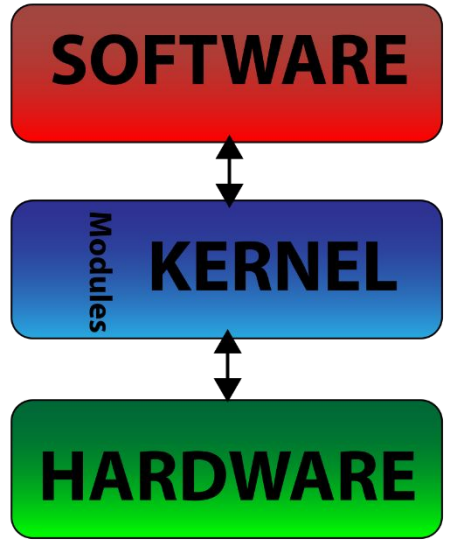
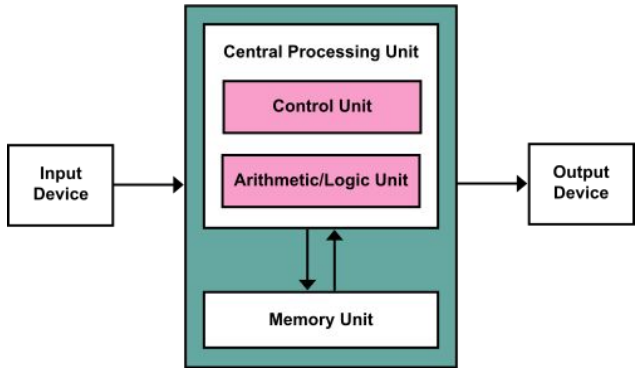
UNIVERSITÀ DEGLI STUDI DELLA BASILICATA

Corso di Sistemi Operativi

Introduzione



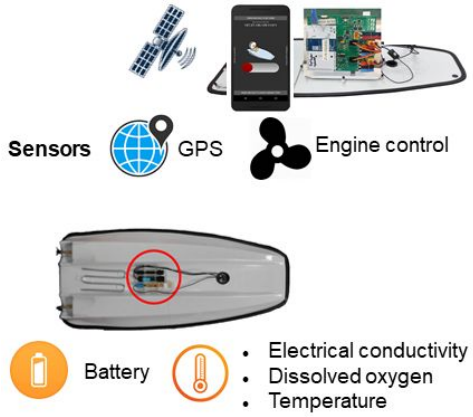
Docente:
Domenico Daniele Bloisi

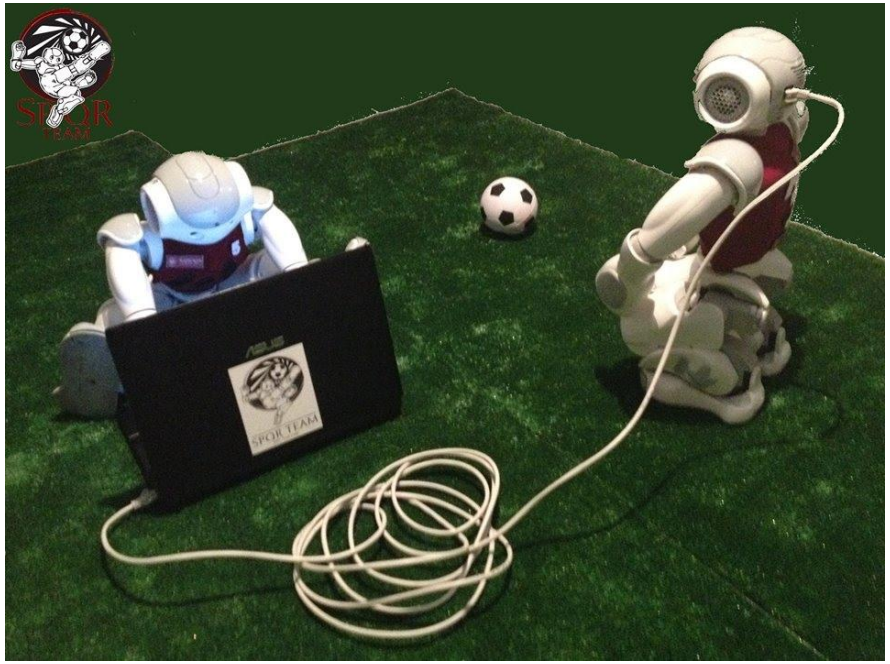


Domenico Daniele Bloisi

- Professore Associato
  Dipartimento di Matematica, Informatica ed Economia
  Università degli studi della Basilicata
  http://web.unibas.it/bloisi

- SPQR Robot Soccer Team
  Dipartimento di Informatica, Automatica e Gestionale Università degli studi di Roma “La Sapienza”
  http://spqr.diag.uniroma1.it



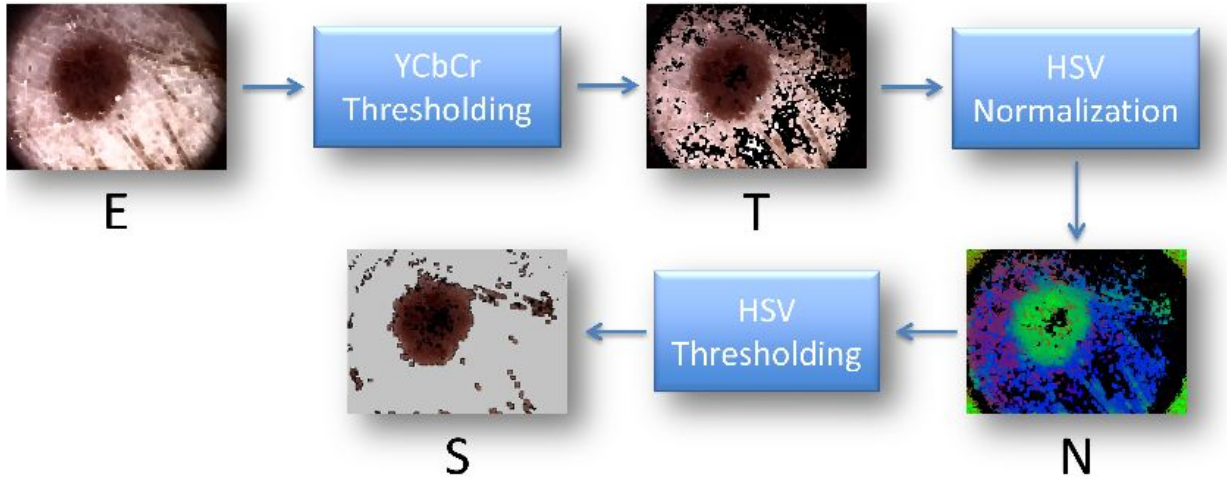


Interessi di ricerca

- Intelligent surveillance
- Robot vision
- Medical image analysis

https://youtu.be/2KHNZX7UIWQ

https://youtu.be/9a70Ucgbi_U



UNIBAS Wolves https://sites.google.com/unibas.it/wolves

- UNIBAS WOLVES is the robot soccer team of the University of Basilicata. Established in 2019, it is focussed on developing software for NAO soccer robots participating in RoboCup competitions.

- UNIBAS WOLVES team is twinned with SPQR Team at Sapienza University of Rome

https://youtu.be/ji0OmkaWh20

Informazioni sul corso

- Home page del corso:
  http://web.unibas.it/bloisi/corsi/sistemi-operativi.html
- Docente: Domenico Daniele Bloisi
- Periodo: I semestre ottobre 2022 – gennaio 2023

  - Lunedì dalle 15:00 alle 17:00 (Aula Leonardo)
  - Martedì dalle 08:30 alle 10:30 (Aula 1)

Ricevimento

- In presenza, durante il periodo delle lezioni:
  Lunedì dalle 17:00 alle 18:00 Edificio 3D, II piano, stanza 15
  Si invitano gli studenti a controllare regolarmente la bacheca degli avvisi per eventuali variazioni
- Tramite google Meet e al di fuori del periodo delle lezioni:
  da concordare con il docente tramite email

Per prenotare un appuntamento inviare una email a
domenico.bloisi@unibas.it

Argomenti del corso

Gli argomenti trattati nel corso riguardano i concetti di base dei sistemi operativi, con particolare riferimento alle tematiche legate alla gestione dei processi e delle memorie nei calcolatori.

Programma – Sistemi Operativi

- Introduzione ai sistemi operativi
- Gestione dei processi
- Sincronizzazione dei processi
- Gestione della memoria centrale
- Gestione della memoria di massa
- File system
- Sicurezza e protezione

Materiale Didattico

- Libro di testo:

  A. Silberschatz, G. Gagne, P.B. Galvin
  "Sistemi operativi. Concetti ed esempi"
  10ª Edizione. Pearson

- Slide delle lezioni:

  disponibili sul sito del corso nella sezione "diario"
  http://web.unibas.it/bloisi/corsi/sistemi-operativi.html#diario

Obiettivi del corso

Il corso intende fornire agli studenti:

- La descrizione delle componenti principali di un moderno sistema operativo
- La capacità di distinguere tra le diverse modalità di gestione processi implementabili in un moderno sistema operativo
- La capacità di identificare le metodologie di sincronizzazione dei processi
- La capacità di valutare le migliori soluzioni per la gestione della memorie nei moderni sistemi operativi

Esame

- Il voto finale viene conseguito svolgendo un esame scritto con 3 domande a risposta aperta e 2 esercizi.

- Gli studenti possono chiedere al docente di svolgere un progetto facoltativo per ottenere un punteggio bonus (fino a tre punti) che verrà sommato al voto ottenuto durante l'esame scritto.

Esempio di esame

Le informazioni sulle date degli appelli e i testi degli esami passati sono disponibili nella sezione "appelli" del sito del corso

Tempo a disposizione: 2 ore e 30 minuti

Domanda 1 (max 5 punti)

Spiegare come avviene la creazione di un processo attraverso la chiamata di sistema `fork()`. Si utilizzi un opportuno esempio per integrare la spiegazione.

Domanda 2 (max 5 punti)

Cos'è il Direct Memory Access (DMA) e quando viene usato? Utilizzare opportuni esempi e schemi grafici per integrare la spiegazione.

Domanda 3 (max 5 punti)

1. Elencare i principali algoritmi di sostituzione delle pagine.
2. Spiegare cosa siano il tasso di page-fault e l'anomalia di Belady, indicando quali algoritmi di sostituzione ne siano affetti.

Esercizio 1 (max 7,5 punti)

Si supponga di avere un hard disk contenente 200 cilindri, numerati da 0 a 199. Il dispositivo sta servendo una richiesta al cilindro 40 e la precedente richiesta si trovava al cilindro 80. La coda di richieste è la seguente (in ordine FIFO)

88, 36, 112, 44, 169, 120, 65, 96, 0, 33

A partire dalla posizione corrente della testina, si disegnino tre grafici che mostrino i movimenti che il braccio dell'hard disk deve compiere per esaudire tutte le richieste nella coda adoperando gli algoritmi di scheduling del disco **FCFS**, **SCAN** e **C-SCAN**.

Esercizio 2 (max 7,5 punti)

Sia data la seguente lista di operazioni che i processi P1, P2, P3 devono eseguire.

P1	**P2**	**P3**
wait(S2)	wait(S3)	wait(S1)
wait(S1)	print("E")	print("A")
print("G")	signal(S1)	signal(S1)
signal(S3)	wait(S3)	print("D")
	print("A")	signal(S2)

Qual è il flusso di esecuzione se inizialmente S1 = 0, S2 = 0 e S3 = 1? Motivare la risposta.

Sistema Operativo

- Un sistema operativo è un software che gestisce l'hardware di un calcolatore.

- Lo scopo di un sistema operativo è quello di fornire all'utente un ambiente nel quale l'esecuzione dei programmi possa avvenire in modo conveniente ed efficace.

Il ruolo del sistema operativo

- Un sistema di elaborazione si può suddividere in quattro componenti:

- Un sistema elaborativo si può anche considerare come l'insieme di

Componenti di un sistema elaborativo

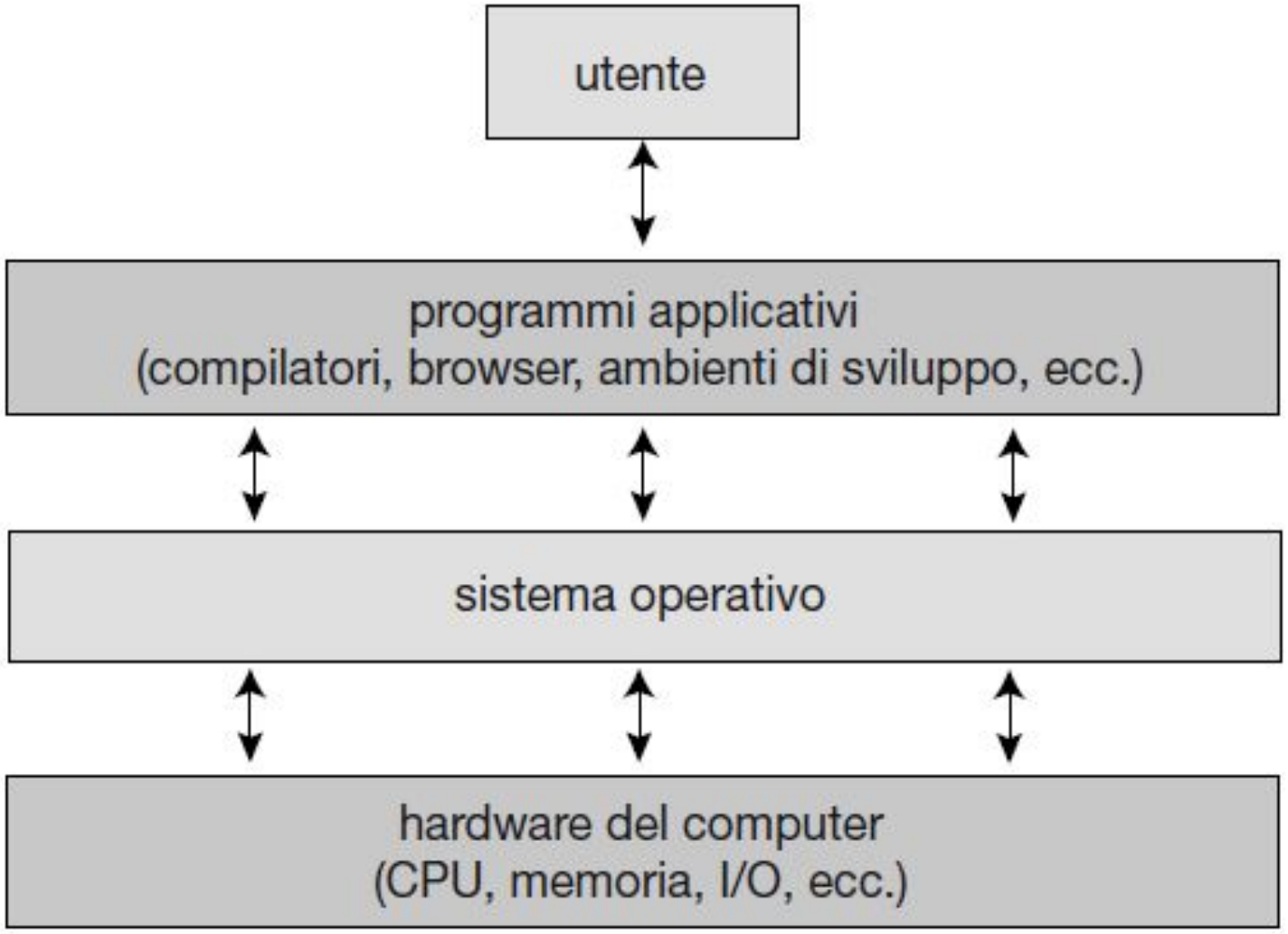
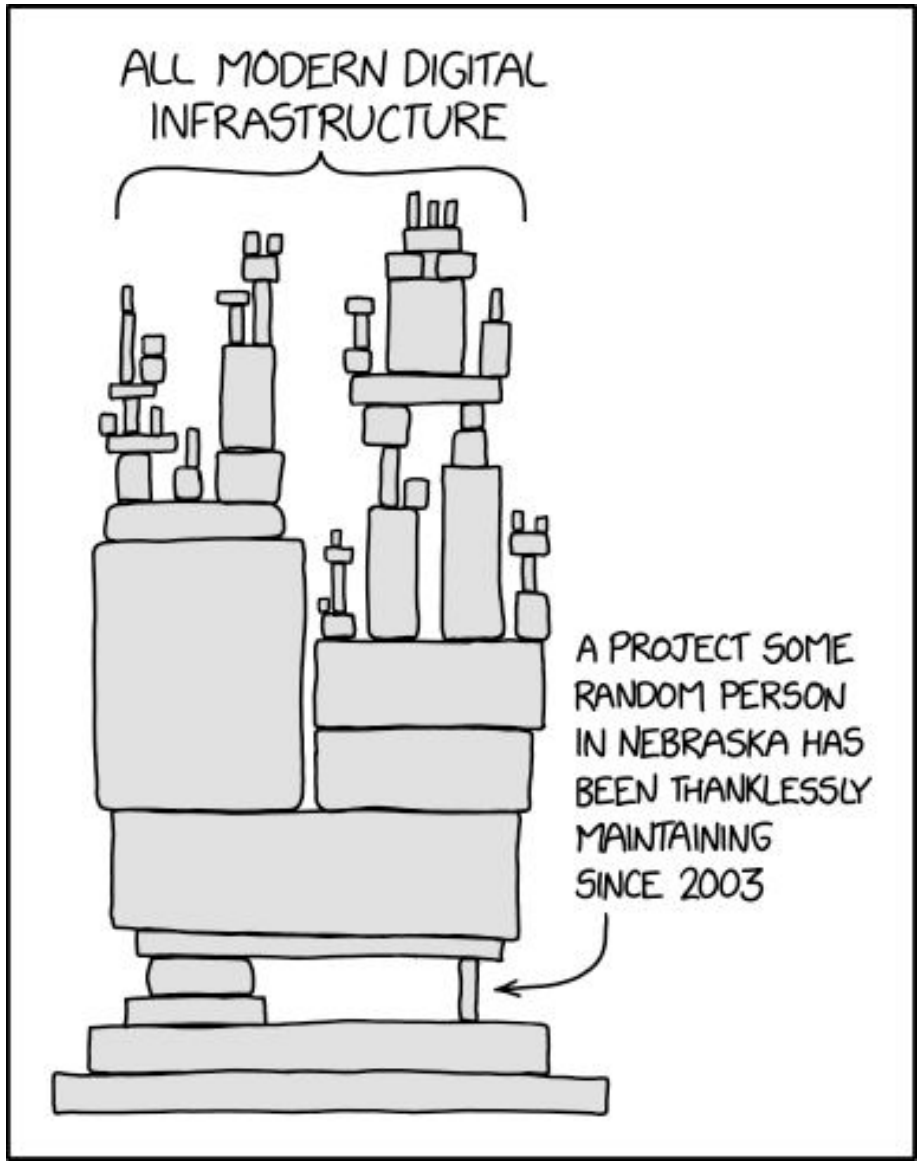


Figura 1.1

Componenti software di un sistema elaborativo



https://xkcd.com/2347/

Definizione di Sistema Operativo

1. I **sistemi operativi** esistono poiché rappresentano una soluzione ragionevole al problema di realizzare un sistema elaborativo che si possa impiegare facilmente, per eseguire i programmi e agevolare la soluzione dei problemi degli utenti.
2. Il sistema operativo è il solo programma che funziona sempre nel calcolatore, generalmente chiamato **kernel** (*nucleo*).
3. Oltre al kernel vi sono due tipi di programmi: i **programmi di sistema**, associati al sistema operativo, ma che non fanno necessariamente parte del kernel, e i **programmi applicativi**, che includono tutti i programmi non correlati al funzionamento del sistema.
4. I sistemi operativi mobili non sono costituiti esclusivamente da un kernel, ma anche da un **middleware**, ovvero da una collezione di ambienti software che fornisce servizi aggiuntivi per chi sviluppa applicazioni.

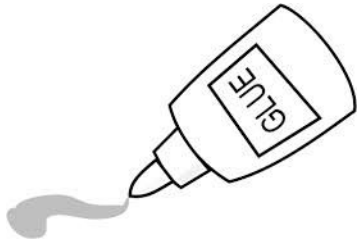
Definizione di Sistema Operativo

Referee

- Manage sharing of resources, protection, isolation
- Resource allocation, isolation, communication

Illusionist

- Provide clean, easy to use abstractions of physical resources
- Infinite memory, dedicated machine
- Higher level objects: files, users, messages
- Masking limitations, virtualization

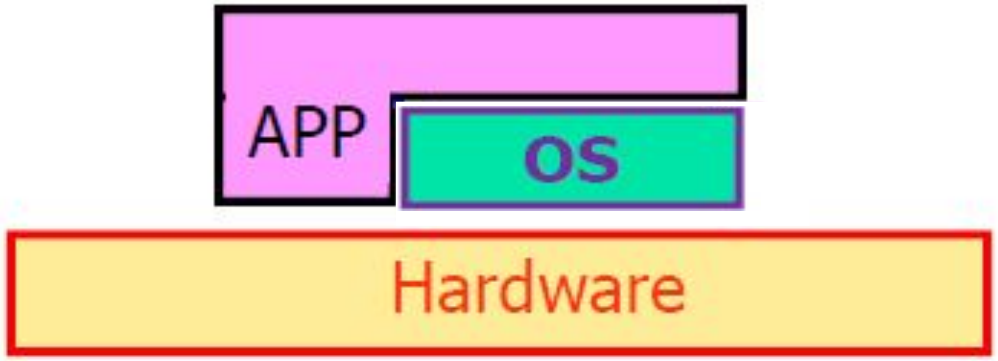


Glue

- Common services
- Storage, Window system, Networking
- Sharing, Authorization
- Look and feel

Prof. Anthony D. Joseph
http://cs162.eecs.Berkeley.edu

History OS: Evolution Step 0



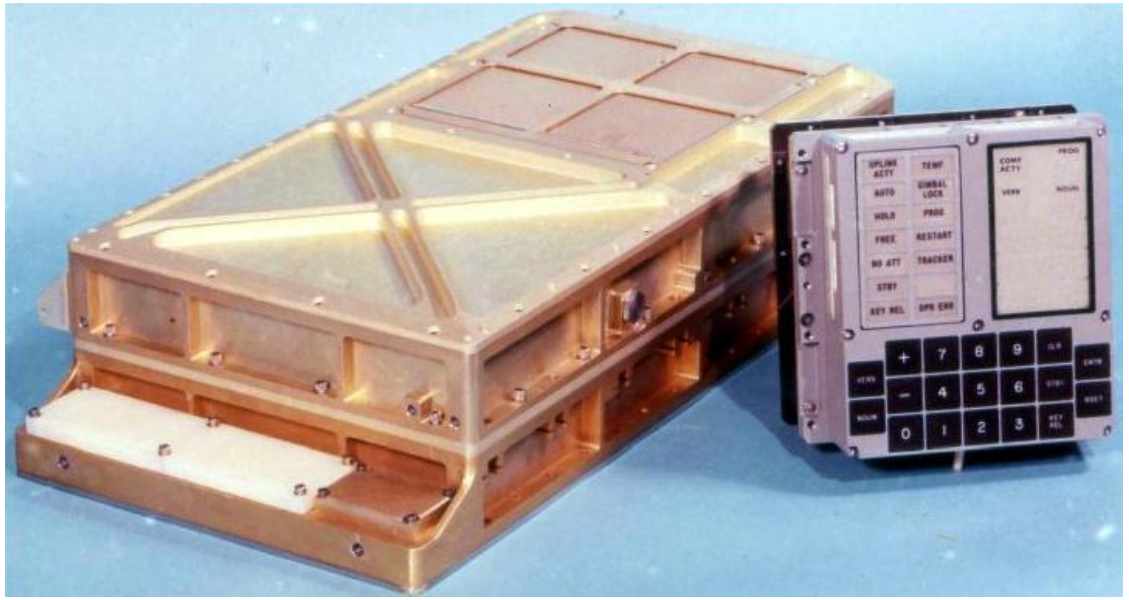
- Simple OS: One program, one user, one machine:
  - examples: early computers, early PCs,
  - embedded controllers such as Nintendo, cars, elevators
  - OS just a library of standard services, e.g. standard device drivers, interrupt handlers, I/O
- Non-problems: No malicious people. No bad programs

  ⇒ A minimum of complex interactions
- Problem: poor utilization, expensive

https://os.itec.kit.edu/downloads/sysarch00a_history.pdf

Apollo Guidance Computer

- AGC software was written in AGC assembly language and stored on rope memory. The bulk of the software was on read-only rope memory and thus could not be changed in operation.

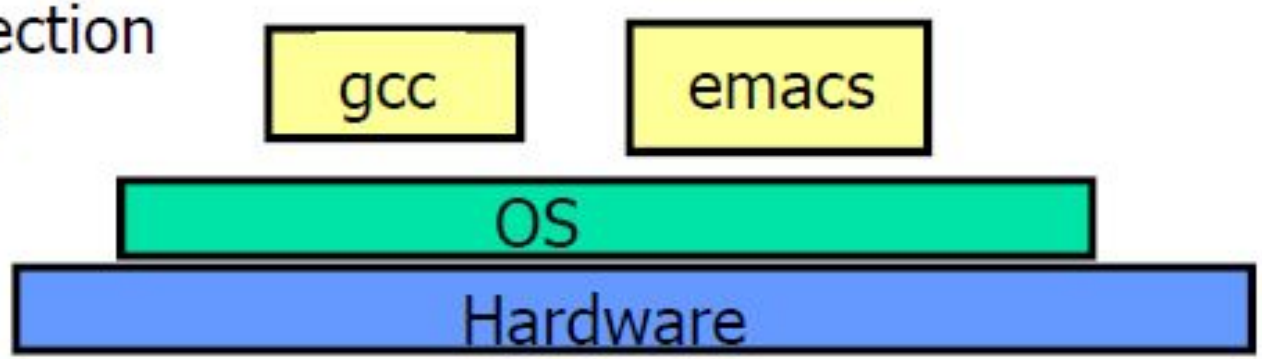


- There was a simple real-time operating system designed by J. Halcombe Laning, consisting of the Exec, a batch job-scheduling using cooperative multi-tasking and an interrupt-driven pre-emptive scheduler called the Waitlist, which could schedule multiple timer-driven 'tasks'. The tasks were short threads of execution which could reschedule themselves for re-execution on the Waitlist, or could kick off a longer operation by starting a "job" with the Exec.

https://en.wikipedia.org/wiki/Apollo_Guidance_Computer

History OS: Evolution Step 1

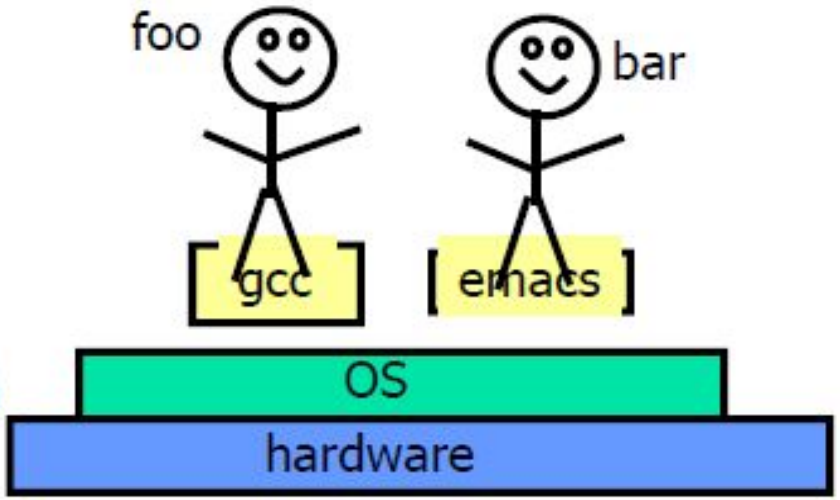
- Simple OS is inefficient:
  - a waiting process blocks everything else on the machine
- (Seemingly) Simple hack:
  - run more than one process at once
  - when one process blocks, switch to another
- A couple of problems: *what if a program*
  - *does infinite loops or*
  - *starts randomly scribbling on memory?*
- OS adds protection
  - Interposition
  - Preemption
  - Privilege



https://os.itec.kit.edu/downloads/sysarch00a_history.pdf

History OS: Evolution Step 2

- Simple OS is too expensive:
  - one user = one computer ⇒
- (Seemingly) simple hack:
  - Allow more than one user at once
  - *Does machine now run n times slower?* Usually not
  - Key observation: users are active in bursts
  - If idle, give resources to others
- Problems: *what if*
  - *users are greedy*
  - *evil*
  - *or just too numerous?*
- OS adds protection
  - (notice: as we try to utilize resources, complexity grows)



https://os.itec.kit.edu/downloads/sysarch00a_history.pdf

Componenti hardware di un sistema elaborativo

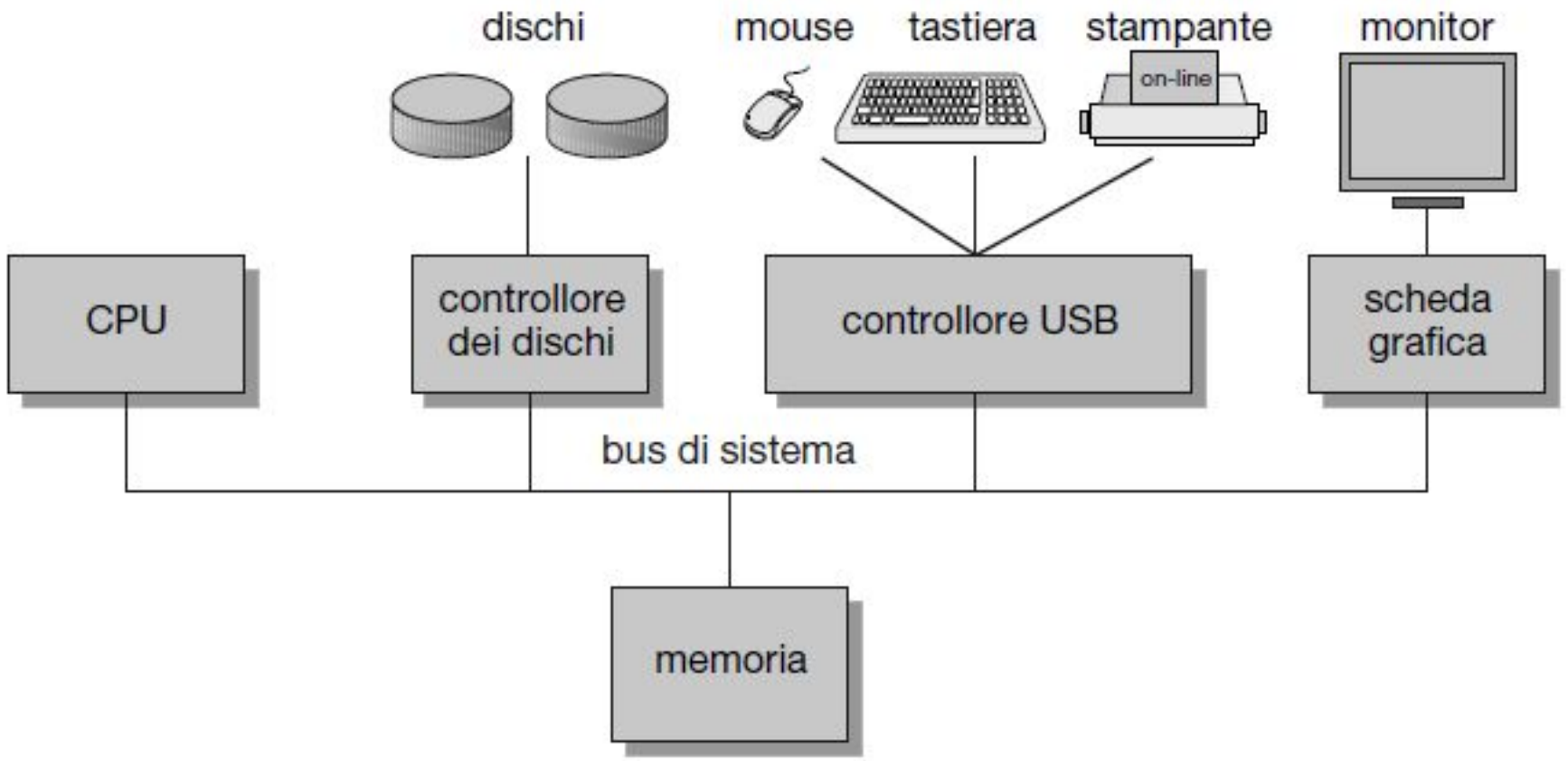


Figura 1.2 Un tipico sistema elaborativo.

Organizzazione di un sistema elaborativo

1. Un moderno calcolatore general-purpose è composto da una o più **CPU** e da un certo numero di **controllori di dispositivi** connessi attraverso un canale di comunicazione comune (*bus)* che permette l'accesso alla memoria condivisa dal sistema (Figura 1.2).
2. I sistemi operativi possiedono in genere per ogni controllore di dispositivo un **driver del dispositivo** che gestisce le specificità del controllore e funge da interfaccia uniforme con il resto del sistema.
3. La **CPU** e i **controllori** possono eseguire operazioni in parallelo, competendo per i cicli di memoria.

interruzioni

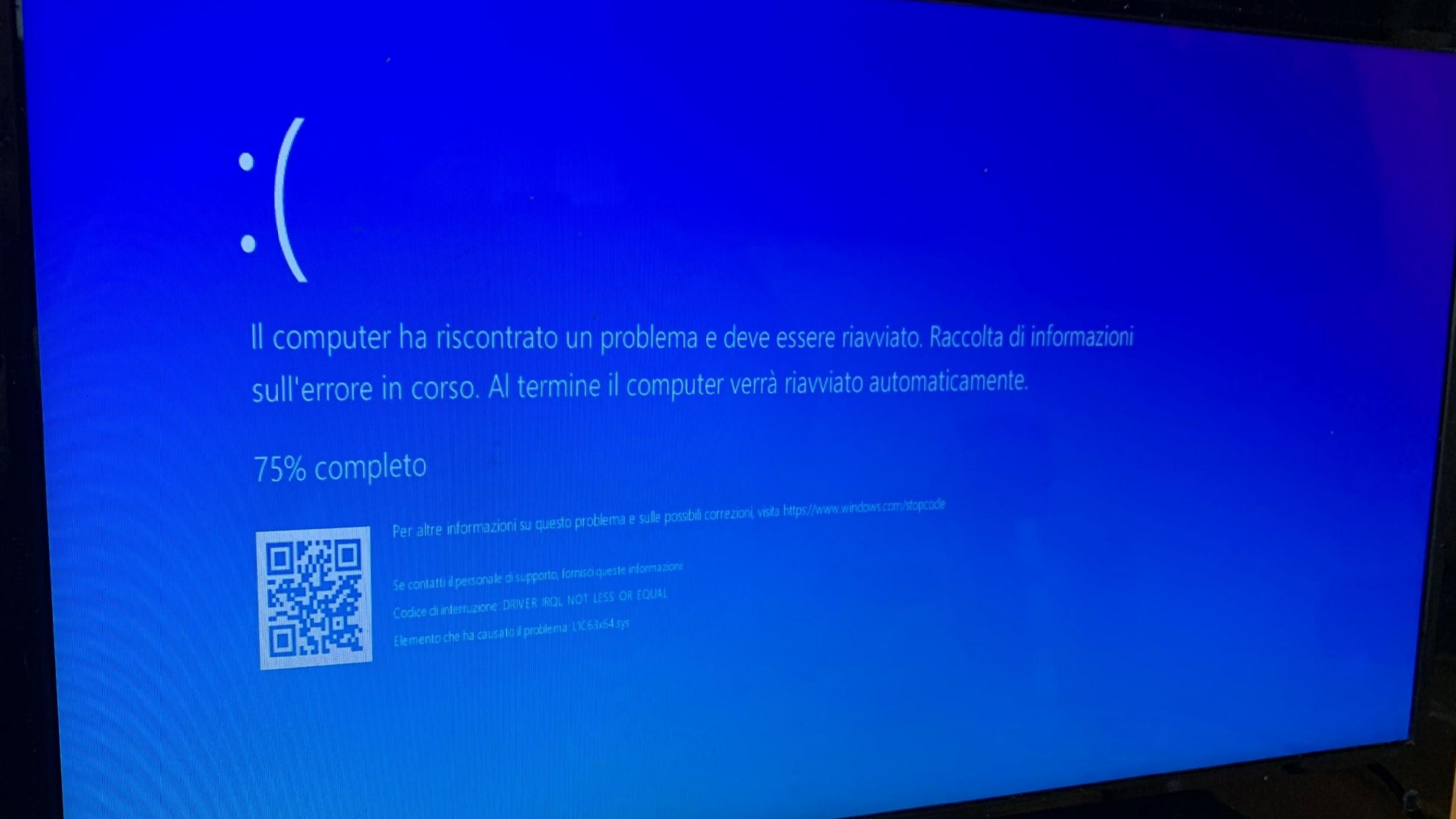
- L'hardware della CPU dispone di un filo chiamato **linea di richiesta di interruzione** (*interrupt-request line*) che la CPU controlla dopo l'esecuzione di ogni istruzione.

- La maggior parte delle CPU ha **due linee di richiesta di interruzione**:

1. non mascherabile (*nonmaskable interrupt*) ⟶ riservata a eventi come errori irreversibili di memoria

2. mascherabile (*maskable interrupt)* ⟶ utilizzata dai controllori dei dispositivi per richiedere un servizio

Concatenamento delle interruzioni (*interrupt chaining*) ⟶ ogni elemento nel vettore delle interruzioni punta alla testa di un elenco di gestori


:(
Il computer ha riscontrato un problema e deve essere riavviato. Raccolta di informazioni sull'errore in corso. Al termine il computer verrà riavviato automaticamente.
75% completo
Per altre informazioni su questo problema e sulle possibili correzioni, visita https://www.windows.com/stopcode
Se contatti il personale di supporto, fornisci queste informazioni:
Codice di interruzione: DRIVER_IRQL_NOT_LESS_OR_EQUAL
Elemento che ha causato il problema: LXC63x64.sys

Interruzioni

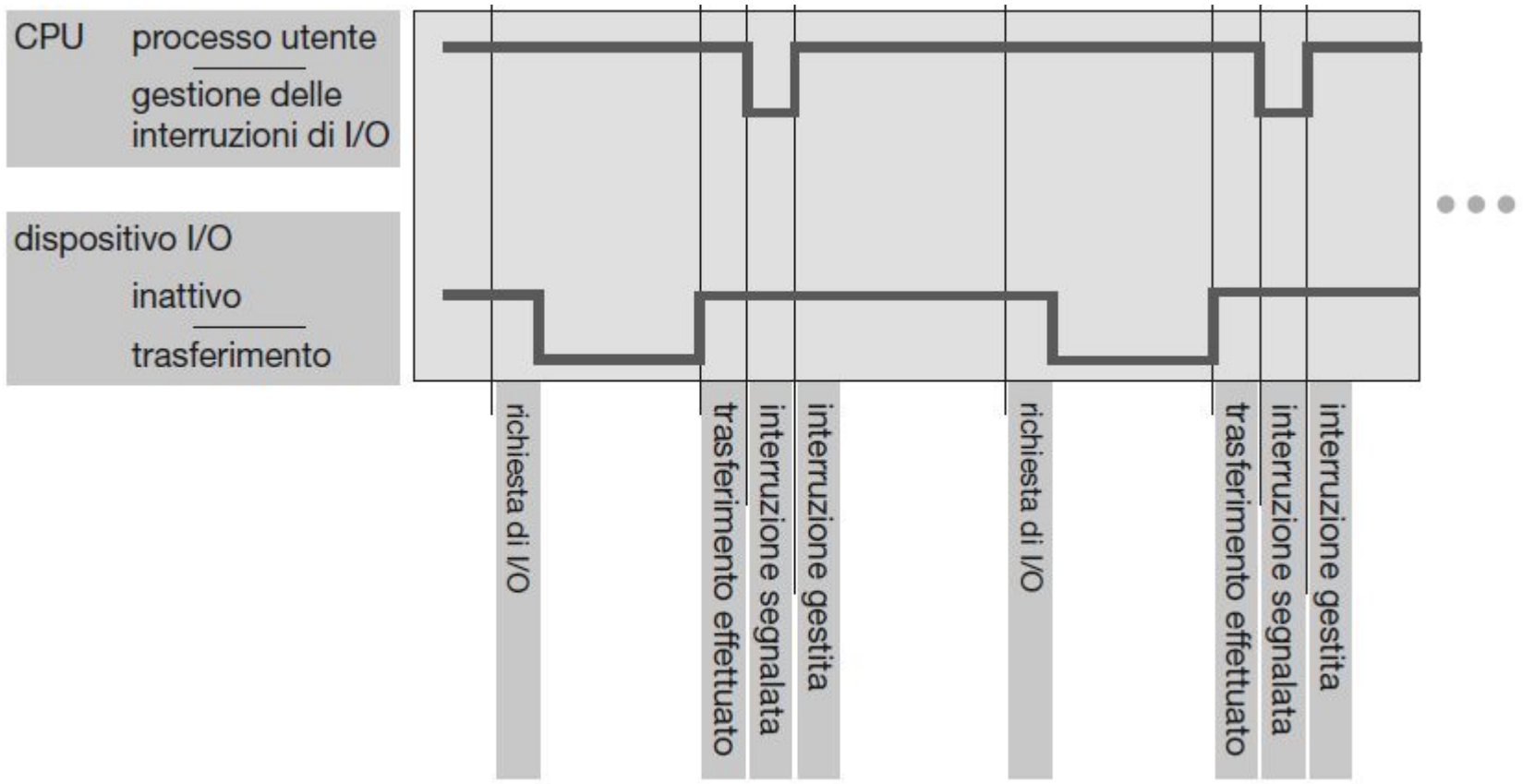


Figura 1.3 Diagramma temporale delle interruzioni per un singolo programma che invia dati in output.

Ciclo di I/O interrupt driven

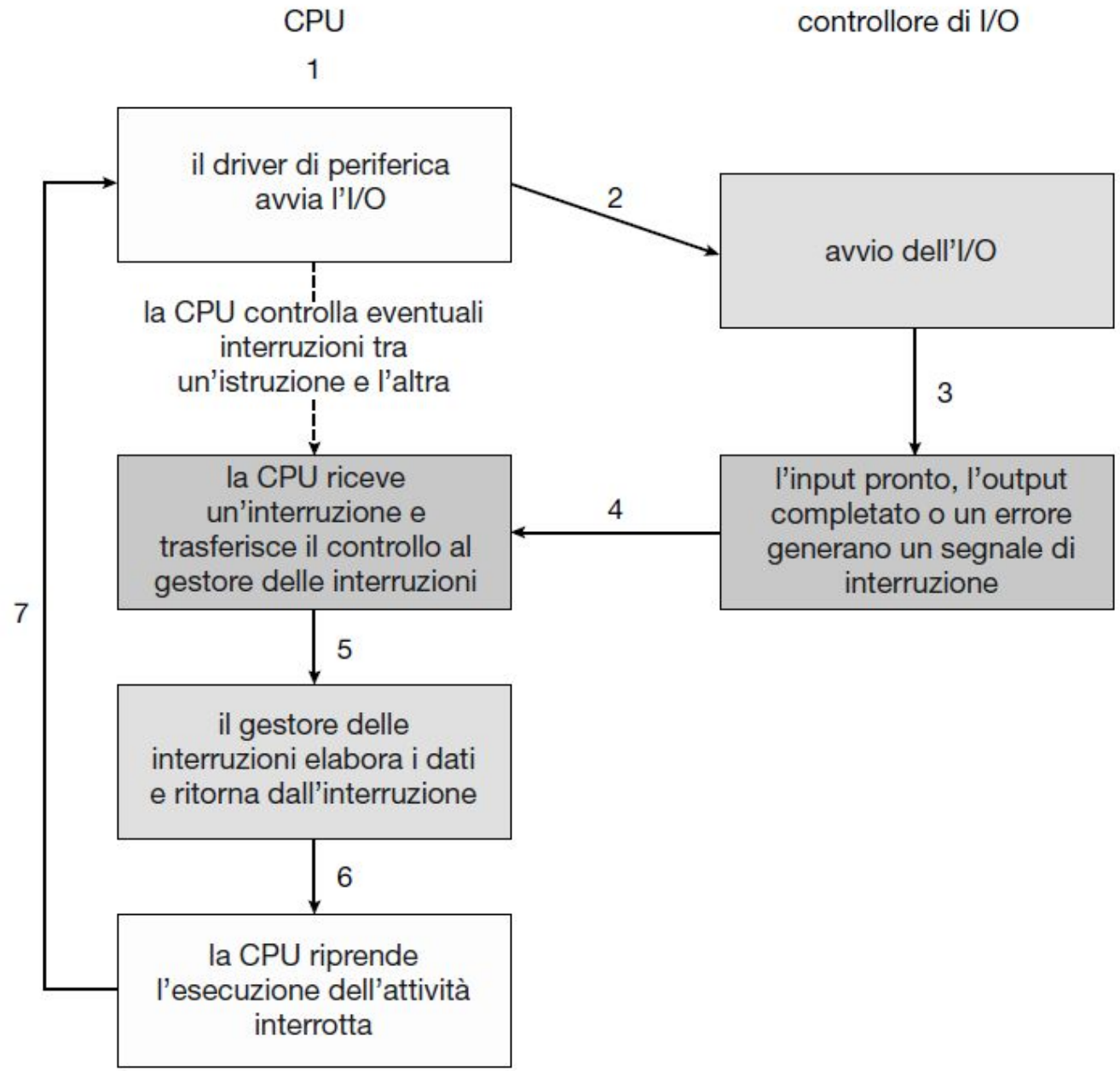


Figura 1.4 Ciclo di I/O guidato dalle interruzioni.

Eventi

numero di vettore	descrizione
0	errore di divisione
1	eccezione di debug
2	interruzione null
3	breakpoint
4	eccezione di overflow
5	eccezione di range exceeded
6	codice operativo non valido
7	dispositivo non disponibile
8	doppio errore
9	overrun del segmento coprocessore (riservato)
10	task state segment (tss) non valido
11	segmento non presente
12	errore di stack
13	protezione generale
14	errore di pagina
15	(riservato Intel, non utilizzare)
16	errore in virgola mobile
17	controllo dell'allineamento
18	controllo della macchina
19–31	(riservato Intel, non utilizzare)
32–255	interruzioni mascherabili

Figura 1.5 Tabella degli eventi di un processore Intel.

Struttura della memoria

MEMORIA PRINCIPALE O **CENTRALE**: **memoria ad accesso casuale** (*random access memory, RAM)* → *memoria volatile*

I computer utilizzano una **memoria di sola lettura elettricamente cancellabile e programmabile (EEPROM)** e altre forme di archiviazione del firmware che vengono riscritte raramente e che non sono volatili.

MEMORIA SECONDARIA: estensione della memoria centrale → capacità di conservare in modo permanente grandi quantità di informazioni

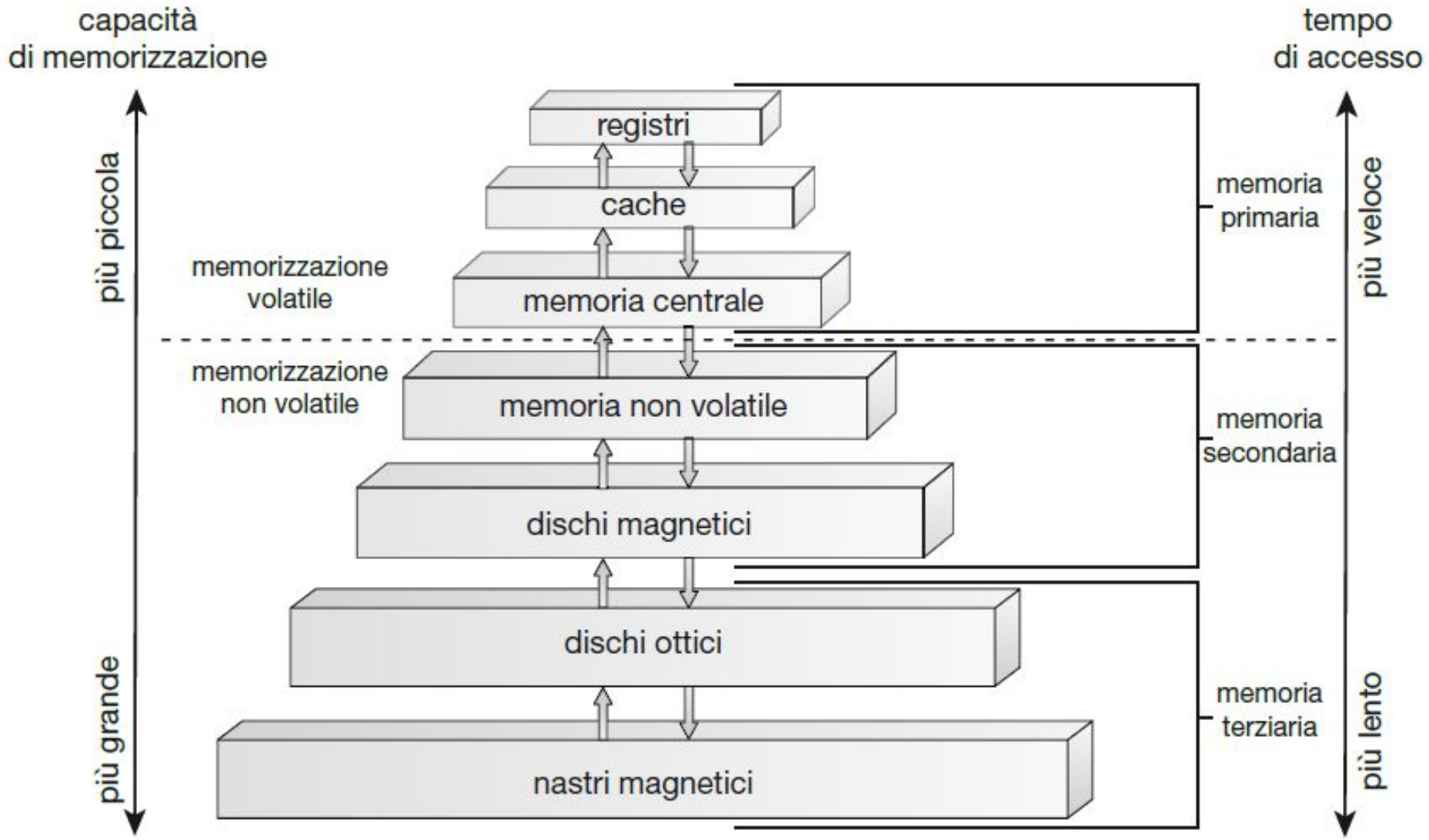


Figura 1.6 Scala gerarchica dei sistemi di memorizzazione.

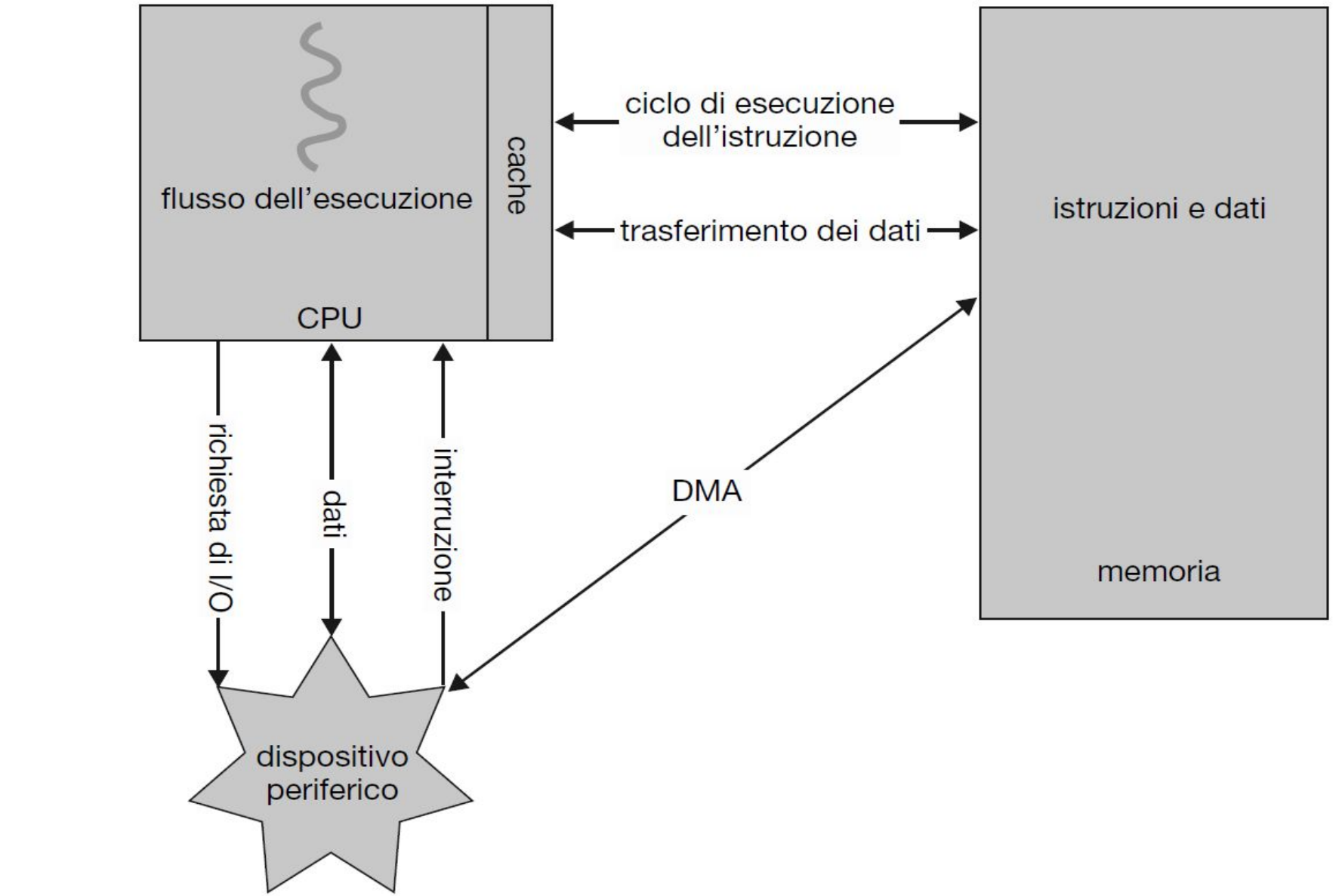


Figura 1.7 Funzionamento di un moderno sistema operativo.

Architettura degli elaboratori

Sistemi monoprocessore

↓

Diversi anni fa la maggior parte dei sistemi utilizzava un solo processore contenente un'unica CPU con un unico nucleo di elaborazione (o unità di calcolo, o *core).*

Sistemi multiprocessore

↓

Multielaborazione simmetrica

La definizione di multiprocessore si è evoluta nel tempo e include ora i sistemi multicore, in cui più unità di calcolo (core) risiedono su un singolo chip.

↓

Architettura **dual-core** = due unità sullo stesso chip

Architettura degli elaboratori

- **CPU**: componente hardware che esegue le istruzioni.
- **Processore**: chip che contiene una o più CPU.
- **Unità di calcolo** (*core*): unità di elaborazione di base della CPU.
- **Multicore**: che include più unità di calcolo sulla stessa CPU.
- **Multiprocessore**: che include più processori.

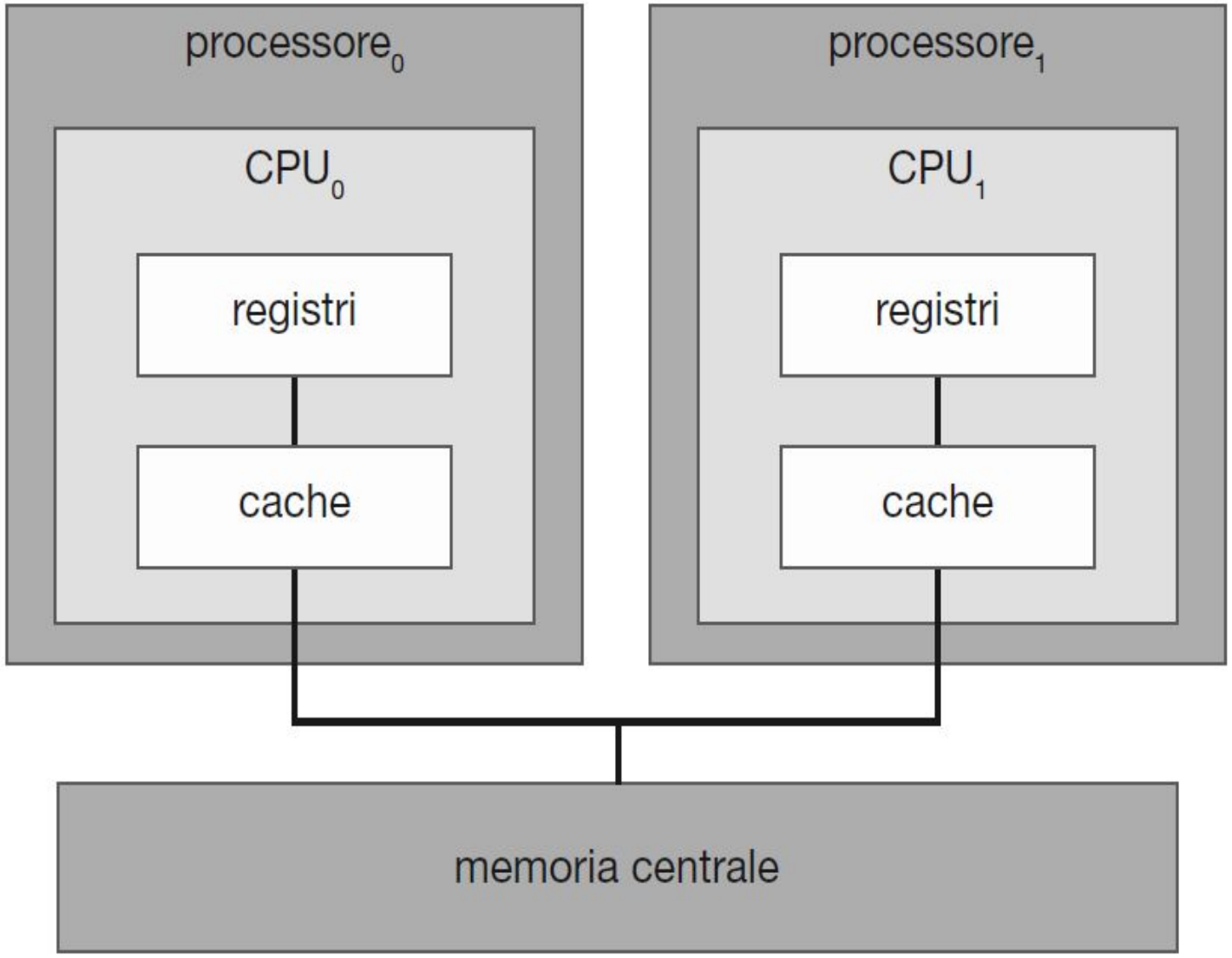


Figura 1.8 Architettura di multielaborazione simmetrica.

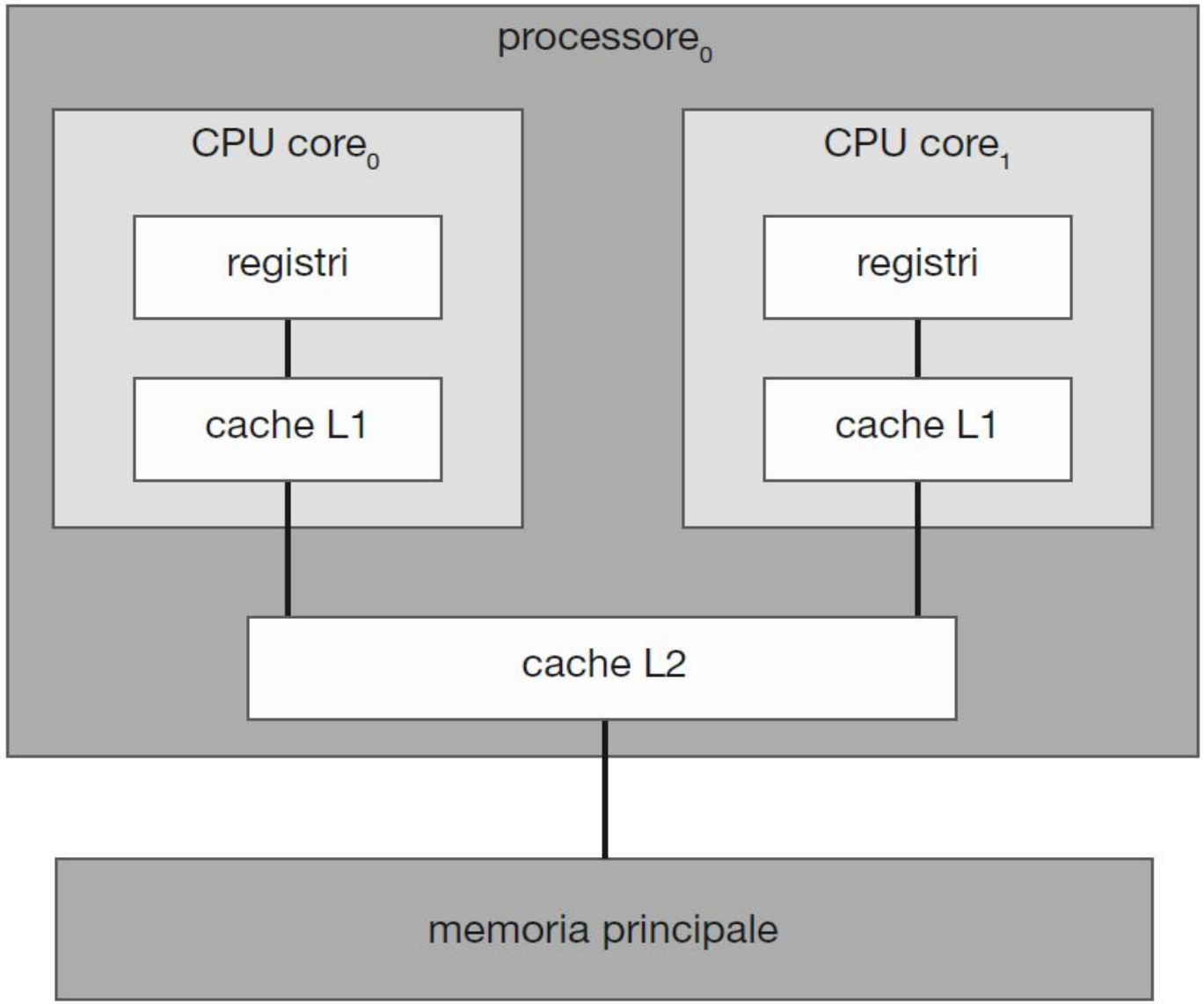


Figura 1.9 Architettura dual-core, con due unità sullo stesso chip.

Aggiungere **nuove CPU** a un multiprocessore ne aumenta la potenza di calcolo, ma ne *peggiora le prestazioni*

fornire a ciascuna CPU (o a ciascun gruppo di CPU) la propria memoria locale accessibile per mezzo di un bus locale piccolo e veloce

accesso non uniforme alla memoria o **NUMA**

i **sistemi NUMA** possono scalare in modo più efficace con l'aggiunta di più processori

↓

sempre più diffusi nei server e nei sistemi di elaborazione ad alte prestazioni

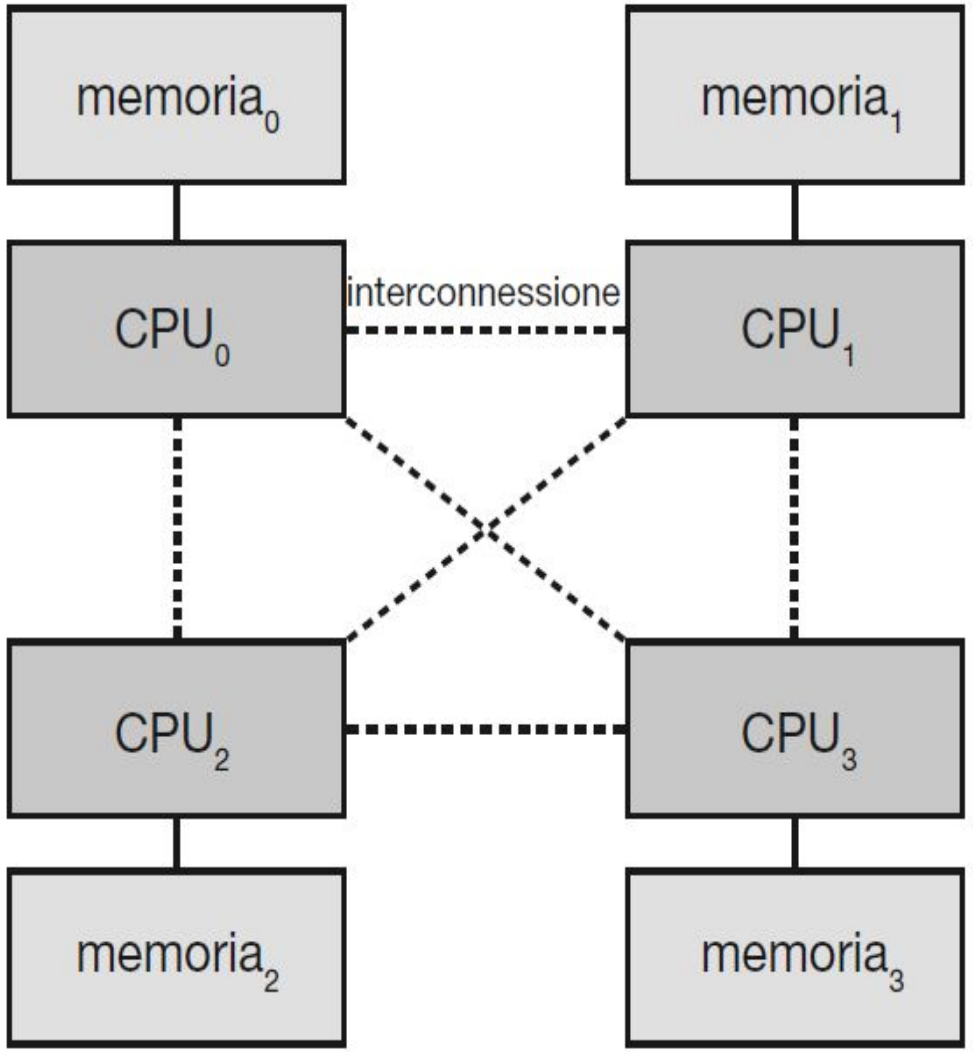


Figura 1.10 Architettura multiprocessore NUMA.

Cluster di elaboratori (*clustered systems)* o **cluster:** un altro tipo di sistemi multiprocessore, basati sull'uso congiunto di più CPU, ma differiscono dai sistemi multiprocessore perché composti di
due o più calcolatori completi – detti **nodi** – collegati tra loro.

↓

debolmente accoppiati

- Cluster asimmetrici
- Cluster simmetrici
- Cluster paralleli

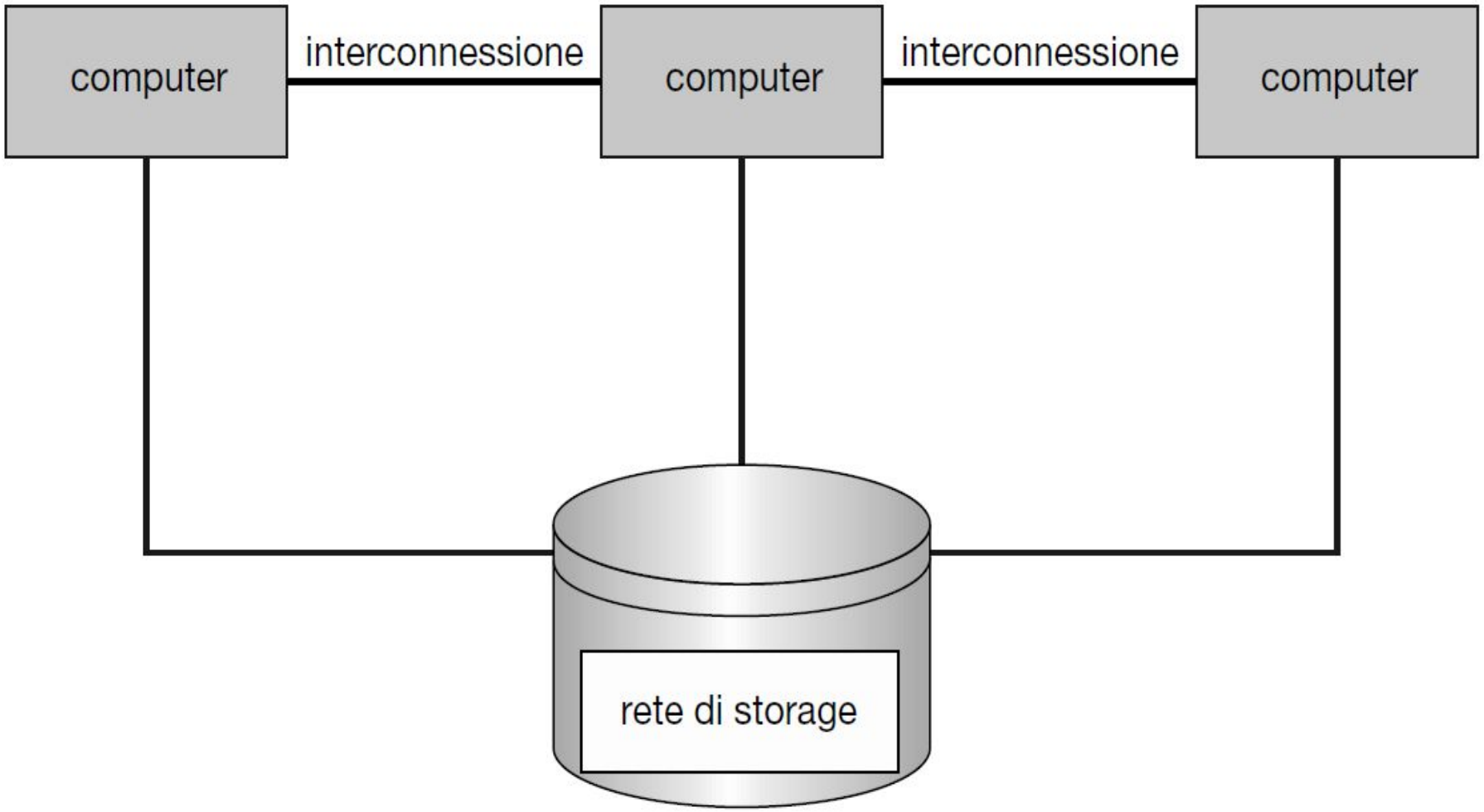


Figura 1.11 Struttura generale di un cluster.

Attività del sistema operativo

1. **Programma di avviamento (*bootstrap program*)**
2. Il **kernel** inizia a offrire servizi al sistema e agli utenti
3. **Interruzioni** ed **eccezioni** (*traps* o *exceptions*)
4. **Chiamata di sistema** (**system call**)
5. **Multiprogrammazione** → *multitasking.*

↓ ↓

file system memoria virtuale

Memoria e multiprogrammazione

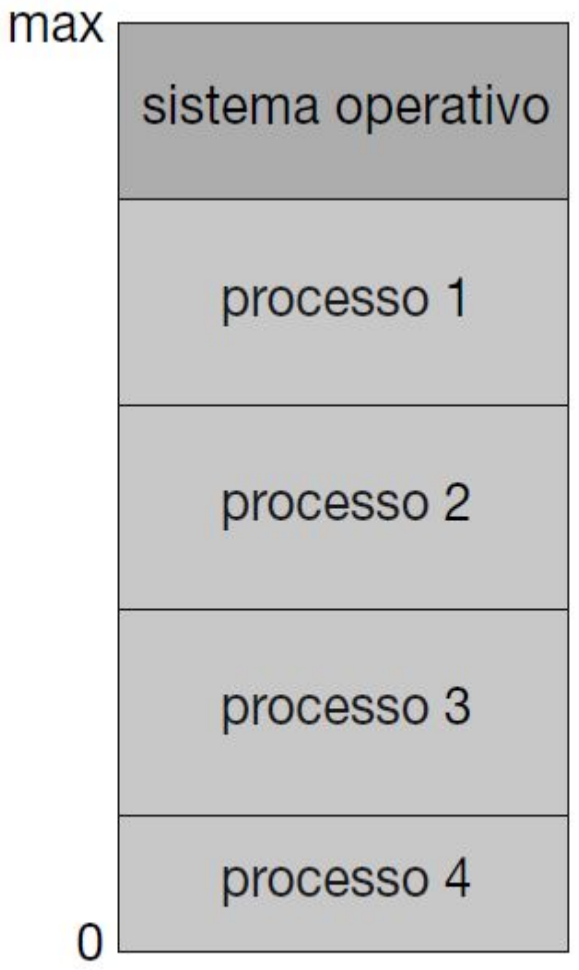


Figura 1.12 Configurazione della memoria per un sistema con multiprogrammazione.

Dual-mode operation

- **modalità utente** e **modalità di sistema** (detta anche *modalità kernel, modalità supervisore, modalità monitor o modalità privilegiata).*

- Per indicare quale sia la modalità attiva, l'architettura della CPU deve essere dotata di un bit, chiamato appunto bit di modalità: **kernel** (0) o **user** (1).

- La duplice modalità di funzionamento (*dual-mode)* consente la protezione del sistema operativo e degli altri utenti dagli errori di un utente.

- Le **chiamate di sistema** (*system call*) sono gli strumenti con cui un programma utente richiede al sistema operativo di compiere operazioni a esso riservate, per conto del programma utente.

- **Timer** → invia un segnale d'interruzione alla CPU a intervalli di tempo specificati e assicura che il sistema operativo mantenga il controllo della CPU.

Dual-mode operation

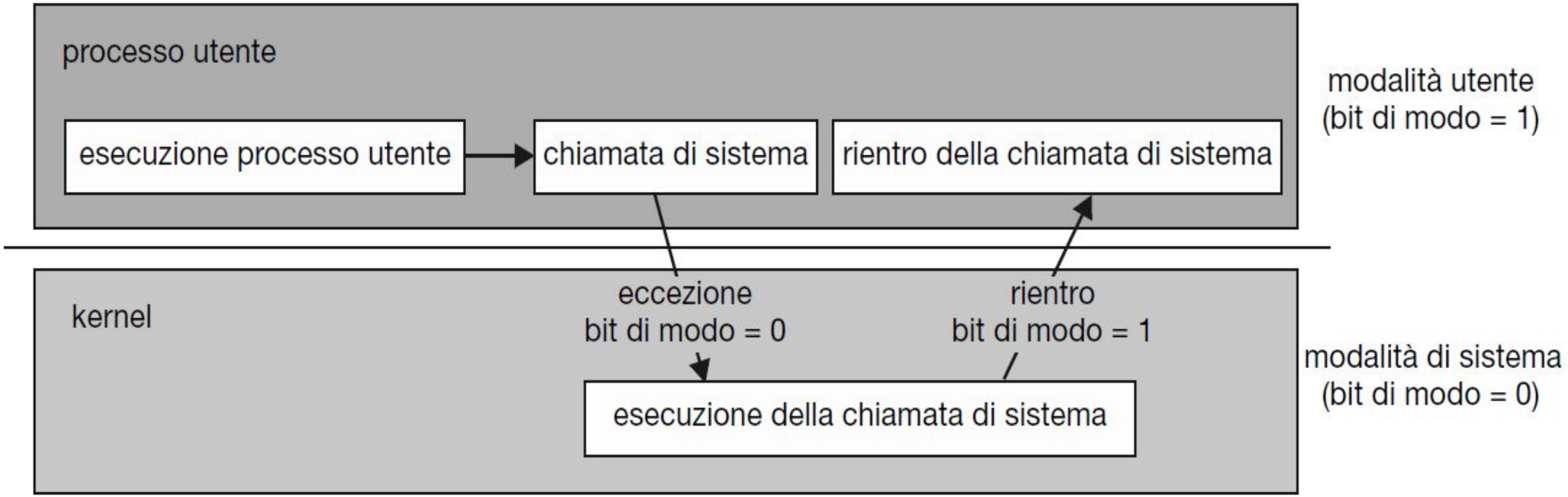


Figura 1.13 Transizione da modalità utente a modalità di sistema.

Gestione delle risorse

Forme di memoria

Livello	1	2	3	4	5
Nome	registri	cache	memoria centrale	disco a stato solido	disco magnetico
Dimensione tipica	< 1 KB	< 16 MB	< 64 GB	< 1 TB	< 10 TB
Tecnologia	memoria dedicata con porte multiple (CMOS)	CMOS SRAM (on-chip o off-chip)	CMOS DRAM	memoria flash	disco magnetico
Tempo d'accesso (ns)	0,25 – 0,5	0,5 – 25	80 – 250	25.000-50.000	5.000,000
Ampiezza di banda (MB/s)	20.000 – 100.000	5000 – 10.000	1000 – 5000	500	20 – 150
Gestito da	compilatore	hardware	sistema operativo	sistema operativo	sistema operativo
Supportato da	cache	memoria centrale	disco	disco	disco o nastro

Figura 1.14 Caratteristiche di varie forme di archiviazione dei dati.

Caricamento delle istruzioni

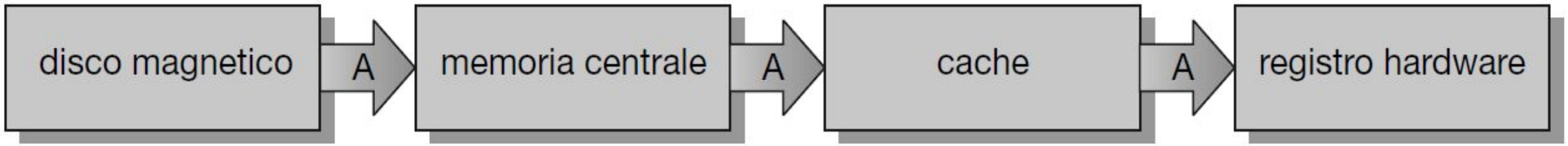


Figura 1.15 Migrazione di un intero A da un disco a un registro.

Sicurezza e protezione

Protezione → ciascun meccanismo di controllo dell'accesso alle risorse possedute da un elaboratore, da parte di processi o utenti.

La **protezione** migliora l'affidabilità rilevando errori nascosti alle interfacce tra i componenti dei sottosistemi.

È compito della **sicurezza** difendere il sistema da attacchi provenienti dall'interno o dall'esterno.

identificatori utente (*user ID*) → identificano univocamente l'utente

Virtualizzazione

Virtualizzazione → tecnica che permette di astrarre l'hardware di un singolo computer in diversi ambienti di esecuzione, creando così l'illusione che ogni distinto ambiente sia in esecuzione sul suo proprio computer.

Permette ai sistemi operativi di funzionare come applicazioni all'interno di altri sistemi operativi

- Con la **virtualizzazione** un sistema operativo compilato per una particolare architettura viene eseguito all'interno di un altro sistema operativo progettato per la stessa CPU.

- **Macchina virtuale (VM)** e **gestore della macchia virtuale**

Virtualizzazione

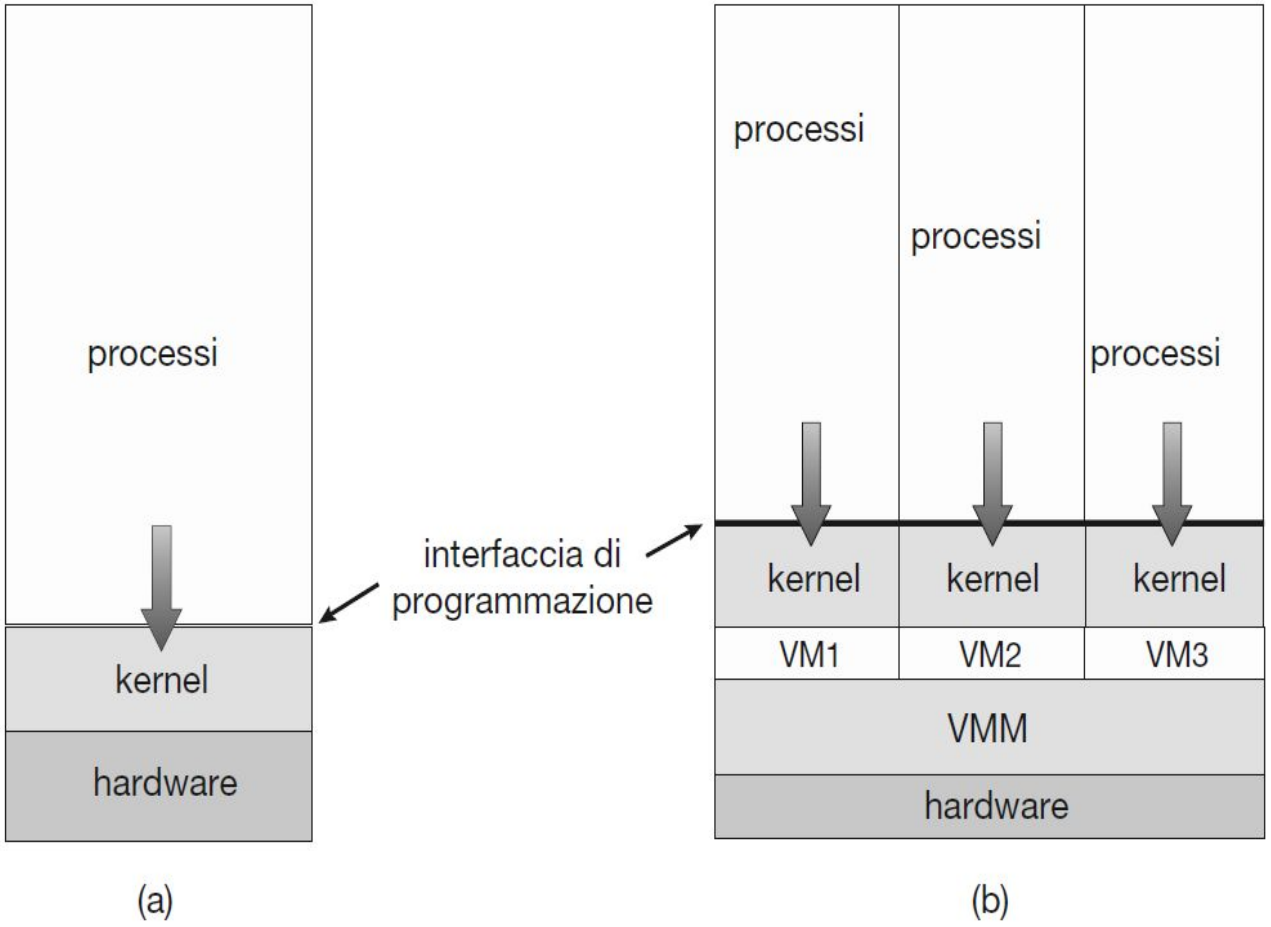


Figura 1.16 Un computer che ha in esecuzione (a) un singolo sistema operativo e (b) tre macchine virtuali.

Sistemi distribuiti

Sistema distribuito → un insieme di elaboratori fisicamente separati e con caratteristiche spesso eterogenee, interconnessi da una rete per consentire agli utenti l'accesso alle varie risorse dei singoli sistemi.

Rete → un canale di comunicazione tra due o più sistemi.

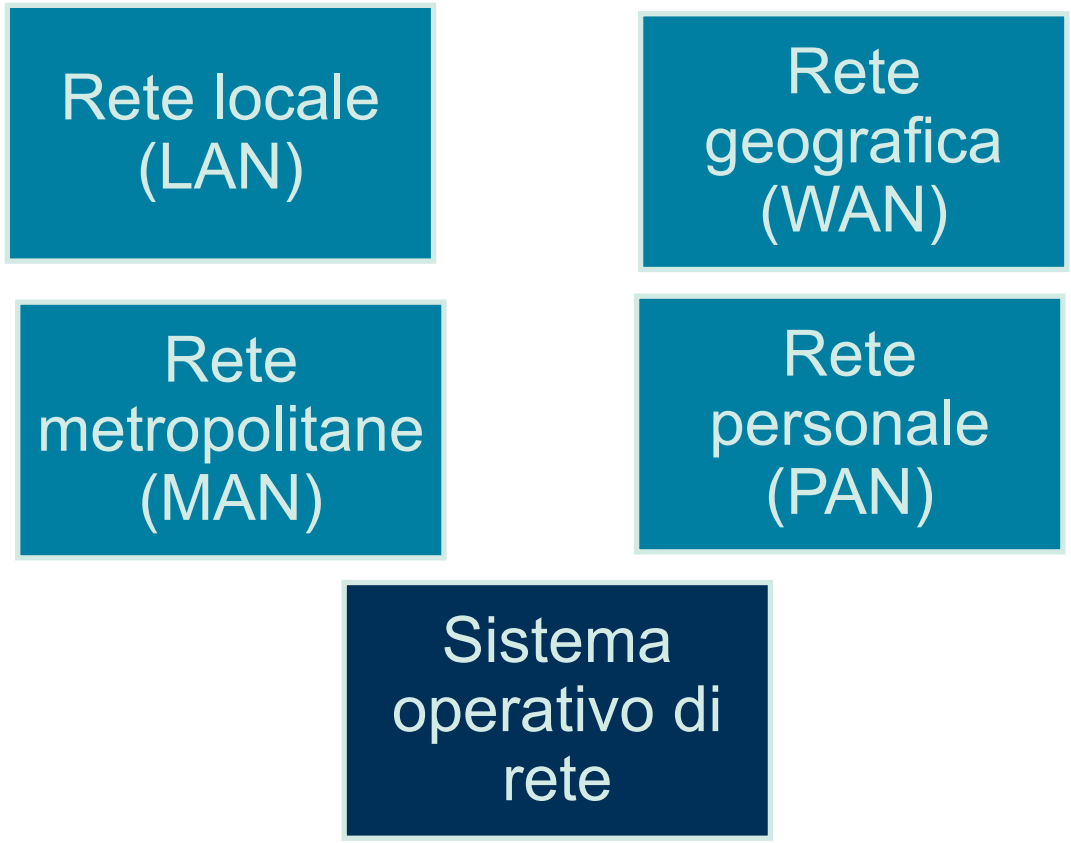
↓

protocollo

↓

TCP/IP

Sistemi distribuiti - Reti



Strutture dati del kernel - Liste

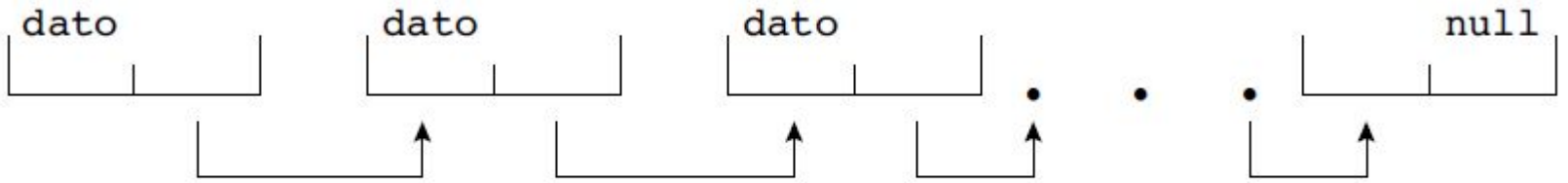


Figura 1.17 Lista semplicemente concatenata.

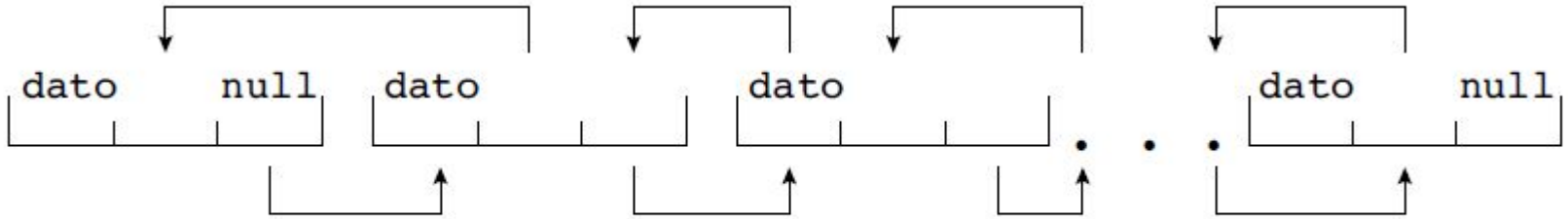


Figura 1.18 Lista doppiamente concatenata.

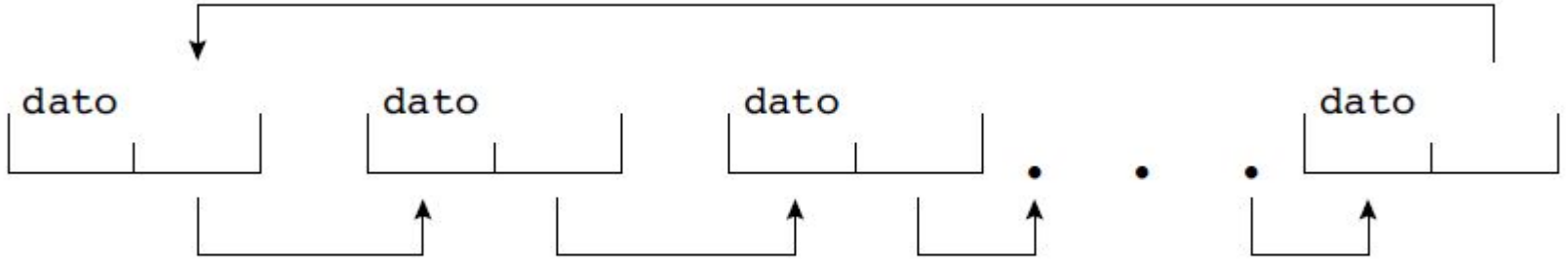
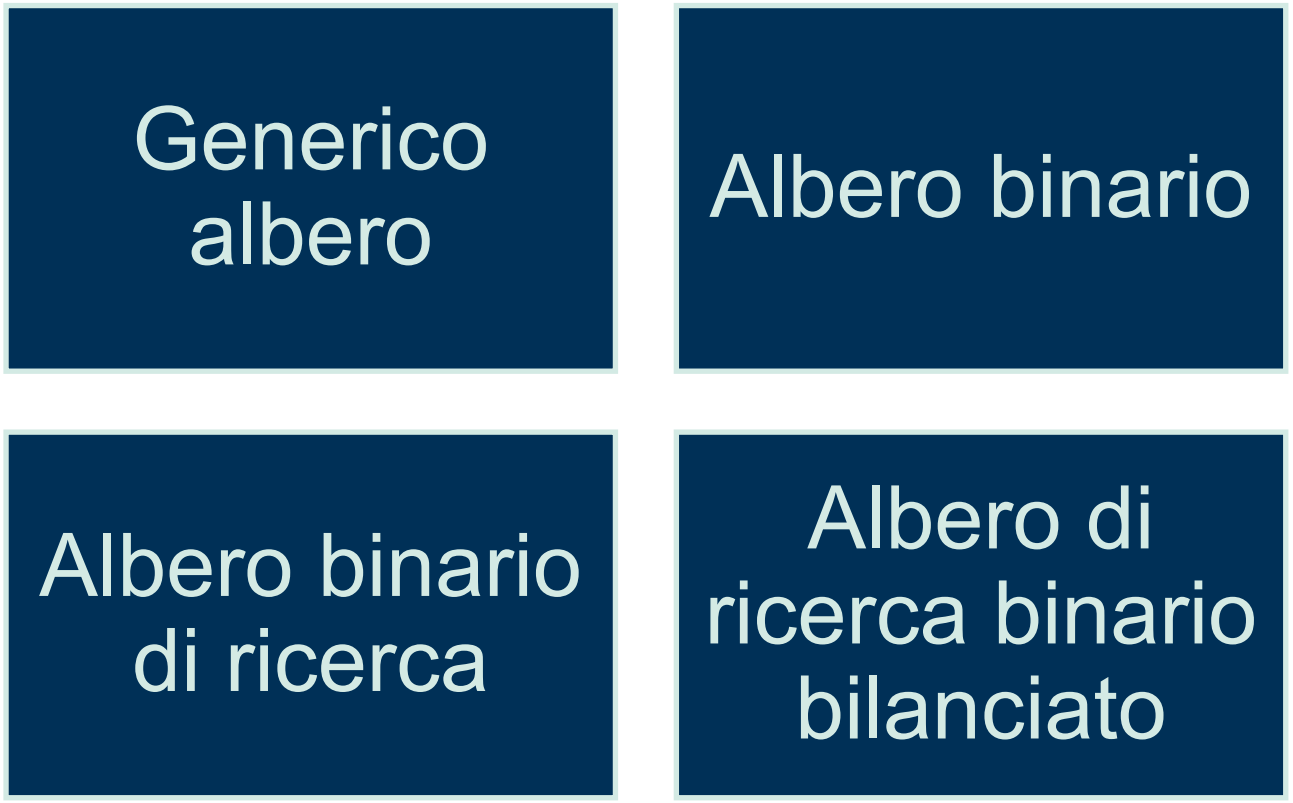


Figura 1.19 Lista circolare.

Alberi

Un **albero** è una struttura dati utilizzabile per rappresentare i dati in maniera gerarchica.



Alberi - esempio

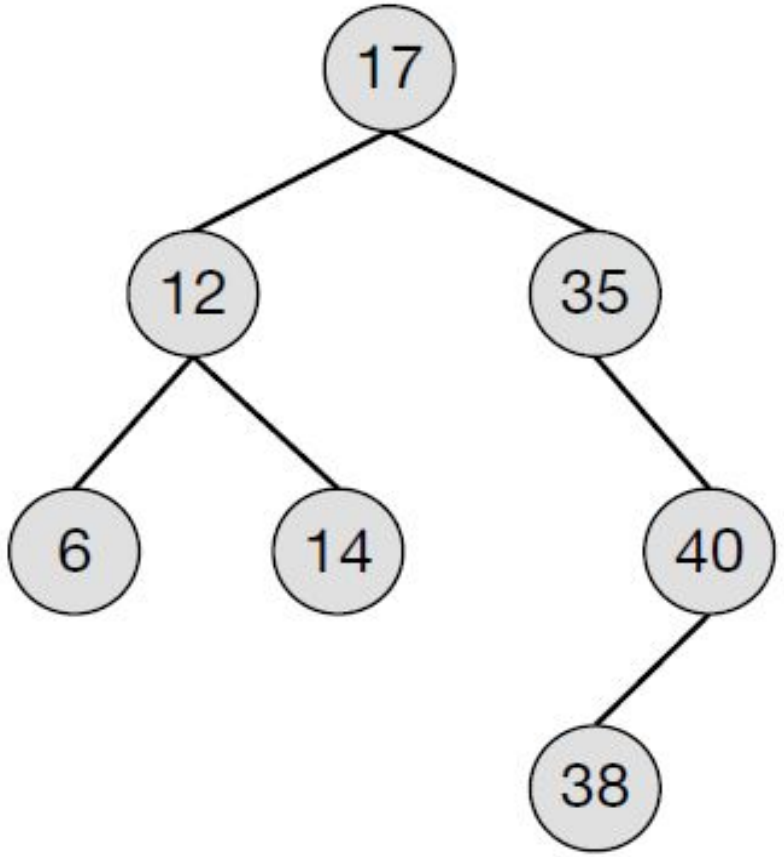
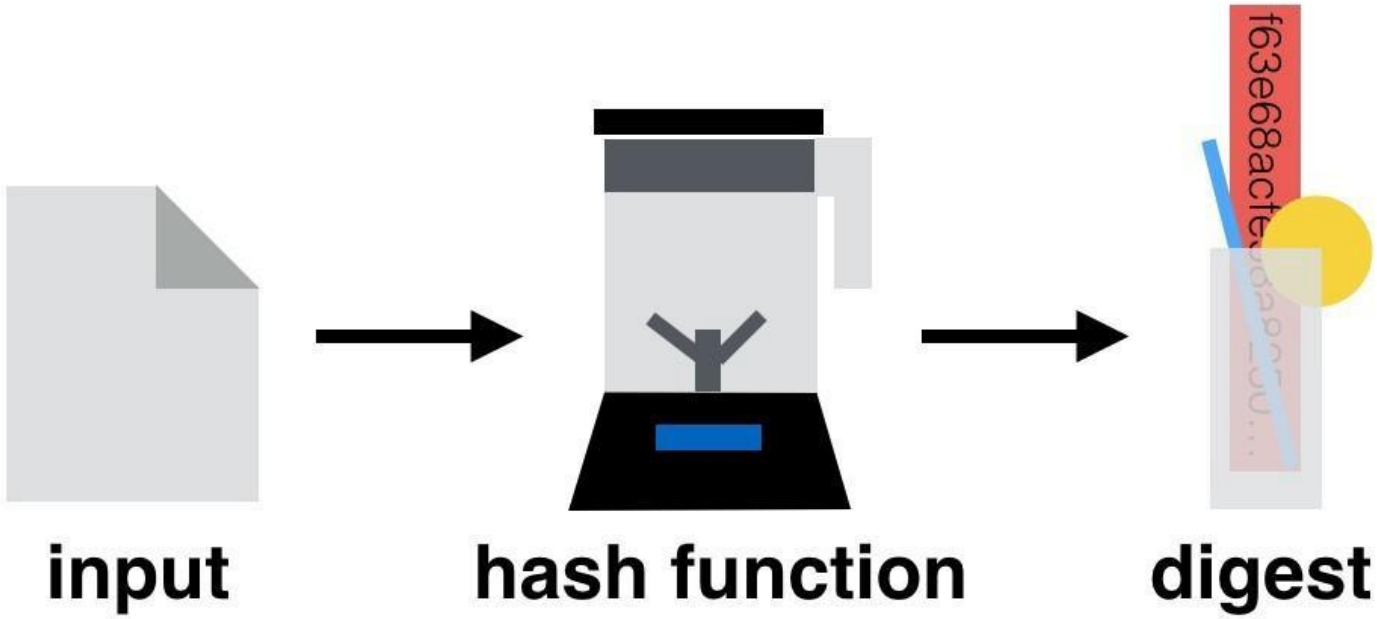


Figura 1.20 Albero binario di ricerca.

Funzioni hash

Una **funzione hash** riceve dati in input, realizza operazioni numeriche sui dati e restituisce un valore numerico.



https://medium.com/@ManningBooks/hash-functions-and-security-8ee0f08602f3

Mappe hash

Una funzione hash può essere utilizzata per creare una **tabella hash** (o mappa hash) che associ (o mappi) coppie [chiave:valore].

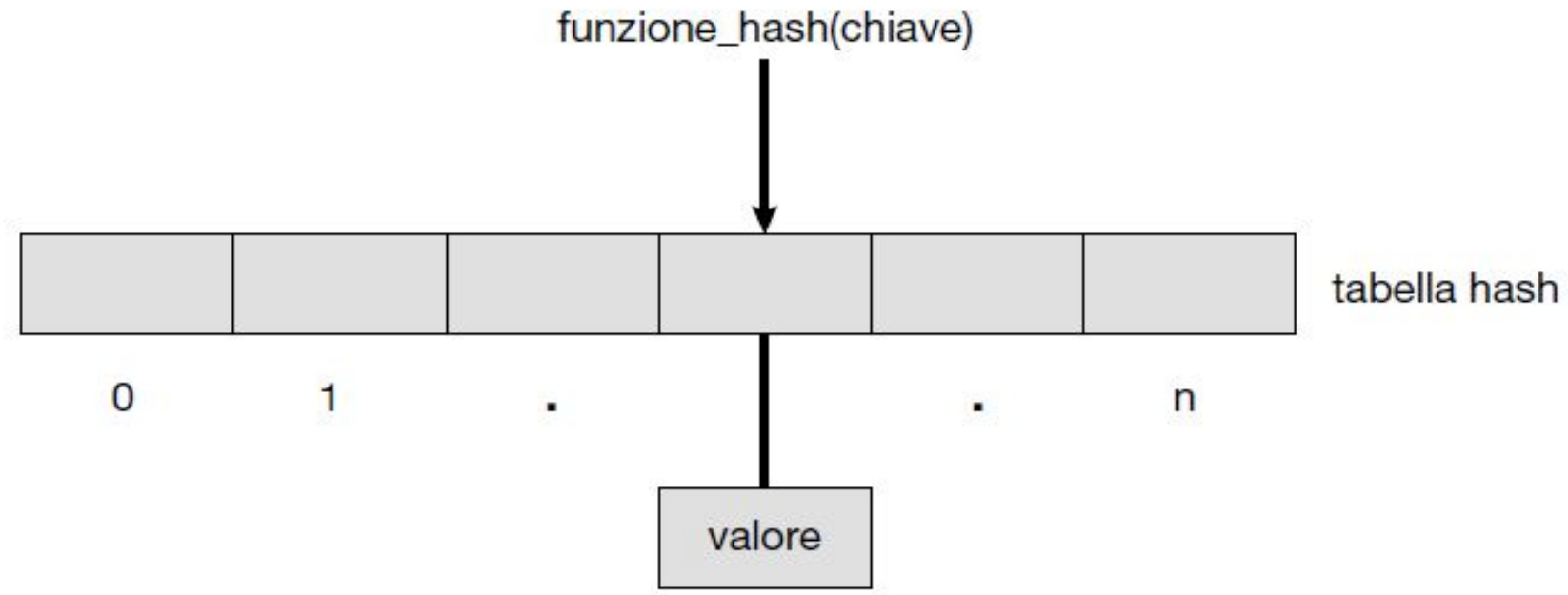


Figura 1.21 Tabella hash.

Elaborazione client-server

Un’architettura di rete contemporanea realizza un sistema in cui alcuni **server** soddisfano le richieste dei **sistemi client**

↓

sistemi client-server

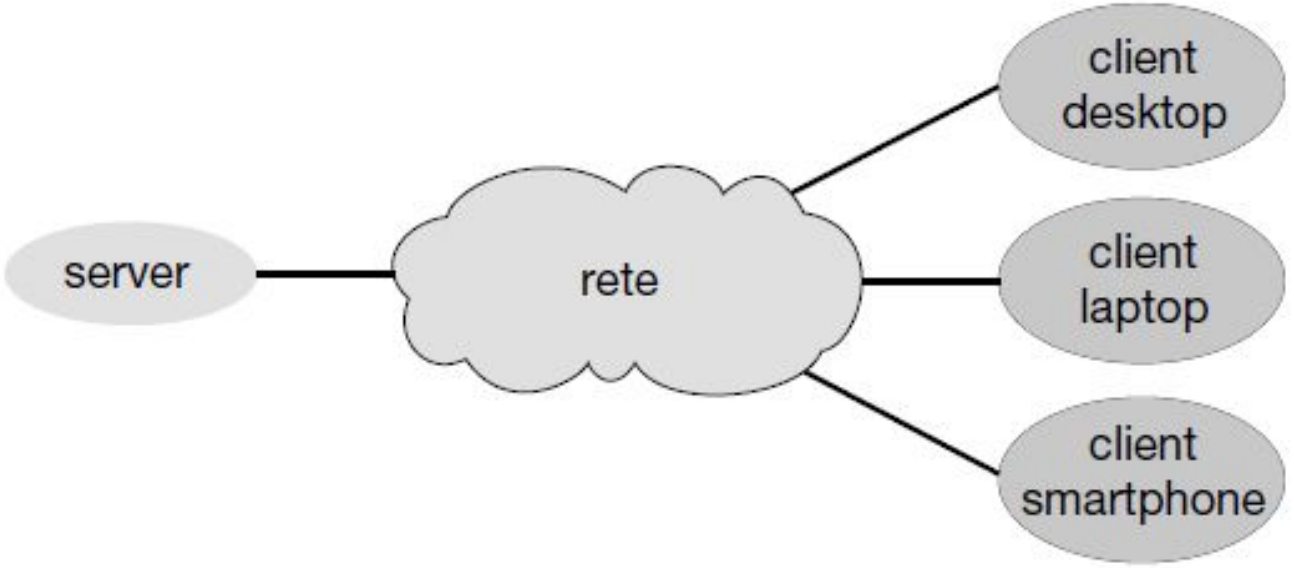


Figura 1.22 Struttura generale di un sistema client-server.

Elaborazione peer-to-peer

peer-to-peer (**P2P**) → cade la distinzione tra client e server; tutti i nodi all'interno del sistema sono su un piano di parità, e ciascuno può fungere ora da client, ora da server, a seconda che stia richiedendo o fornendo un servizio.

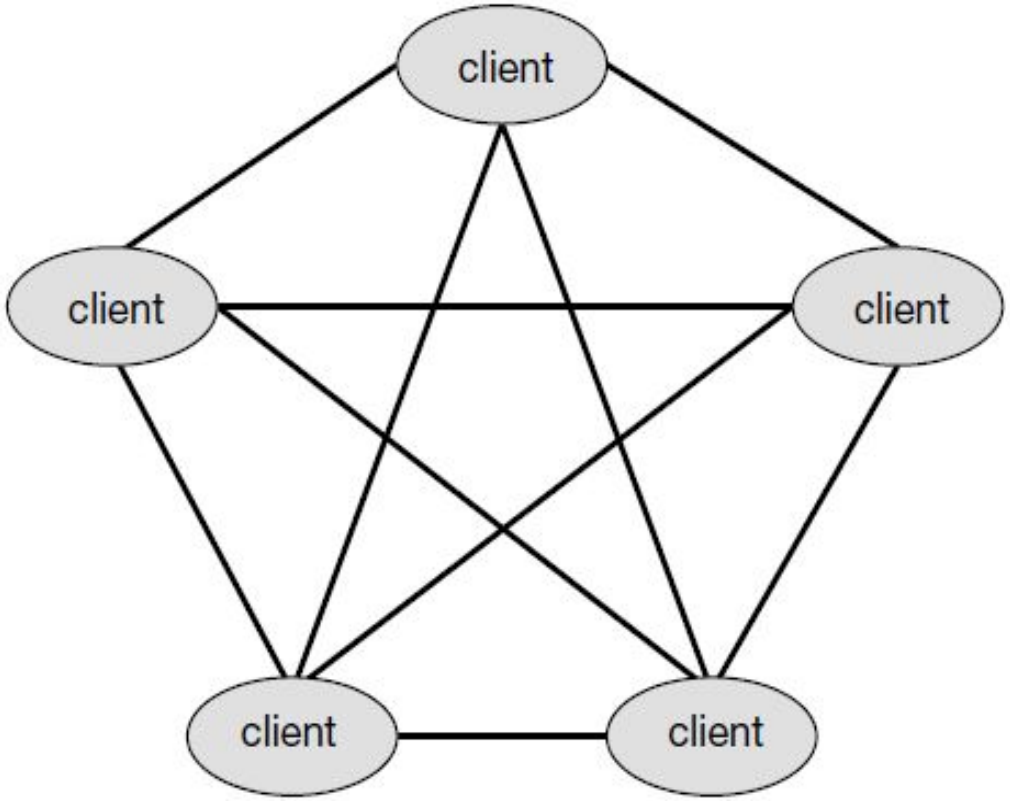
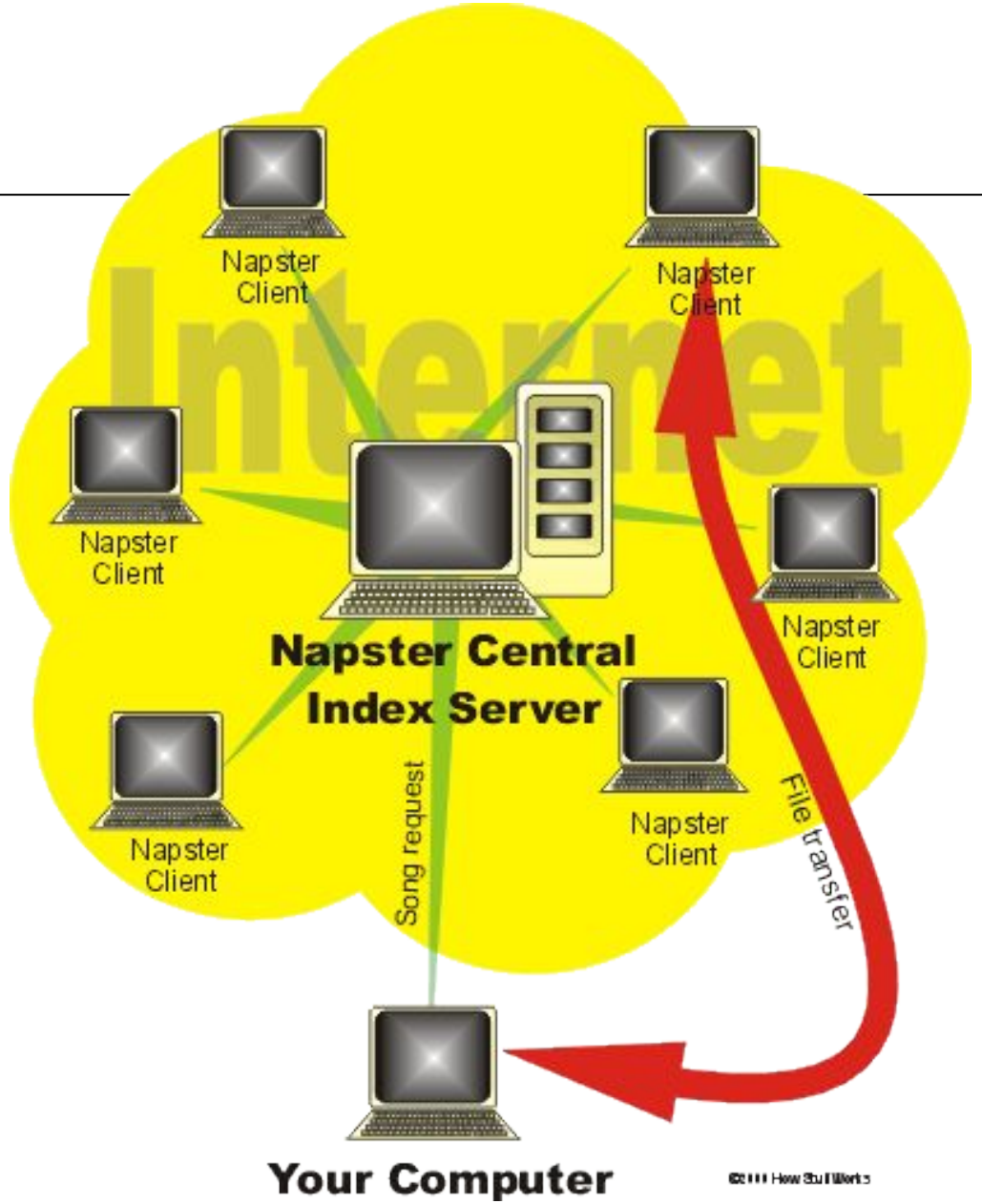


Figura 1.23 Sistema peer-to-peer senza servizi centralizzati.

Napster

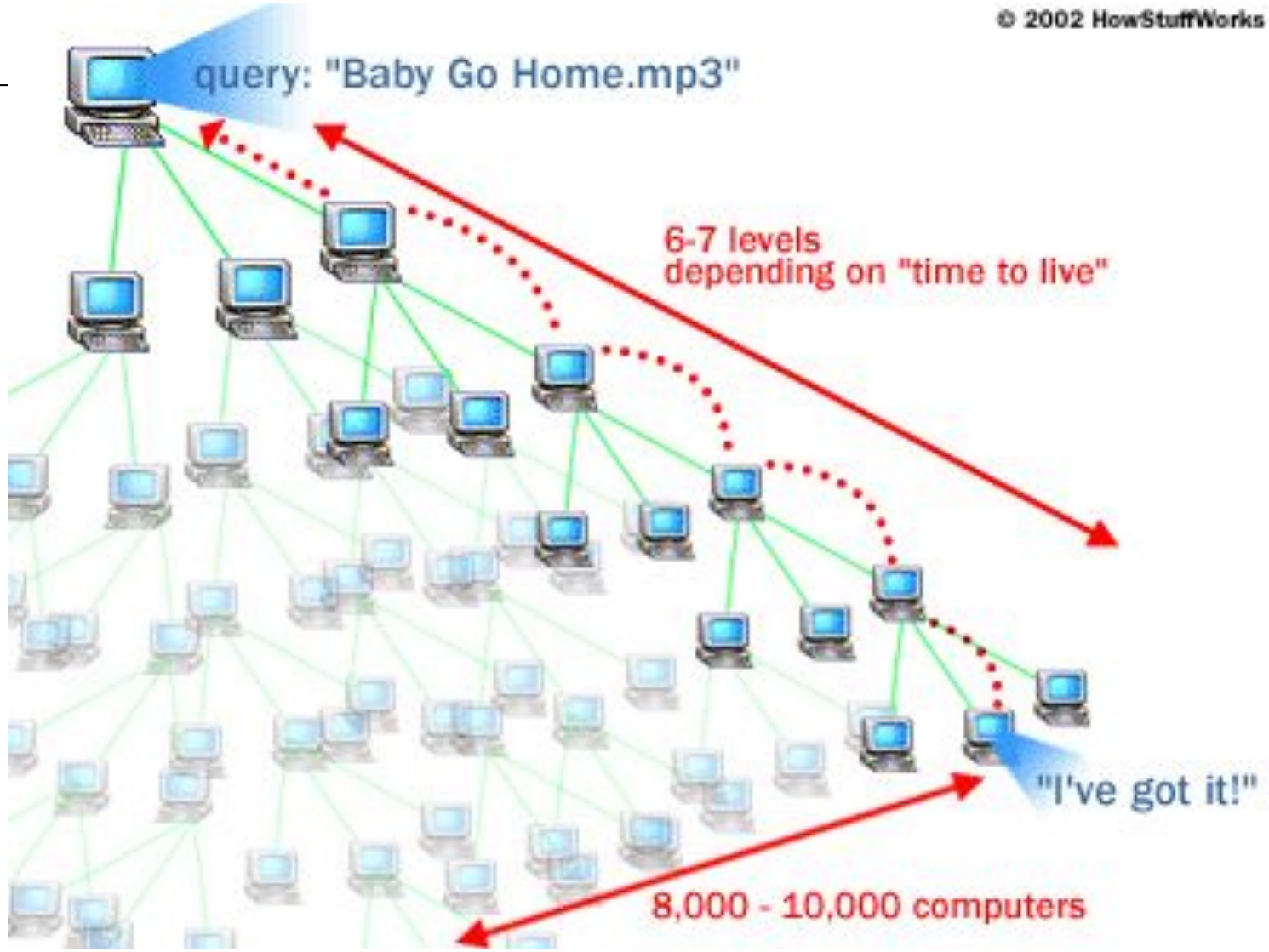


https://computer.howstuffworks.com/file-sharing.htm

https://www.wired.it/attualita/tech/2019/06/01/napster-20-anni-dopo-storia-sean-parker

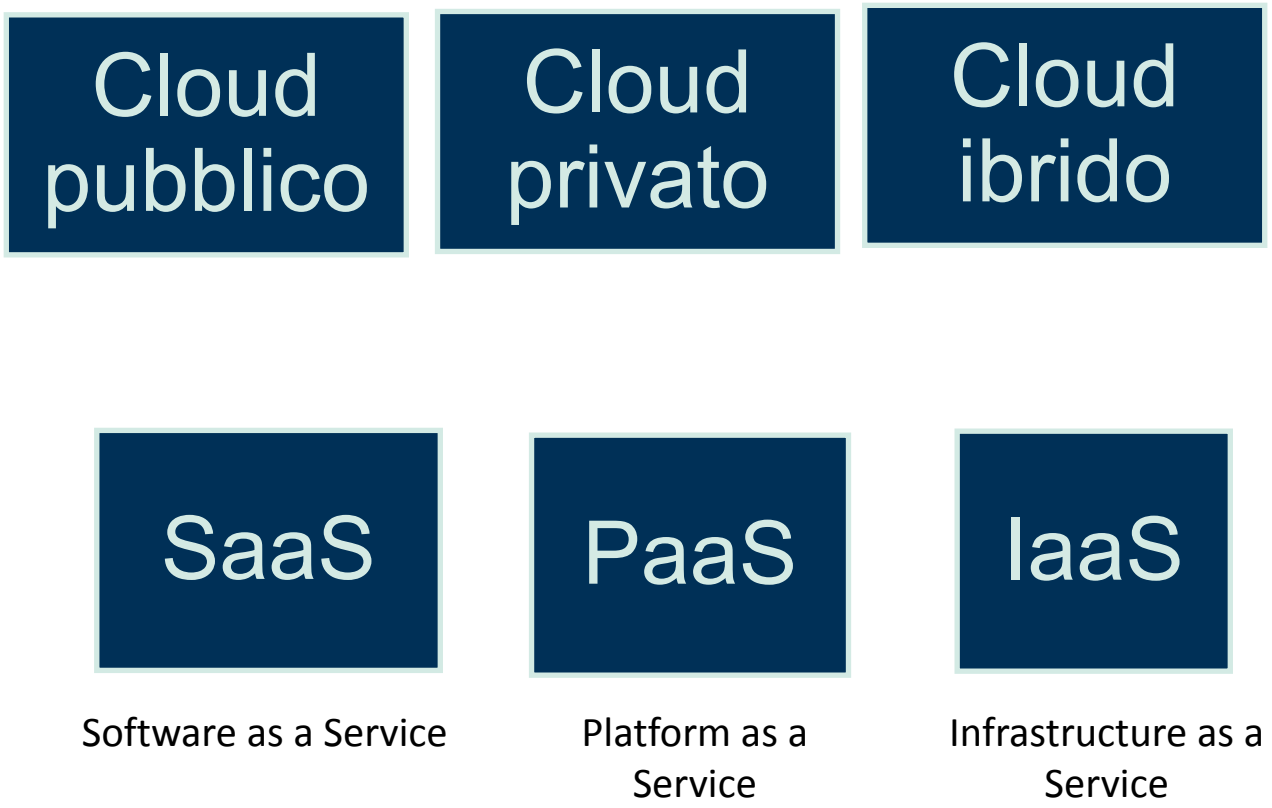
Gnutella

LimeWare



https://computer.howstuffworks.com/file-sharing.htm

Cloud computing



Cloud computing

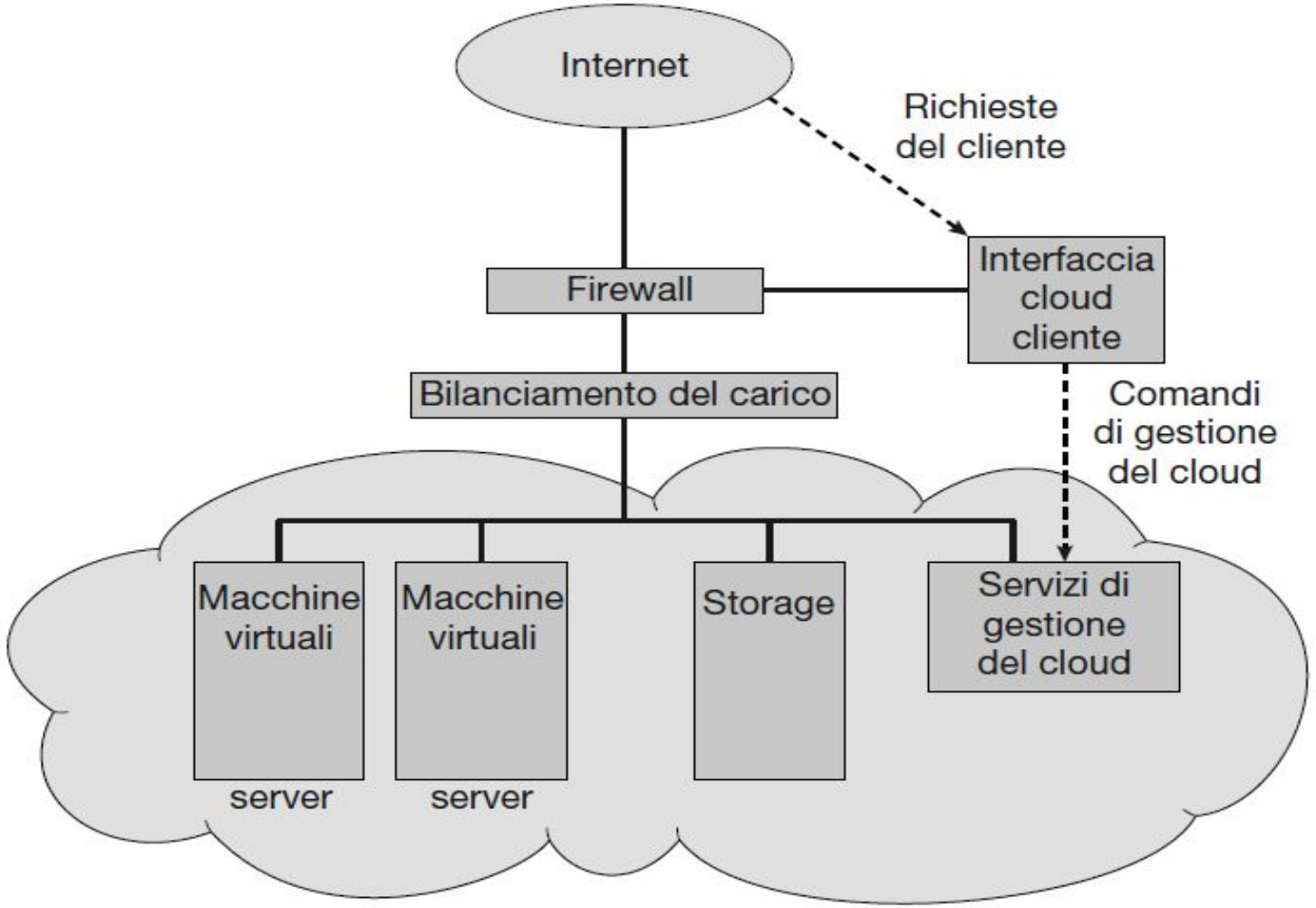


Figura 1.24 Cloud computing.

Sistemi operativi liberi e open-source

Sia i **sistemi operativi liberi** sia i **sistemi operativi open-source** sono disponibili in formato sorgente anziché come codice binario compilato.

Software libero → non solo rende il codice sorgente disponibile, *ma è* anche dotato di una licenza che consente l'uso, la ridistribuzione e la modifica senza costi.

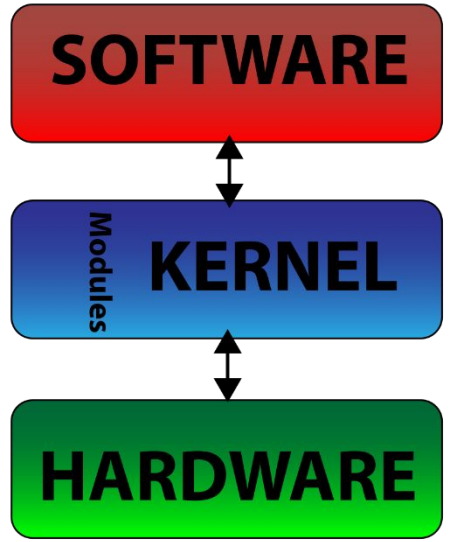
Il **software open-source** non offre necessariamente tale licenza

- **GNU/Linux**, **FreeBSD** e **Solaris** sono esempi diffusi di sistemi operativi open-source
- Microsoft Windows è un **software proprietario**

UNIVERSITÀ DEGLI STUDI DELLA BASILICATA

Corso di Sistemi Operativi

Introduzione



Docente:
Domenico Daniele Bloisi