

# **DALLE ARCHITETTURE AI SISTEMI OPERATIVI**

**Danilo Croce**

Ottobre 2023



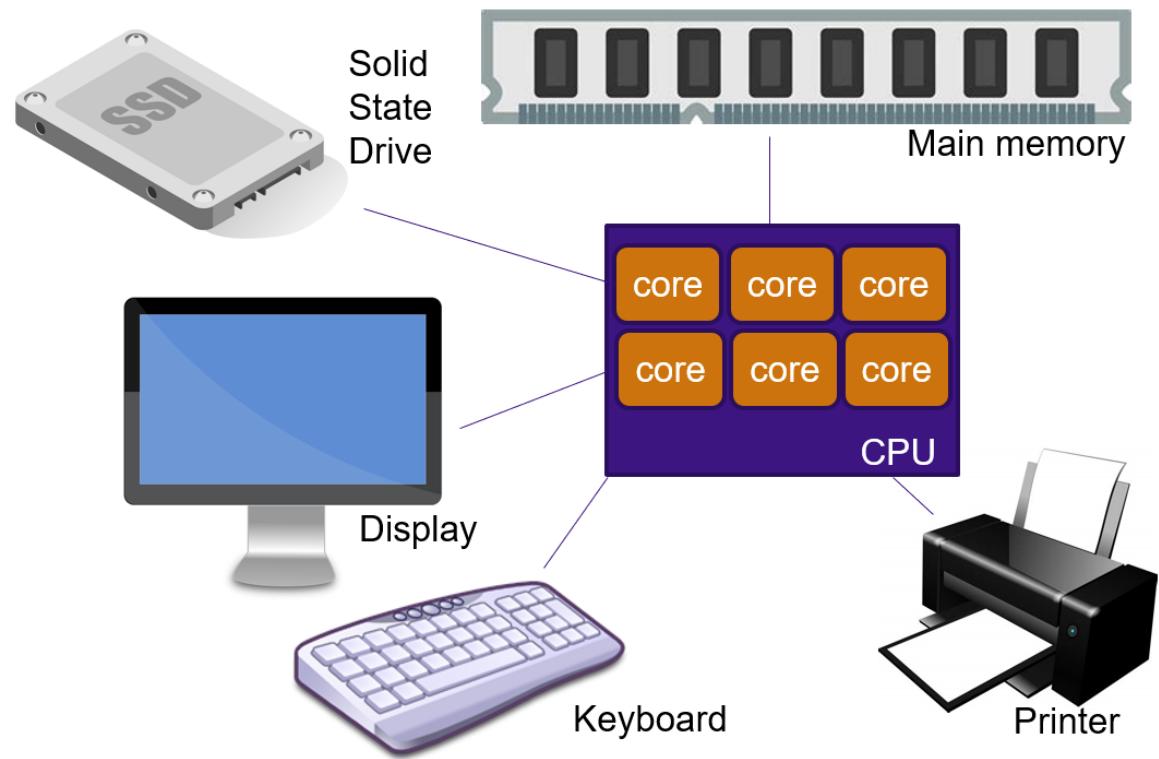
# COS'È UN SISTEMA OPERATIVO?

- 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**.



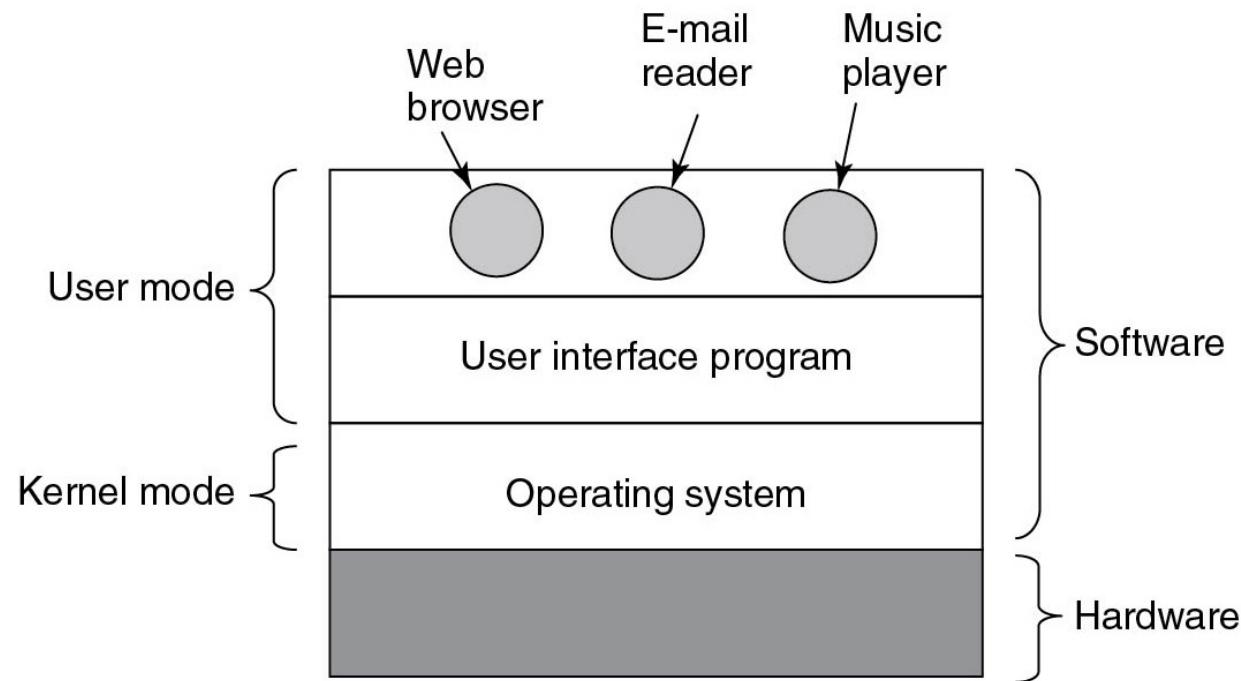
# COMPONENTI DI UN COMPUTER MODERNO (1)

- Uno o più processori
- Memoria principale
- Dischi o unità flash
- Stampanti
- Tastiera
- Mouse
- Display
- Interfacce di rete
- Dispositivi di I/O



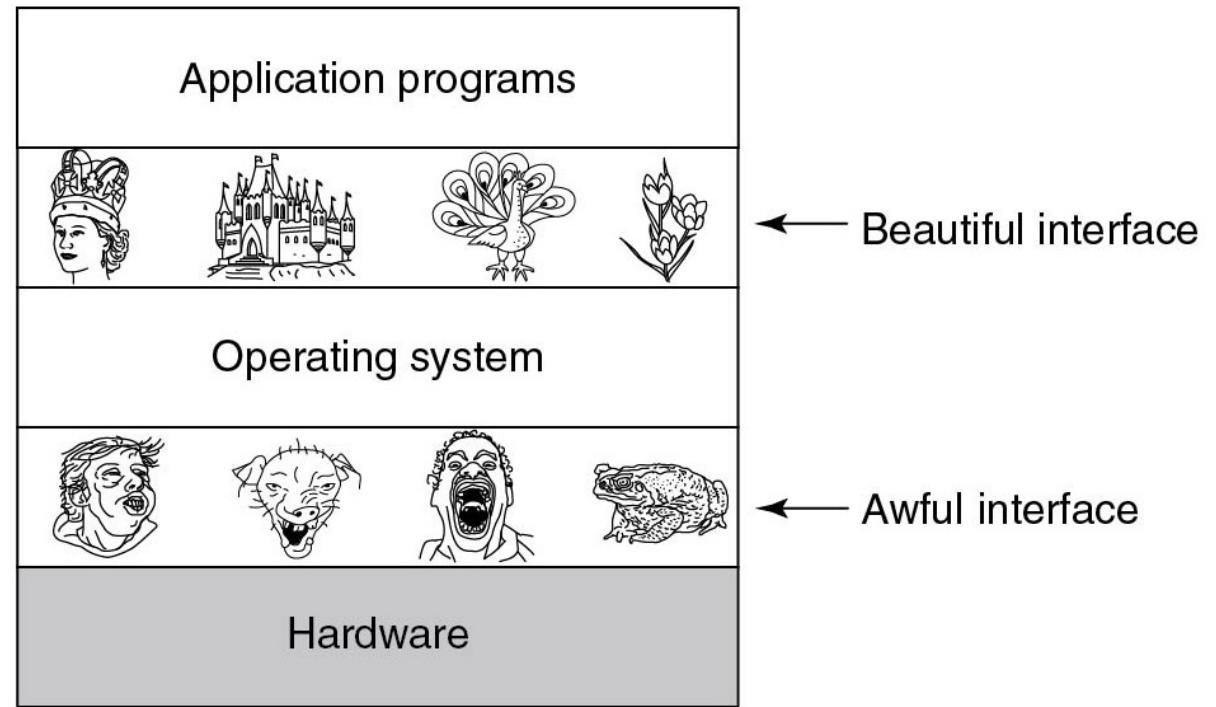
# COMPONENTI DI UN COMPUTER MODERNO (2)

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



# THE OPERATING SYSTEM AS AN EXTENDED MACHINE

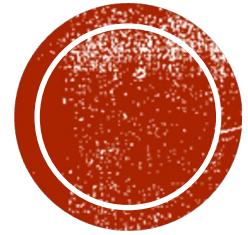
- Idea di **astrazione**
  - Il sistema operativo si pone tra l'Hardware e le Applicazioni
- Visione **top-down**
  - Il SO fornisce astrazioni ai programmi applicativi
- Vista **bottom-up**
  - Il SO gestisce parti di un sistema complesso
  - e fornisce un'allocazione ordinata e controllata delle risorse



# 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**:
  - nel **tempo**: CPU, stampante,...
  - nello **spazio**: memoria centrale, disco,...





# BREVE STORIA DEI SISTEMI OPERATIVI

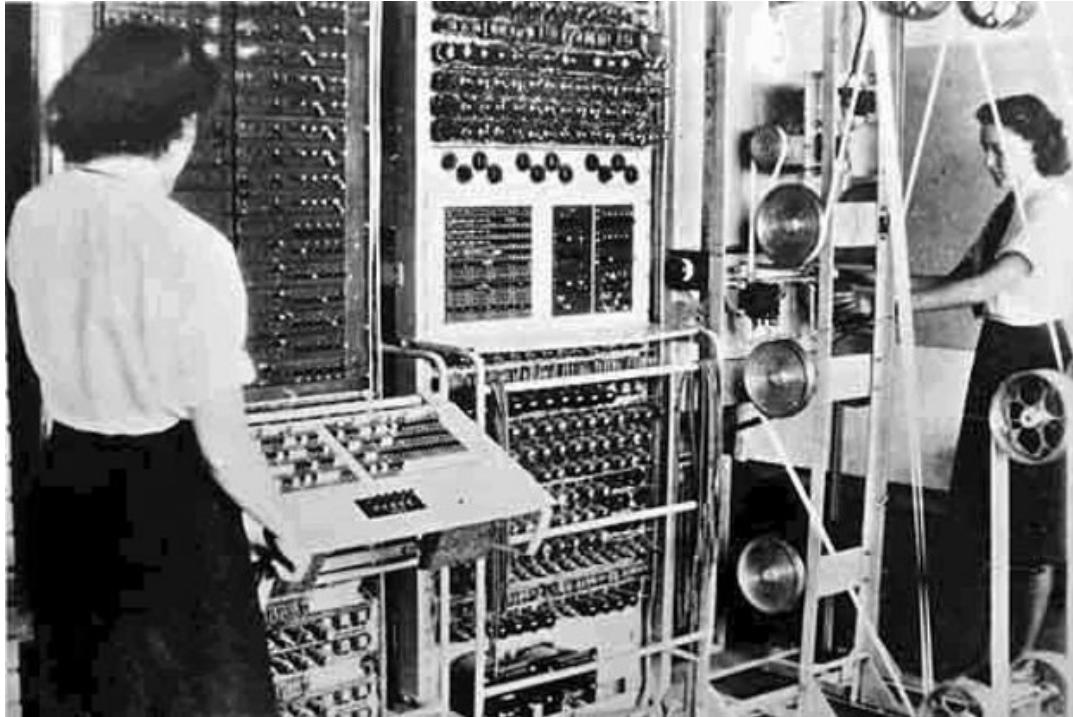


# GENERAZIONI DI SISTEMI OPERATIVI

- The first generation (1945-55): Vacuum tubes
- The second generation (1955-65): Transistors and batch systems
- The third generation (1965-1980): ICs and multiprogramming
- The fourth generation (1980-present): Personal computers
- The fifth generation (1990-present): Mobile computers



# FIRST GENERATION: VACUUM TUBES

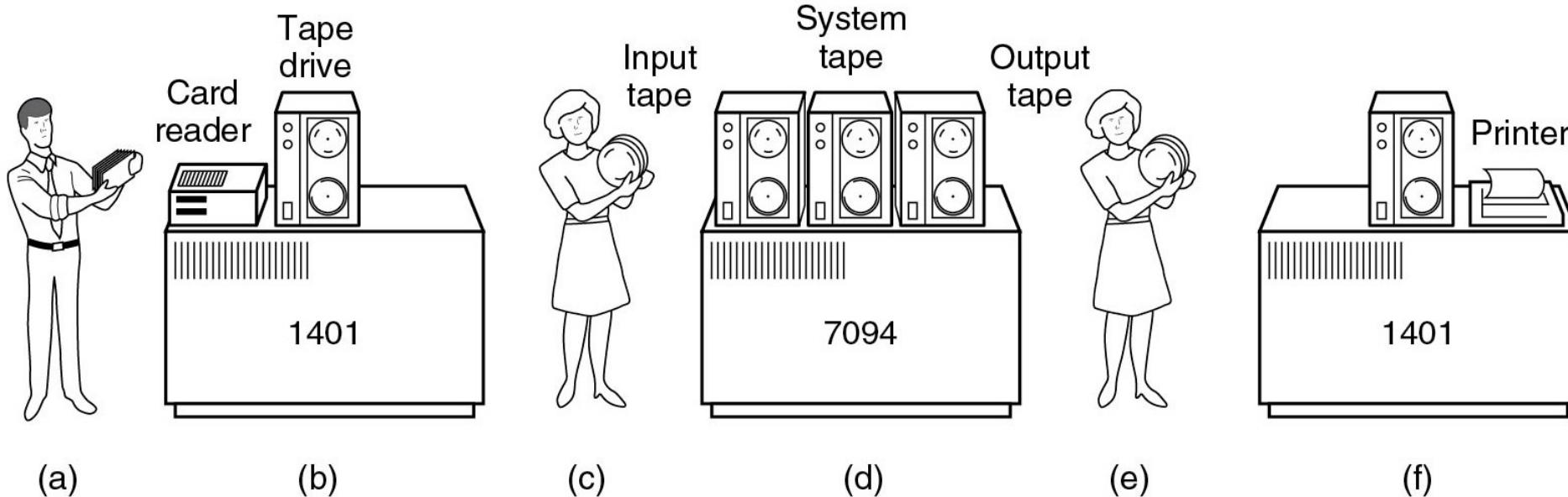


Annotated photographs of the COLOSSUS electronic digital computer/Kew/National Archives and Records Administration (N A R A) [F O 850/234]

**The Colossus Mark 2 computer: Cryptanalysis of the Lorenz Cipher**



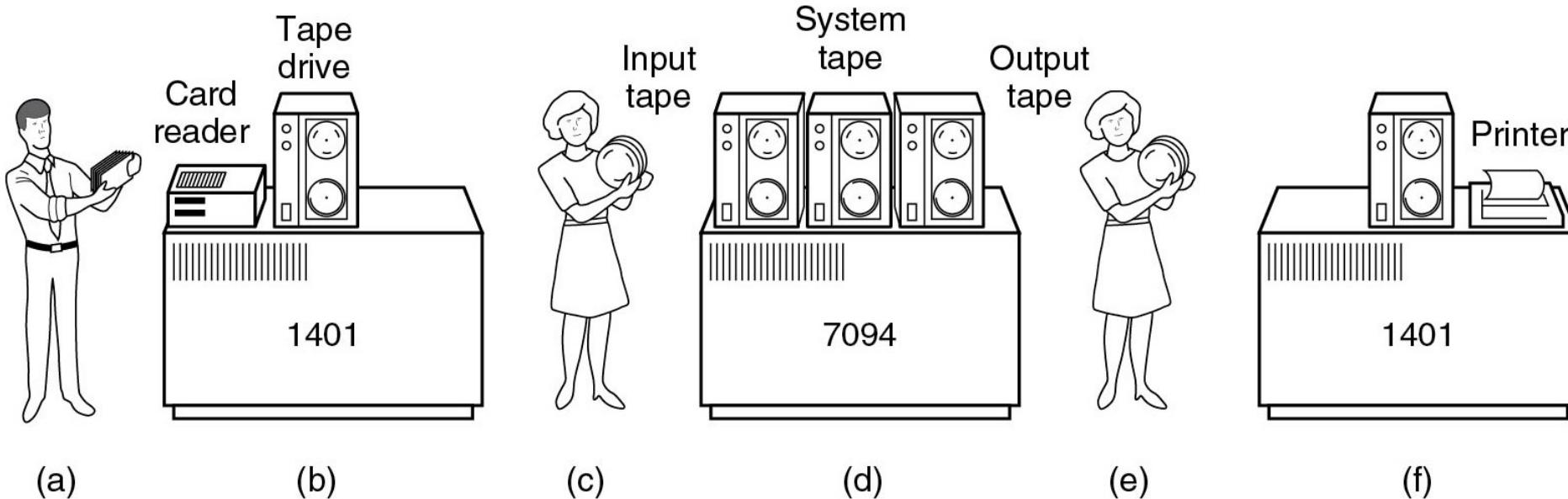
# SECOND GENERATION: TRANSISTORS AND BATCH SYSTEMS (1)



- Un primo sistema batch (manuale?)
  - (a) I programmatore portano le schede al 1401.
  - (b) Il 1401 legge un lotto di lavori su nastro...



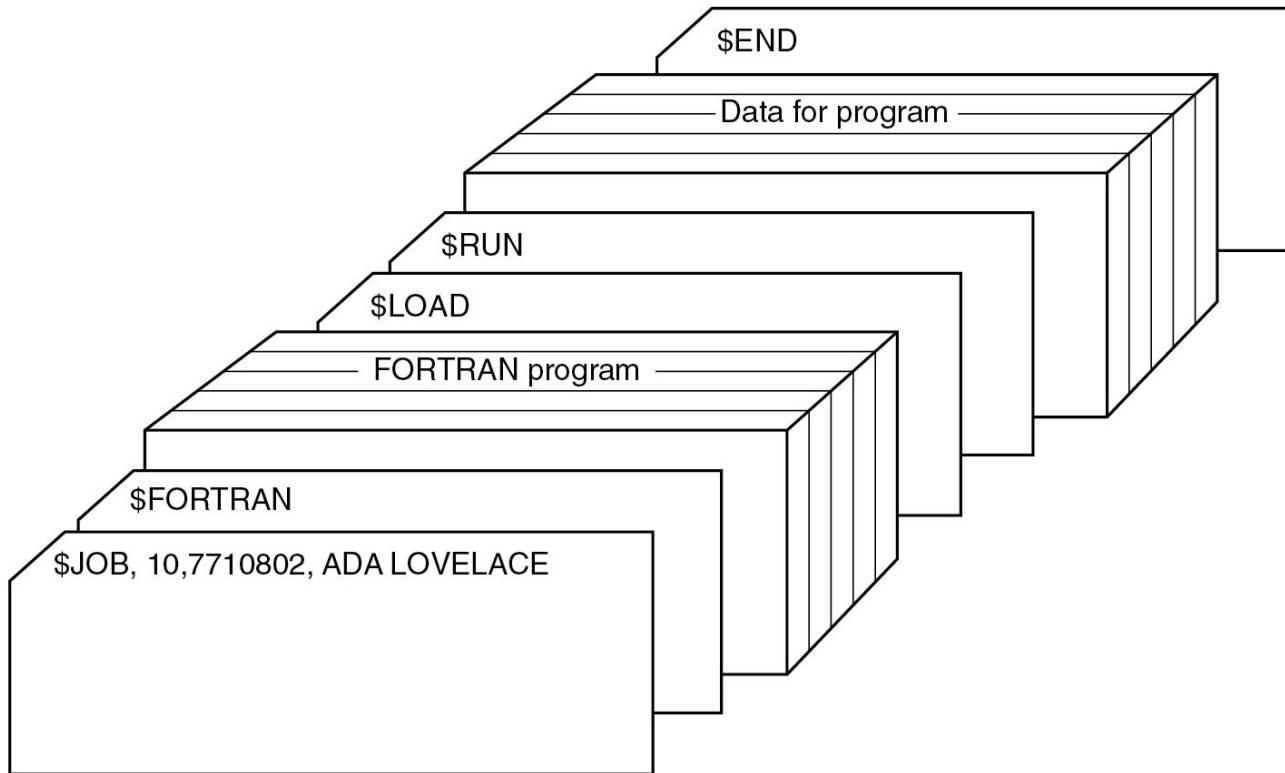
# SECOND GENERATION: TRANSISTORS AND BATCH SYSTEMS (2)



- (c) L'operatore porta il nastro di input al 7094.
- (d) Il 7094 esegue i calcoli.
- (e) L'operatore porta il nastro di output al 1401.
- (f) Il 1401 stampa l'output.

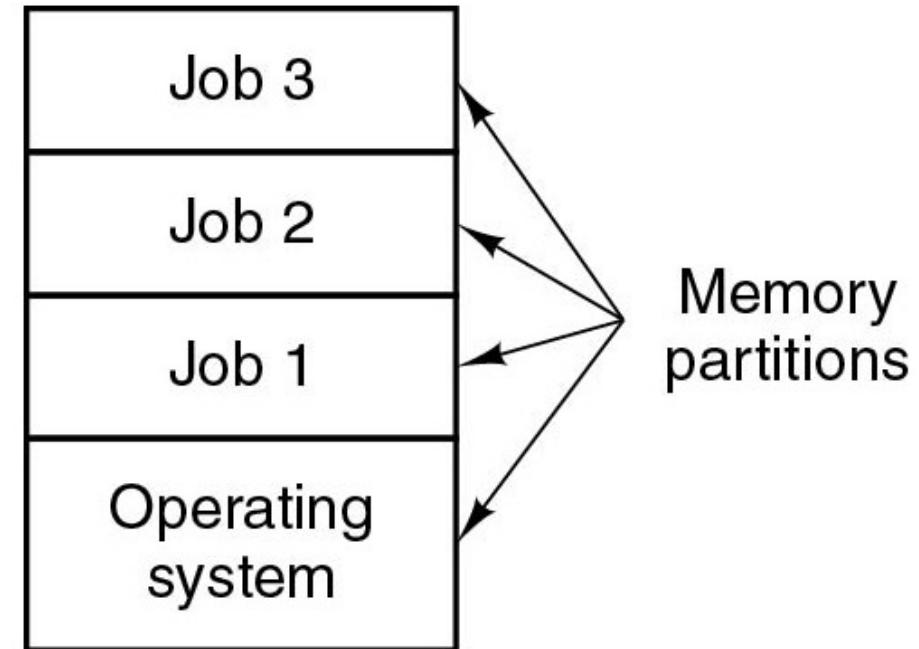


# SECOND GENERATION: TRANSISTORS AND BATCH SYSTEMS (3)



# THIRD GENERATION: ICS AND MULTIPROGRAMMING

- Problema delle architetture batch: CPU, inattiva in attesa di I/O.
  - Per calcoli scientifici, tempo sprecato insignificante.
  - Per dati commerciali, attesa I/O 80-90%.
- **Memoria partizionata per job multipli.**
  - **Spooling** (*simultaneous peripheral operation on line*) per caricare nuovi job senza interruzioni.
- **Problema:** Tempi lunghi tra l'inserimento di un job e l'output.
  - **Soluzione:** **Timesharing** per risposta rapida.
    - CPU assegnata a job utenti attivi.
    - Protezione hardware necessaria per vero *timesharing*.
- **MULTICS**, sistema per multiplexing e servizi informatici.
  - Non ebbe mai veramente successo ma i suoi principi permangono nel tempo



# TIMESHARING AND MULTICS

- “I remarked to Dennis that easily half the code I was writing in Multics was error recovery code”.

—Tom Van Vleck, 1973

- Multics (uno dei primi sistemi operative ad introdurre l’idea di Timesharing) introducesse molte caratteristiche innovative che oggi ritroviamo nei sistemi operative moderni.
  - Era purtroppo anche estremamente complesso.

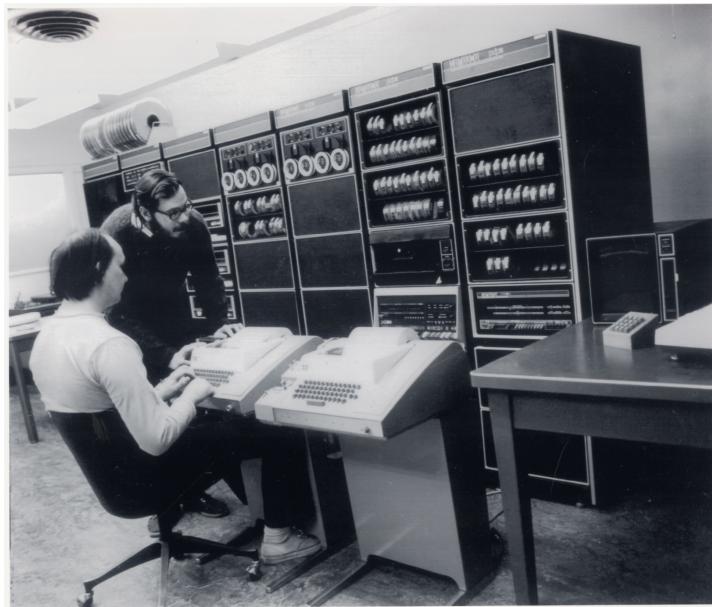


# UNIX – A SIMPLER OPERATING SYSTEM

- Unix è un sistema operativo **multiutente** e con **multiprogrammazione**.
- Storia
  - MULTICS (MULTIplexed Information and Computing Service, 1965).
  - Ken Thompson (Bell Laboratories -1969).
    - PDP-7, scritto in ASSEMBLER.
    - UNICS (UNiplexed Information and Computing Service).
  - Ken Thompson, Dennis Ritchie (1970-1974).
    - UNIX, PDP-11.
    - Ritchie sviluppa il linguaggio C (partendo dal linguaggio B).
    - La terza versione di UNIX è scritta in C.
    - Un articolo su UNIX viene pubblicato nel 1974 (ACM Turing Award 1984).



# UNIX – A SIMPLER OPERATING SYSTEM



- Ken Thompson crea versione monoutente di MULTICS su PDP-7.
- Questa versione evolve in sistema operativo UNIX®.
- UNIX® diventa popolare in ambito accademico e aziendale.
- Emergenza di diverse versioni UNIX, System V e BSD.
  - Sebbene inizialmente condiviso e libero, portò a molte versioni diverse e al CAOS
- IEEE sviluppa standard POSIX per garantire compatibilità inter-sistemi.

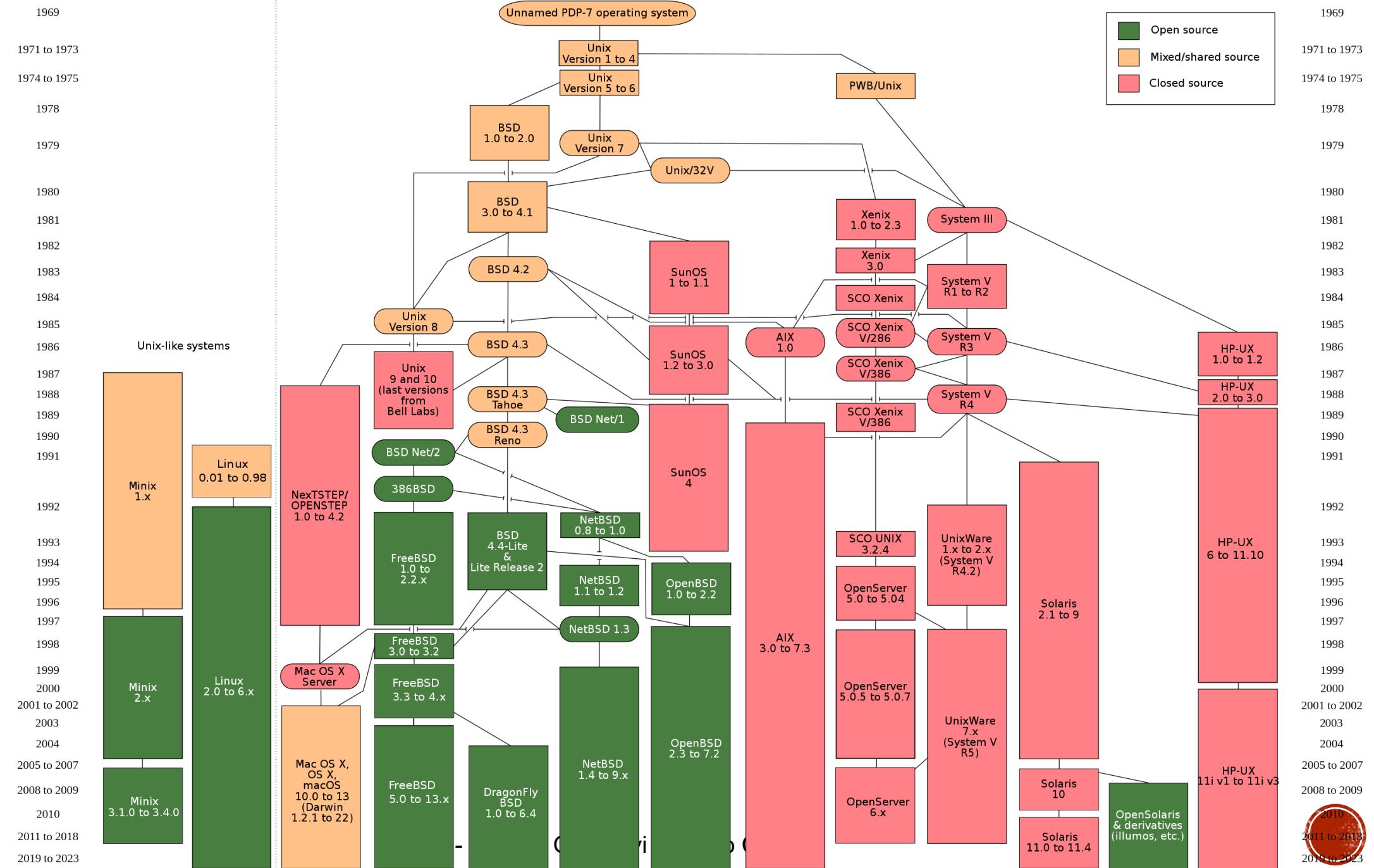


# I PROGETTI DI STANDARDIZZAZIONE

- Diversi progetti di standardizzazione mirano a consolidare i vari aspetti di UNIX con lo scopo di ottenere un'interfaccia standard per programmare in UNIX.
- I più importanti sono:
  - POSIX (Portable Operating System): fusione di System V e BSD (1984).
  - VIBM, DEC, Hewlett-Packard creano OSF (Open Software Foundation) e il loro sistema UNIX è OSF/1 (1988).
  - X/OPEN definisce la Single UNIX specification (1993) e il sistema relativo ha il marchio UNIX 95.
  - "Open group (fusione di Open Software Foundation e X/OPEN; <http://www.opengroup.com> 1996).
    - Definizione della seconda versione della Single UNIX specification (1997) col marchio UNIX 98.

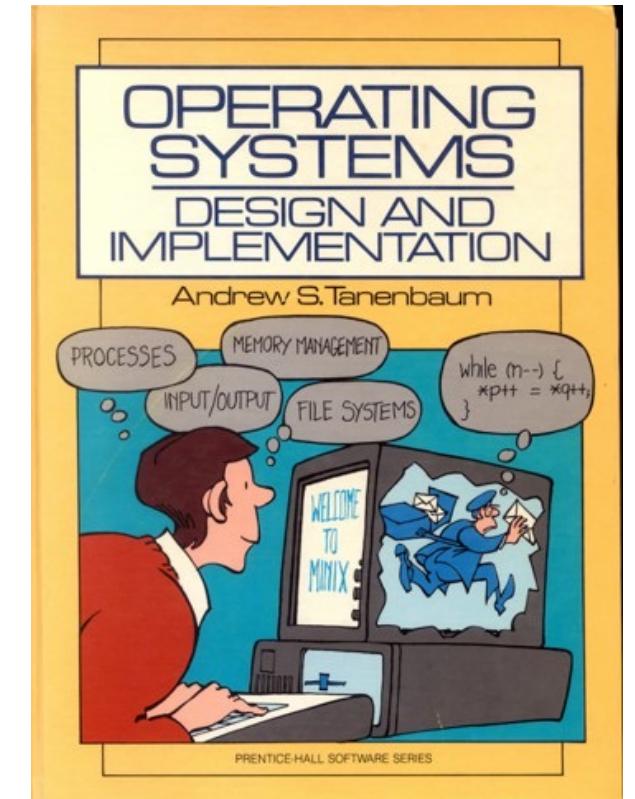


# A SUCCESS STORY



# UNA VARIANTE DEL SISTEMA UNIX

- Sebbene ci siano molte versioni di UNIX, le compagnie più importanti forniscono una versione basata su UNIX System V Release 4 (SVR4).
  - es. Solaris 2.x è l'implementazione di UNIX più diffusa e di più largo successo da un punto di vista commerciale.
- Questi sistemi sono molto grossi e complicati (al contrario dell'idea originaria di Thompson) e in alcuni casi anche costosi.
- Così, Tanenbaum sviluppa MINIX (1987), un piccolo sistema UNIX (11800 righe di codice C e 800 righe di codice Assembler) compatibile con gli standard POSIX.
  - MINIX è un sistema sviluppato a scopo didattico basato sul modello a micro-kernel ([www.cs.vu.nl/~ast/minix.html](http://www.cs.vu.nl/~ast/minix.html)).
  - Ancora usato nei processori moderni



# DA MINIX A LINUX (1990S)

*Hello everybody out there using minix -*

*I'm doing a (free) operating system (just a hobby, won't be big and professional like gnu) for 386(486) A T clones. This has been brewing since april, and is starting to get ready. I'd like any feedback on things people like/dislike in minix, as my OS resembles it somewhat (same physical layout of the file-system (due to practical reasons) among other things).*

*I've currently ported bash(1.08) and g c c(1.40), and things seem to work. This implies that I'll get something practical within a few months, and I'd like to know what features most people would want. Any suggestions are welcome, but I won't promise I'll implement them :-)*

*Linus (torvalds@kruuna.helsinki.fi)*

*P.S. Yes - it's free of any minix code, and it has a multi-threaded fs. It is NOT portable (uses 386 task switching etc), and it probably never will support anything other than A T-harddisks, as that's all I have :-(*



# E NEL FRATTEMPO MICROSOFT E APPLE?

- Anni 70: Sviluppo dei circuiti LSI e nascita dei personal computer.
  - (Large Scale Integration, integrazione su larga scala), chip contenenti migliaia di transistor su un centimetro quadrato di silicio
- Intel introduce l'8080, necessità di un sistema operativo.
  - Gary Kildall crea CP/M e fondò Digital Research.
  - Digital Research adatta CP/M per microcomputer (o PC anti-litteram)
- Anni 80: IBM cerca sistema operativo per il PC, incontra Digital Research.
  - Kildall rifiuta incontro con IBM, opportunità mancata.
- Microsoft acquisisce DOS da Seattle Computer Products.
  - Nasce MS-DOS e domina il mercato dei PC IBM.



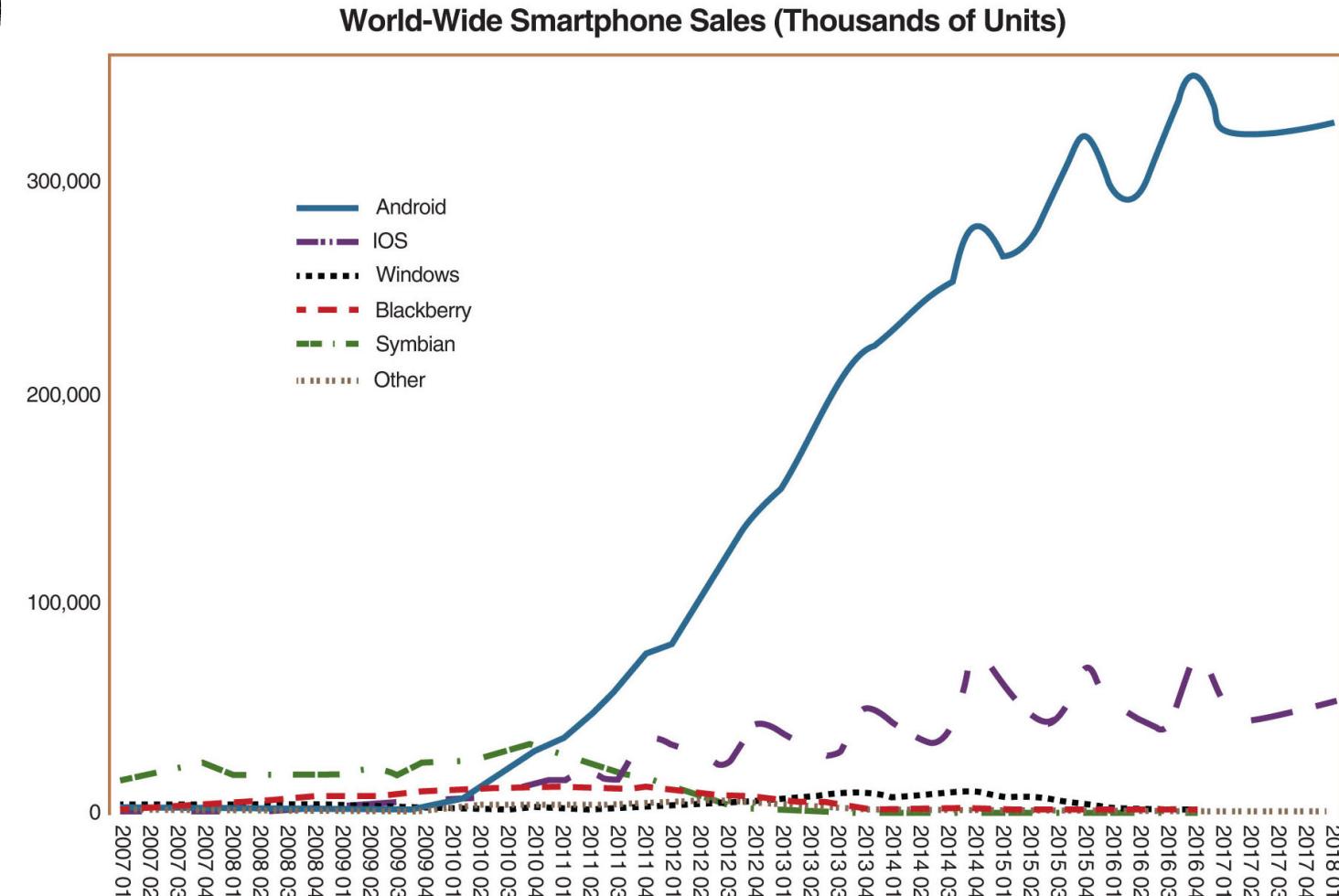
# E NEL FRATTEMPO MICROSOFT E APPLE? (2)

- Anni 80/90:
  - GUI introdotta da Doug Engelbart e adottata da Xerox PARC.
  - Steve Jobs ispirato da GUI, sviluppa Apple Macintosh.
  - Microsoft crea Windows come ambiente grafico su MS-DOS.
- Evoluzioni:
  - Apple consolida il suo sistema operativo basato su UNIX ma *architecture-specific*
  - Windows NT influenzato da VAX VMS e sfida legale con DEC.
  - Windows 2000 e successori consolidano la posizione di Microsoft.
  - Vista e Windows 8 incontrano resistenza, successo con Windows 10.



# DA ALLORA, LINUX (E ANDROID) HANNO PROSPERATO

- Soprattutto in ambienti
  - Server
  - Mobile



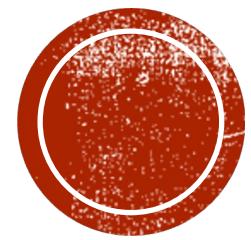
# MA ANCHE MINIX

This is Google's cache of [ftp://ftp.supermicro.com/CDR-X11-UP\\_1.10\\_for\\_Intel\\_X11\\_UP\\_platform/Intel/ME/Other\\_Licenses/Minix3\\_License.txt](ftp://ftp.supermicro.com/CDR-X11-UP_1.10_for_Intel_X11_UP_platform/Intel/ME/Other_Licenses/Minix3_License.txt). It is a snapshot of the page as it appeared on Mar 23, 2017 15:18:02 GMT.

License Copyright (c) 1987, 1997, 2006, Vrije Universiteit, Amsterdam, The Netherlands  
All rights reserved. Redistribution and use of the MINIX 3 operating system in source and  
binary forms, with or without modification, are permitted provided that the following  
conditions are met:  
\* Redistributions of source code must retain the above copyright  
notice, this list of conditions and the following disclaimer.  
\* Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following  
disclaimer in the documentation and/or other materials provided with the distribution.  
\* Neither the name of the Vrije Universiteit nor the names of the software authors or  
contributors may be used to endorse or promote products derived from this software

- MINIX 3 è stato adottato da Intel per il suo motore di gestione ed è ora presente in desktop, server e laptop.



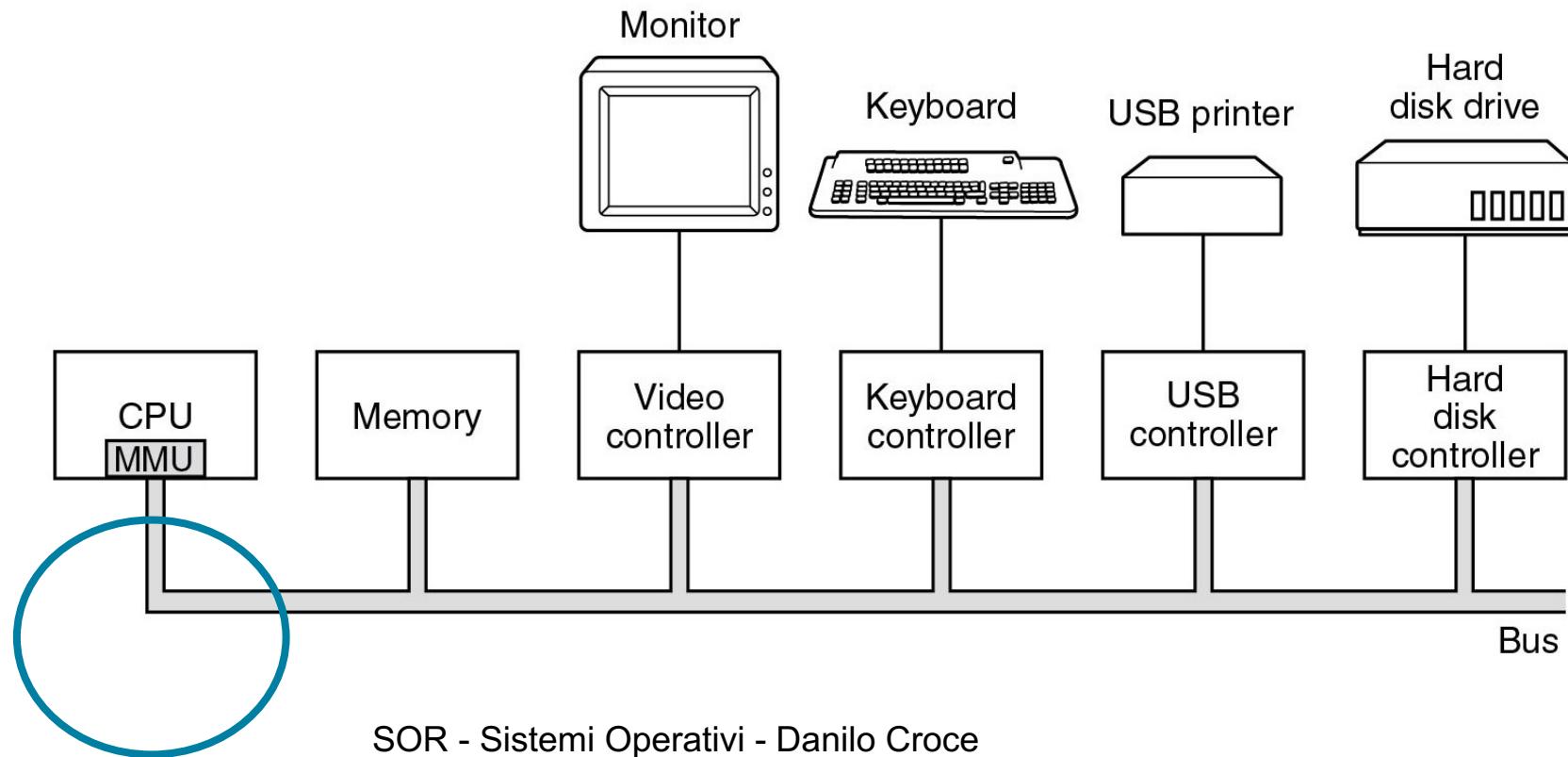


# UNO SGUARDO ALL'hardware



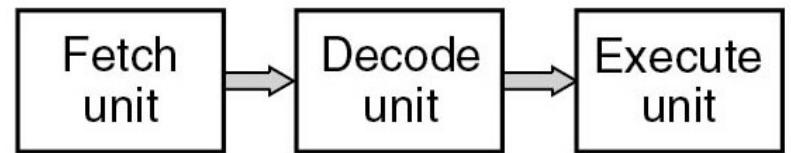
# UNO SGUARDO DI INSIEME

- Architettura (semplificata) di un calcolatore



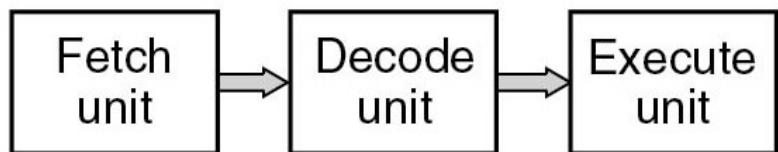
# IL PROCESSORE

- La CPU è il cervello del computer, esegue istruzioni dalla memoria.
  - Il ciclo base della CPU: **preleva** (*fetch*), **decodifica** (*decode*), **esegue** (*execute*) istruzioni.
  - I programmi vengono eseguiti in questo ciclo.
- Le CPU eseguono un set specifico di istruzioni.
  - Registri interni memorizzano dati importanti e risultati.
  - I set di istruzioni includono funzione per il caricamento/salvataggio dati dalla memoria.
- Registri speciali come il **program counter** indicano l'istruzione successiva.
- Lo **stack pointer** punta alla cima dello stack in memoria.
  - Lo stack contiene frame di procedure con parametri e variabili locali.
- Il **registro Program Status Word (PSW)** contiene informazioni sullo stato del programma.
  - Il PSW è fondamentale per chiamate di sistema e I/O.

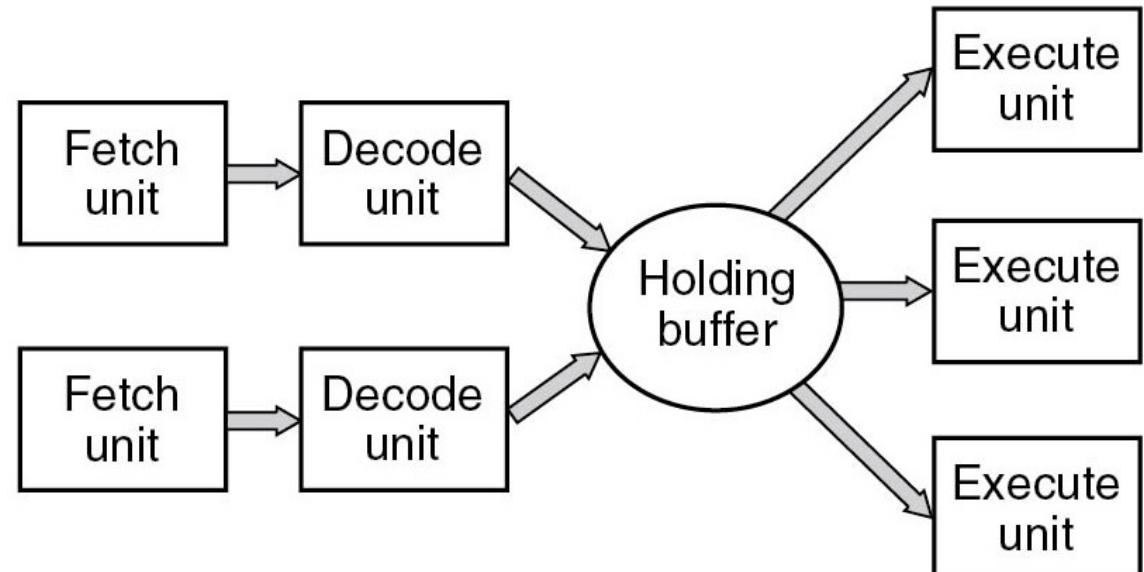


# IL PROCESSORE

- Il sistema operativo gestisce il *multiplexing* temporale della CPU.
  - Durante il *multiplexing*, il sistema operativo salva e ripristina i registri.
  - Ciò permette al sistema operativo di eseguire programmi in modo efficiente.
- Progettazioni avanzate: **pipeline**
  - non del tutto trasparenti al SO



(a)

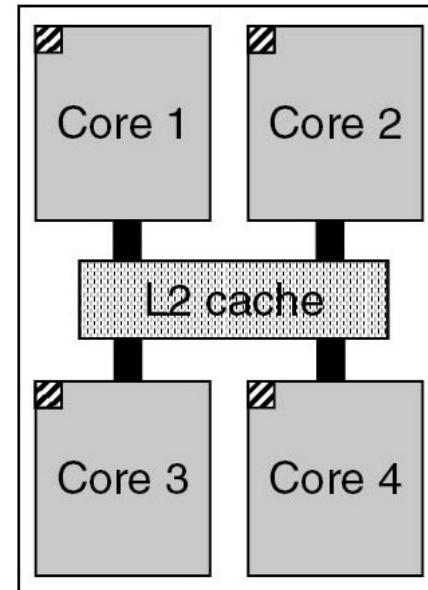


(b)

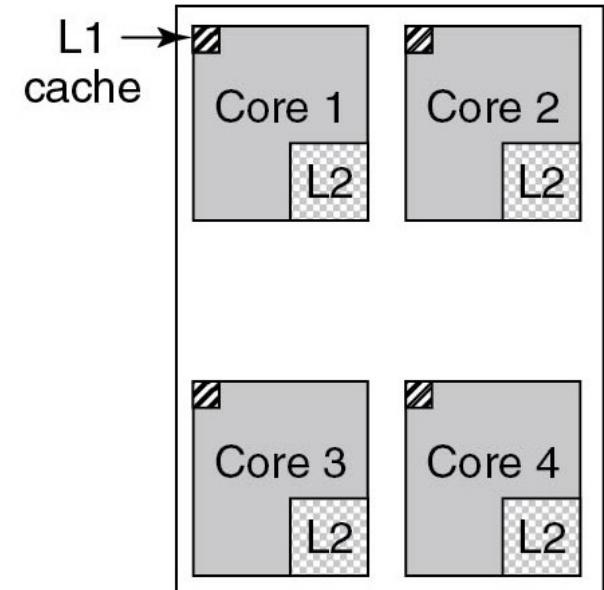


# PIÙ DI UN PROCESSORE

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



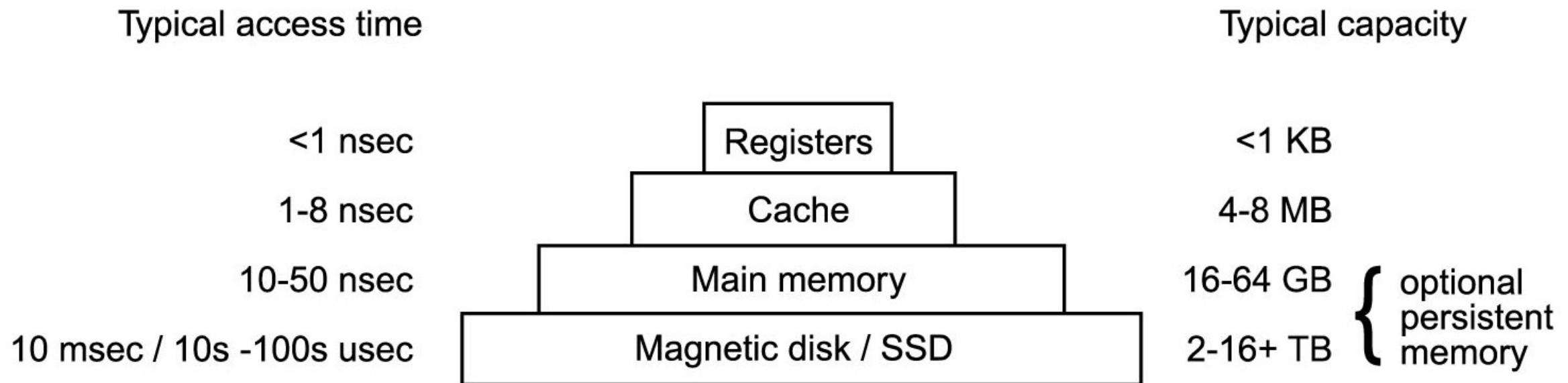
(a)



(b)



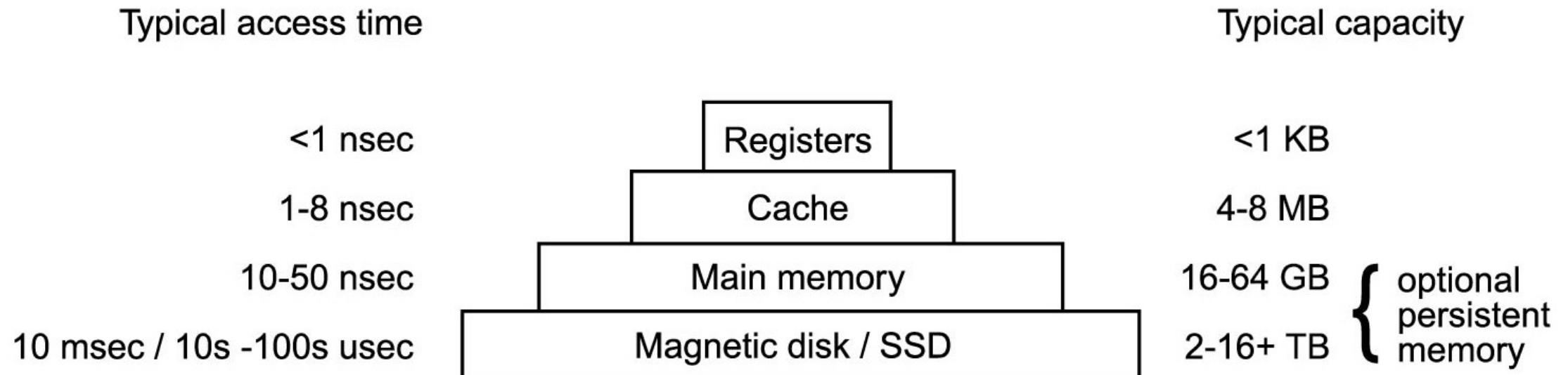
# LA MEMORIA DI UN CALCOLATORE



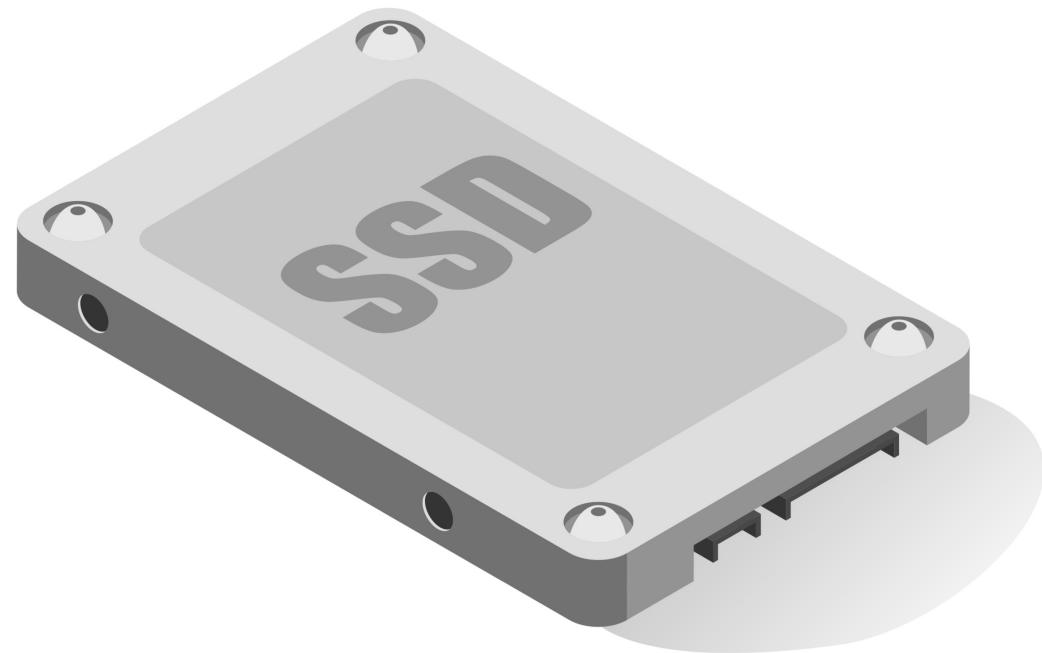
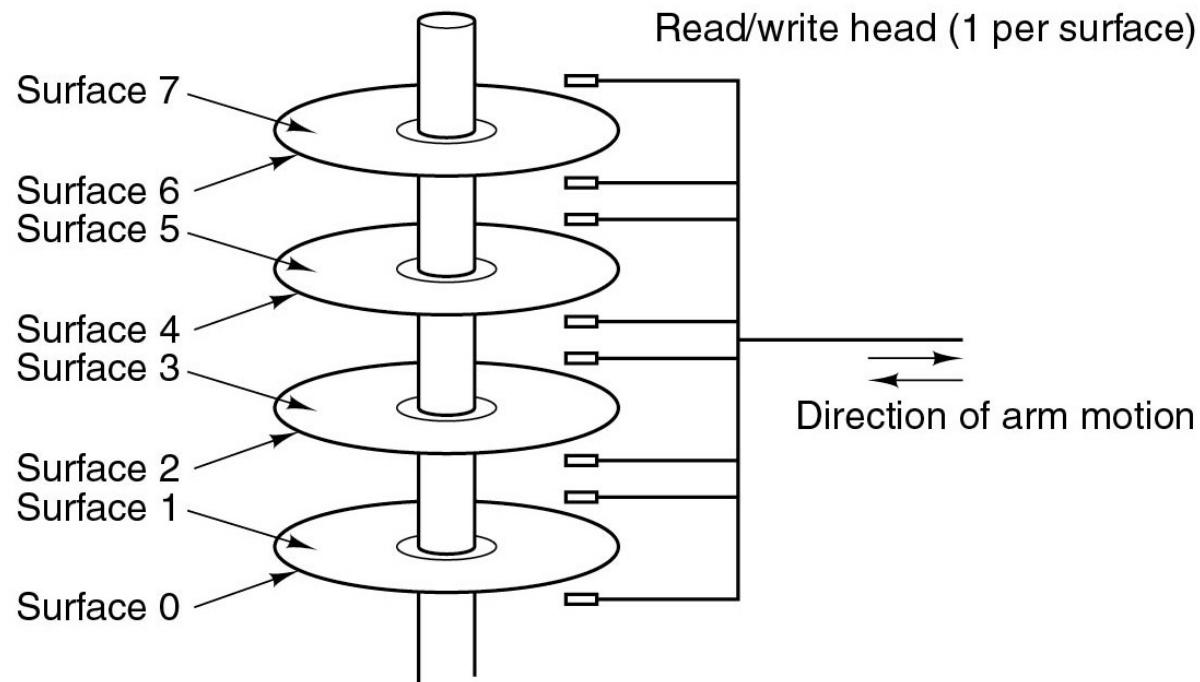
# LA MEMORIA DI UN CALCOLATORE (2)

- Problemi del sistema di cache:

- Quando inserire un nuovo elemento nella cache?
- In quale riga della cache inserire il nuovo elemento?
- Quale elemento rimuovere dalla cache quando è necessario uno slot.
- Dove mettere un elemento appena eliminato nella memoria più grande.



# NONVOLATILE STORAGE



Spesso (erroneamente) indicato anche come disco.  
Nessuna parte in movimento, dati in memoria elettronica  
(flash). Molto più veloce dei dischi magnetici.



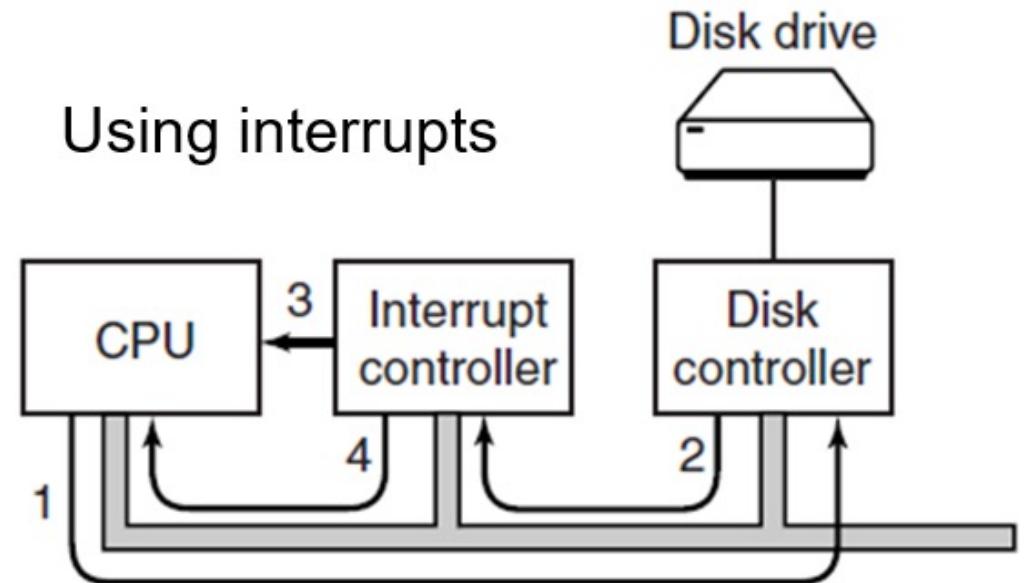
# DISPOSITIVI DI I/O

- Si individuano due componenti:
  - il **controller**: più semplice da usare per il SO;
  - il **dispositivo** in sé: interfaccia elementare ma complicata da pilotare.
    - esempio: dischi SATA.
- Ogni controller ha bisogno di un *driver* per il S.O.
- Il **driver interagisce con il controller** attraverso le porte di I/O:
  - istruzioni tipo IN / OUT;
  - mappatura in memoria.



# DISPOSITIVI DI I/O (2)

- Per eseguire l'I/O:
  - Il processo esegue la chiamata di sistema
  - Il kernel effettua una chiamata al driver
  - Il driver avvia l'I/O
- e interroga il dispositivo per vedere se ha finito (è occupato in attesa)
- o chiede al dispositivo di generare un interrupt quando ha finito (e restituisce qualche informazion)
- più avanzato: fa uso di hardware speciale
  - DMA



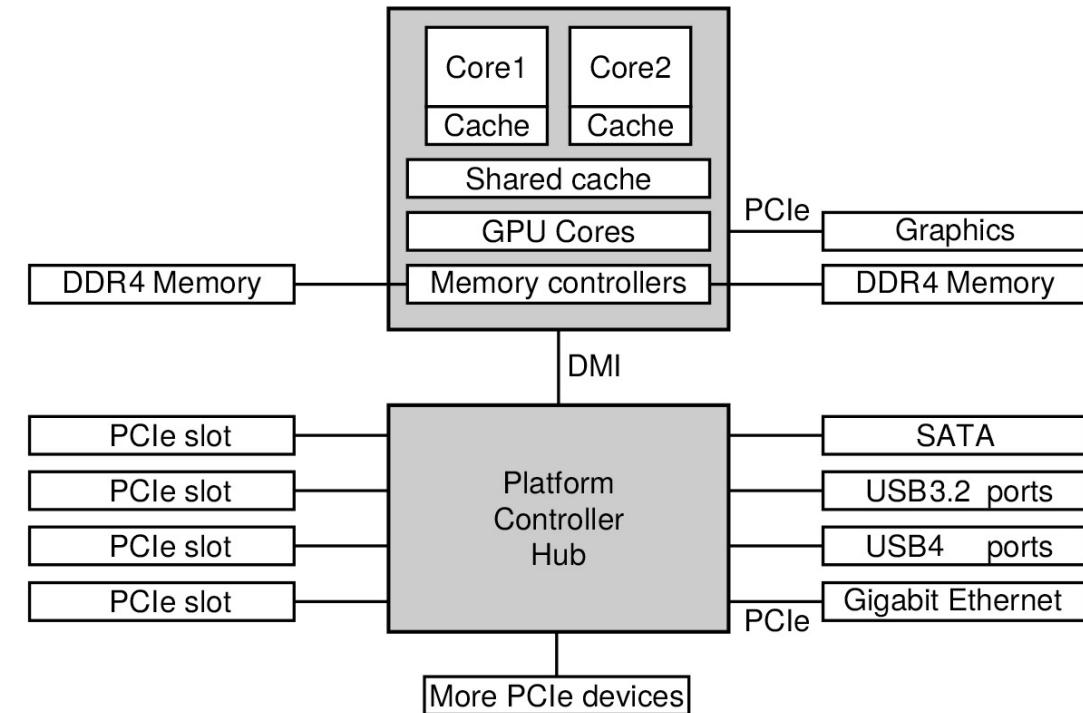
# IL DMA

- DMA è l'acronimo di "*Direct Memory Access*" ed è un dispositivo hardware speciale.
- Consente ai componenti di accedere direttamente alla memoria del computer senza coinvolgere la CPU.
- Migliora l'efficienza ed aumenta le prestazioni nelle operazioni di input/output (I/O) ad alta velocità.
- Utilizzato per il trasferimento veloce di dati tra memoria e dispositivi periferici.
- Riduce il carico sulla CPU durante le operazioni di I/O, consentendole di concentrarsi su altri compiti critici.



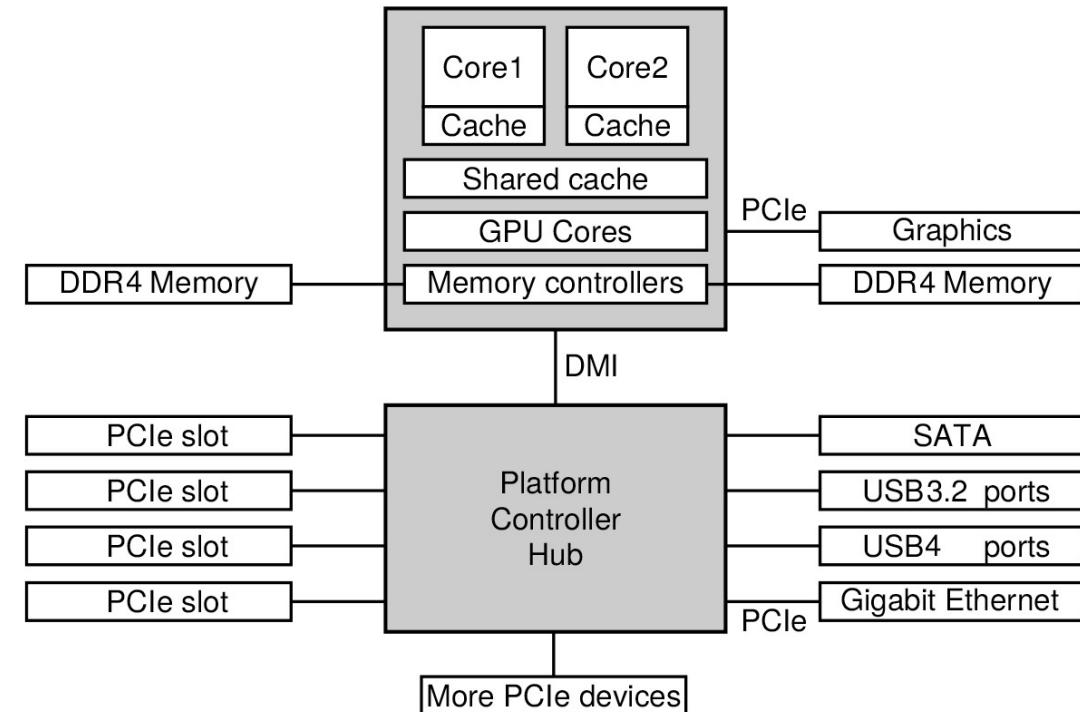
# BUSES

- L'evoluzione dei computer ha portato all'aggiunta di bus supplementari per gestire l'aumento del traffico dati.
  - Un sistema x86 moderno ha diversi bus con funzioni e velocità di trasferimento diverse.
  - **la CPU comunica con la memoria attraverso un bus veloce DDR4, con una periferica grafica esterna sul bus PCIe (Peripheral Component Interconnect Express)**
  - con tutti gli altri dispositivi attraverso un hub su un bus DMI (Direct Media Interface).
- Il bus PCIe è il principale e più veloce bus di comunicazione nei computer attuali.
  - **PCIe utilizza un'architettura a connessioni punto a punto dedicate**, migliorando l'efficienza rispetto ai bus condivisi.



# BUSES

- Dispositivi *legacy* possono essere collegati a un processore hub separato.
- USB (Universal Serial Bus) è stato sviluppato per connettere dispositivi lenti al computer, ma le versioni moderne possono raggiungere velocità fino a 40 Gbps.
- USB utilizza un connettore a 4-11 conduttori per alimentazione e comunicazione.
- USB consente il collegamento immediato dei dispositivi senza necessità di riavvio del sistema.



# AVVIO DEL SISTEMA (BOOT)

- La memoria flash della scheda madre contiene il firmware (il BIOS).
- Dopo aver premuto il pulsante di accensione, la CPU esegue il BIOS che
  - Inizializza RAM e altre risorse
  - Esegue la scansione dei bus PCI/PCIe e inizializza i dispositivi
  - Imposta il firmware *runtime* per i servizi critici (ad esempio, I/O a basso livello) che il sistema deve utilizzare dopo l'avvio.
- Il BIOS cerca la posizione della tabella delle partizioni sul secondo settore del dispositivo di avvio
  - Contiene le posizioni di altre partizioni
- Il BIOS è in grado di leggere semplici file system (ad esempio, FAT-32) e avvia il primo programma di *bootloader* (dalla partizione indicata dal boot manager).
  - Il *bootloader* può caricare altri programmi di *bootloading*.
- Alla fine viene caricato l'OS

