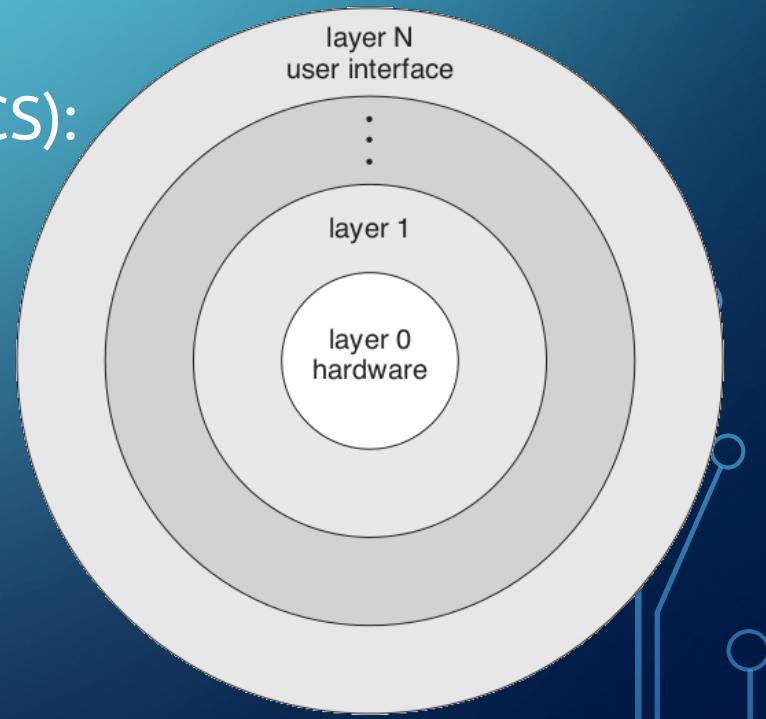
- Si utilizza una **gerarchia di livelli**;

- 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;

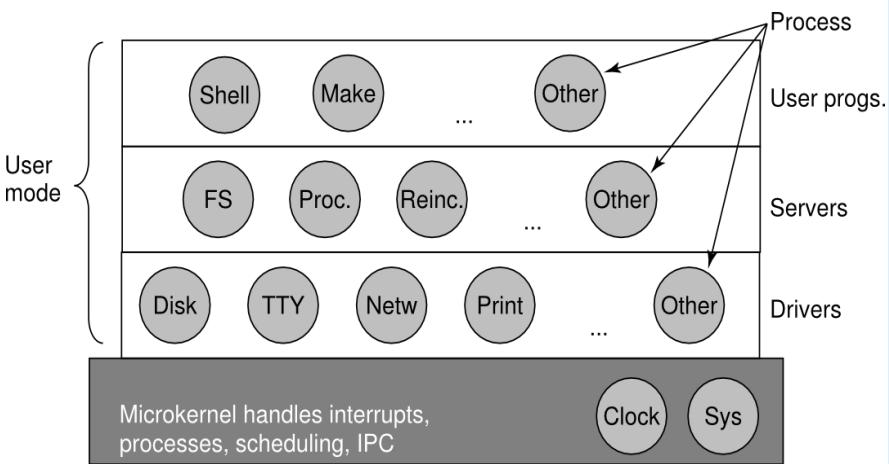
- variante ad **anelli concentrici** (MULTICS):

- **separazione forzata** dall'hardware;

- **problemi di prestazioni** dovuti alle chiamate nidificate e al relativo overhead.



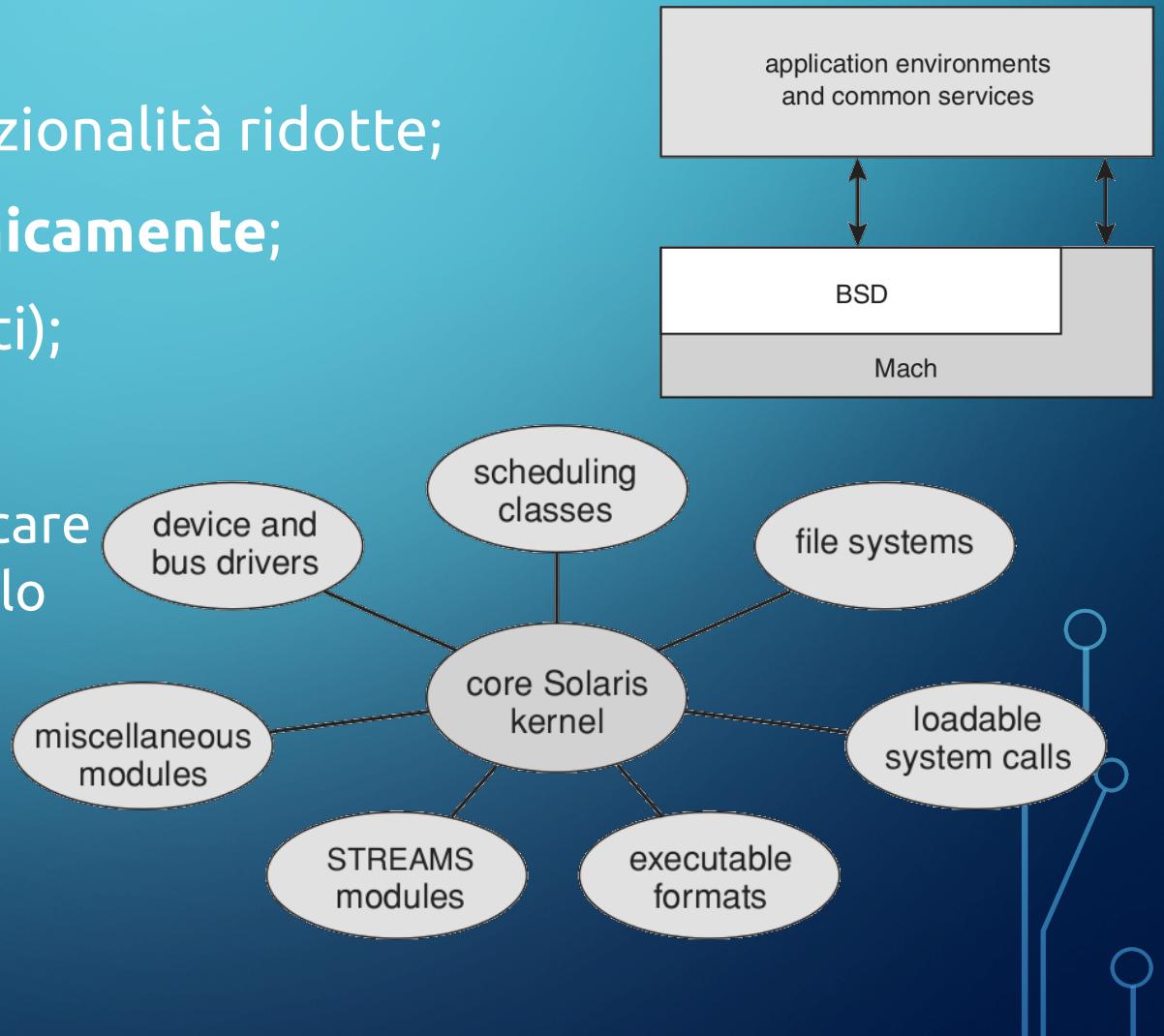
MICROKERNEL



- 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.**

