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"
- ed una "definizione dal punto di vista funzionale" (...)

Previously on Tecniche della Programmazione

"CALCOLATORE: definizione dal punto di vista dell'utente"

Il calcolatore e` una **MACCHINA PROGRAMMABILE**
cioe` una macchina capace di eseguire diversi programmi, anche piu` in contemporanea

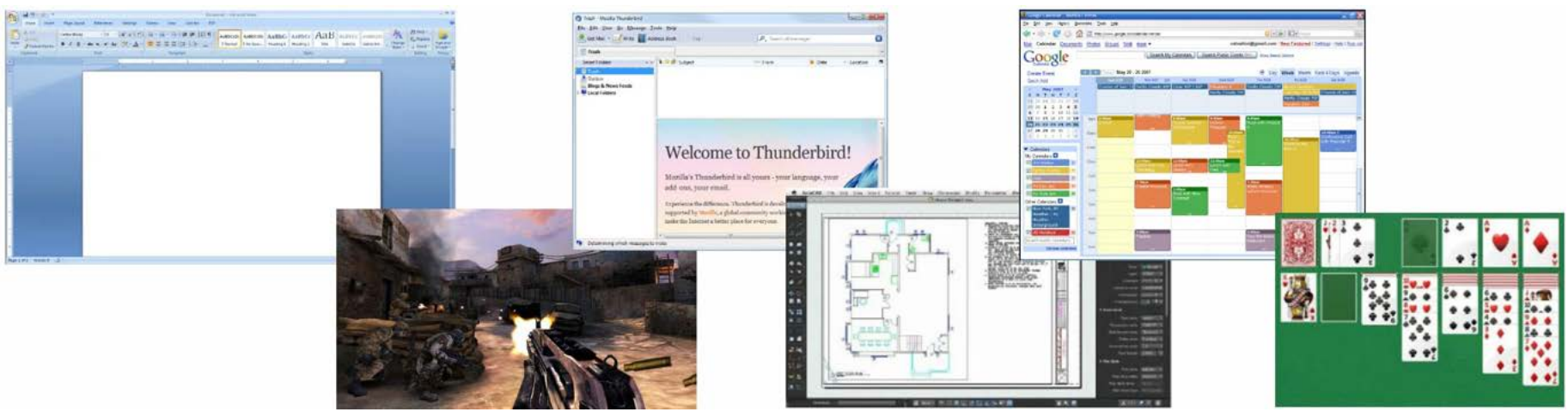
Un programma e` scritto in un LINGUAGGIO DI PROGRAMMAZIONE.

Un particolare linguaggio di programmazione e` il LINGUAGGIO MACCHINA, le cui istruzioni sono direttamente eseguibili dal calcolatore.

Un programma scritto in un linguaggio diverso dal linguaggio macchina (linguaggio a piu` alto livello), deve essere tradotto in linguaggio macchina, per essere eseguito dal calcolatore.

L'utente interagisce con il calcolatore mediante i programmi che vi "girano" (che vi vengono eseguiti). Questi programmi sono "il SOFTWARE" che il calcolatore esegue.

Esempi di programmi ...



Previously on Tecniche della Programmazione

"CALCOLATORE: definizione dal punto di vista FUNZIONALE"

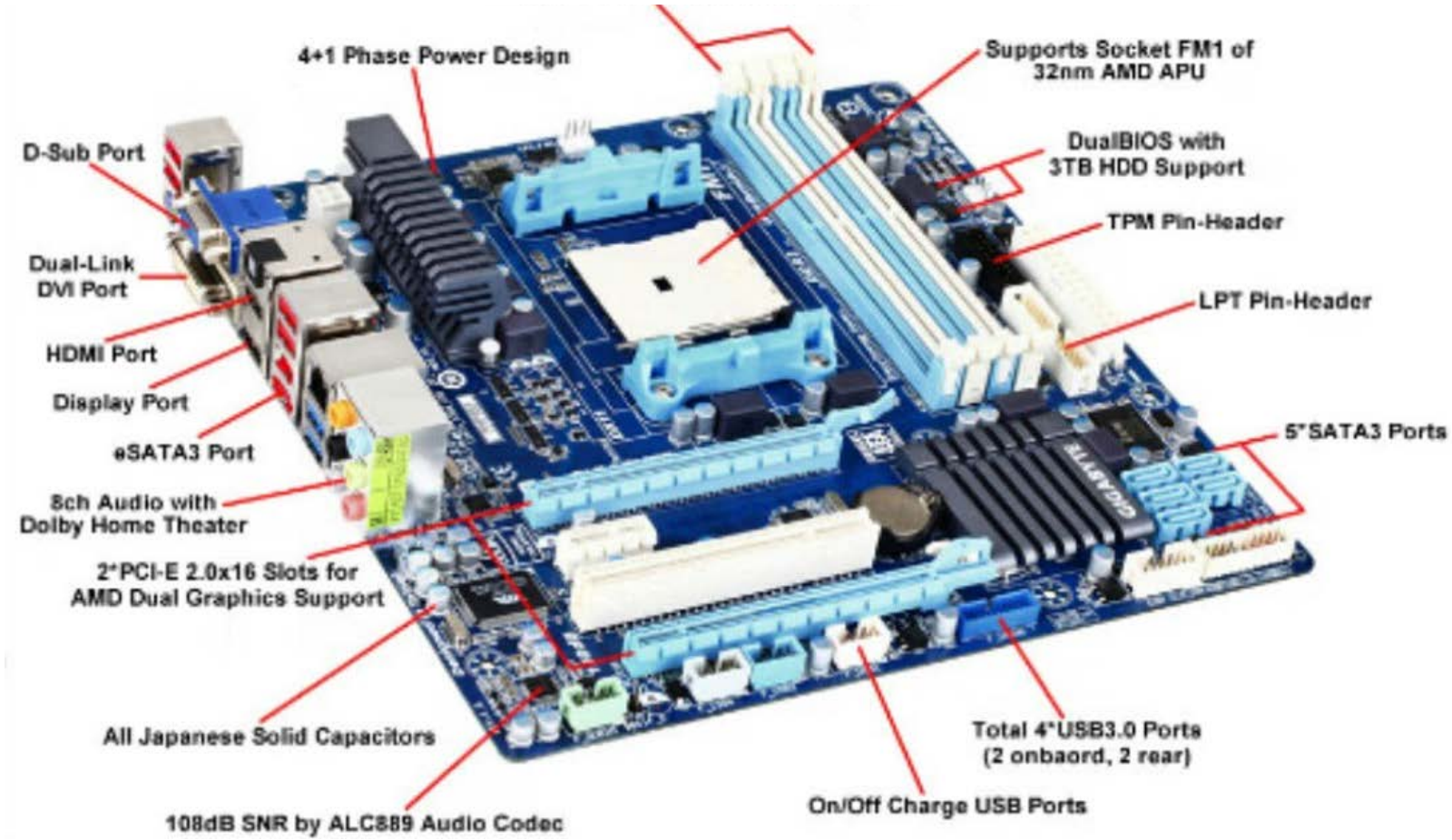
Il calcolatore e` un **SISTEMA**, composto da varie componenti fisiche.

Le componenti fisiche (memoria centrale, CPU, interfacce verso le periferiche, memoria di massa) sono guidate dalla CPU, in modo da collaborare per eseguire programmi, e fornire all'utente un comportamento (che e` quello definito (programmato) nei programmi che girano sul calcolatore (Software).

Previously on Tecniche della Programmazione

"CALCOLATORE: definizione dal punto di vista FUNZIONALE"

Il calcolatore e` un **SISTEMA**, composto da varie componenti fisiche.
Le componenti fisiche (memoria centrale, CPU, interfacce verso le periferiche, memoria di massa) sono guidate dalla CPU, in modo da collaborare per eseguire programmi, e fornire all'utente un comportamento (che e` quello definito (programmato) nei programmi che girano sul calcolatore (Software).



and now ...

Tecniche della Programmazione, lez. 2

- Architettura del calcolatore (Macchina di Von Neumann)
  - MEMORIA CENTRALE
  - CPU (Central Processing Unit)
  - BUS
  - (Interfacce per le) PERIFERICHE

- Software di Base e Software Applicativo

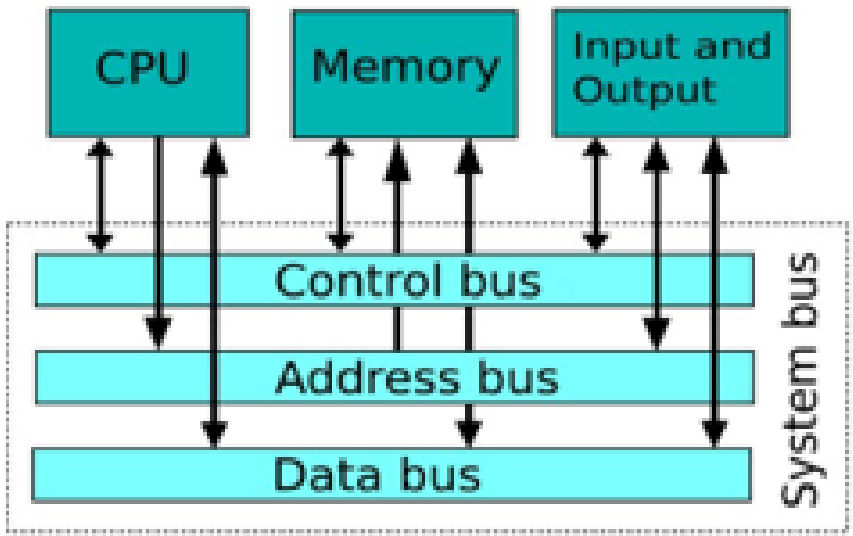
- Linguaggio Macchina (Addendum)
  - Esempio di linguaggio macchina
  - Programma scritto in linguaggio macchina
  - Esecuzione di un programma scritto in linguaggio macchina

- Programma, processo, linguaggio, prima esperienza di programmazione

Architettura del Calcolatore

... descrive

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:

cosi` , chi vuole usare le risorse del computer puo` 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, gia` 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 piuo` 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, spostamennto 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 funzionalita` (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 attivita` dedicate: scrivere un documento, leggere le email, navigare il web, videogiocare, usare un foglio elettronico, fare le slide per la lezione …

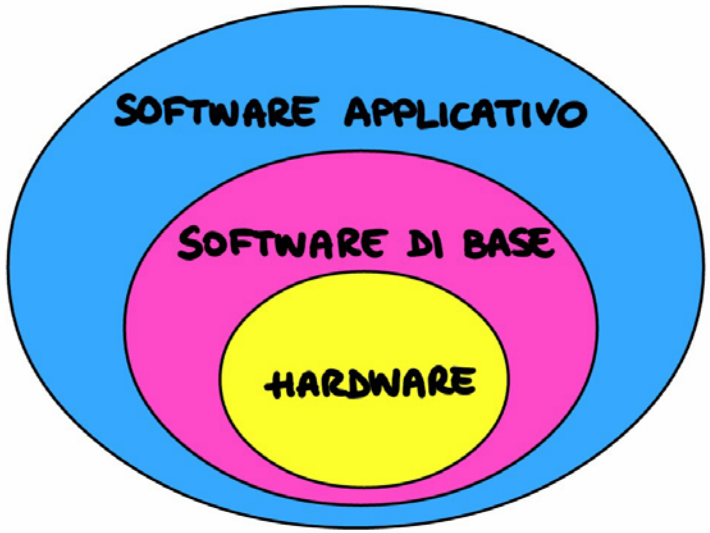
Questi programmi usano le funzionalita` offerte dal Software di base per portare a compimento le proprie … ad esempio,

mentre scriviamo un documento, il Word Processor (Applicazione) dialoga con il Sistema Operativo, per ricevere tramite tastiera (unita` 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 (unita` periferica di memoria)

Livelli funzionali

ciascun LIVELLO offre un insieme di funzionaita`

ciascun LIVELLO realizza le proprie funzionaita` 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.

In qualunque momento del suo funzionamento il calcolatore sta eseguendo programmi

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.

- Nell'eseguire programmi, il calcolatore
  riceve dati in ingresso (DATI DI INPUT) e
  "li trasforma" in dati in uscita (DATI DI OUTPUT)

- I dati sono entita` numeriche memorizzate (cioe` contenute, rappresentate!) nella memoria (una delle componenti) del calcolatore.

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

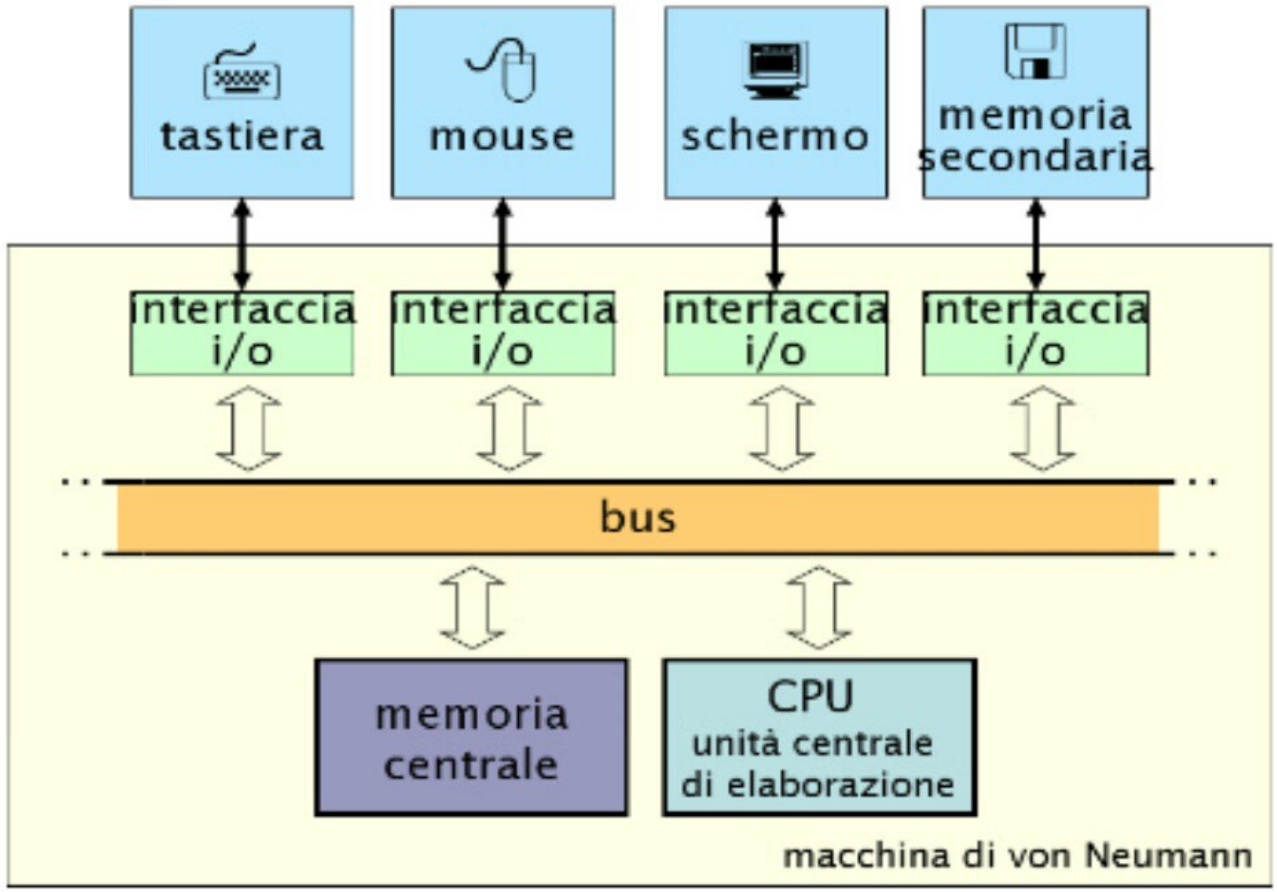
Macchina di Von Neumann

Un modello semplificato … che ancora vale …

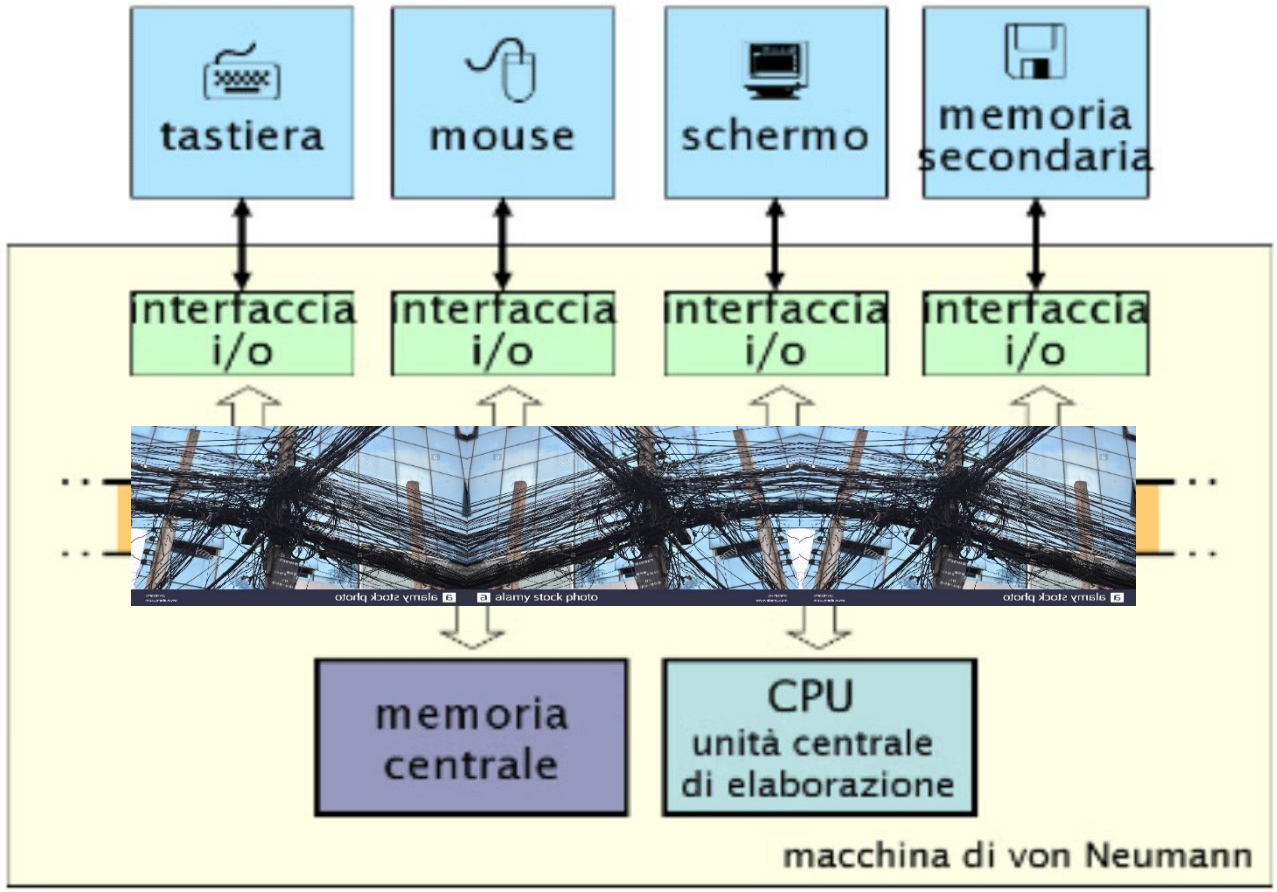
- Il calcolatore riceve dati in ingresso (DATI DI INPUT) e "li trasforma" in dati in uscita (DATI DI OUTPUT)
- I dati sono entita` numeriche rappresentate nella memoria (una delle componenti) del calcolatore.
  (l'INFORMAZIONE e` un dato o un gruppo di dati che hanno una *semantica* associata)

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

Componenti della Macchina di Von Neumann

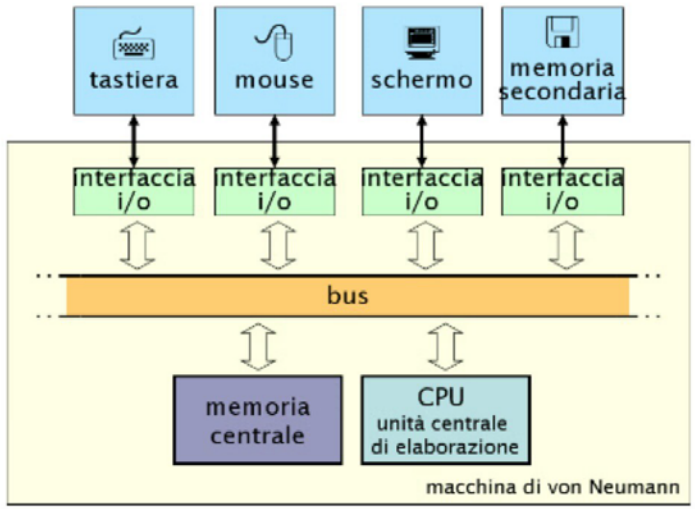


Componenti della Macchina di Von Neumann



Componenti della Macchina di Von Neumann

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

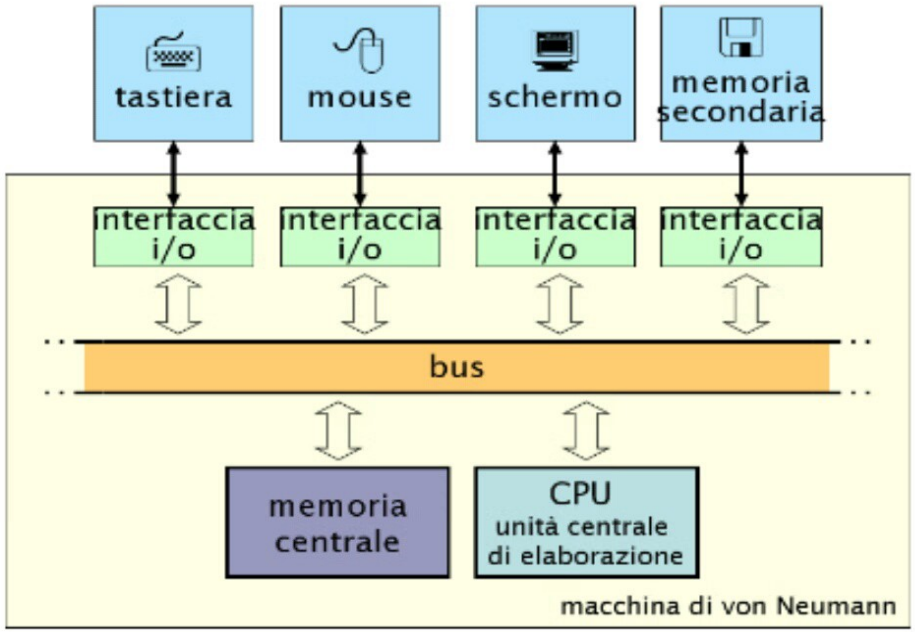


2) **CPU o PROCESSORE**: ESEGUE LE ISTRUZIONI PER L'ELABORAZIONE DEI DATI. INOLTRE CONTROLLA E COORDINA IL FUNZIONAMENTO DELLE ALTRE PARTI.

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.

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

E` fatta di **LOCAZIONI DI MEMORIA**

Ogni **LOCAZIONE** e` composta da BIT (BInary digiT)

OGNI **LOCAZIONE** 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*)

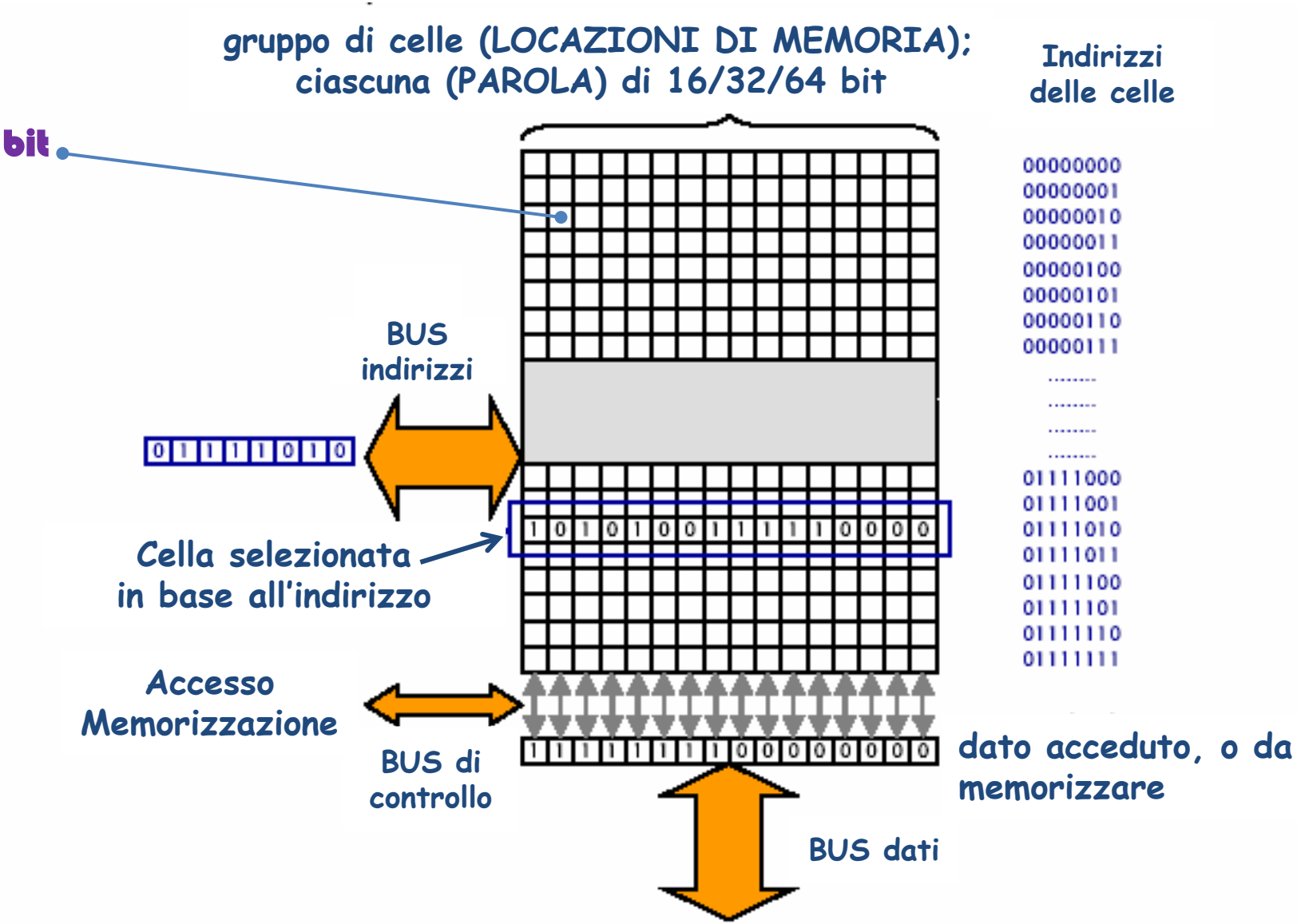
- Operazione di MEMORIZZAZIONE: un dato/istruzione viene copiato in un'area di memoria (Locazione di Memoria)
- Operazione di ACCESSO: il valore contenuto in una Locazione di memoria viene "acceduto", ad esempio viene consultato per copiarlo altrove al fine di usarlo per qualche altra operazione

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"

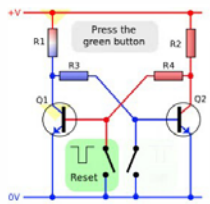
Struttura semplificata della memoria centrale



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 [ ]

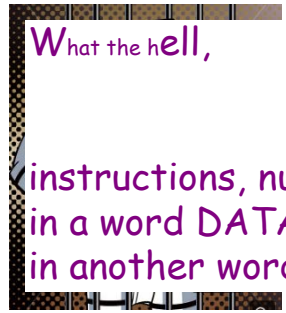
UNA CELLA HA UNA DIMENSIONE FISSA (LA DIMENSIONE TIPICA ATTUALE È [ ] BIT).
UNA CELLA CORRISPONDE AD UNA [ ], OVVERO RAPPRESENTA UNA PORZIONE DI MEMORIA CHE VIENE (TIPICAMENTE) UTILIZZATA INTERAMENTE (OVVERO LETTURE, SCRITTURE E TRASFERIMENTI VENGONO EFFETTUATI IN GRUPPI DI 32 o 64 BIT)

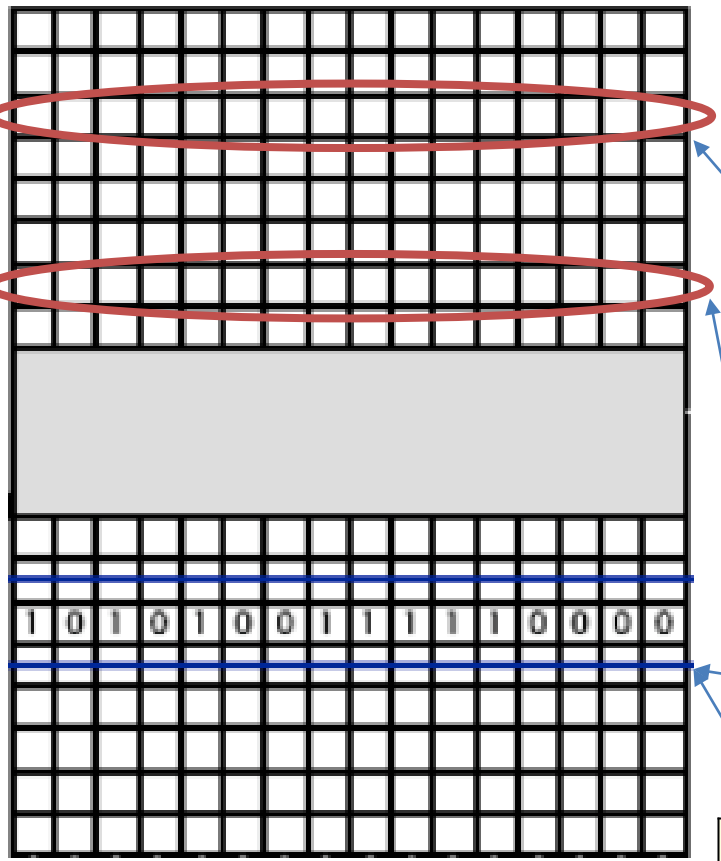
[Una cella contiene un ...] (LA SEQUENZA DI BIT MEMORIZZATA AL SUO INTERNO).

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

vedi in fondo

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





000000000 (Load)

codice operazione	IND	REG

istr. macchina (load) da memoria a registro

0000000010 (add) R1 R2

Codice operazione	REG1	REG2

istr. macchina per operazione tra due registri

```
1010100111110000
0000000000000000
1111000000000000
0000111100001111
```

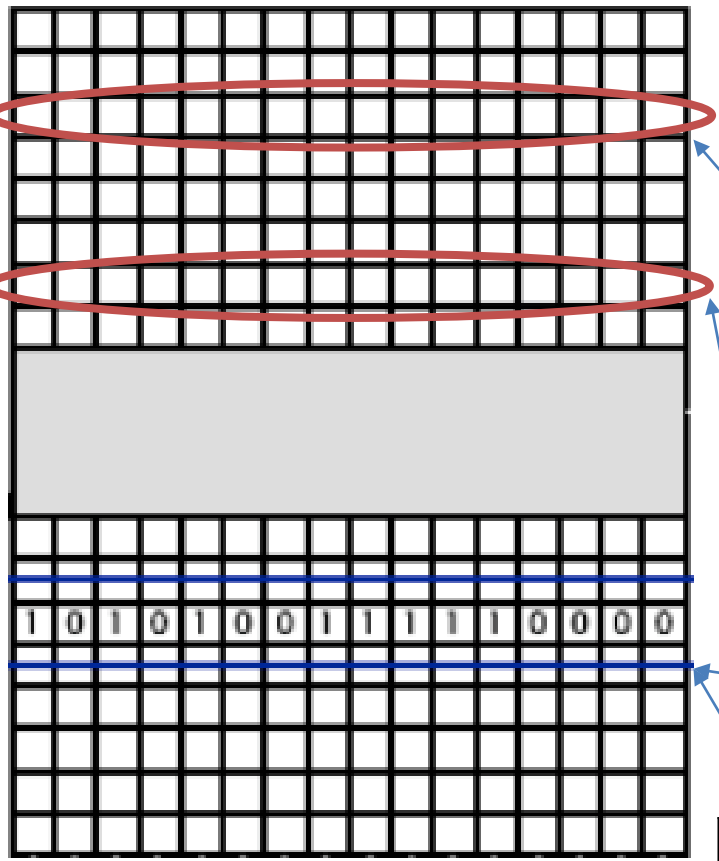
numero

```
1010100111110000
0000000000000000
```

numero

NB

numero: un dato, che sicuramente rappresenta un'informazione ...; in questi casi un numero reale rappresentato in Floating Point (1), oppure un intero rappresentato in complement a due su 32 bit (2), oppure un intero positivo rappresentato in forma binaria pura su 4 bit (3) (4), o su 16,32,64 ... bit (?), oppure un carattere, rappresentato come un intero in forma binaria su 8 bit (6), oppure quattro caratteri consecutivi (7)…



000000000 (Load)

codice operazione (3)	IND numero	REG (4)

istr. macchina (load) da memoria a registro

0000000010 (add)

Codice operazione (3)	R1 REG1 (4)	R2 REG2 (4)

istr. macchina per operazione tra due registri

```
1010100111110000
0000000000000000
1111000000000000
0000111100001111
```
(1)

```
1010100111110000
0000000000000000
```
(7)

(2)

```
10101001
```
(6)

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.

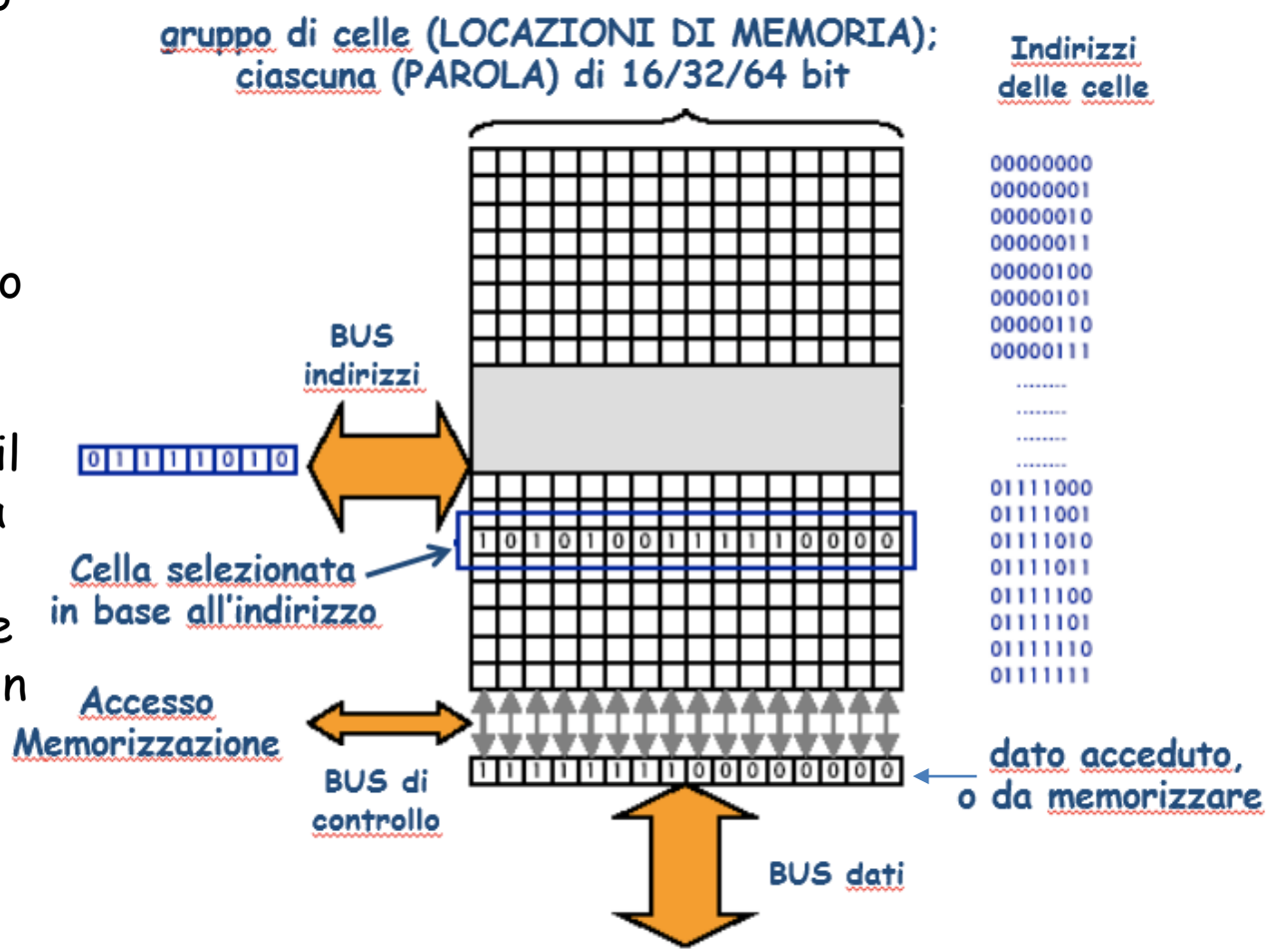
TUTTI I COMPONENTI SONO COLLEGATI AL BUS!

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 e` 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 (cioe` indipendente dall'indirizzo: … remember, Accesso Diretto: Random Access Memory

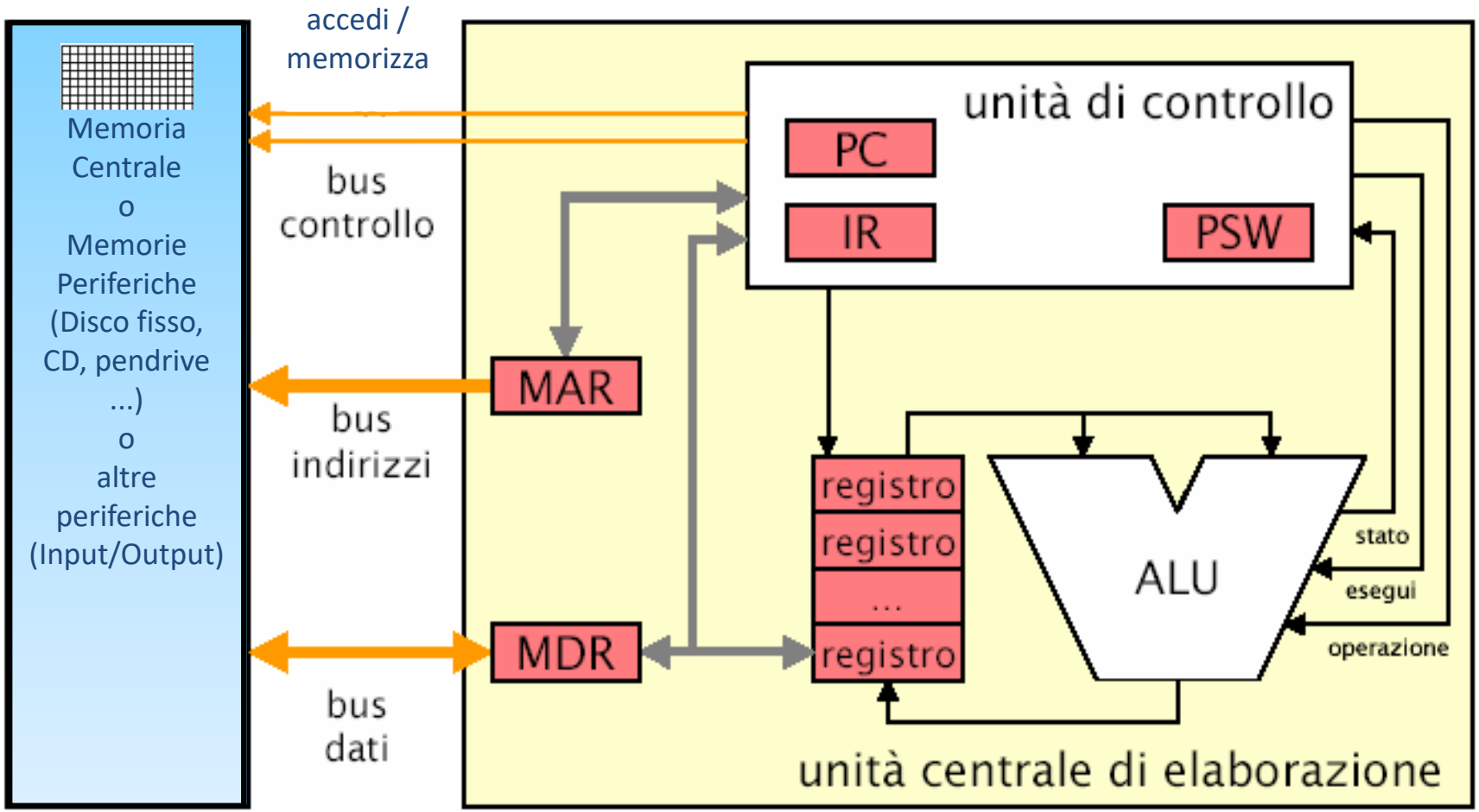
Processore


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 SEMICONDUTTORE, TIPICAMENTE SILICIO.

Struttura della CPU



Elementi di una 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 (piu`, 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'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

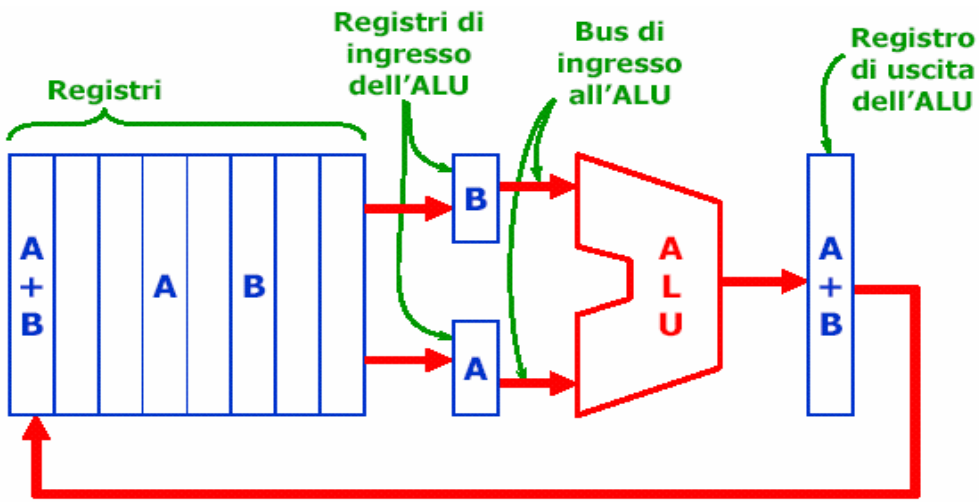
Registri (2/2)

- MAR: 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

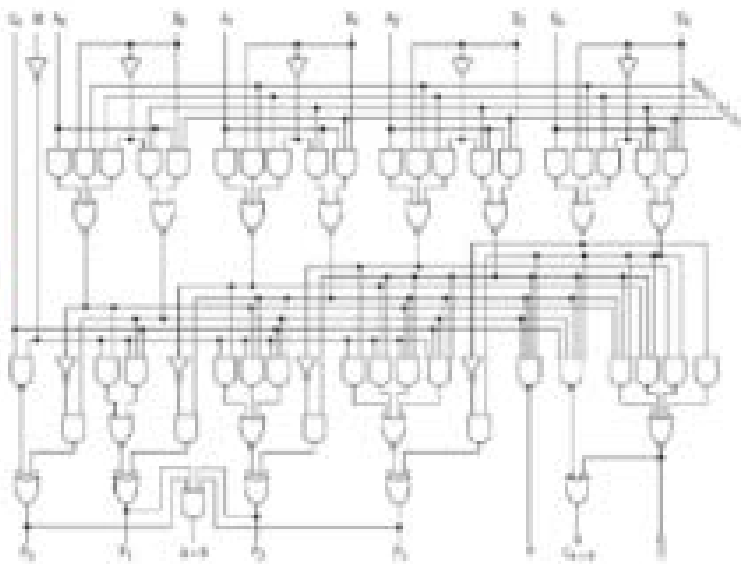
Unità Aritmetico-Logica

L'**U**nità **A**ritmetico-**L**ogica (**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, ...)

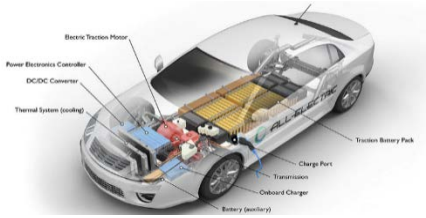
ESEGUE Istr. MACCHINA per controllo flusso esecuz.

- 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.

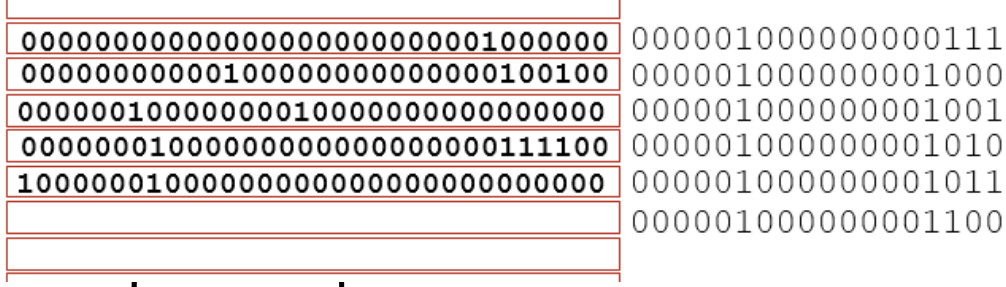
Attraverso il BUS, e i registri, i circuiti possono interagire con la memoria e con le interfacce di I/O …
Yawn …

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!



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,**
- **calcolare dati intermedi e finali,**
- **produrre dati di output** che rappresentano la soluzione.

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.

Continua con i **Cenni sul linguaggio macchina**

Esecuzione di un'istruzione macchina, da parte della CPU

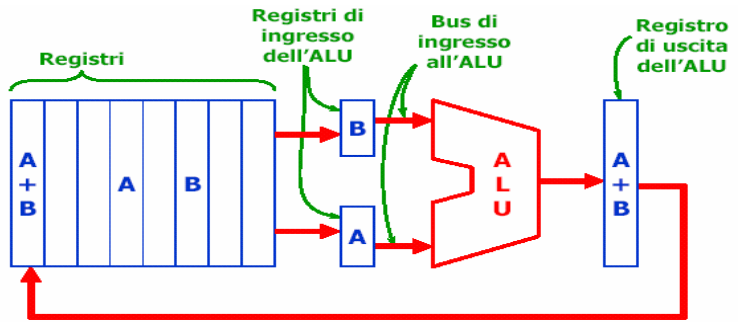
L'istruzione e`

"somma A, che sta in RAM ad un certo indirizzo, con B, che sta in RAM ad un certo indirizzo"

(per definizione il risultato verra` memorizzato dalla CU nel registro 1 della CPU)

La CU deve eseguire vaie operazioni elementari

- operazioni di decodifica dell'istruzione
- operazioni per far arrivare A in MDR (trasf. da RAM a MDR)
- copia di MDR nel registro 4, che ora contiene A
- operazioni per far arrivare B in MDR
- copia di B nel registro 6, da MDR
- copia di A dal registro 4 al registro di ingresso ALU
- copia di B dal registro 6 al registro di ingresso ALU
- attivazione del circuito giusto (indicato dal codice dell'istruzione)
- il circuito deposita il risultao nel registro di uscita
- copia del registro di uscita nel registro 1
- e poi probabilmente un'altra istruzione del programma copiera` A+B in una locazione della RAM, da dove forse un'altra istruzione del programma potrebbe eseguire una "stampa in output" del contenuto dei quella locazione ...



Continua con i Cenni sul linguaggio macchina

Tempo Macchina

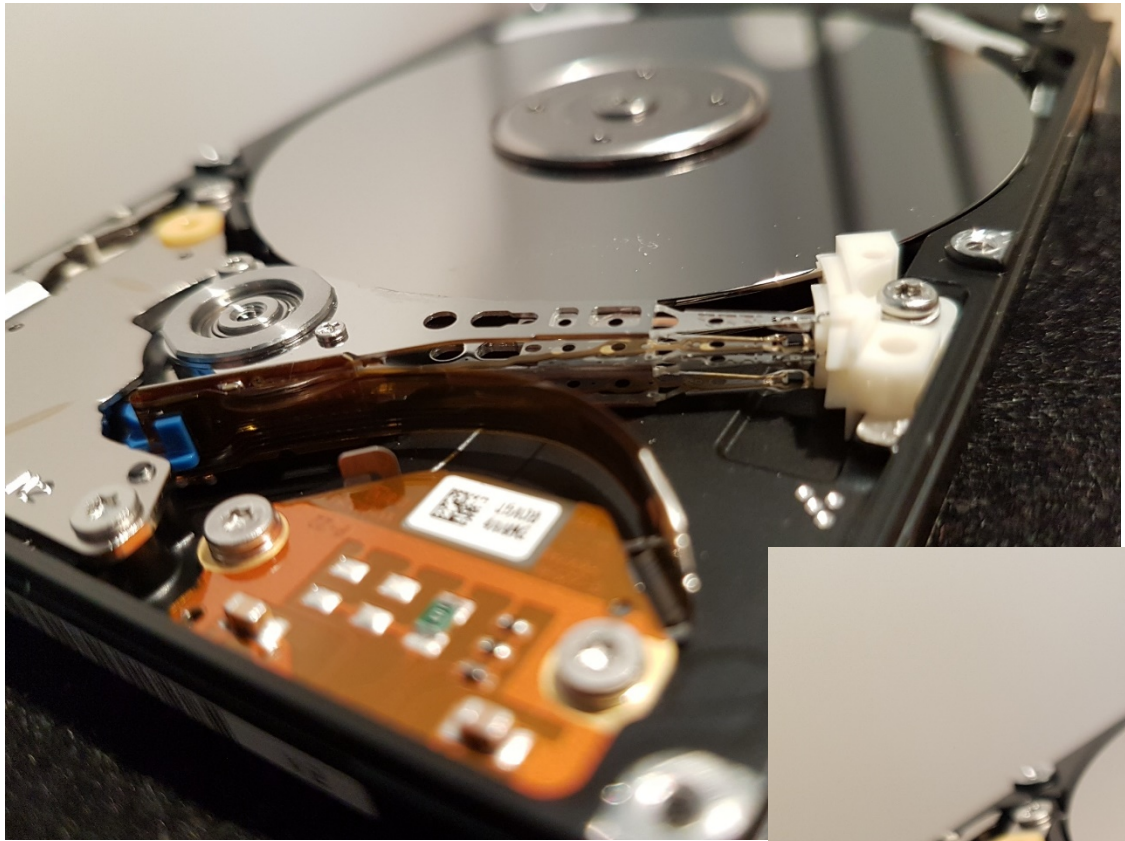
Quante istruzioni puo` eseguire la CPU in un certo tempo?

Cosa e` il clock?

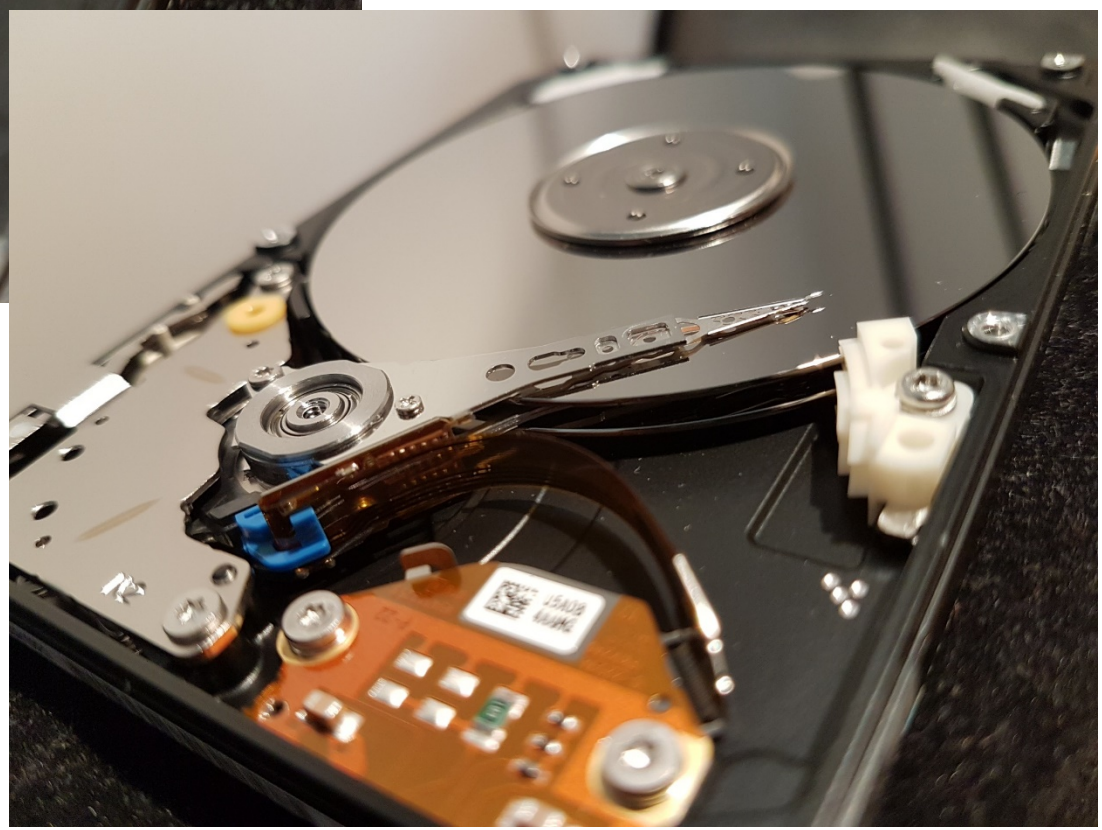
E la cache?

Memoria Secondaria: e` poco importante?

(Vedi in fondo: Approfondimenti)



normale HD



Memorie Secondarie

HARD-DISK

CD-ROM

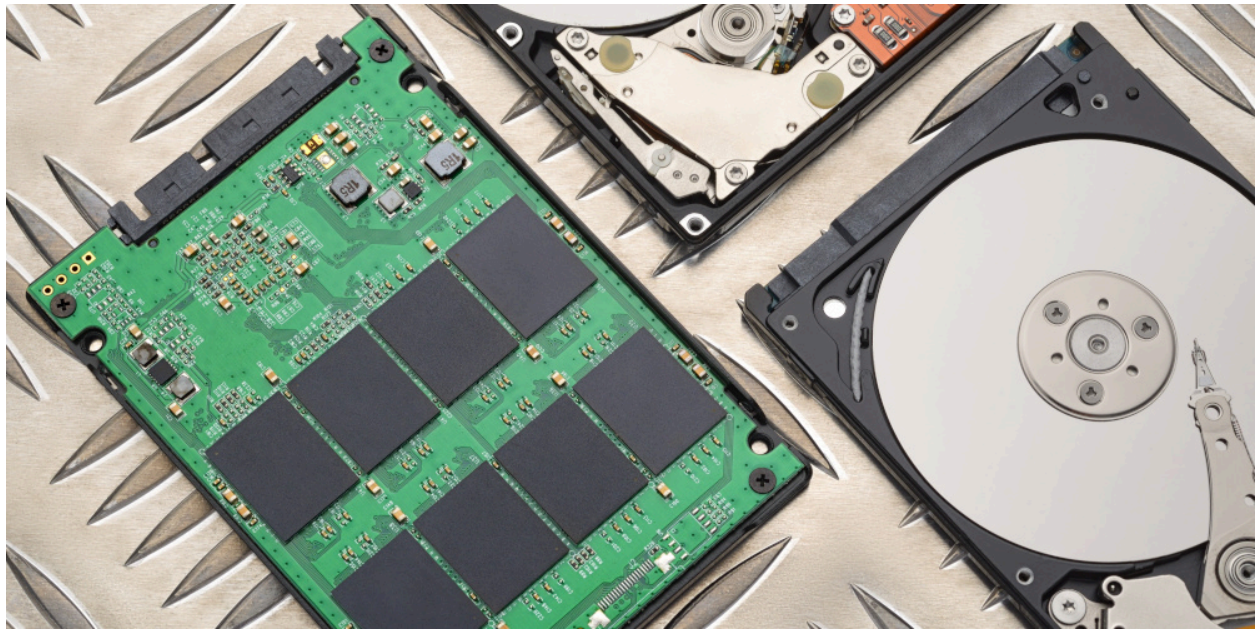
DVD

NASTRO MAGNETICO

USB FLASH DRIVE

Memorie Secondarie

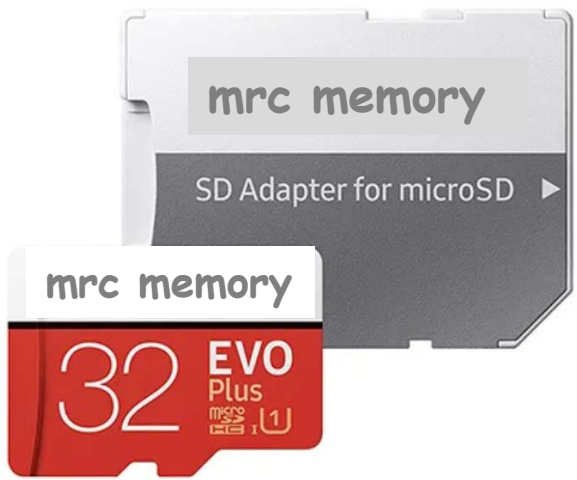
normale HD



SSD

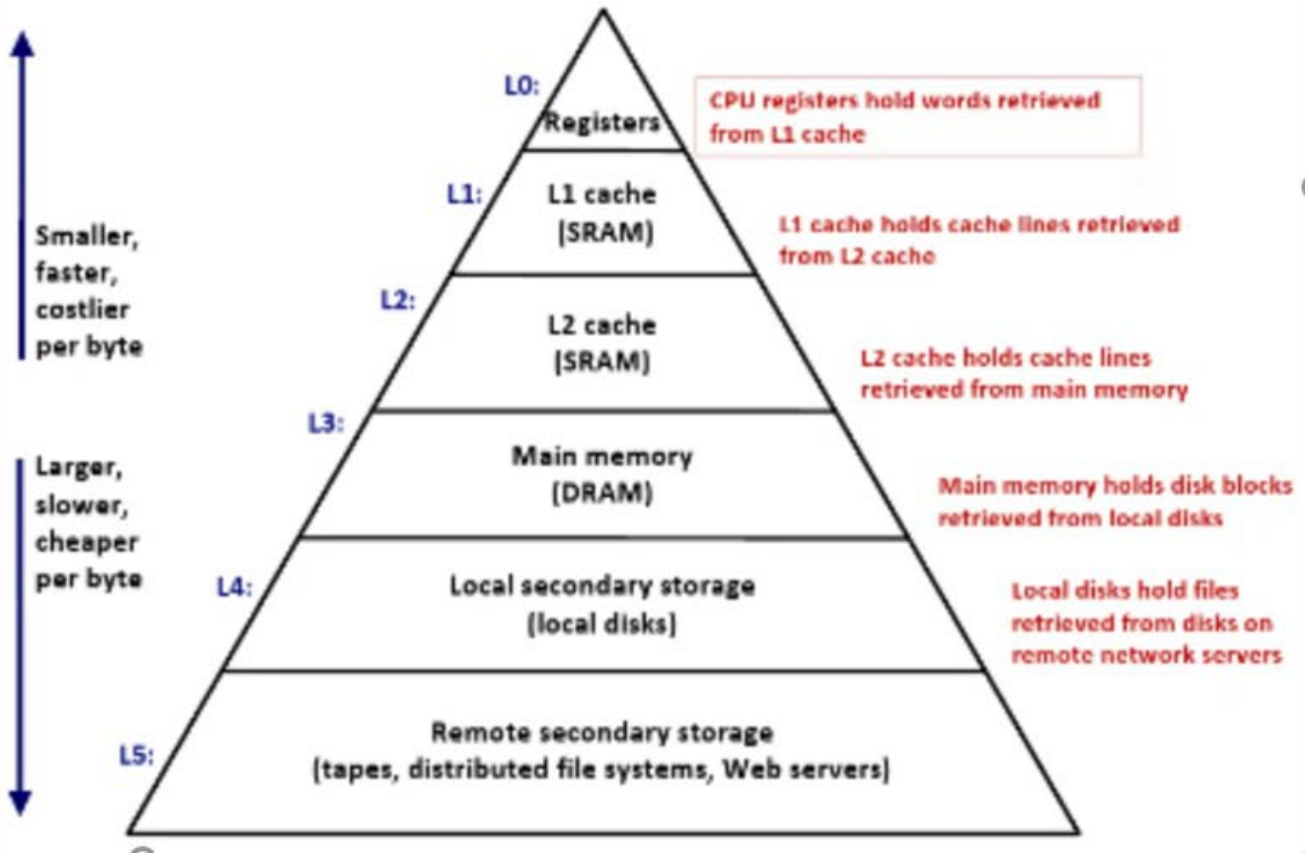
scheda SD, SDHC

micro SD



4 cose importanti riguardo alle memorie …

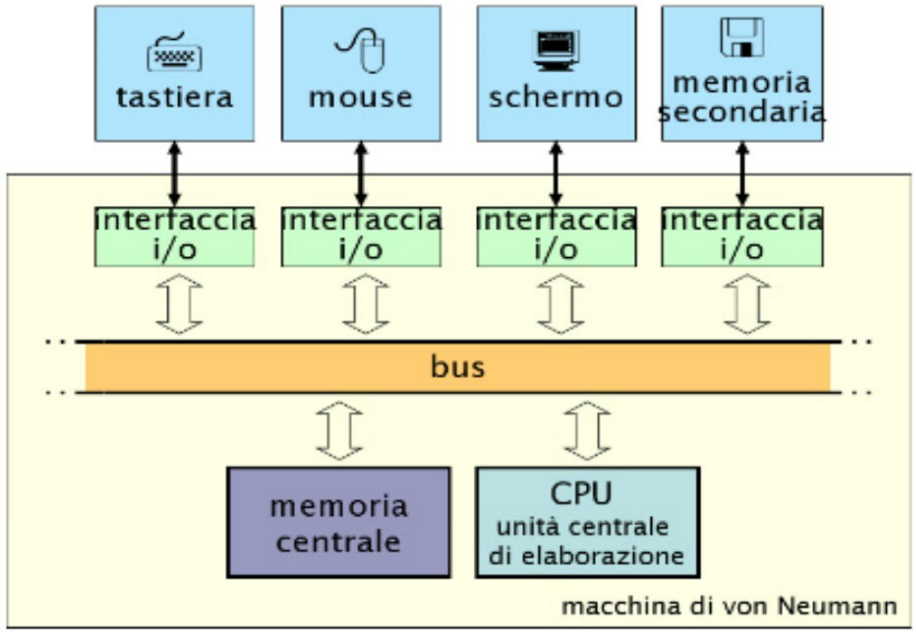
- capacita` … KB, MB, GB, TB …
- tempo di accesso
- velocità (da memoria a RAM)
- costo



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

1ns = 1 miliardesimo di sec. 10^{-9}
1 μs = 1 milionesimo di sec.: 10^{-6}
1 ms = 1 millesimo di secondo : 10^{-3}

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 COMPRENSIBILE ALLA PERIFERICA E VICE VERSA. POSSONO ESSERE PENSATE COME DELLE PICCOLE CPU DEDICATE AL FUNZIONAMENTO DELLE PERIFERICHE.

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.

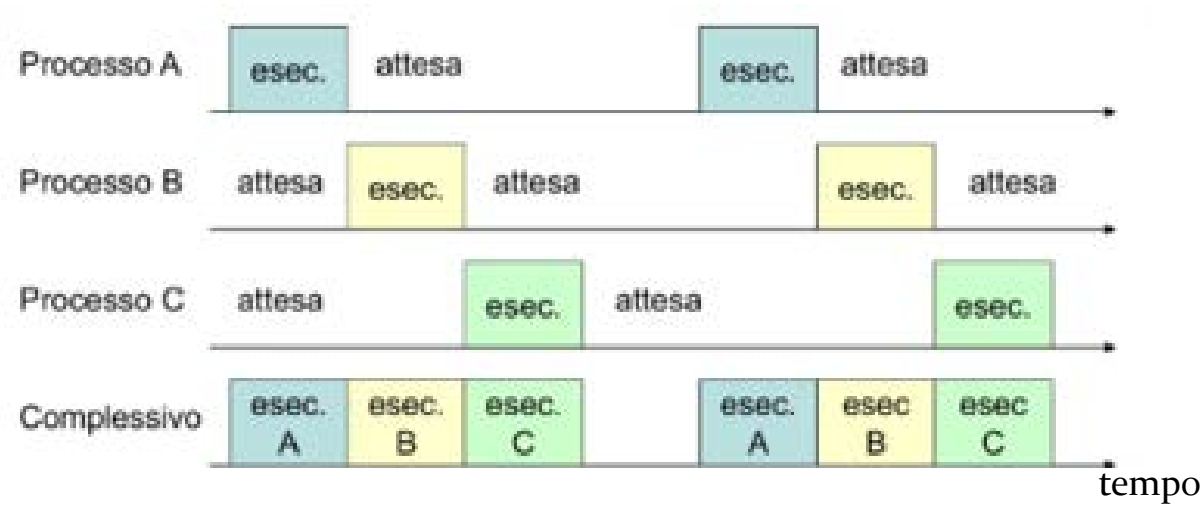
Multitasking

Ignoriamo per un momento i processori "multi-core".
Abbiamo un solo processore, che puo` eseguire un solo processo alla volta.

Come fanno molti programmi, ovvero molti processi, a "girare" in contemporanea sul calcolatore?

Ad esempio, il Sistema Operativo (SO) gira sempre ... quando lanciamo in esecuzione un programma ... anche quello gira ... E non si puo` aspettare che un processo finisca, per poi eseguire il seguente

Il trucco e` che l'esecuzione dei processi
viene organizzata ("*Scheduling*") in modo che in ogni istante un solo processo sia in effettiva esecuzione, mentre tutti gli altri sono "sospesi".

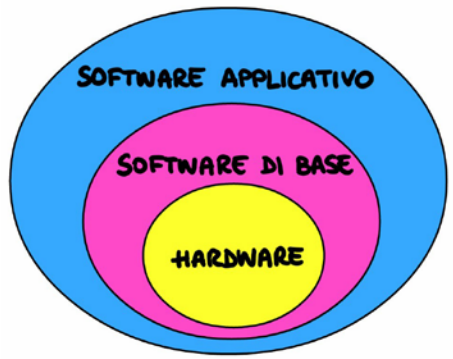


Il SISTEMA OPERATIVO

In sostanza, e` il software di base, quello che gira sul calcolatore sin da subito dopo il "bootstrap" e che permette di usare il calcolatore.

Come lo permette?

- "Gestisce" le risorse del calcolatore (tutte ... le memorie, le cpu/gpu, le interfacce delle periferiche ... e quindi le periferiche ...)
  - per l'esecuzione di un programma
  - per l'uso da parte dell'utente
  - per l'uso da parte del software applicativo



- "Virtualizza", rende piu` astratte, quelle risorse, in modo da permettere di usarle, all'utente e soprattutto al software applicativo
  - fornisce l'interfaccia di base tra l'utente e il calcolatore
  - fornisce l'interfaccia di base tra il software applicat. e il calcolatore

SO e l'esecuzione di un programma

Prima abbiamo detto che il SO "gestisce e virtualizza" le risorse del calcolatore ... (bla bla)

Un esempio di virtualizzazione e` il modo in cui il SO ci "mostra" e "fa usare" le risorse. Ad esempio con il meccanismo di cartelle ("directory") con cui interagiamo per aprire file e chiedere l'esecuzione di programmi (il doppio click).
Oppure con l'analogo meccanismo "a linea di comando".

eh? linea di che? ...

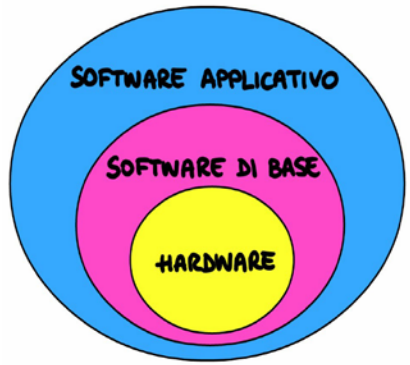
Ecco cosa fa il SO quando cerchiamo di eseguire un programma: noi facciamo doppio click ... e il SO

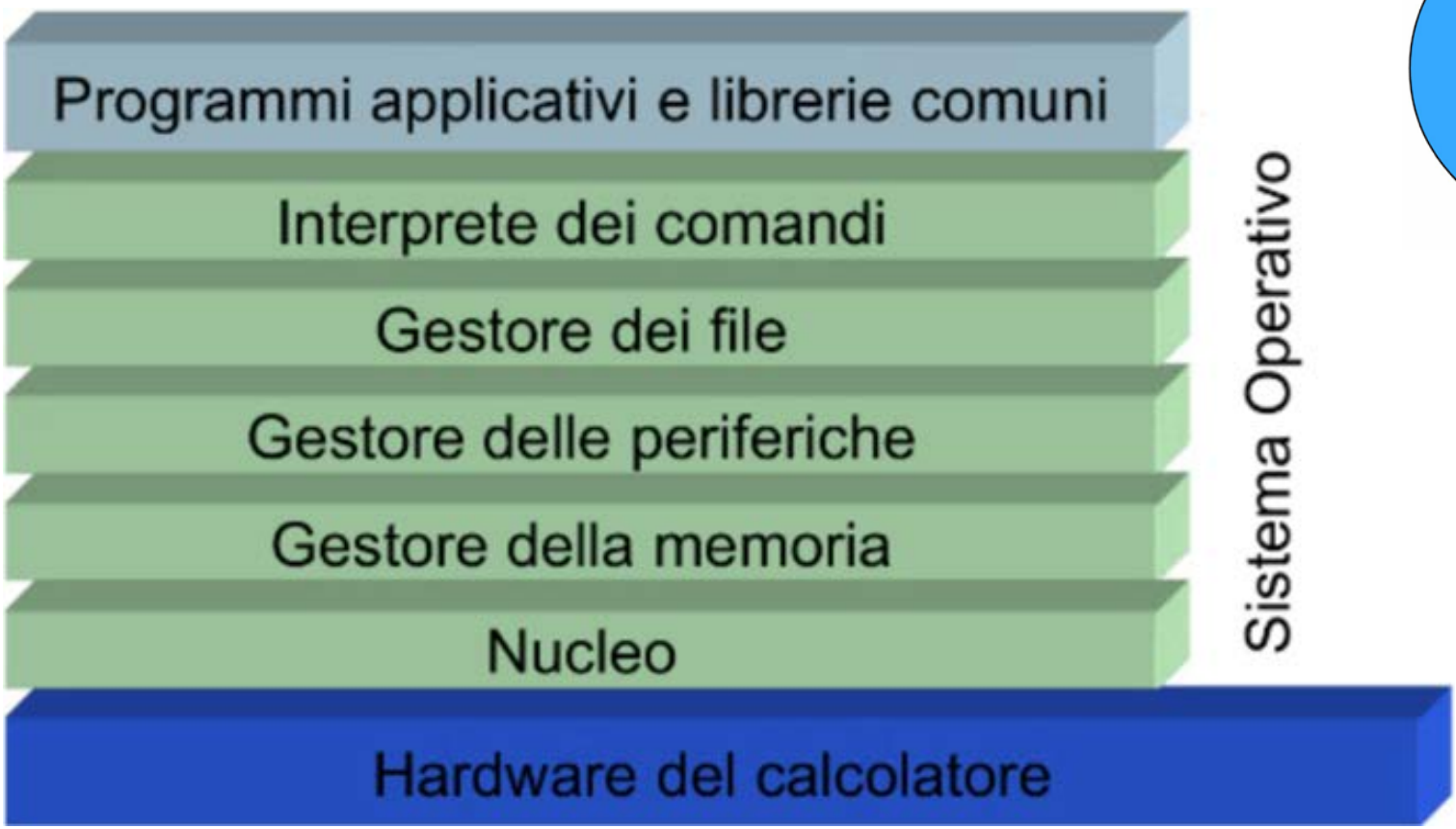
- intercetta il doppio click e le sue coordinate, in base alle quali capisce quale programma eseguibule va eseguito
- in una tabella opportuna (del file system) individua l'indirizzo iniziale del programma eseguibile nella memoria secondaria;
- carica il programma eseguibile in memoria centrale, dopo aver deciso quale parte della RAM occupare
- ora il programma e` in RAM, come sequenza di istruzioni macchina, a partire da una certa locazione della RAM, cioe` a partire dall'indirizzo di quella locazione;
- viene richiesto alla CPU di iniziare ad eseguire (RUN) il programma
- poi entrera` in azione il multitasking, attraverso lo scheduler nel nucleo del SO
- ...

e noi vediamo solo il programma in esecuzione ...

Il SISTEMA OPERATIVO (struttura a livelli)

Il SO in realta` e` un conglomerato di programmi che si occupano di ... tutto, ciascuno per la sua area di responsabilita` .



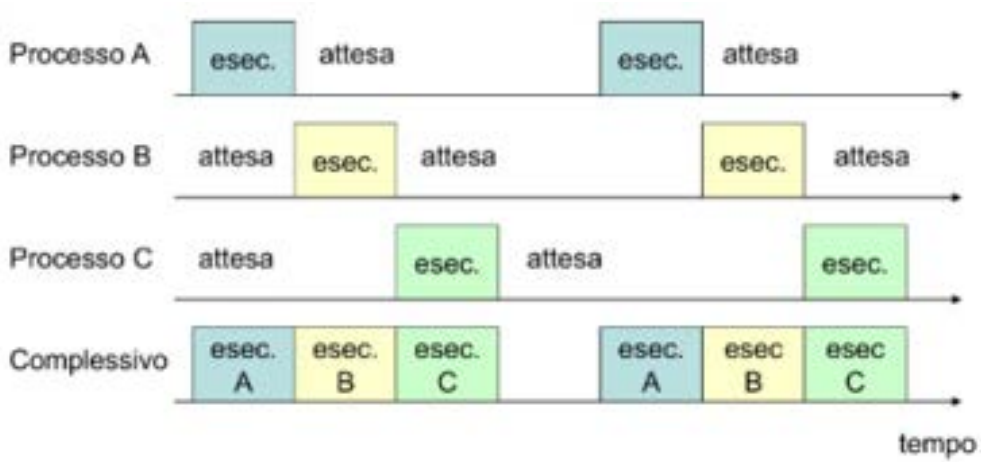


Nucleo (Kernel) del SO

E` il responsabile di tutte le operazioni di "basso livello", cioe` quelle che impegnano/usano l'harwdare.

Dialoga con le altre componenti, per gestire l'uso della memoria centrale e della cpu, la gestione dei processi (attivazione, accesso alle proprie parti di memoria), la gestione delle periferiche (anche di memoria: file system)

E` il nucleo del SO (in particolare lo "scheduler") a *gestire* l'*impegno* del processore,
cioe` la concessione ai processi dell'uso, per un certo tempo, del processore, o dei processori, o dei core del processore



SO – Gestore della Memoria Centrale (e non solo)

La RAM e` limitata.

Quando un programma viene eseguito, il suo eseguibile e` caricato in memoria, ed occupa spazio ... da un certo indirizzo della RAM, fino ad un altro certo indirizzo.

Il Gestore cura l'assegnazione di spazi di RAM ai processi.
Ci possono essere molti processi contemporaneamente in RAM. E lo spazio deve essere gestito attentamente.

Il Gestore cura anche i trasferimenti da Memoria Secondaria a Memoria Centrale

... e viceversa. Gia`, perche' a volte la RAM proprio non basta.

Allora, per ogni processo, solo parti di esso possono essere presenti in RAM, mentre il resto viene conservato in un'area di "swap" nella Memoria Secondaria.
La memoria secondaria coinvolta e` chiamata anche **Memoria Virtuale**.
Le porzioni di codice di un processo spostate tra RAM e memoria virtuale vengono chiamate pagine (oppure segmenti, se sono di dimensione flessibile).
Il processo, curato dal gestore della memoria, di spostamento di queste pagine e` chiamato **paginazione/segmentazione**.

SO – Gestore delle Periferiche

E` il sottoinsieme del SO che gestisce le INTERFACCE I/O con le quali le Periferiche sono associate al calcolatore. Delle periferiche di memoria parliamo dopo.)

Il Gestore in questo caso permette che l'utente possa usare le periferiche mediante un semplice meccanismo di interazione, basato ad esempio su un software di interazione (cosi` usiamo una stampante, ad esempio).

Un driver viene istallato sul calcolatore per permettere la gestione della periferica. E` un programma, o un insieme di programmi, che permette al Gestore di ... gestire la periferica.

Cosa fa il gestore?

- **Virtualizzazione della periferica:** l'utente non deve conoscere i particolari tecnici del funzionamento della periferiche: basta che interagisca attraverso il software apposito.
- **Trasferimento/Traduzione dati:** Oltre che in seguito ad interazioni con l'utente, la periferica potrebbe essere usata anche da processi in corso di esecuzione, senza particolari input diretti dall'utente. Il Gestore cura il trasferimento delle richieste (da utente o da processo) e la trtaduzione delle richieste in comandi direttamente eseguibili dalla periferica (attraverso il driver).
- **Gestione conflitti tra processi:** Il gestore stabilisce la priorita` d'uso della periferica e fa in modo che richieste da processi/utenti diversi non collidano.

SO – Gestore dei File: FILE SYSTEM

E` il gestore delle periferiche di memoria secondaria (il DISCO FISSO, ad esempio)

Il Gestore e` responsabile della gestione dei dati memorizzati nelle unita` di memoria secondaria.

I dati sono memorizzati in sequenze di byte. Il nome generico di queste sequenze e` FILE.
Fisicamente un FILE e` individuato dal suo indirizzo di inizio nella memoria, e da altre informazioni, tra cui quanto e` grande, o l'indirizzo dove termina.

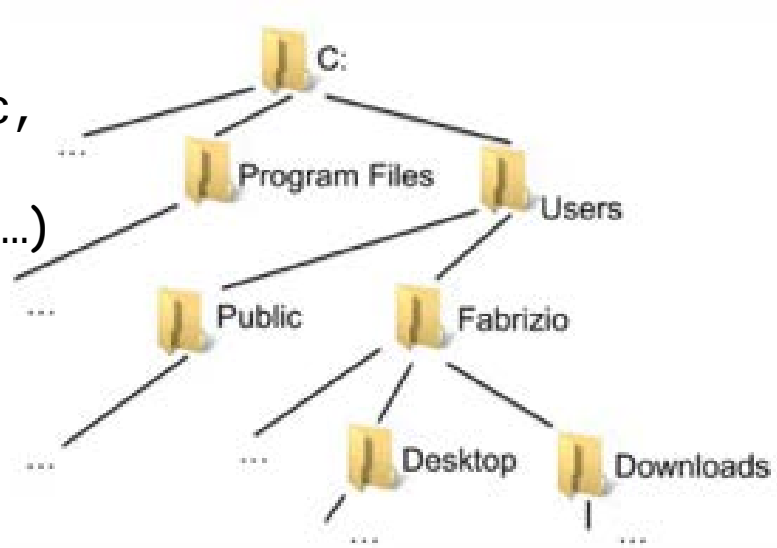
Le informazioni fisiche dette sopra vengono virtualizzate, in mod da offrire all'utente un meccanismo di selezione ed accesso ai file, gerarchico e basato sulle metafore di FILE e CARTELLA (*DIRECTORY).
Un file e` contenuto in una cartella, una cartella in un'altra ... e in cima a questa gerarchia c'e` la periferica di memoria (ad esempio C:\ ...)

File, Directory, Volume

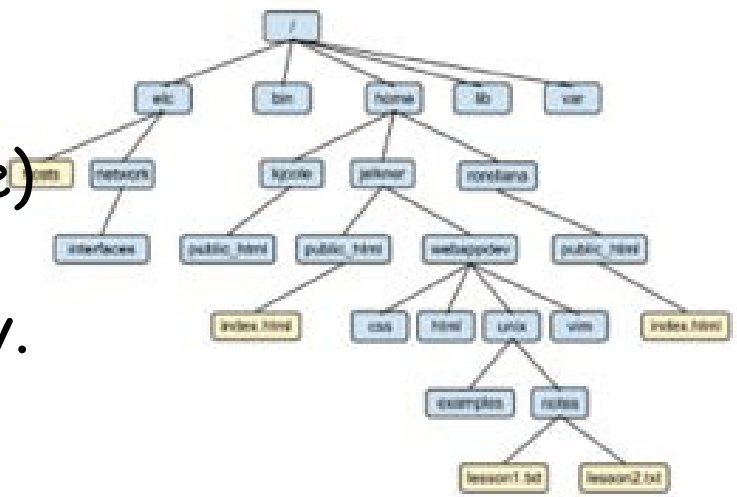
("logica" = ha nome e posizione in memoria …)

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, …`)



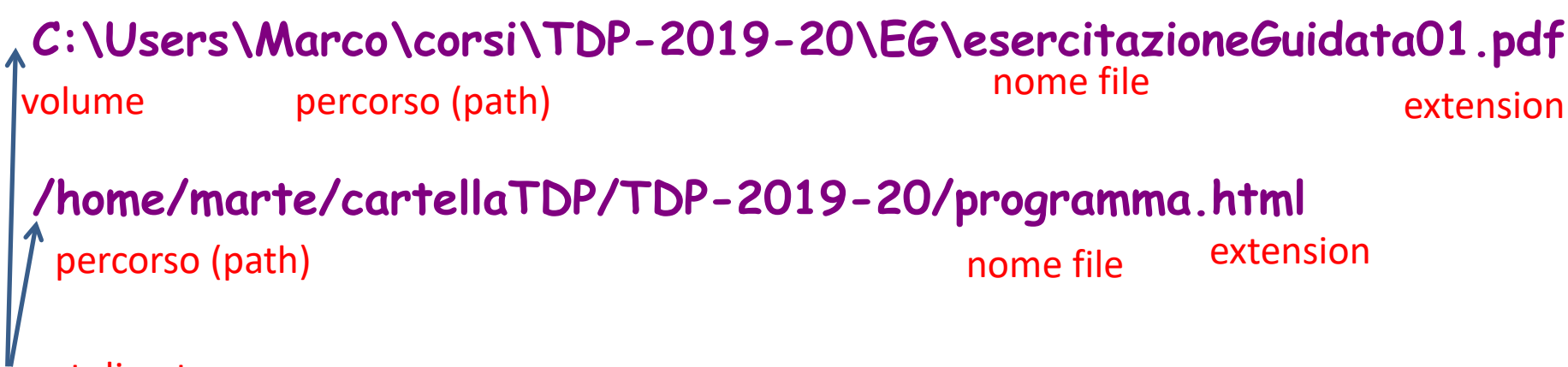
Una **DIRECTORY ("cartella")** e` il "contenitore logico di file" (*files*)
tutti i file (files) sono in directory (directories), organizzate in modo gerarchico, "ad albero"



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.



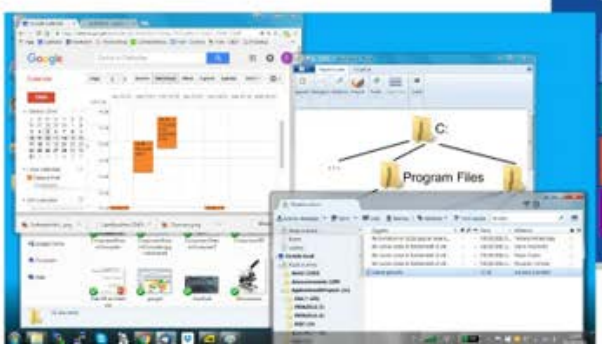
SO – Interprete dei comandi

E` collegato all'interfaccia tra SO e utente.

Attraverso l'**interfaccia**, l'utente capisce dove sono i dati e i programmi e puo` attivare programmi

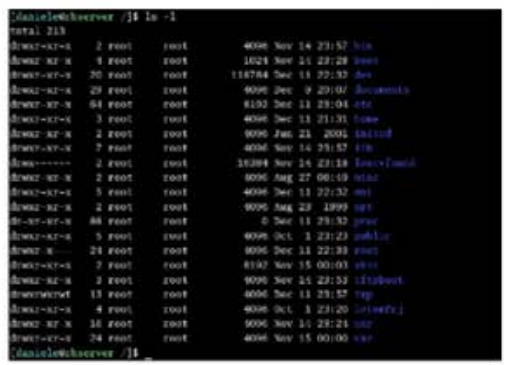
Questa interfaccia puo` essere grafica o "a linea di comando" ma comunque serve ad inviare richieste di consultazione di dati, o attivazione programmi, che poi vengono debitamente interpretate da questo sottoinsieme del SO, per dar loro seguito.

interfaccia di software



interfaccia SO

interfaccia SO



Tecniche della Programmazione, lez. 2

- Approfondimenti

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**.

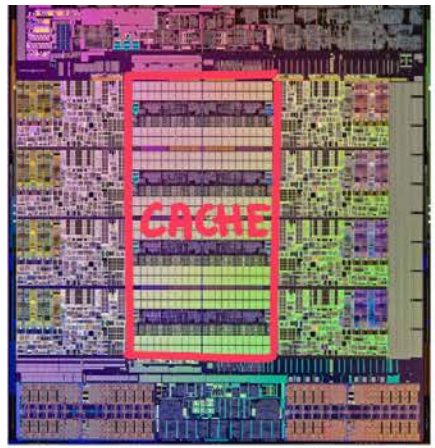
Come funziona?
Parti contigue della RAM vengono trasferite nella cache:
la CPU interagisce con quelle.
Quando i dati che servono non si trovano in cache, altre parti di RAM vengono messe in cache, al posto di quelle inutili.

Cosi` 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 PRINCÌPI:

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.

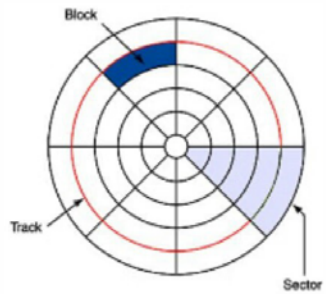
Memorie Secondarie (o "di massa" se proprio insistete)

Rispetto alla RAM, sono

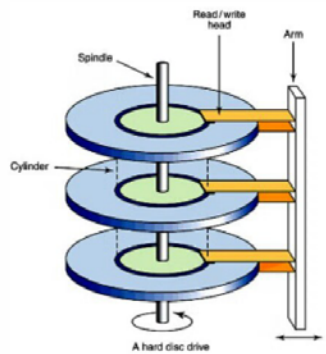
- PIÙ CAPIENTI (CENTINAIA DI GB CONTRO QUALCHE GB NEI COMPUTER ATTUALI)
- PERMANENTI (CONSERVANO IL CONTENUTO ANCHE DOPO LO SPEGNIMENTO DEL COMPUTER)
- PIÙ ECONOMICHE
- PIÙ LENTE (IL TEMPO DI ACCESSO AI DATI CONTENUTI IN MEMORIA SECONDARIA È MOLTO MAGGIORE DEL TEMPO DI ACCESSO AI DATI CONTENUTI IN MEMORIA 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.

Tecniche della Programmazione, lez. 2

- soluzioni

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.

Esercizio: vedi e riempi le due prossime slide, senza guardare quelle precedenti; poi controllo se quello che hai scritto e` giusto.

...

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 (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