

Previously on Tecniche della Programmazione

Hardware: Struttura fisica del calcolatore, componenti meccaniche, elettriche, elettroniche, magnetiche ...

Software: il programmi che il calcolatore ha o puo' avere in esecuzione

(ware = merce, articolo di mercanzia ... oggetto, manufatto ...)

abbiamo visto

- una "definizione dal punto di vista dell'utente" (MACCHINA PROGRAMMABILE)
- ed una "definizione dal punto di vista funzionale" (...)

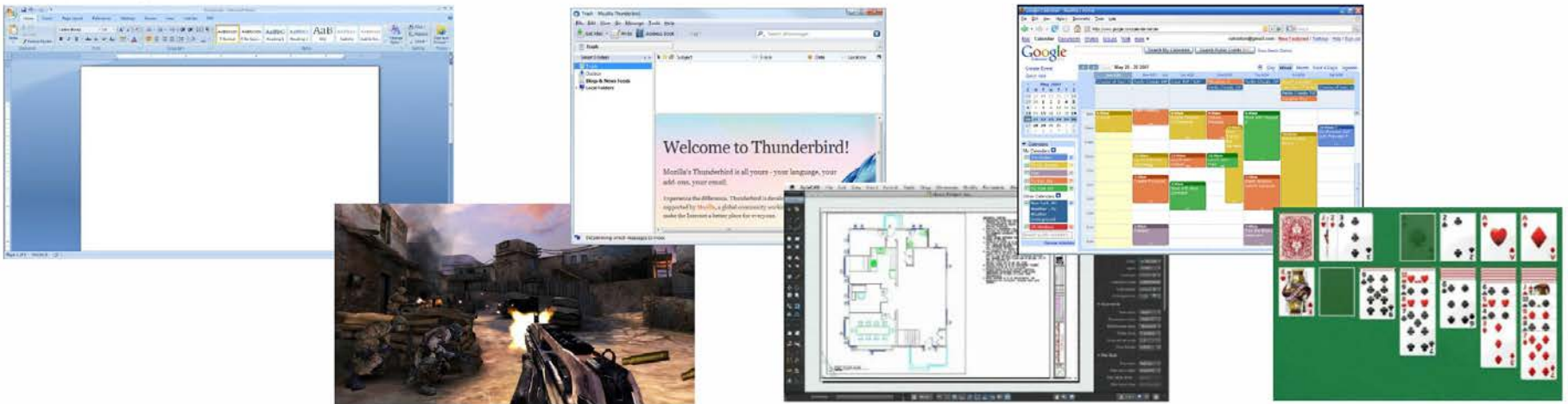


Previously on Tecniche della Programmazione

CALCOLATORE (o ELABORATORE o COMPUTER)

CHE COSA È ? UNA MACCHINA PROGRAMMABILE, OVVERO IN GRADO DI ESEGUIRE PROGRAMMI. DEFINIZIONE DAL PUNTO DI VISTA DELL'UTENTE, CHE INTERAGISCE CON IL CALCOLATORE ATTRAVERSO I PROGRAMMI.

CIASCUN PROGRAMMA CONSENTE DI FAR SVOLGERE AL CALCOLATORE DELLE OPERAZIONI E METTE A DISPOSIZIONE DELL'UTENTE DEGLI STRUMENTI CHE L'UTENTE PUÒ UTILIZZARE PER PERSEGUIRE UN CERTO SCOPO.



Previously on Tecniche della Programmazione

CALCOLATORE (o ELABORATORE o COMPUTER)

CHE COSA È ? UN SISTEMA, OVVERO UN OGGETTO COSTITUITO DA UN INSIEME DI COMPONENTI CHE INTERAGISCONO FRA LORO PER OTTENERE UN CERTO COMPORTAMENTO.

DEFINIZIONE DAL PUNTO DI VISTA FUNZIONALE.

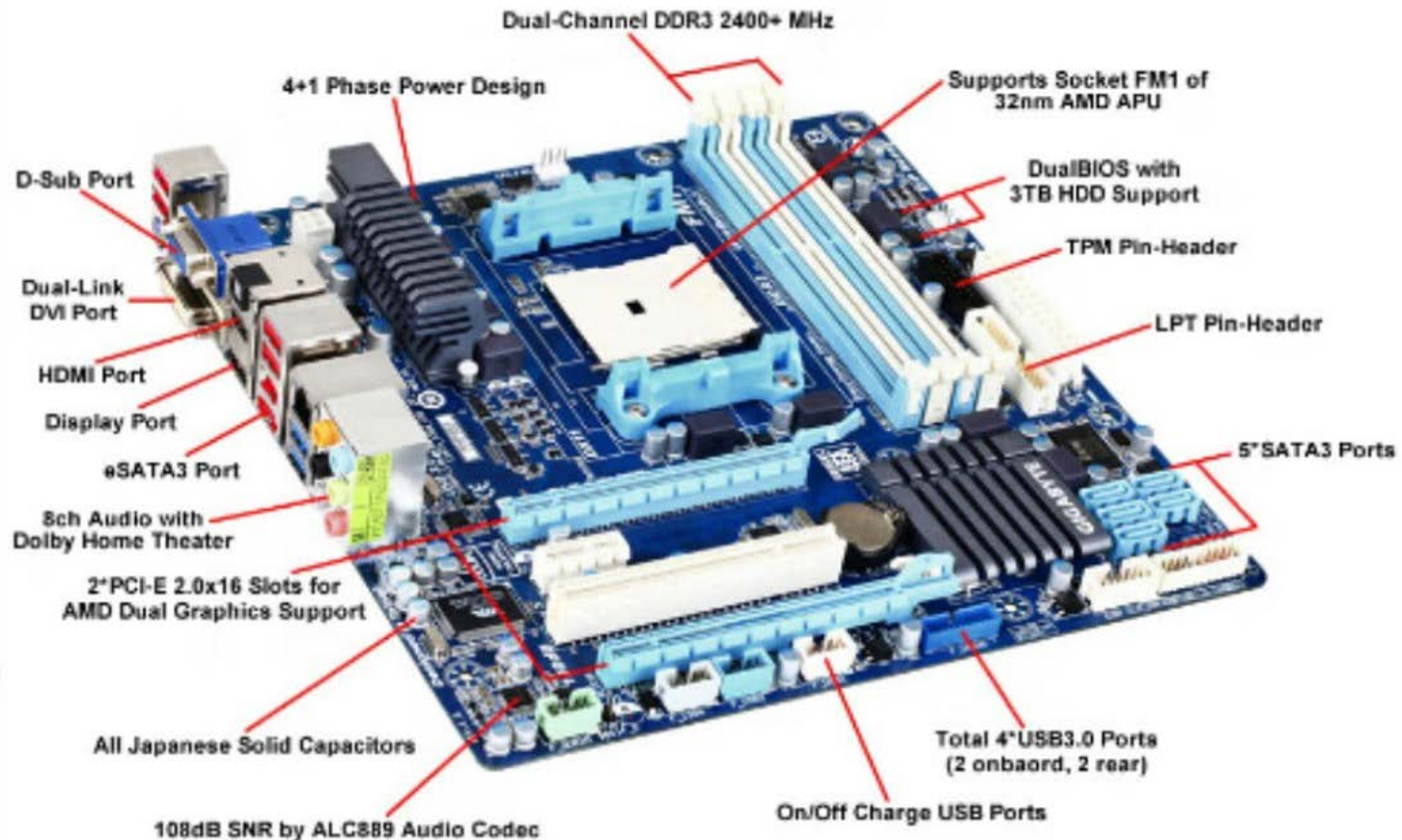


Previously on Tecniche della Programmazione

CALCOLATORE (o ELABORATORE o COMPUTER)

CHE COSA È ? UN SISTEMA, OVVERO UN OGGETTO COSTITUITO DA UN INSIEME DI COMPONENTI CHE INTERAGISCONO FRA LORO PER OTTENERE UN CERTO COMPORTAMENTO.

DEFINIZIONE DAL PUNTO DI VISTA FUNZIONALE.



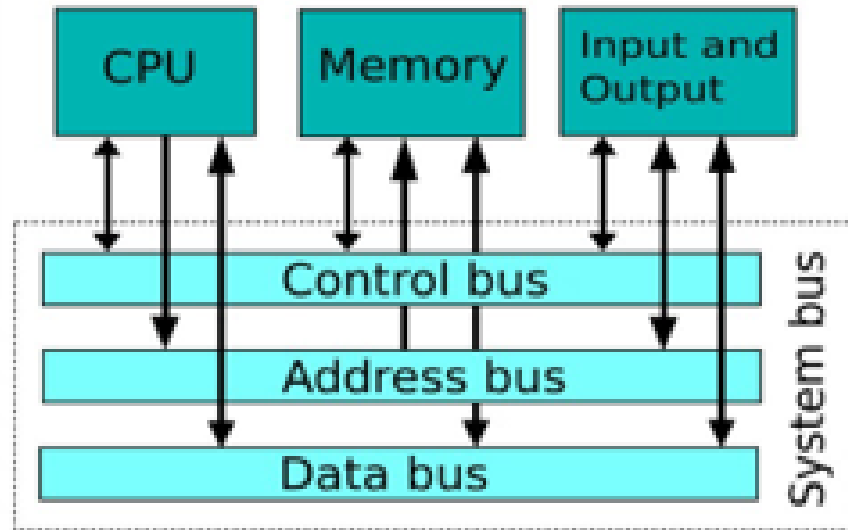
And now

- **Software di Base e Software Applicativo**
- **Architettura del calcolatore (Macchina di Von Neumann)**
 - **MEMORIA CENTRALE**
 - **CPU (Central Processing Unit)**
 - **BUS**
 - **(Interfacce per le) PERIFERICHE**
- **Linguaggio Macchina**
 - **Esempio di linguaggio macchina**
 - **Programma scritto in linguaggio macchina**
 - **Esecuzione di un programma scritto in linguaggio macchina**
- **Programma, processo, linguaggio, prima esperienza di programmazione**

Architettura del Calcolatore

... describe

QUALI sono le componenti funzionali del sistema
si dice anche le "risorse" del computer



... e

COME INTERAGISCONO fra loro queste componenti

Software di Base

mainly ... Operating System

programmi ... pronti a funzionare al momento giusto, o funzionanti in continuazione

Propone una

Visione VIRTUALIZZATA delle risorse del computer:

così, chi vuole usare le risorse del computer può chiederlo mediante comandi prestabiliti, senza conoscere il funzionamento effettivo delle componenti

Software Applicativo

Singoli programmi (*Applicazioni Software*) che sono stati aggiunti (istallati) nel computer

- possono essere eseguiti dal computer ... basta un click ... o due
- per funzionare e interagire con l'utente, **usano le risorse del computer**
- e per usare le risorse dialogano con il (chiedono al) **Software di Base**

vedi **esempio nella prossima slide**

Software di Base

mainly ... Operating System

Tanti programmi, già disponibili sul computer e pronti a funzionare (vengono caricati nella memoria al momento del "bootstrap").

Usano le risorse del computer, o meglio ... consentono di usarle, ad esempio ... vedi sotto

Il tutto **attraverso una visione semplificata (VIRTUALIZZATA) delle risorse del computer**: l'utente può usarle senza conoscerne il funzionamento concreto

Usi delle risorse?

- utente (DIR, click su una cartella, click per muovere cose ... click per eseguire un programma applicativo, TYPE per stampare il contenuto di un file testuale, spostamento di qualcosa nel cestino ...
- Software Applicativo ... usa altri comandi, meno chiari ... per accedere alla memoria, scrivere file, muovere cose sullo schermo ... ed eseguire le proprie funzionalità (scrivere file, vedere un pdf, giocare a PacMan, o a Call of Duty Modern Warfare ...

Software Applicativo

Singoli programmi (*Applicazioni Software*) che permettono di usare il computer per svolgere attività dedicate: scrivere un documento, leggere le email, navigare il web, videogiochi, usare un foglio elettronico, fare le slide per la lezione ...

Questi programmi usano le funzionalità offerte dal Software di base per portare a compimento le proprie ... ad esempio,

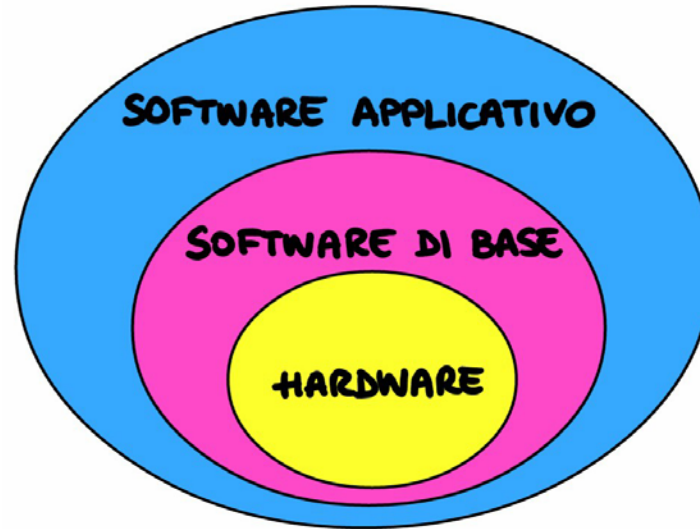
mentre scriviamo un documento, il Word Processor (Applicazione) dialoga con il Sistema Operativo, per **ricevere tramite tastiera** (unità periferica di Input) **quel che scrivo**;

per "salvare" il documento, per futuri usi, noi chiediamo di salvare, e il Word Processor dialoga con il SO, per **eseguire la memorizzazione** dei dati del documento, da Memoria Centrale a **Memoria di Massa** (unità periferica di memoria)

Livelli funzionali

ciascun LIVELLO offre un insieme di funzionalita`

ciascun LIVELLO realizza le proprie funzionalita` usando le funzionalita` proprie e quelle del livello precedente



le funzionalita` di ciascun LIVELLO sono realizzate mediante il linguaggio di quel livello, che permette di scrivere frasi in cui si chiamano ad essere eseguite funzionalita` di quel livello (che possono anche chiamare quelle del livello sottostante)

Macchina (Architettura) di Von Neumann



Un modello di architettura: il modello semplificato che denota il funzionamento del calcolatore e le relazioni tra le sue componenti.

- Il calcolatore riceve **dati in ingresso (DATI DI INPUT)** e "li trasforma" in **dati in uscita (DATI DI OUTPUT)**
- I **dati** sono **entità numeriche memorizzate** (cioè contenute, **rappresentate!**) nella **memoria** (una delle componenti) del calcolatore.

(MEMENTO: l'INFORMAZIONE è un dato o un gruppo di dati che hanno una *semantica associata*)



Macchina di Von Neumann



Un modello semplificato ... che ancora vale ...

- Il calcolatore riceve **dati in ingresso** (DATI DI INPUT) e "li trasforma" in **dati in uscita** (DATI DI OUTPUT)
- I dati sono entita` numeriche rappresentate nella memoria (una delle componenti) del calcolatore.
(l'INFORMAZIONE e` un dato o un gruppo di dati che hanno una *semantica associata*)
- La trasformazione (**ELABORAZIONE**) dei dati avviene mediante l'esecuzione di un **programma**.

Diversi **programmi** corrispondono a diverse elaborazioni ...

(Anche i programmi risiedono nella **memoria**, quando devono essere eseguiti)

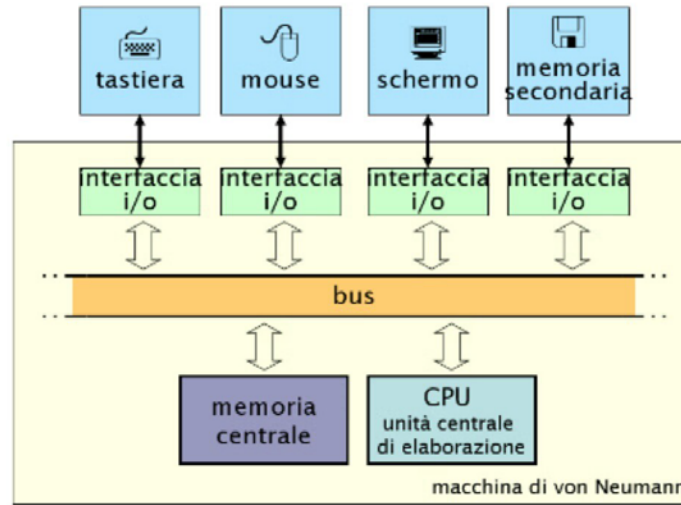


Componenti della Macchina di Von Neumann

1) MEMORIA : MEMORIZZA E FORNISCE L'ACCESSO A DATI E PROGRAMMI.

3) INTERFACCE I/O : COMPONENTI DI COLLEGAMENTO CON LE PERIFERICHE DEL CALCOLATORE.

ATTENZIONE : MOUSE, TASTIERA, SCHERMO, ALTOPARLANTI, DISCO, ...
NON FANNO PARTE DELLA MACCHINA IN QUESTO MODELLO.



2) CPU o PROCESSORE : ESEGUE LE ISTRUZIONI PER L'ELABORAZIONE DEI DATI. INOLTRE CONTROLLA E COORDINA IL FUNZIONAMENTO DELLE ALTRE PARTI.

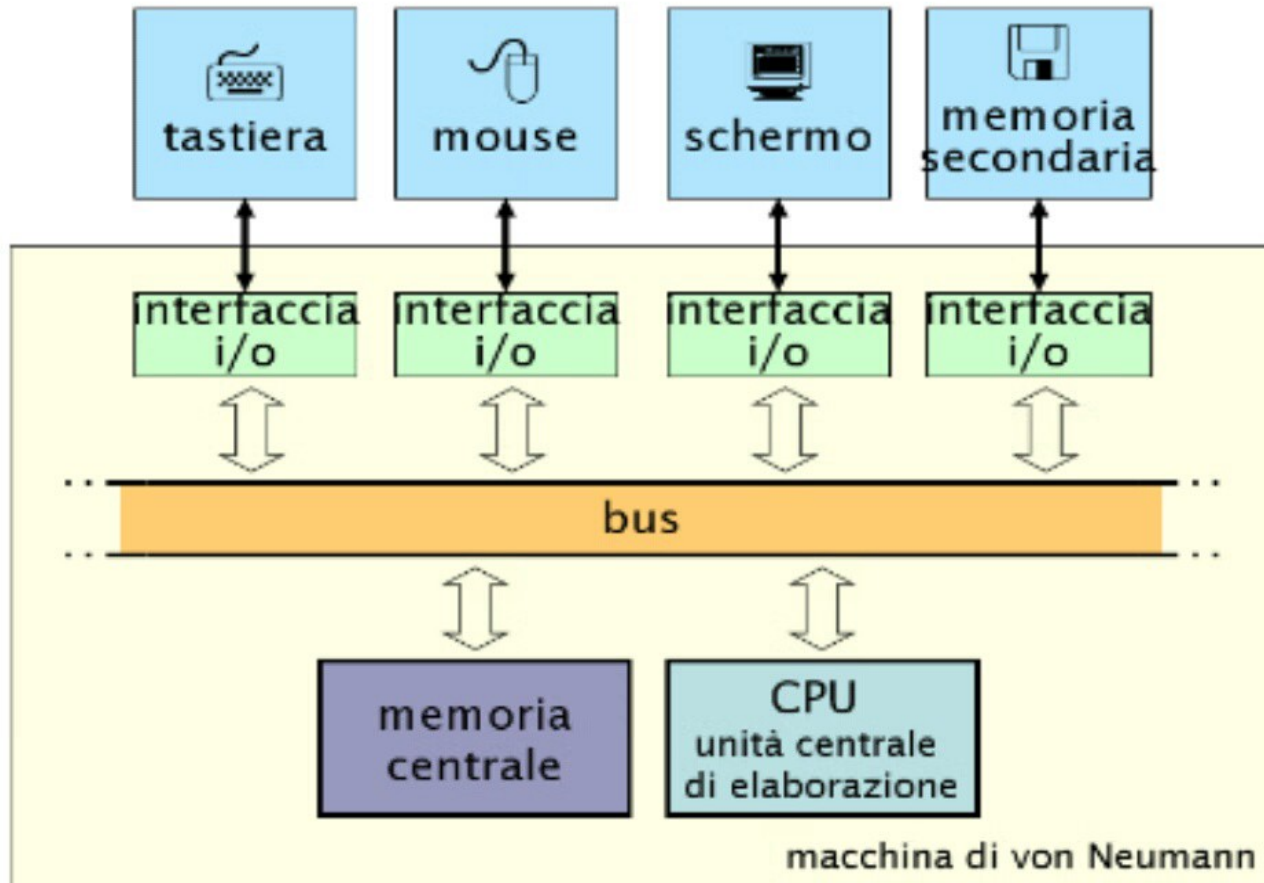
4) BUS : SVOLGE LA FUNZIONALITÀ DI TRASFERIMENTO DI DATI ED INFORMAZIONI DI CONTROLLO TRA LE ALTRE COMPONENTI.

Componenti della Macchina di Von Neumann

1) MEMORIA: MEMORIA
E FORNISCE L'
A DATI E PRO

3) INTERFACCE
COMPONENTI
GAMENTO CON
PERIFERICHE

ATTENZIONE
SCHERMO, ALT
NON FANNO
IN QUESTO



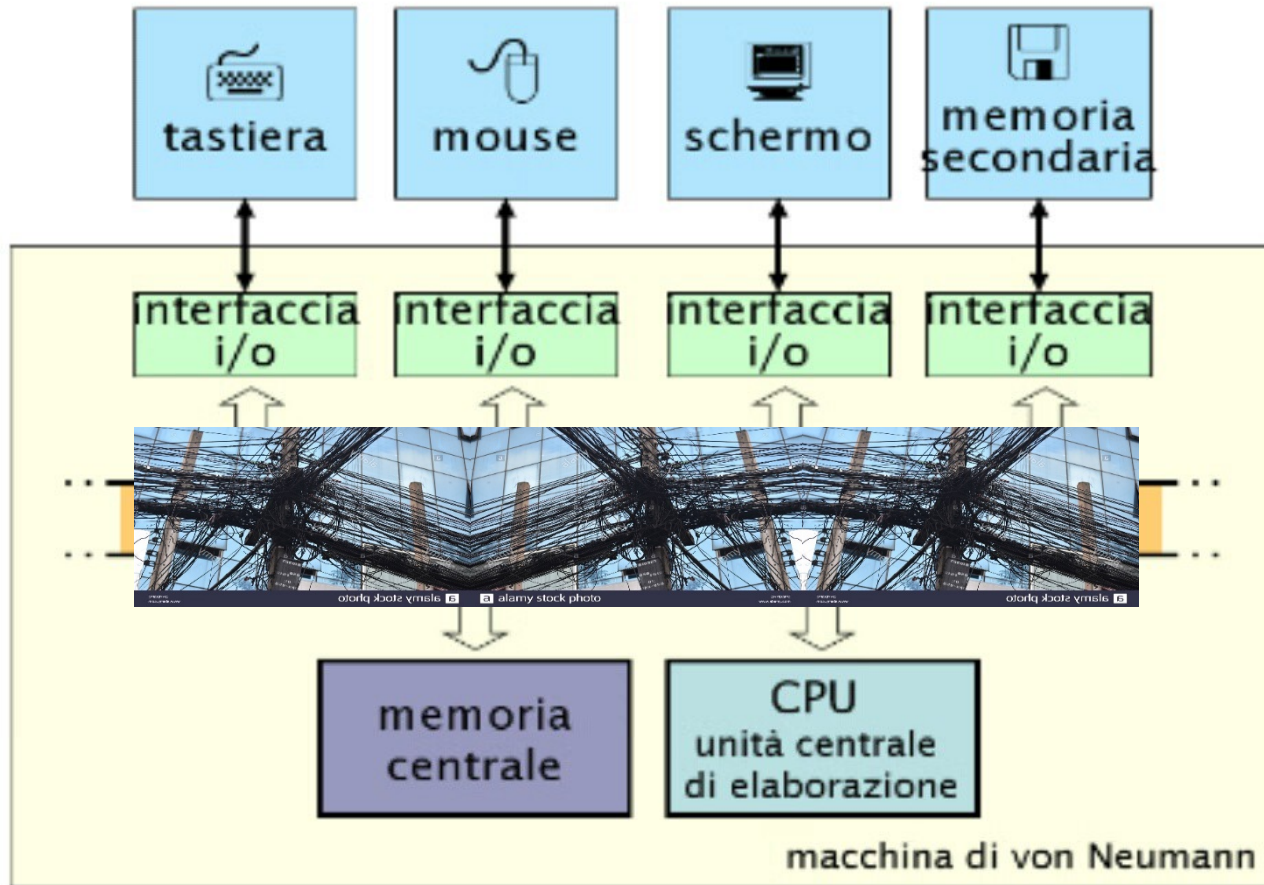
ESSORE:
TRUZIONI
RAZIONE
OLTRE
COORDINA
LENTO DELLE
DI
FORMAZIONI
COMPONENTI.

Componenti della Macchina di Von Neumann

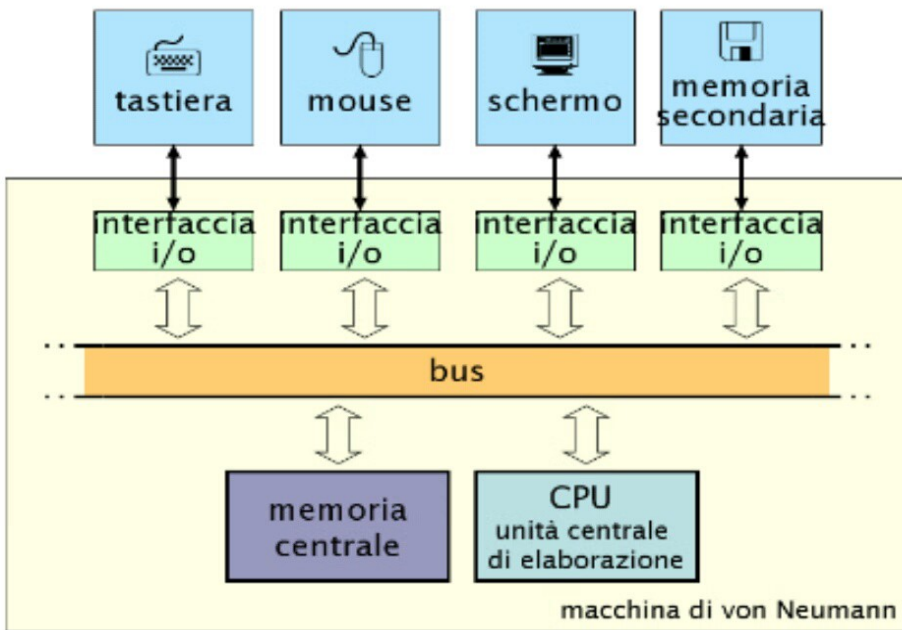
1) MEMORIA: MEMORIA
E FORNISCE L'
A DATI E PRO

3) INTERFACCE
COMPONENTI
GAMENTO CON
PERIFERICHE

ATTENZIONE
SCHERMO, ALT
NON FANNO
IN QUESTO



ESSORE:
TRUZIONI
RAZIONE
OLTRE
COORDINA
IENTO DELLE
DI
FORMAZIONI
COMPONENTI.



Recap: Funzionalità delle varie componenti:

Elaborazione dei dati

CPU

Coordinamento delle elaborazioni e degli spostamenti di dati

CPU + BUS

Memorizzazione dati

avviene nella Memoria Centrale (MC)

Accesso/Memorizzazione dati in MC

CPU + BUS

Memorizzazione nella Memoria Esterna (secondaria, o "di massa")

CPU + MC + BUS + Interfaccia I/O

Trasferimento dati da e per I/O

CPU + MC + BUS + Interfaccia I/O

Memoria Centrale

Consente di registrare (MEMORIZZARE ...) dati/istruzioni in modo temporaneo:

VOLATILE



- MEMORIZZAZIONE

- ACCESSO

La chiamano RAM: Random Access Memory;

OGNI LOCAZIONE DI MEMORIA HA UN indirizzo

"La CPU accede direttamente alla RAM, attraverso il BUS"

Memoria Centrale

Consente di registrare (MEMORIZZARE ...) dati/istruzioni in modo temporaneo: il contenuto scompare quando la memoria non e' alimentata (la memoria e' *VOLATILE*)



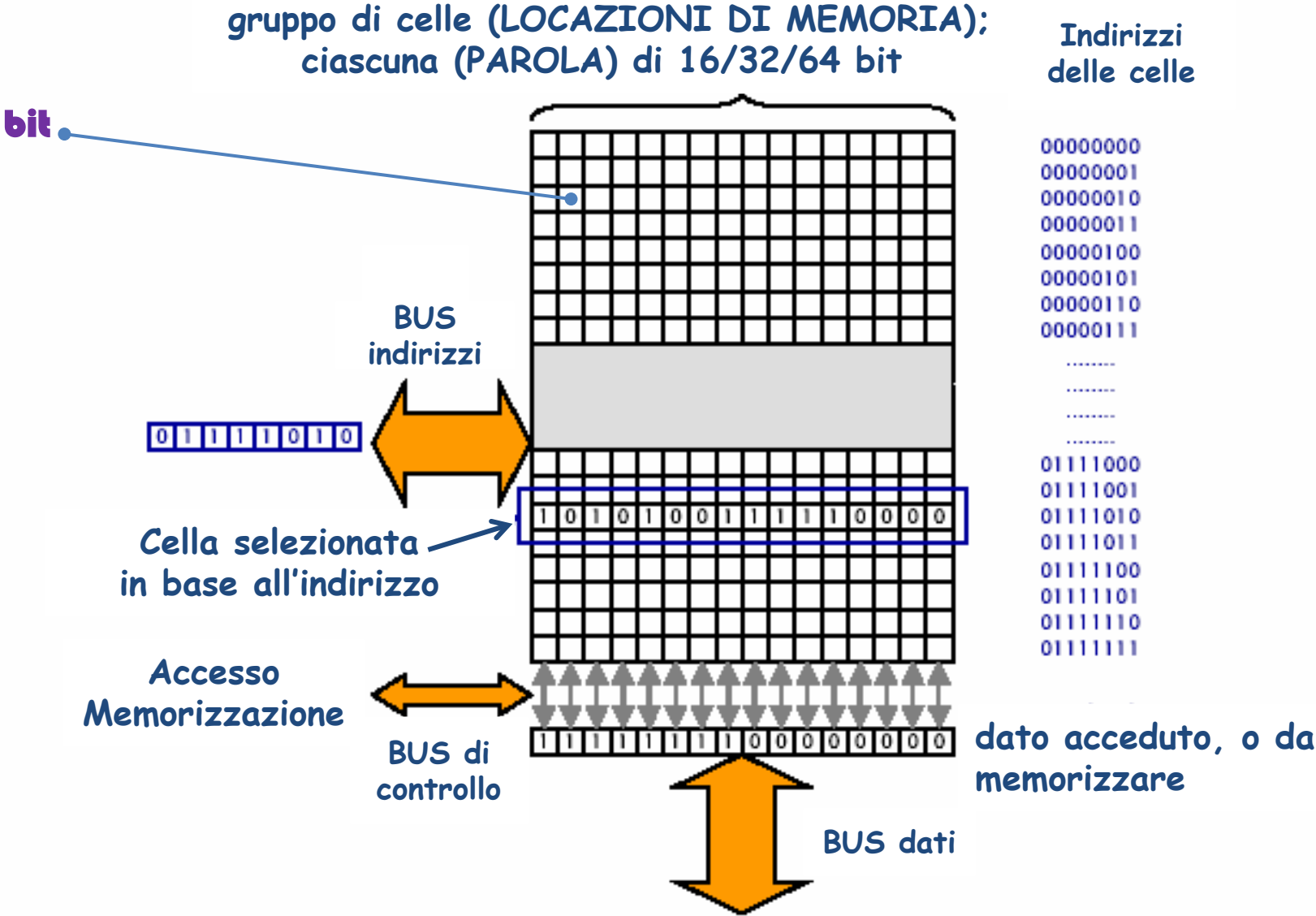
- Operazione di **MEMORIZZAZIONE**: un dato/istruzione viene copiato in un'area di memoria (Locazione di Memoria)
- Operazione di **ACCESSO**: il valore contenuto in una Locazione di memoria viene "acceduto", ad esempio viene consultato per copiarlo altrove al fine di usarlo per qualche altra operazione

La chiamano **RAM**: Random Access Memory; non vuol dire che si accede a caso ... vuol dire che le operazioni di accesso e memorizzazione sono possibili ovunque, senza un ordine obbligatorio che imponga di visitare un'area prima di un'altra a priori.


Le **LOCAZIONI** di memoria sono identificate da un **indirizzo**, un numero Mediante il suo indirizzo, una Locazione puo' essere indicata univocamente, per chiedere di effettuarvi un accesso o una memorizzazione.

"La CPU accede direttamente alla RAM, attraverso il BUS"

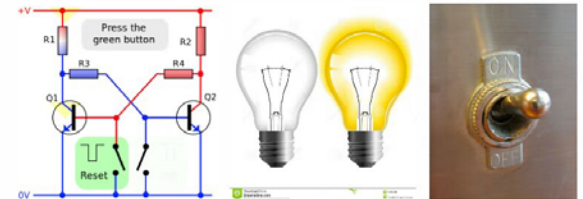
Struttura semplificata della memoria centrale



Qualche particolare

COSTITUITA DA UNO O PIÙ **CHIP** , CIASCUNO DEI QUALI CONTIENE UN NUMERO ELEVATISSIMO DI **BIT** (LE RAM TIPICHE ATTUALI HANNO QUALCHE DECINA O UN CENTINAIO DI MILIARDI DI BIT).

BIT: ELEMENTO BASE DELLA MEMORIA, PUÒ RAPPRESENTARE **2 VALORI** (IN GENERE INDICATI DA **0** E **1**). UN BIT È **FISICAMENTE REALIZZATO** DA UN QUALCHE TIPO DI DISPOSITIVO CHE HA **2 STATI** (PER ESEMPIO LE DUE POSIZIONI DI UN INTERRUTTORE, I DUE LIVELLI DI VOLTAGGIO O DI CORRENTE DI UN CIRCUITO, DUE LIVELLI DISTINTI DI INTENSITÀ LUMINOSA, DUE DIREZIONI DI MAGNETIZZAZIONE...)



Multipli del byte					
Prefissi SI			Prefissi binari		
Nome	Simbolo	Multiplo	Nome	Simbolo	Multiplo
kilobyte	kB	10^3	kibibyte	KiB	2^{10}
megabyte	MB	10^6	mebibyte	MiB	2^{20}
gigabyte	GB	10^9	gibibyte	GiB	2^{30}
terabyte	TB	10^{12}	tebibyte	TiB	2^{40}
petabyte	PB	10^{15}	pebibyte	PiB	2^{50}
exabyte	EB	10^{18}	exbibyte	EiB	2^{60}
zettabyte	ZB	10^{21}	zebibyte	ZiB	2^{70}
yottabyte	YB	10^{24}	yobibyte	YiB	2^{80}

<https://en.wikipedia.org/wiki/Kilobyte>

stop, hai già visto l'esecuzione del programma nell'ultima parte della lezione 1?

Organizzazione della Memoria

I BIT DELLA SONO RAGGRUPPATI IN CELLE O

UNA CELLA HA UNA DIMENSIONE FISSA (LA DIMENSIONE TIPICA ATTUALE È BIT).
UNA CELLA CORRISPONDE AD UNA , OVERO RAPPRESENTA UNA
PORZIONE DI MEMORIA CHE VIENE (TIPICAMENTE) UTILIZZATA INTERAMENTE (OVERO
LETTURE, SCRITTURE E TRASFERIMENTI VENGONO EFFETTUATI IN GRUPPI DI 32 O 64 BIT)

Una cella contiene un ... (LA SEQUENZA DI BIT MEMORIZZATA AL SUO INTERNO).

UNA CELLA HA UN CHE PERMETTE AL PROCESSORE DI RIFERIRSI A QUELLA
CELLA PER EFFETTUARE UNA LETTURA O UNA SCRITTURA.

Organizzazione della Memoria

I BIT DELLA MEMORIA CENTRALE SONO RAGGRUPPATI IN CELLE O LOCAZIONI DI MEMORIA.

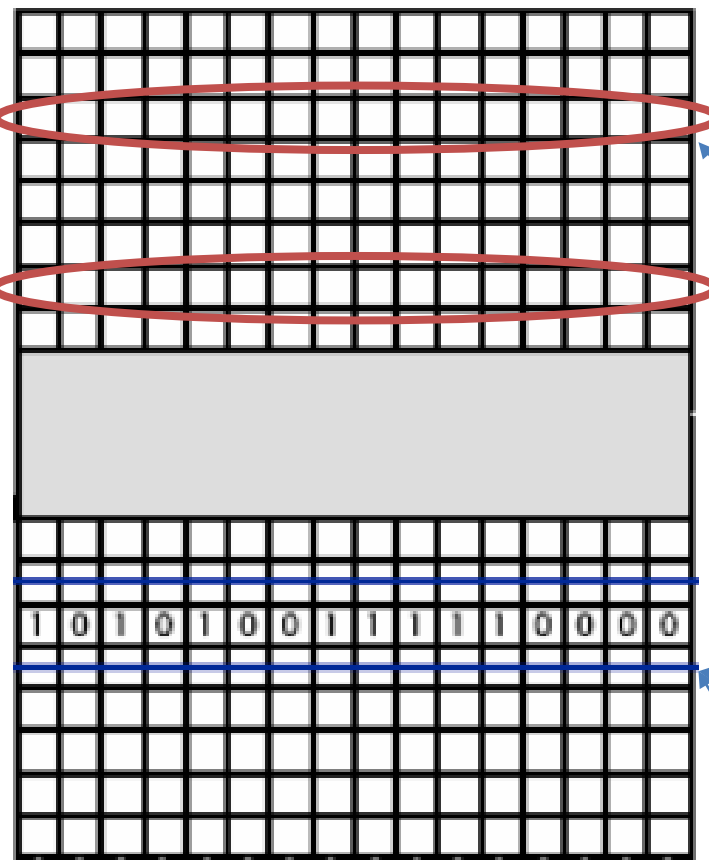
UNA CELLA HA UNA DIMENSIONE FISSA (LA DIMENSIONE TIPICA ATTUALE È 32 O 64 BIT).
UNA CELLA CORRISPONDE AD UNA PAROLA DI MEMORIA, OVVERO RAPPRESENTA UNA PORZIONE DI MEMORIA CHE VIENE (TIPICAMENTE) UTILIZZATA INTERAMENTE (OVVERO LETTURE, SCRITTURE E TRASFERIMENTI VENGONO EFFETTUATI IN GRUPPI DI 32 O 64 BIT)

UNA CELLA HA UN VALORE (LA SEQUENZA DI BIT MEMORIZZATA AL SUO INTERNO).

UNA CELLA HA UN INDIRIZZO CHE PERMETTE AL PROCESSORE DI RIFERIRSI A QUELLA CELLA PER EFFETTUARE UNA LETTURA O UNA SCRITTURA.

Si', ma ... Cosa c'e' in una cella?

What the hell,
 instructions, numbers ...
 in a word DATA;
 in another word INFORMATION



00000000 (Load)

codice operazione	IND	REG
-------------------	-----	-----

istr. macchina (load) da memoria a registro

0000000010 (add)

Codice operazione	R1 Reg1	R2 Reg2
-------------------	------------	------------

istr. macchina per operazione tra due registri

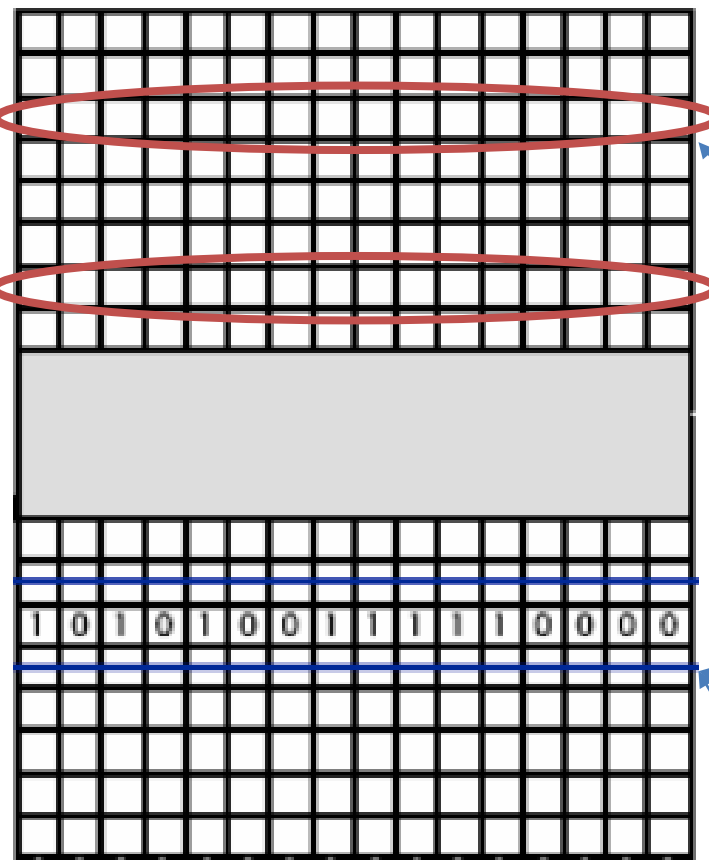
1010100111110000
0000000000000000
1111000000000000
0000111100001111

1010100111110000
0000000000000000

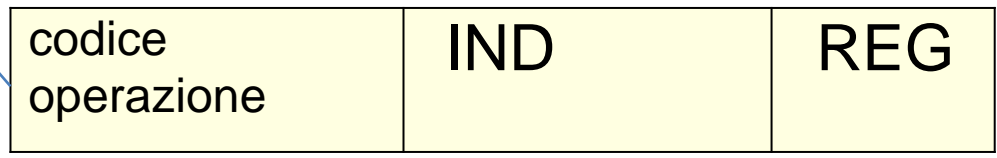
numero:

Si', ma ... Cosa c'e' in una cella?

What the hell,
 instructions, numbers ...
 in a word DATA;
 in another word INFORMATION

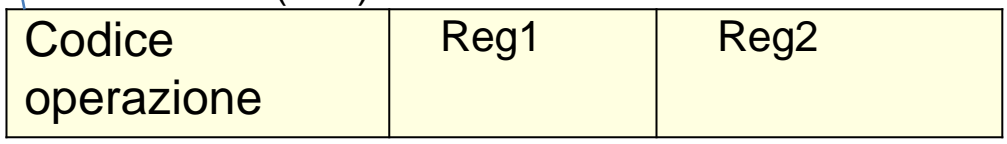


00000000 (Load)

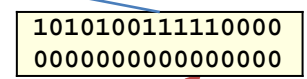
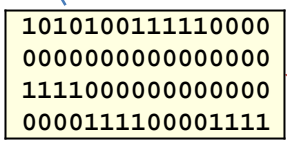


istr. macchina (load) da memoria a registro

0000000010 (add)



istr. macchina per operazione tra due registri



numero: un dato, che sicuramente rappresenta un'informazione ...; in questi casi un numero reale rappresentato con segno, esponente e mantissa, oppure un intero rappresentato in complement a due su 32 bit, oppure un intero positivo su 16 bit, oppure due caratteri ...

Bus

COMPONENTE DEL CALCOLATORE DEDICATO AL TRASFERIMENTO DI DATI (BUS DATI), DI INDIRIZZI (BUS INDIRIZZI) E DI INFORMAZIONI DI CONTROLLO (BUS DI CONTROLLO) FRA I COMPONENTI DEL CALCOLATORE. È L'INSIEME DEI COLLEGAMENTI SU CUI VENGONO TRASFERITI DATI, INDIRIZZI ED INFORMAZIONI DI CONTROLLO.



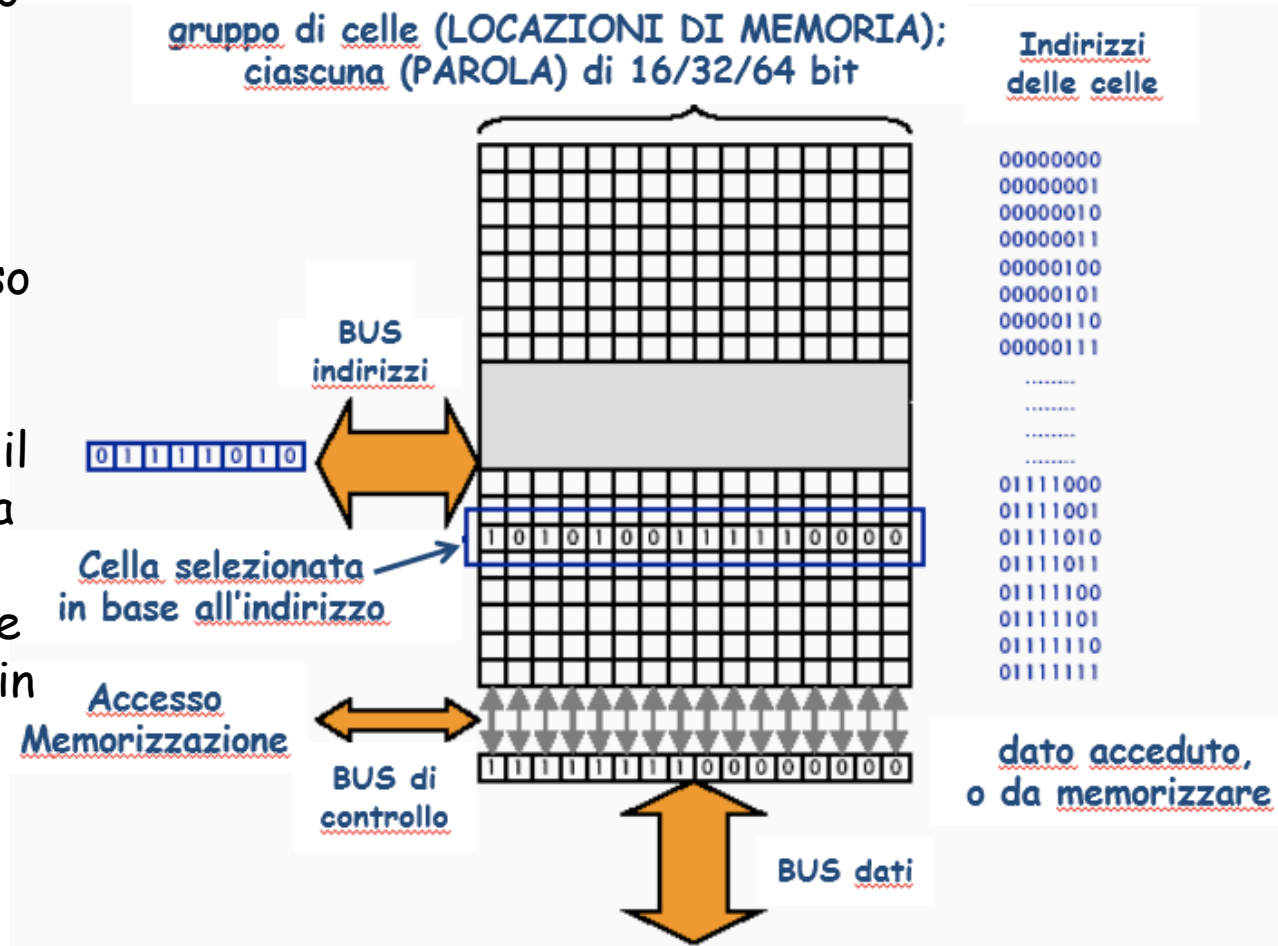
TUTTI I COMPONENTI SONO COLLEGATI AL BUS!

MANIERA MODULARE ED ESPANDIBILE DI COLLEGARE TUTTI I COMPONENTI DI UN CALCOLATORE. IN PARTICOLARE, QUANDO VIENE AGGIUNTO UN NUOVO COMPONENTE AL CALCOLATORE, È SUFFICIENTE COLLEGARLO AL BUS PER METTERLO IN COMUNICAZIONE CON TUTTI GLI ALTRI COMPONENTI.

Funzionamento di ACCESSO e MEMORIZZAZIONE

In ACCESSO, il processore invia l'indirizzo della cella da visitare, attraverso il BUS INDIRIZZI. E il valore acceduto è reso disponibile attraverso il BUS DATI.

In MEMORIZZAZIONE, il processore invia il dato da copiare in memoria, attraverso il BUS DATI, e l'indirizzo della locazione in cui eseguire la copia, attraverso il BUS INDIRIZZI. E il valore viene copiato nella cella indicata.



Ciascun accesso avviene in un tempo INDIPENDENTE DALLA LOCAZIONE (cioè indipendente dall'indirizzo: ... remember, Accesso Diretto: Random Access Memory)

Processore

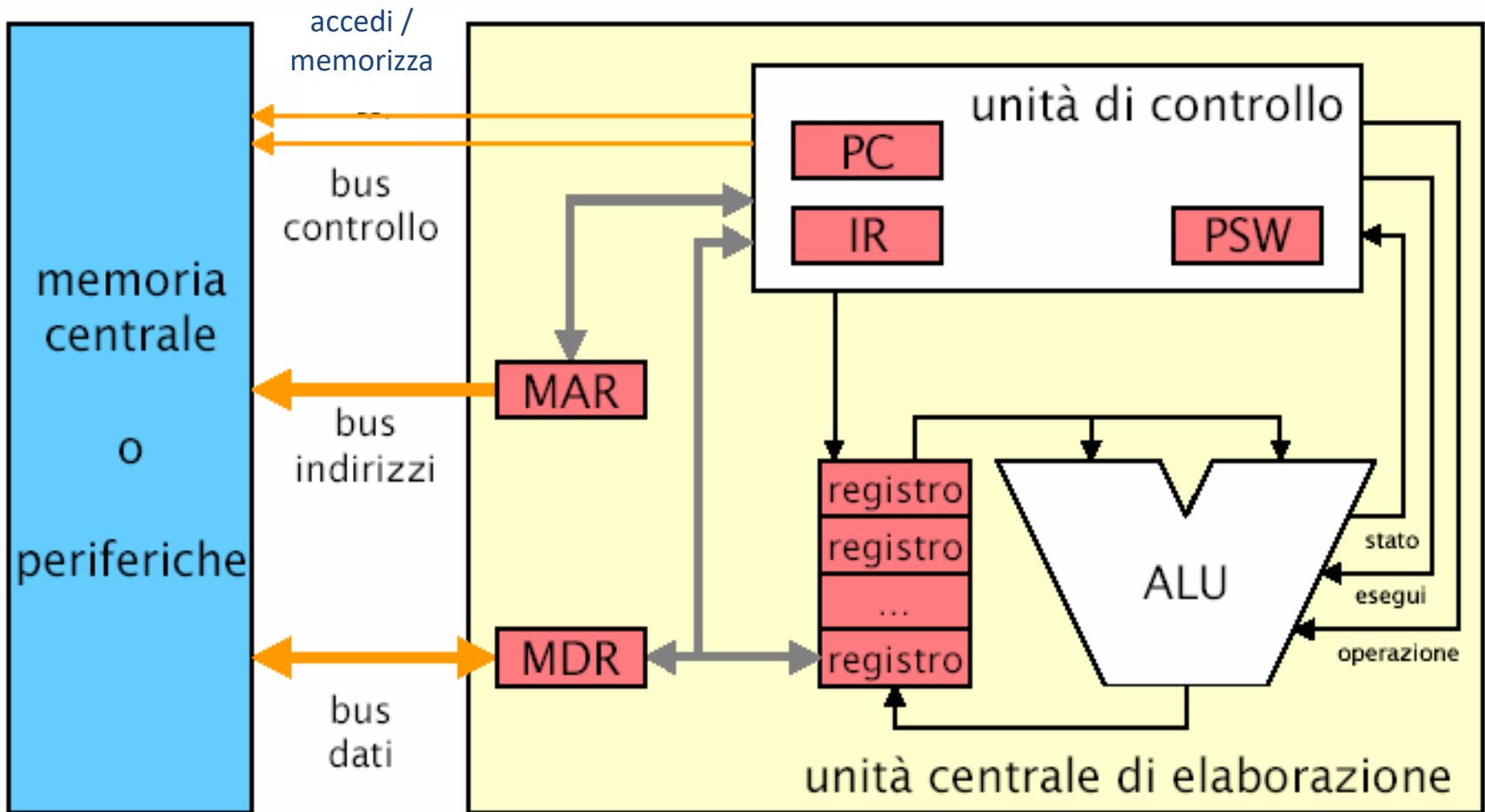


ANCHE CHIAMATO **CPU (CENTRAL PROCESSING UNIT)** È RESPONSABILE DELLA **ELABORAZIONE DEI DATI** E DEL **COORDINAMENTO FRA I COMPONENTI** DEL CALCOLATORE. LA CPU È COMPOSTA DA DUE PARTI:

- LA **UNITÀ ARITMETICO-LOGICA (ALU- ARITHMETIC LOGIC UNIT)**, RESPONSABILE DELL' ESECUZIONE DELLE ISTRUZIONI
- LA **UNITÀ DI CONTROLLO (CU - CONTROL UNIT)**, RESPONSABILE DEL COORDINAMENTO FRA I COMPONENTI

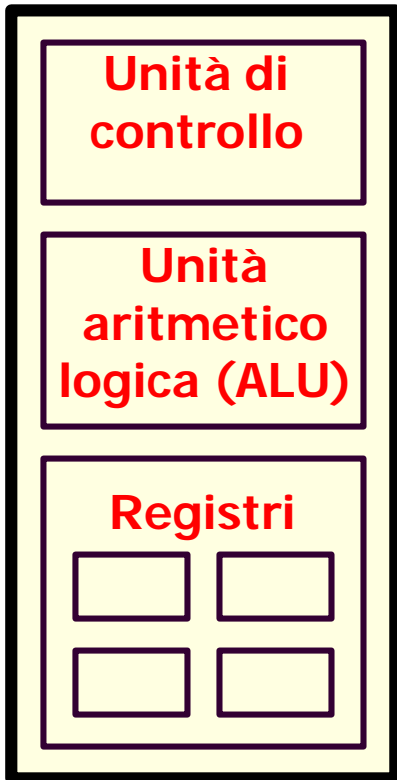
⇒ NEI COMPUTER ATTUALI LA **CPU** PUÒ ESSERE COSTITUITA DA DIVERSI **CORE**, CIASCUNO DEI QUALI PUÒ SVOLGERE TUTTE LE FUNZIONI DESCRITTE SOPRA E CONTIENE DIVERSE ALU. IL PROCESSORE È FISICAMENTE UN **SINGOLO CIRCUITO INTEGRATO** O **CHIP**, CONSISTENTE DI TRANSISTOR, DIODI, CONDENSATORI E RESISTORI COLLEGATI TRA LORO PER MEZZO DELLA TECNOLOGIA **VLSI (VERY LARGE SCALE INTEGRATION)** E COLLOCATI SU UNO STRATO DI SEMICONDUITTORE, TIPICAMENTE SILICIO.

Struttura della CPU



Elementi di una CPU

CPU



Unità di controllo Control Unit – CU)

- Svolge funzioni di controllo, decide quali istruzioni eseguire. Per ogni istruzione da eseguire, prepara l'esecuzione, spostando i dati nei registri opportuni.

Unità aritmetico–logica (Arithmetic-Logic Unit – ALU)

- esegue le operazioni aritmetico-logiche (più, meno, ..., confronto).

Registri

- **memoria ad alta velocità** usata per risultati temporanei e informazioni di controllo;
- il **valore massimo** memorizzabile in un registro è determinato dalle **dimensioni** del registro;
- esistono registri di uso generico e registri specifici:

Registri (1/2)

Esistono registri di uso generico e registri specifici :

- **PC**: contatore delle istruzioni (program counter)
 - contiene l' da eseguire
- **IR**: registro delle istruzioni (instruction register)
 - contiene l'
- **PSW**: program status word
 - contiene informazioni, opportunamente codificate, sull'esito dell'ultima istruzione che è stata eseguita

Registri (1/2)

Esistono registri di uso generico e registri specifici :

- **PC**: contatore delle istruzioni (program counter)
 - contiene l'indirizzo della prossima istruzione da eseguire
- **IR**: registro delle istruzioni (instruction register)
 - contiene l'istruzione che deve essere eseguita
- **PSW**: program status word
 - contiene informazioni, opportunamente codificate, sull'esito dell'ultima istruzione che è stata eseguita

Registri (2/2)

- **MAR:** registro indirizzi della memoria
 - indirizzo della che deve essere acceduta o memorizzata
- **MDR:** registro dati della memoria
 - che è stato acceduto o che deve essere memorizzato
- registri generali
 - per memorizzare di una operazione

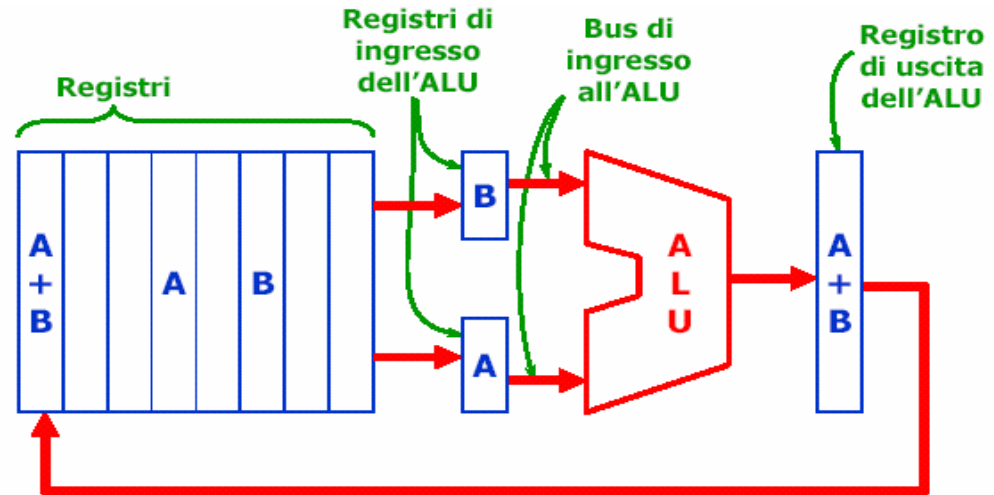
Registri (2/2)

- **MAR: memory address register**
 - indirizzo della cella di memoria che deve essere acceduta o memorizzata
- **MDR: memory data register**
 - dato che è stato acceduto o che deve essere memorizzato
- **registri generali**
 - per memorizzare gli operandi ed il risultato di una operazione

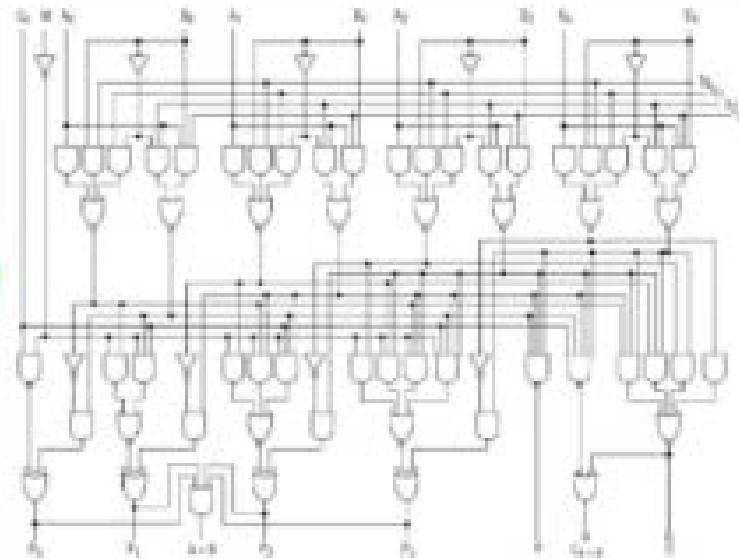
Unità Aritmetico-Logica

L'Unità Aritmetico-Logica (**ALU**) è costituita da un insieme di circuiti in grado di svolgere le operazioni di tipo aritmetico e logico

La ALU accede ai valori nei registri, esegue operazioni tra quei valori, memorizza risultati delle operazioni in registri



**ESEMPIO DI SCHEMA
CIRCUITALE DI UNA ALU** →



si', ma che fa la CPU?

(che fa la ALU in collaborazione con la CU?)

esegue le istruzioni del programma,
(scritto/memorizzato in linguaggio
macchina ...)

ESEGUE Istruzioni MACCHINA per **elaborazione dati**

- aritmetiche
- logiche (AND, OR, NOT)
- relazionali (maggiore, minore, uguale, ...)

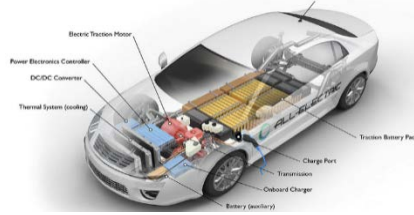
ESEGUE Istr. MACCHINA per **controllo flusso** esecuz.

- sequenza
- selezione
- ciclo
- salto (*Jump*) ad una data istruzione

ESEGUE Istruzioni MACCHINA per **trasferimento** dati

- dati ed istruzioni fra CPU e memoria
- dati fra CPU e dispositivi di I/O (tramite interfacce)

Istruzioni macchina?



I **circuiti della CPU** collegano registri ed eseguono operazioni di accesso e memorizzazione sui medesimi.

Attraverso il BUS, e i registri, i circuiti possono interagire con la memoria e con le interfacce di I/O ...

Yawn ...

Piu' o meno ogni **circuito** corrisponde ad una delle **operazioni** *(compiti, lavori, cose che possono essere fatte)* che la **CPU** puo' eseguire usando e controllando le altre strutture del calcolatore.

Yes! Queste sono le istruzioni eseguibili dalla CPU ...

E Yes! Queste sono le **istruzioni macchina** ...

Istruzioni macchina!

00000000000000000000000001000000	0000010000000000111
0000000000010000000000000100100	000001000000001000
000001000000010000000000000000	000001000000001001
000000100000000000000000111100	000001000000001010
10000010000000000000000000000000	000001000000001011
	000001000000001100

E queste sono le **istruzioni macchina**, nel senso che un programma scritto in linguaggio macchina, elenca la sequenza di istruzioni macchina che devono essere eseguite per

- acquisire i dati di input,
- risolvere il problema,
- produrre dati di output .

Per essere eseguito, il programma va **memorizzato** (cioe` messo in memoria).

Ogni **istruzione** va scritta in memoria specificando

- il suo codice (codice operativo che identifica il circuito)
- e quali registri sono coinvolti nell'operazione da eseguire
- e, se serve, anche quali indirizzi di RAM devono essere coinvolti nell'operazione.

Istruzioni macchina!

00000000000000000000000000000001000000	0000010000000000000111
0000000000000100000000000000000100100	00000100000000001000
0000001000000000100000000000000000	00000100000000001001
0000000100000000000000000000000111100	00000100000000001010
100000010000000000000000000000000000	00000100000000001011
	00000100000000001100

E queste sono le **istruzioni macchina**, nel senso che un programma scritto in linguaggio macchina, elenca la sequenza di istruzioni macchina che devono essere eseguite per

- acquisire i dati di input,
- risolvere il problema,
- produrre dati di output .

Per essere eseguito, il programma va **memorizzato** (cioe` messo in memoria).

Ogni **istruzione** va scritta in memoria specificando

- il suo codice (codice operativo che identifica il circuito)

(per questo si parla di codice sorgente, codice di programma, "codifica" intendendo "scrittura di un programma" ...)

- e quali registri sono coinvolti nell'operazione da eseguire
- e, se serve, anche quali indirizzi di RAM devono essere coinvolti nell'operazione.

Tempo Macchina

Quante istruzioni puo` eseguire la CPU in un certo tempo?

Cosa e` il clock?

E la cache?

Memoria Secondaria: e` poco importante?

POCHE OPERAZIONI, MA **MOLTO EFFICIENTI** – I PROCESSORI ATTUALI POSSONO TIPICAMENTE ESEGUIRE QUALCHE DECINA DI MILIARDI DI OPERAZIONI PER SECONDO.
AD ESEMPIO UN INTEL CORE I7 2600K PUÒ ESEGUIRE CIRCA 117.000 **MIPS** – MILLIONS OF INSTRUCTIONS PER SECOND.

Clock

LE ATTIVITÀ DEL PROCESSORE VENGONO SCANDITE DA UN **CLOCK**, UN **SEGNALE** CHE È USATO COME UN METRONOMO PER **COORDINARE** E **SINCRONIZZARE** LE OPERAZIONI DELLE VARIE PARTI DELLA CPU.

UNA CARATTERISTICA FONDAMENTALE DEL CLOCK È LA SUA **FREQUENZA**, OVVERO IL NUMERO DI SEGNALI CHE VENGONO INVIATI OGNI SECONDO. NEI CALCOLATORI ATTUALI QUESTA FREQUENZA È DELL'ORDINE DI QUALCHE **GIGAHERTZ**, OVVERO QUALCHE MILIARDO DI SEGNALI AL SECONDO.

IL **TEMPO** CHE INTERCORRE FRA UN SEGNALE E L'ALTRO, DENOTATO DA **T_p** , È L'**INVERSO DELLA FREQUENZA** E DEVE ESSERE MAGGIORE DEL TEMPO CHE IMPIEGA IL SEGNALE AD ARRIVARE IN OGNI PARTE DELLA CPU. COINCIDE CON IL TEMPO CHE IMPIEGA LA CPU AD ESEGUIRE UNA **OPERAZIONE ELEMENTARE**.

Memoria Cache

La **velocità** con cui la **CPU** esegue le sue operazioni è la **massima** possibile in tutto il sistema di elaborazione.

Il **BUS** trasmette dati alla propria velocità, che è **minore**.
La **memoria** gestisce le proprie locazioni a velocità **ancora minore**.
Le operazioni che coinvolgono le **periferiche** (unità di I/O, memoria esterna ...) sono diversificate ma sempre (molto) **più lente**.

Così, in quest'immensità, può succedere che il processore debba fermarsi ad aspettare che altre componenti del calcolatore finiscano il loro lavoro.

Ad esempio se la CPU ha richiesto un dato alla RAM, la risposta arriverà un bel po' di clock dopo, lasciando la CPU inattiva per quel po' di tempo.

Per diminuire i "**clock di inattività della CPU**", la soluzione consiste nell'usare una **memoria più veloce** della RAM e far interagire questa con la CPU.

MA questa memoria è più costosa e quindi più piccola della RAM: non si può solo sostituire la RAM con questa memoria più veloce ...che si chiama **CACHE**.

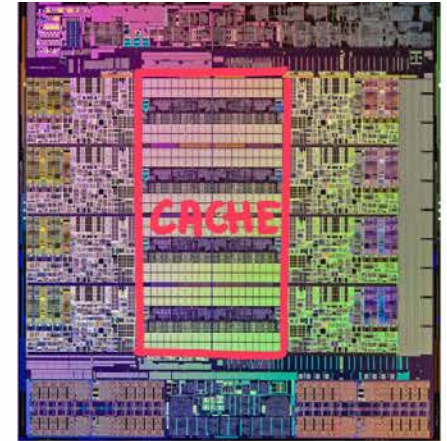
Come funziona?

Parti contigue della RAM vengono trasferite nella cache:
la CPU interagisce con quelle.
Quando i dati che servono non si trovano in cache, altre parti di RAM vengono messe in cache, al posto di quelle inutili.

Così la CPU interagisce con una memoria ad alta velocità, invece che con la RAM.

Memorie Cache

- SPESSO LA **MEMORIA CACHE** È COSTRUITA **DENTRO AL CHIP** DEL PROCESSORE
- SPESSO CI SONO **DIVERSE MEMORIE CACHE**, CON DIVERSI ORDINI DI IMPORTANZA



LA **MEMORIA CACHE** FUNZIONA BENE GRAZIE A DUE PRINCIPI:

- 1) **LOCALITÀ TEMPORALE**: UN DATO CHE È STATO RECENTEMENTE USATO MOLTO PROBABILMENTE VERRÀ DI NUOVO USATO A BREVE.
⇒ LA CACHE MANTIENE I DATI USATI PIÙ RECENTEMENTE.
- 2) **LOCALITÀ SPAZIALE**: SE UN DATO È STATO APPENA USATO, MOLTO PROBABILMENTE DATI AD ESSO ADIACENTI VERRANNO PRESTO USATI.
⇒ INTERE PORZIONI DI MEMORIA CENTRALE SONO TRASFERITE IN CACHE.

Memorie Secondarie (o "di massa" se proprio insistete)

Rispetto alla RAM, sono

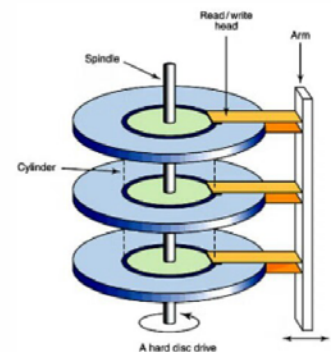
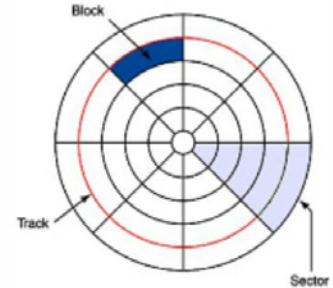
- **PIÙ CAPIENTI** (CENTINAIA DI GB CONTRO QUALCHE GB NEI COMPUTER ATTUALI)
- **PERMANENTI** (CONSERVANO IL CONTENUTO ANCHE DOPO LO SPEGNIMENTO DEL COMPUTER)
- **PIÙ ECONOMICHE**
- **PIÙ LENTE** (IL TEMPO DI ACCESSO AI DATI CONTENUTI IN MEMORIA SECONDARIA È MOLTO MAGGIORE DEL TEMPO DI ACCESSO AI DATI CONTENUTI IN MEMORIA CENTRALE)
⇒ QUANDO L'UTENTE CHIEDE DI ESEGUIRE UN PROGRAMMA, QUESTO VIENE **COPIATO DALLA MEMORIA SECONDARIA ALLA MEMORIA CENTRALE**.

Memorie Secondarie ... "Hard Disk"

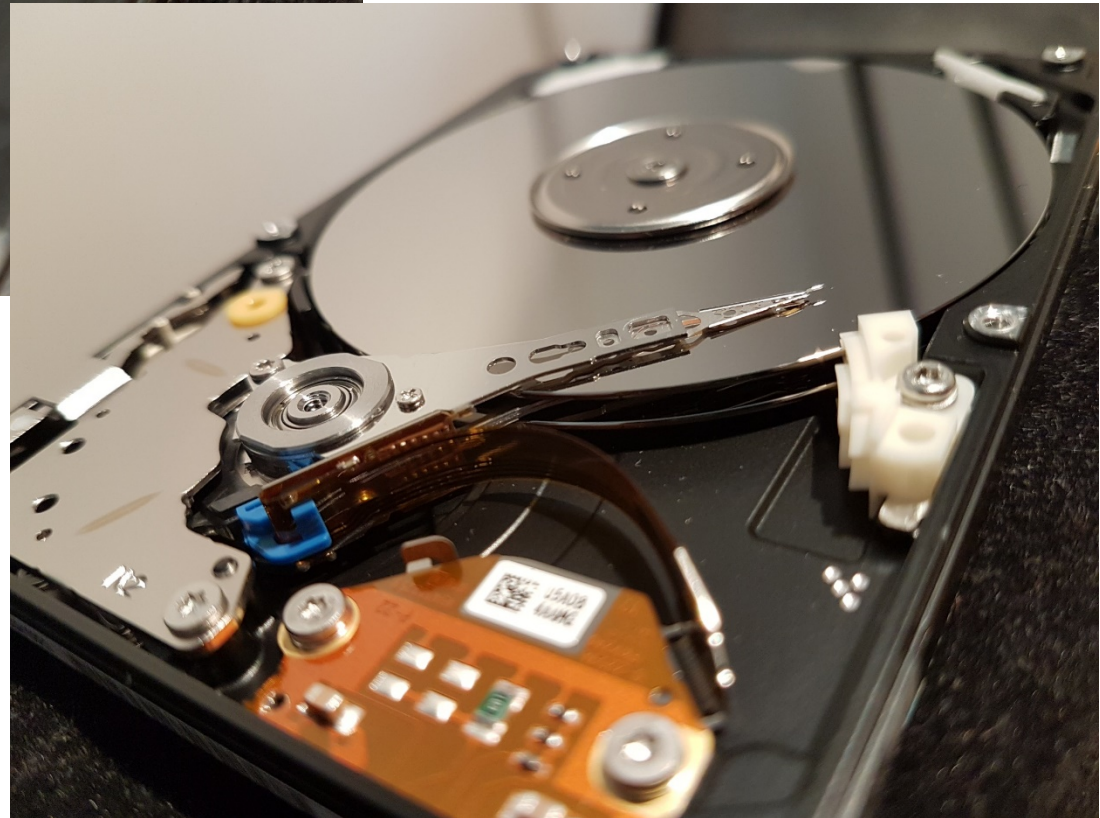
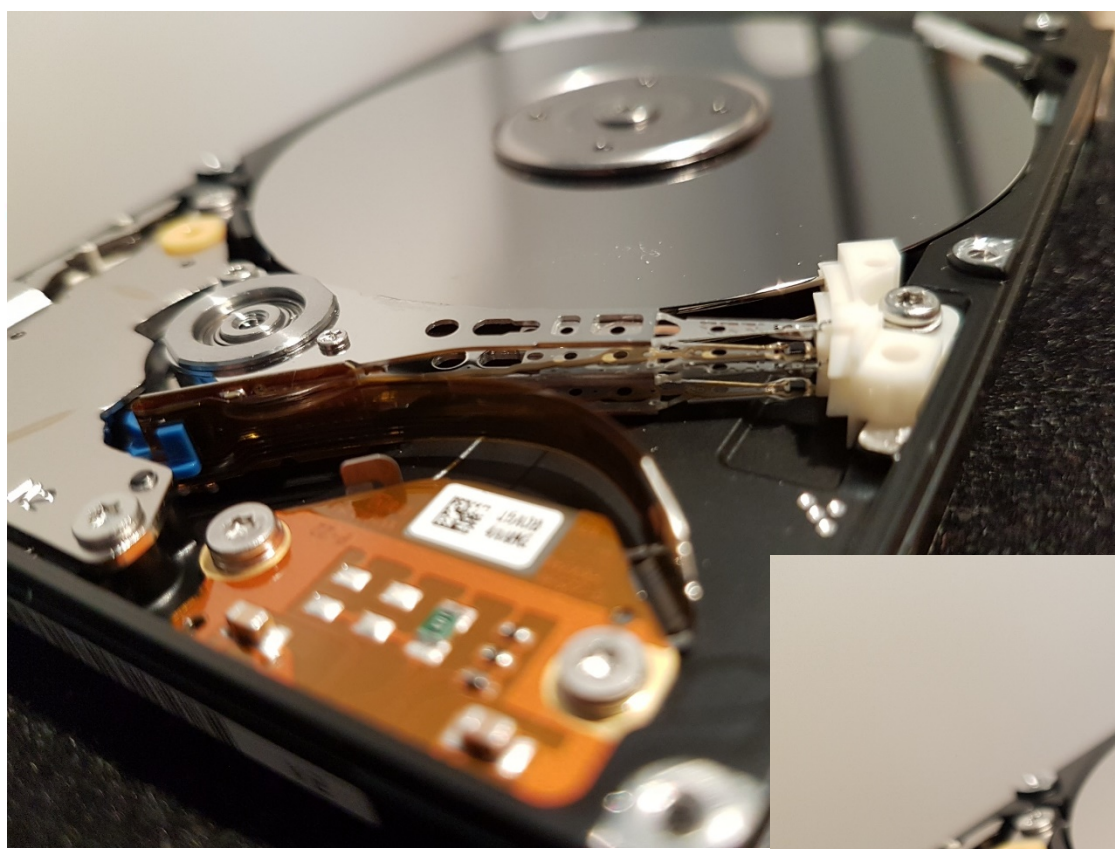
ANCHE NOTO COME **DISCO FISSO** o **DISCO RIGIDO**, È COSTITUITO DA UN INSIEME DI **DISCHI MAGNETICI** CHE RUOTANO INTORNO AD UN ASSE.

LA SUPERFICIE DI UN DISCO MAGNETICO È DIVISA IN **SETTORI**, CHE HANNO LA FORMA DI "SPICCHI", ED IN UNA SEQUENZA DI **TRACCE**, CHE SONO CAMMINI CIRCOLARI LUNGO I QUALI VENGONO MEMORIZZATI I BIT.

LETTURA E SCRITTURA VENGONO EFFETTUATE PER MEZZO DI **TESTINE**, UNA PER CIASCUN DISCO MAGNETICO. LE TESTINE SI ACCORCIANO E SI ALLUNGANO PER ARRIVARE SULLA TRACCIA GIUSTA, MENTRE I DISCHI RUOTANO PER FAR SÌ CHE LE TESTINE SIANO POSIZIONATE SUL SETTORE GIUSTO.



normale HD



Memorie Secondarie

HARD-DISK



CD-ROM



DVD



NASTRO MAGNETICO

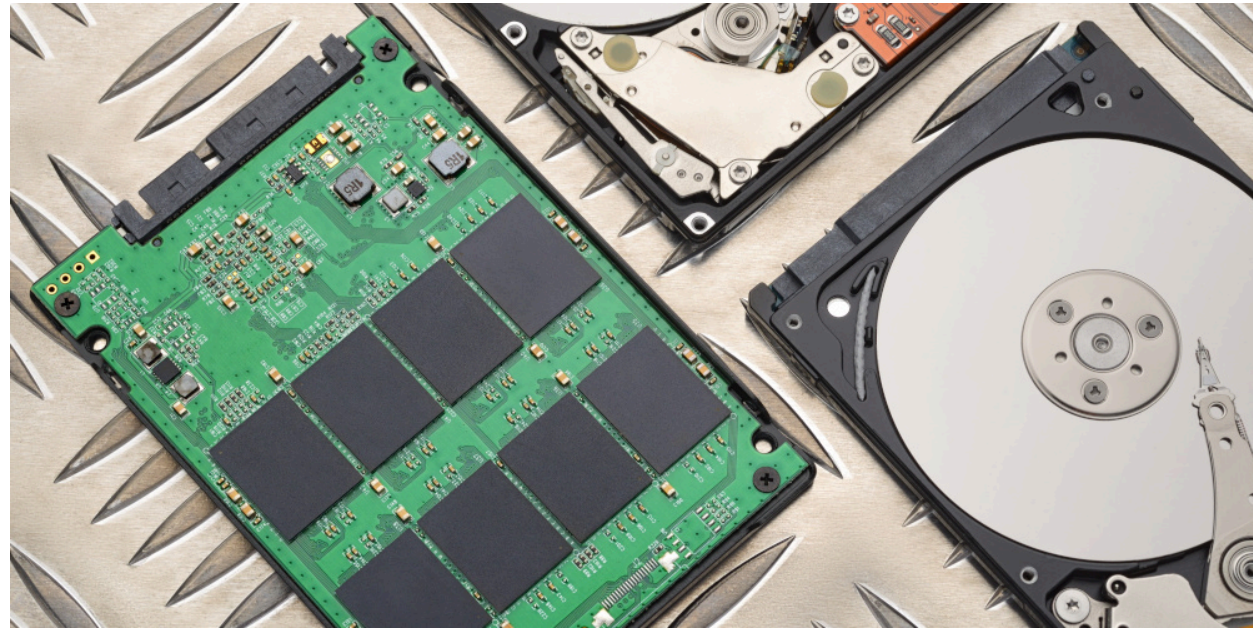


USB FLASH DRIVE

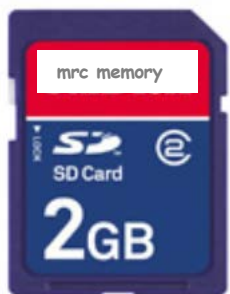


Memorie Secondarie

normale HD



SSD



scheda SD, SDHC



micro SD

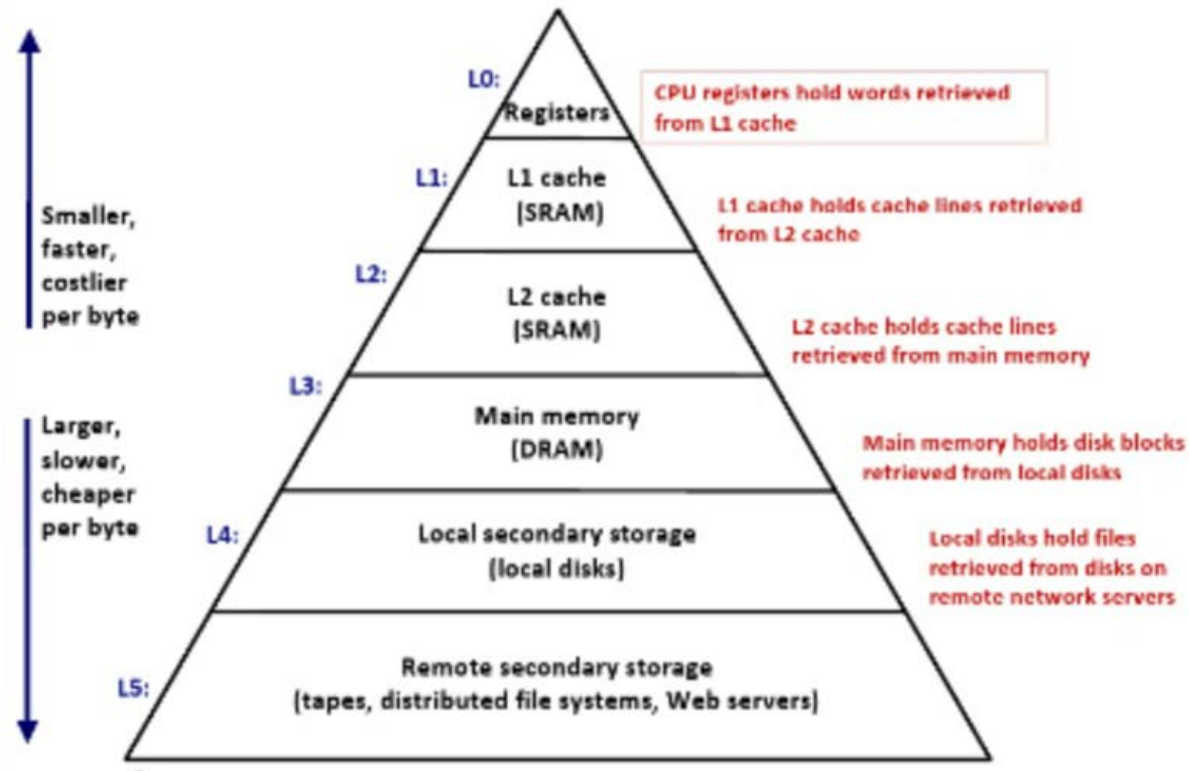
4 cose importanti riguardo alle memorie ...

- capacita' ... KB, MB, GB, TB ...

- tempo di accesso

- velocita'
(da memoria a RAM)

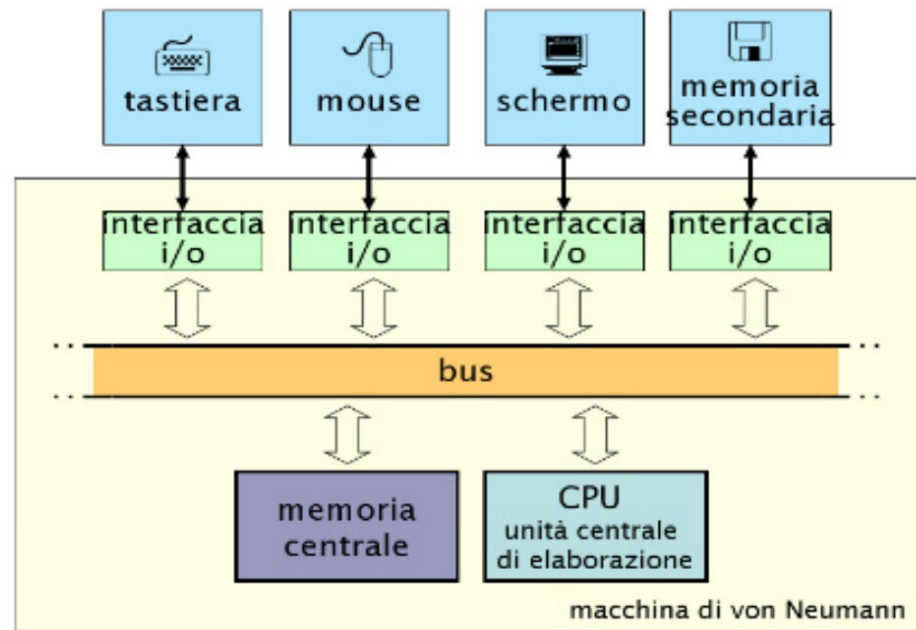
- costo



indicativamente ...

registry	Kb	1ns
cache	MB	10ns
ram	GB	100ns
HD	100GB-nTB	1-10ms
nastro, CD	GB	100ms

Interfacce di I/O



UN CALCOLATORE È IN GENERE COLLEGATO A DIVERSE PERIFERICHE DI INPUT (MOUSE, TASTIERA, SCANNER, MICROFONO, TELECAMERA, ...), CHE PERMETTONO AL CALCOLATORE DI RICEVERE DATI ED ISTRUZIONI DALL'UTENTE, ED A DIVERSE PERIFERICHE DI OUTPUT (MONITOR, STAMPANTI, ALTOPARLANTI, ...), CHE PERMETTONO AL CALCOLATORE DI COMUNICARE ALL'UTENTE I RISULTATI DEL LAVORO SVOLTO.

LE INTERFACCE DI I/O CONTROLLANO IL FUNZIONAMENTO DELLE PERIFERICHE DI I/O, IN PARTICOLARE DEVONO TRADURRE I SEGNALI INTERNI AL CALCOLATORE IN UN FORMATO COMPRESIBILE ALLA PERIFERICA E VICE VERSA. POSSONO ESSERE PENSATE COME DELLE PICCOLE CPU DEDICATE AL FUNZIONAMENTO DELLE PERIFERICHE.

SISTEMA OPERATIVO

È IL PRINCIPALE COMPONENTE DEL SOFTWARE DI BASE.

HA DUE SCOPI PRINCIPALI:

- 1) **GESTIRE LE RISORSE DEL CALCOLATORE** - LA CPU, LE PERIFERICHE, LA MEMORIA CENTRALE, LA MEMORIA DI MASSA...
- 2) **VIRTUALIZZARE LE RISORSE** - LE RISORSE DEL CALCOLATORE DEVONO ESSERE RESE PIÙ SEMPLICI DI QUANTO NON SIANO IN REALTÀ: GLI UTENTI, MA ANCHE I PROGRAMMI APPLICATIVI, NON DEVONO ESSERE A CONOSCENZA DEL FUNZIONAMENTO DELL'HARDWARE DEL CALCOLATORE ED IL SISTEMA OPERATIVO DEVE ASTRARRE LE RISORSE MESSE A DISPOSIZIONE DALL'HARDWARE.

ESEMPIO DI VIRTUALIZZAZIONE: AVVIO DELL'ESECUZIONE DI UN PROGRAMMA

DAL PUNTO DI VISTA DELL'UTENTE: BASTA UN DOPPIO CLICK E L'ESECUZIONE DEL PROGRAMMA VIENE AVVIATA.

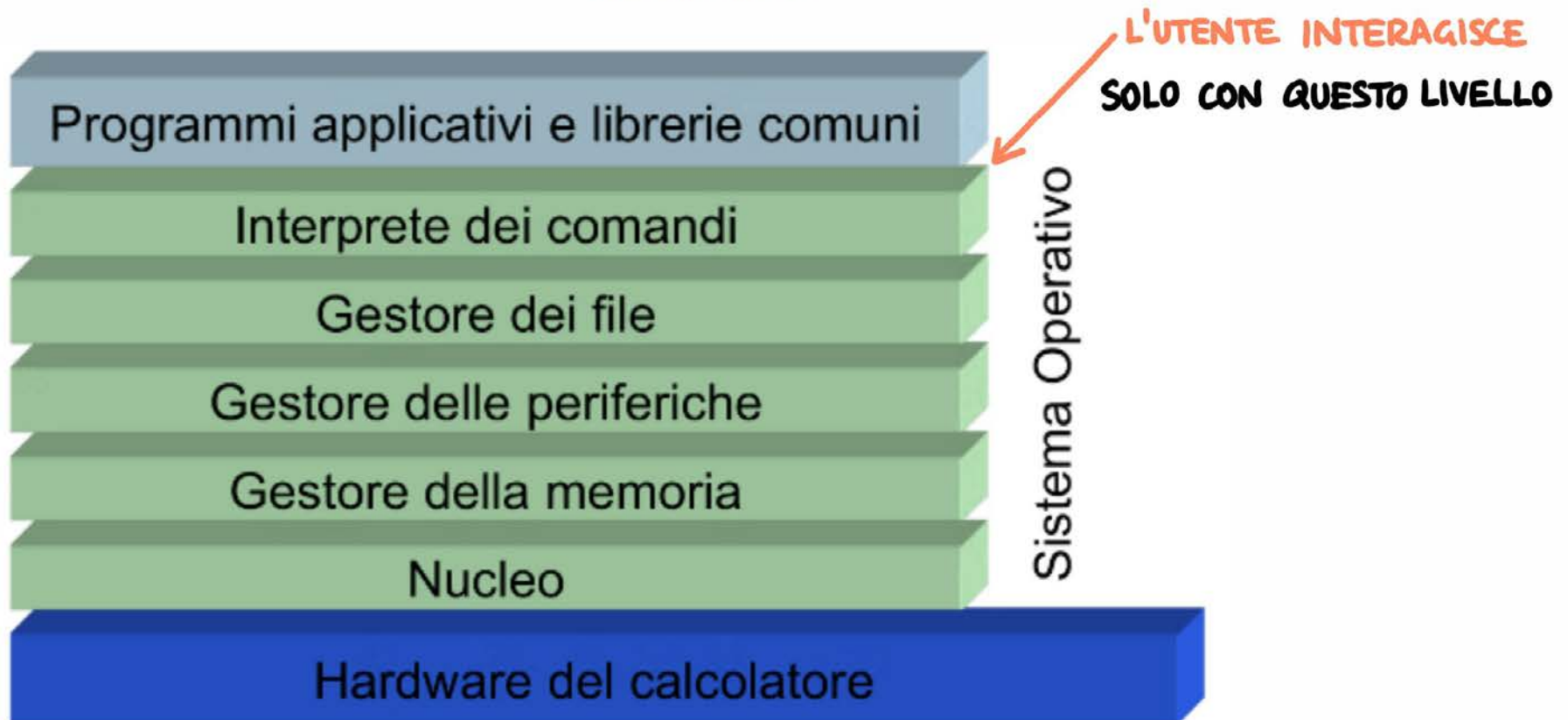


QUELLO CHE FA IL SISTEMA OPERATIVO:

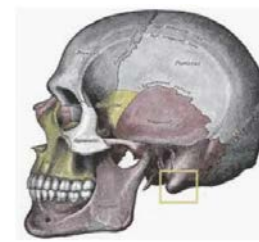
- **INDIVIDUA LA SEQUENZA DI ISTRUZIONI CHE COSTITUISCONO IL PROGRAMMA IN MEMORIA SECONDARIA**
 - **ASSEGNA RISORSE AL PROGRAMMA - AD ESEMPIO DELLO SPAZIO IN MEMORIA CENTRALE**
 - **COPIA LE ISTRUZIONI DEL PROGRAMMA IN MEMORIA CENTRALE**
 - **FA SÌ CHE LA CPU ESEGA LE PRIME ISTRUZIONI DEL PROGRAMMA.**
- ⇒ **TUTTO QUESTO AVVIENE IN MANIERA NON VISIBILE ALL'UTENTE.**

STRUTTURA DEL SISTEMA OPERATIVO

PUÒ ESSERE DESCRITTA IN MODO GERARCHICO



Programma e Processo



Processo mastoideo - Wikipedia



Come abbiamo visto, un programma e' una sequenza di istruzioni.

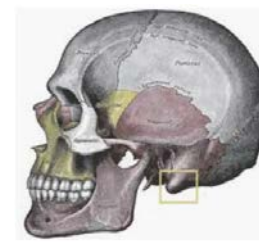
Il **PROGRAMMA** e' memorizzato in qualche supporto di memoria

Un programma puo' essere eseguito molte volte, ed e' sempre il medesimo ... una cosa statica, sempre uguale a se stessa.

Un programma in esecuzione costituisce un **PROCESSO**, o *thread*, attivo nel calcolatore.

Ogni esecuzione del programma corrisponde ad un nuovo processo.

Programma e Processo



Processo mastoideo - Wikipedia



Come abbiamo visto, un programma e` una sequenza di istruzioni.

Il **PROGRAMMA** e` memorizzato in qualche supporto di memoria

- Memorizzato in memoria secondaria (sul disco fisso, su disco esterno, dvd, cd, Pennetta usb, ...), quando non serve
- Memorizzato in memoria centrale, mentre e` in esecuzione.

... o e` uno o e` l'altro ... in qualunque momento della sua esistenza (senno` non esiste).

Un programma puo' essere eseguito molte volte, ed e' sempre il medesimo ... una cosa statica, sempre uguale a se stessa.

Un programma in esecuzione costituisce un **PROCESSO**, o *thread*, attivo nel calcolatore.

Ogni esecuzione del programma corrisponde ad un nuovo processo.

Al medesimo programma possono corrispondere molti processi ...

Inoltre altri processi possono essere creati da un processo attivo.

Mentre il calcolatore e` acceso, processi nascono e muoiono in continuazione, alcuni originati da comandi dell'utente, altri da processi attivi del Sistema Operativo.

VITA DEI PROCESSI

I **PROCESSI NASCONO** QUANDO I **PROGRAMMI** VENGONO MANDATI IN ESECUZIONE E **MUOIONO** QUANDO L'ESECUZIONE TERMINA. CHE SUCCEDA **DURANTE LA VITA DI UN PROCESSO?**
⇒ LA **CPU ESEGUE LE ISTRUZIONI DEL PROCESSO.**

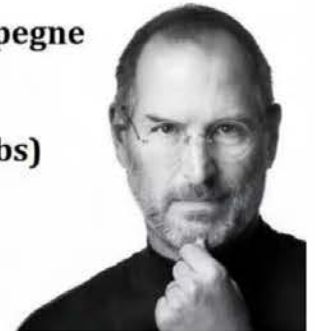


PROBLEMA: ABBIAMO **1 SOLA CPU** E **MOLTI PROCESSI** DA ESEGUIRE. COME POSSIAMO ESEGUIRE TUTTI I PROCESSI?

UNIPROGRAMMAZIONE: UNA POSSIBILITÀ È ESEGUIRE **1 PROCESSO PER VOLTA**. SOLUZIONE ESTREMAMENTE SCONVENIENTE!

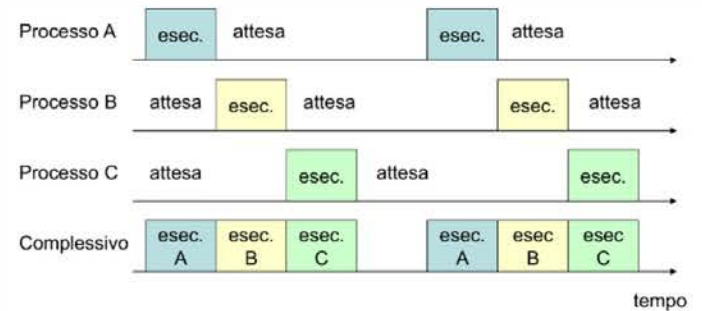
Prova un po' a spegnere e riaccende

(Steve Jobs)



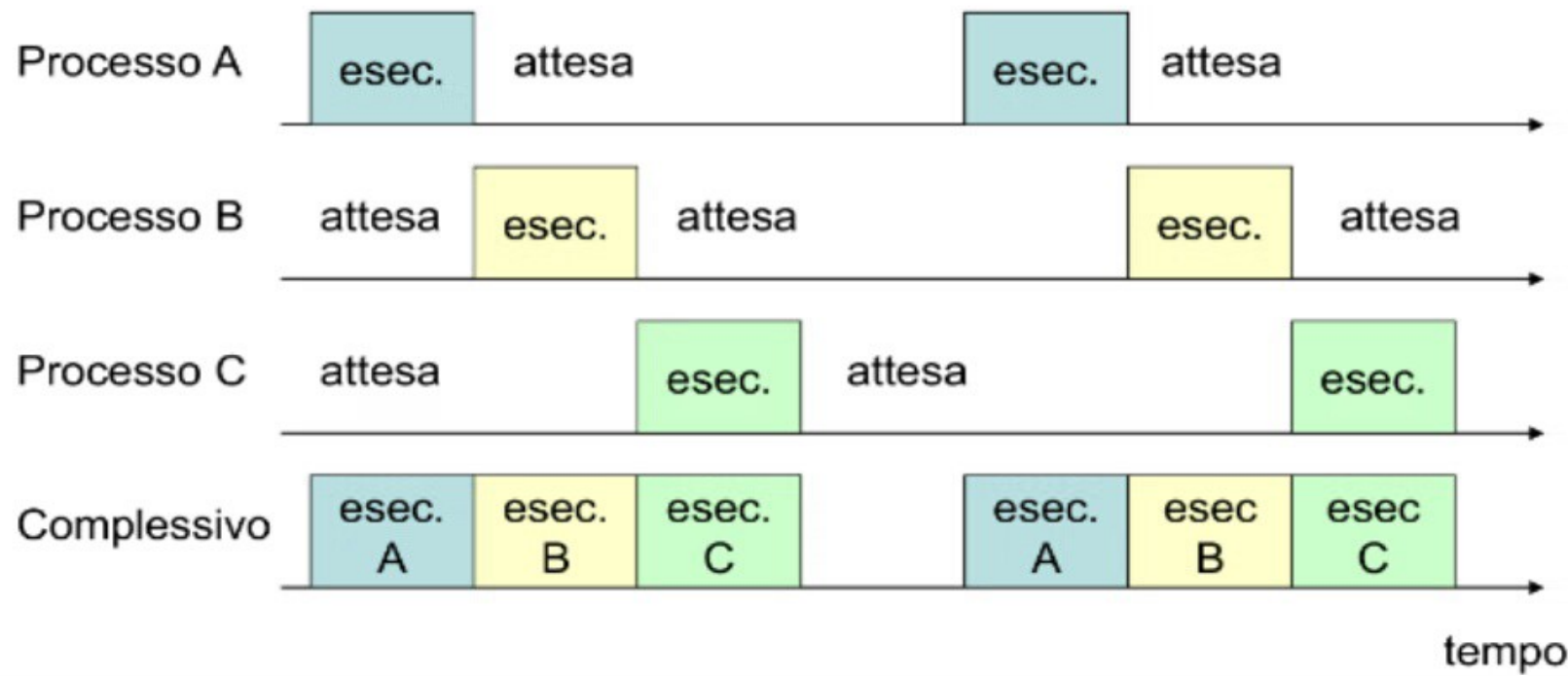
MULTIPROGRAMMAZIONE E NUCLEO

ANCHE DETTA **MULTITASKING**, LA MULTIPROGRAMMAZIONE FORNISCE ALL'UTENTE L'IMPRES-
SIONE CHE CI SIANO **DIVERSI PROCESSORI VIRTUALI**, OGNUNO DEDICATO AD UN PROCESSO.
IL **PROCESSORE SALTA DA UN PROCESSO ALL'ALTRO**,
DANDO L'ILLUSIONE ALL'UTENTE CHE TUTTI I
PROCESSI SIANO IN ESECUZIONE CONTEMPORANEA,
MENTRE IN OGNI ISTANTE LA CPU PUÒ ESEGUIRE
UN SOLO PROCESSO: TUTTI GLI ALTRI SONO IN
ESECUZIONE VIRTUALE (OVVERO SONO IN ATTESA CHE LA CPU NE ESEGUA LE ISTRUZIONI).



IL **NUCLEO** O **KERNEL** O **GESTORE DEI PROCESSI** È RESPONSABILE
DELLA **GESTIONE DEL PROCESSORE** (O DEI PROCESSORI), IN
PARTICOLARE FA DA **SCHEDULER**, OVVERO PROGRAMMA L'ACCES-
SO DEI PROCESSI ALLA RISORSA PROCESSORE.

MU
ANC
SION
IL
DAN
PRO
MEN
UN
ESE



ES.
→
→
→
→
mpo
NI).



IL NUCLEO O KERNEL O GESTORE DEI PROCESSI È RESPONSABILE DELLA GESTIONE DEL PROCESSORE (O DEI PROCESSORI), IN PARTICOLARE FA DA SCHEDULER, OVVERO PROGRAMMA L'ACCESSO DEI PROCESSI ALLA RISORSA PROCESSORE.

Mem. Secondaria

GESTORE DELLA MEMORIA



È RESPONSABILE DELLA GESTIONE DELLA MEMORIA CENTRALE E DI RENDERE EFFICIENTI GLI SPOSTAMENTI DALLA MEMORIA DI MASSA ALLA MEMORIA CENTRALE E VICEVERSA.

LA QUANTITÀ DI MEMORIA DI CUI UN PROCESSO HA BISOGNO POTREBBE ESSERE MAGGIORE DELLA MEMORIA CENTRALE DISPONIBILE.

⇒ A CIASCUN PROCESSO VIENE ASSEGNATA UNA AREA DI MEMORIA VIRTUALE, GRANDE A SUFFICIENZA DA PERMETTERE L'ESECUZIONE DEL PROCESSO. IL GESTORE DELLA MEMORIA GESTISCE LA CORRISPONDENZA TRA MEMORIA VIRTUALE E MEMORIA CENTRALE: IN OGNI ISTANTE SOLO DELLE PORZIONI ("PAGINE") DEL PROGRAMMA SONO REALMENTE IN MEMORIA CENTRALE. PORZIONI POCO USATE VENGONO MOSSE IN UNA AREA DI SWAP, CHE È PARTE DELLA MEMORIA SECONDARIA.

GESTORE DELLE PERIFERICHE

- Virtualizzazione della periferica
- Gestione conflitti tra processi
- Trasferimento/Traduzione dati



È RESPONSABILE DELLA GESTIONE DELLE PERIFERICHE.

UTILIZZA UN INSIEME DI DRIVER ED HA I SEGUENTI SCOPI:

- TRASFORMARE LE PERIFERICHE IN PERIFERICHE VIRTUALI, OVVERO L'UTENTE NON DEVE ESSERE A CONOSCENZA DELLA MODALITÀ DI FUNZIONAMENTO FISICO DELLE PERIFERICHE PER POTERLE UTILIZZARE.
- GESTIRE LA TRADUZIONE DEI DATI DAL FORMATO INTERNO AL CALCOLATORE A QUELLO DELLE PERIFERICHE.
- GESTIRE LE COMUNICAZIONI DEI SEGNALI DA E VERSO LE PERIFERICHE.
- GESTIRE I CONFLITTI LEGATI A PROCESSI MULTIPLI CHE VOGLIONO ACCEDERE ALLA STESSA PERIFERICA.

GESTORE DEI FILE • FILE SYSTEM



È RESPONSABILE DELLA GESTIONE DELLE INFORMAZIONI MEMORIZZATE NELLA MEMORIA SECONDARIA, OVVERO NEI DISCHI RIGIDI.

* DAL PUNTO DI VISTA FISICO, I DISCHI SONO ORGANIZZATI IN SUPERFICI, SETTORI E TRACCE

* DAL PUNTO DI VISTA DELL'UTENTE, LE INFORMAZIONI HANNO UN'ORGANIZZAZIONE GERARCHICA (OVVERO AD ALBERO) IN FILE E CARTELLE

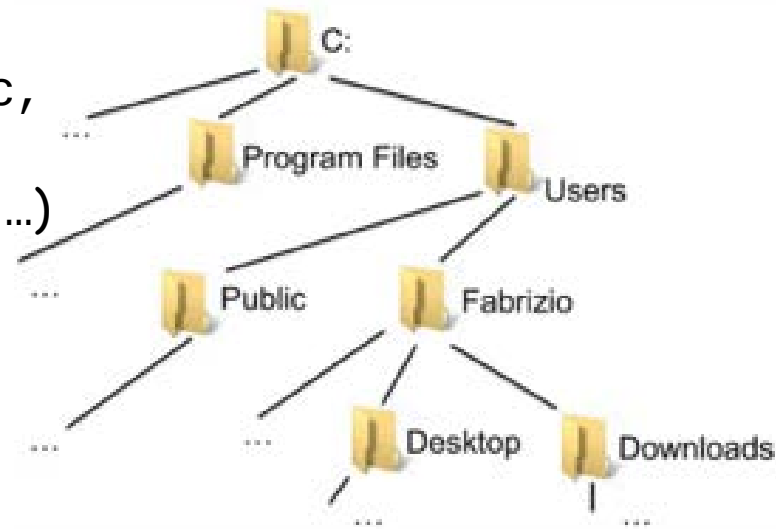
* È COMPITO DEL FILE SYSTEM VIRTUALIZZARE LA MEMORIA SECONDARIA, COSÌ CHE VENGA PRESENTATA ALL'UTENTE SECONDO L'ORGANIZZAZIONE GERARCHICA.

File, Directory, Volume

Un **FILE** ("archivio") e' la "unita' logica di memorizzazione"

- Una risorsa contenente dati (conf.txt, lettera.docx, sonnes.mp3, bombshell.c, acro.py, lettera.pdf, ...)
- Un programma (acro.exe, bombshell.o, ...)

("logica" = ha nome e posizione in memoria ...)



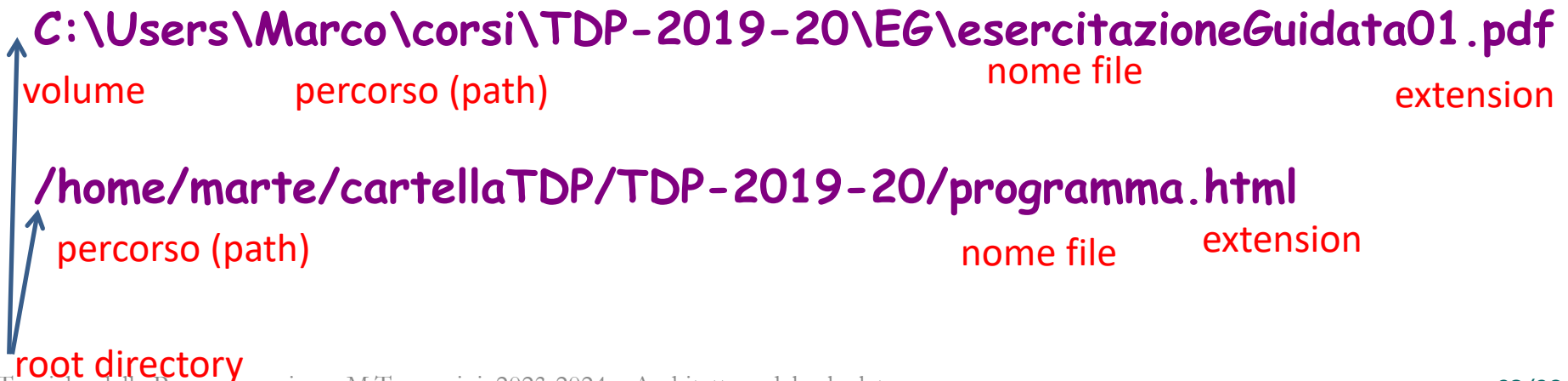
Una **DIRECTORY** ("cartella") e' il "contenitore logico di file" (*files*)
tutti i file (*files*) sono in directory (*directories*),
organizzate in modo gerarchico, "ad albero"



Un **Volume** e' una porzione (partizione) della memoria secondaria, alla quale corrisponde una gerarchia di directory.
(ex. ... E' la radice dell'albero ... **C:**)

FILES

- HANNO UN **CONTENUTO**, CHE È UNA SEQUENZA DI BYTE DI LUNGHEZZA VARIABILE. I DATI CONTENUTI IN UN FILE VENGONO **INTERPRETATI** SULLA BASE DEL **FORMATO DEL FILE**, CHE INDICA LA **CONVENZIONE** CON LA QUALE IL CONTENUTO DEL FILE VIENE INTERPRETATO.
- HANNO UN **NOME** O **IDENTIFICATORE** CHE DEVE ESSERE **UNIVOCO** ALL'INTERNO DELLA DIRECTORY DI CUI FANNO PARTE E LA CUI PARTE FINALE, DETTA **ESTENSIONE** (ES: `.txt`, `.exe`, `.pdf`), INDICA IL **FORMATO DEL FILE** IN MOLTI SISTEMI OPERATIVI.
- HANNO UN **PERCORSO** O **CAMMINO** CHE CORRISPONDE AL CAMMINO DALLA RADICE ALLA DIRECTORY CHE CONTIENE IL FILE.



INTERPRETE DEI COMANDI



È L' **INTERFACCIA UTENTE** DEL SISTEMA OPERATIVO: **DEFINISCE E GESTISCE LE OPERAZIONI A DISPOSIZIONE DELL'UTENTE FINALE, OVVERO I COMANDI A DISPOSIZIONE DELL'UTENTE** COME L'ESECUZIONE DI UN'APPLICAZIONE O L'APERTURA DI UN DOCUMENTO.

Interfacce Grafiche
(GUI = Graphic User Interface")



Interfaccia a
"linea di comando"

