

# Fondamenti di Informatica

---

## CAP. 3

Accademia di belle Arti di Verona

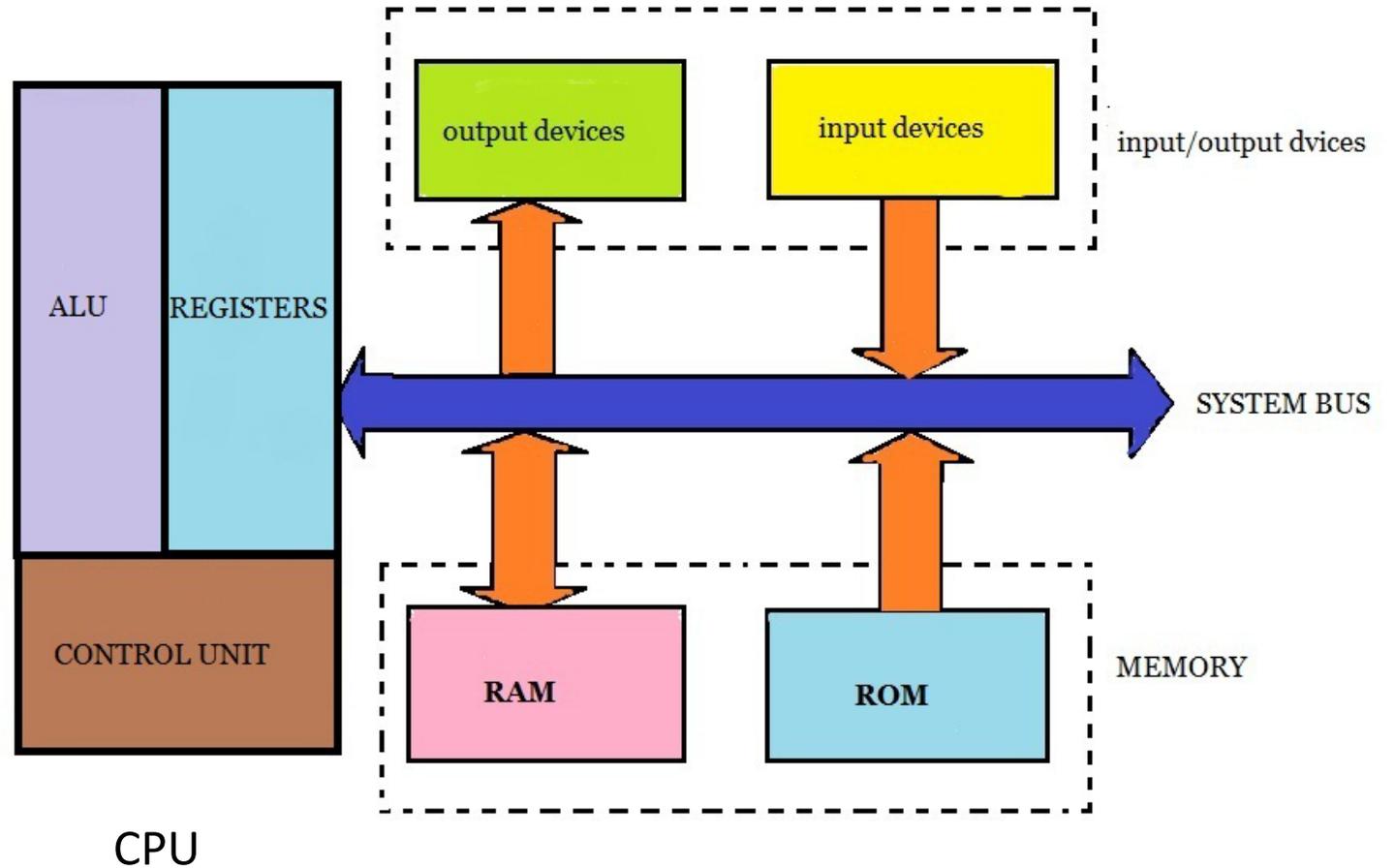
Università degli Studi di Verona

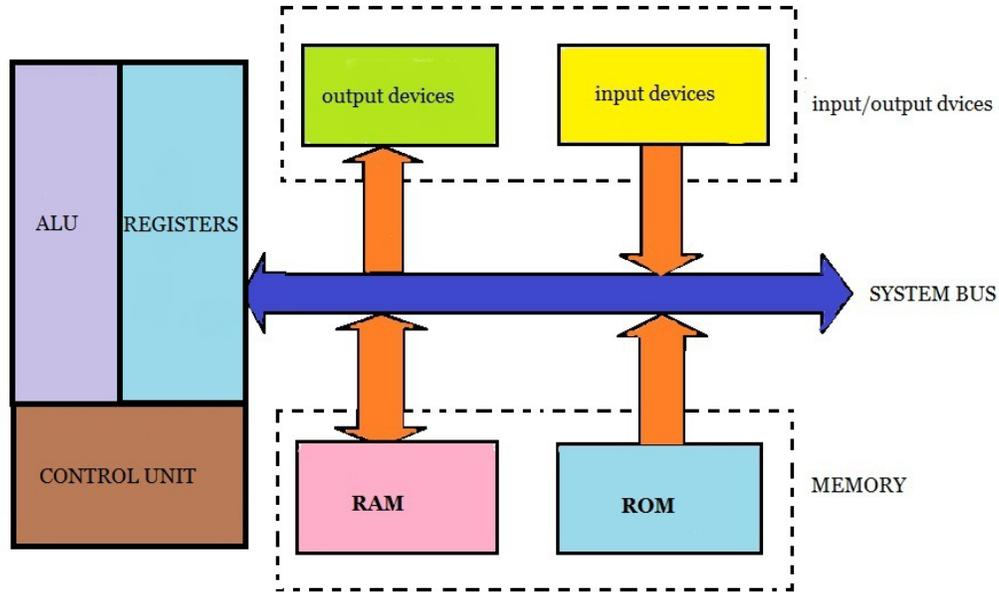
A.A. 2021-2022

Docente - Vincenzo Giannotti

# CAPITOLO 3 – COME FUNZIONA IL COMPUTER

# La CPU (Central Processing Unit)





# Struttura della CPU

- La CPU (microprocessore) rappresenta il cuore del Computer e per questo è detta Unità di Elaborazione «Centrale».
- La CPU governa il Computer impartendo «ordini» a tutti gli altri organi e interpretando le istruzioni che noi stessi gli diamo: direttamente mediante comandi del SO o indirettamente con dei programmi.



## Operazioni logiche fondamentali Tabelle di verità

<u>A</u>	<u>B</u>	<u>A or B</u>	<u>A</u>	<u>B</u>	<u>A and B</u>	<u>A</u>	<u>not A</u>
0	0	0	0	0	0	0	1
0	1	1	0	1	0	1	0
1	0	1	1	0	0	<i>(negazione)</i>	
1	1	1	1	1	1		

*(somma logica)*                      *(prodotto logico)*

Le tabelle elencano tutte le possibili combinazioni in ingresso e il risultato associato a ciascuna combinazione

## La CPU – ALU (Arithmetic and Logic Unit)

Innanzitutto il nostro computer deve poter svolgere delle operazioni. Se dobbiamo fare una operazione aritmetica, per esempio una somma, abbiamo bisogno di una «calcolatrice» che ci permetta di farlo. Questa calcolatrice risiede all'interno della CPU ed è la ALU. Poiché l'acronimo di ALU è Unità Aritmetica e Logica se ne deduce che questa parte della CPU non solo può effettuare operazioni di tipo aritmetico, ma anche operazioni di tipo logico, dette anche «**operazioni booleane**».

# La CPU - ALU

• Oltre agli operatori fondamentali possiamo trovare anche altri operatori booleani:

- **NAND** (NOT AND)

A	B	
0	0	1
0	1	1
1	0	1
1	1	0

- **NOR** (NOT OR)

A	B	
0	0	1
0	1	0
1	0	0
1	1	0

- **XOR** (OR esclusivo)

A	B	
0	0	0
0	1	1
1	0	1
1	1	0

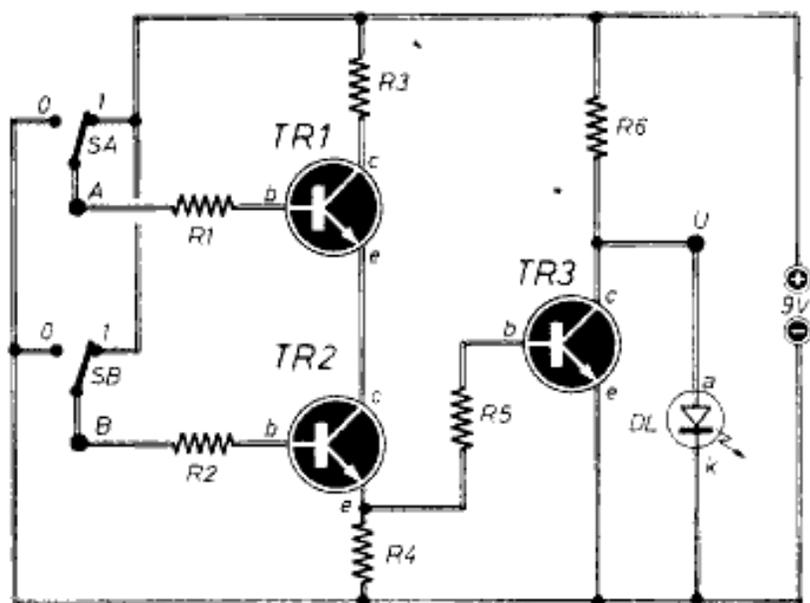
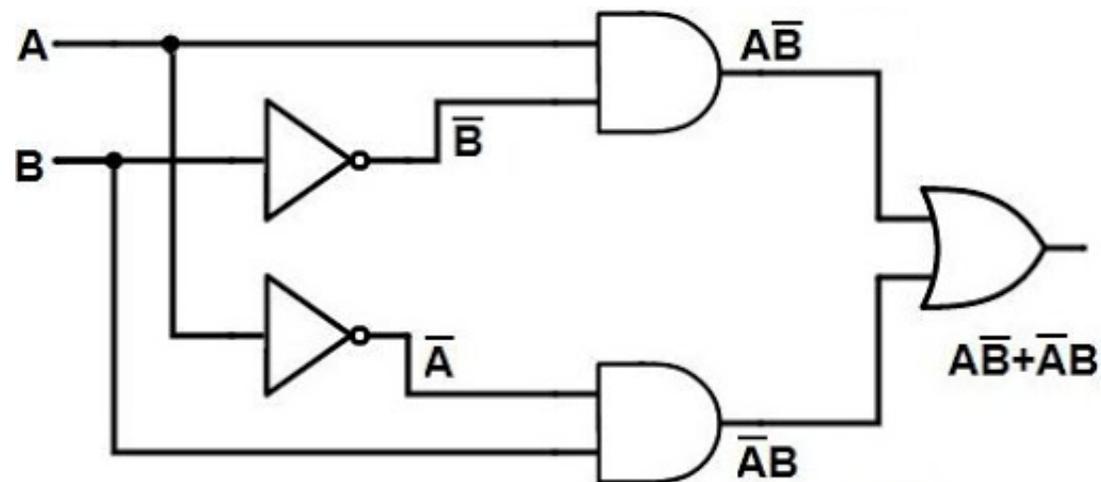


Tabella verita NAND

A	B	U
0	0	1
0	1	1
1	0	1
1	1	0



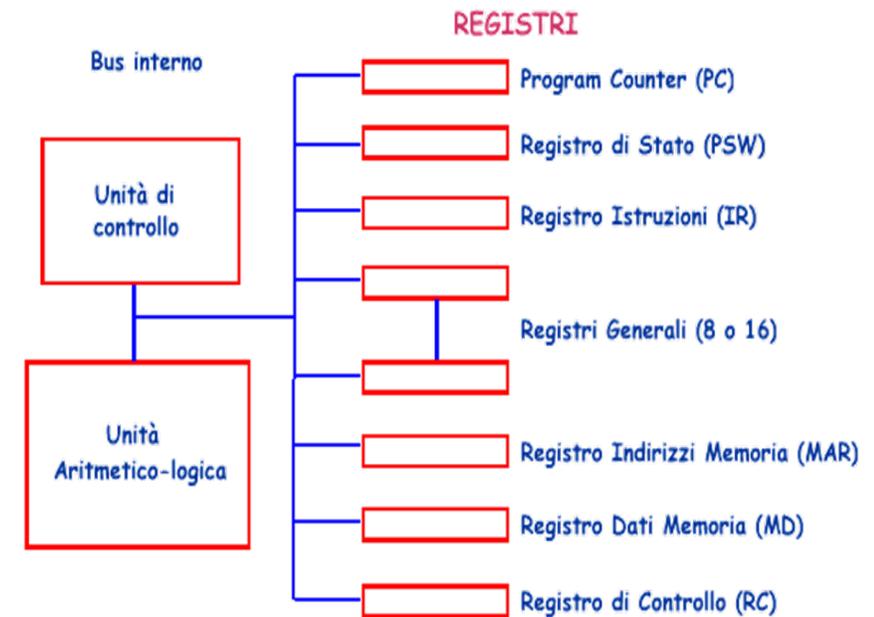
Figura tratta da [https://www.microst.it/Tutorial/funzioni\\_logiche\\_5.html](https://www.microst.it/Tutorial/funzioni_logiche_5.html)



# La CPU – Registri

I Registri sono delle piccole unità di memoria locale che servono a mantenere fisicamente vicini alla ALU le informazioni che in quel momento sono essenziali per eseguire la elaborazione richiesta (i.e. metto gli operandi nei registri prima di sommarli). Il valore di tutti i registri ad un dato istante, rappresentano lo «**stato**» della CPU in quell'istante.

- **program counter (PC)**: memorizza l'indirizzo della prossima istruzione;
- **instruction register (IR)**: contiene l'istruzione da eseguire;
- **memory address register (MAR)**: contiene l'indirizzo della locazione di memoria da leggere o scrivere;
- **memory data register (MDR o MD)**: serve a scambiare dati tra la CPU e la memoria;
- **program status word (PSW)**: è il registro dei flag che memorizza particolari condizioni (riporto, overflow, risultato +/-, controllo parità);
- **control register (RC)** è il registro per i segnali di controllo (ready, wait, write, read,...).



# La CPU – Control Unit

L'ultima componente della CPU che funge da vero e proprio «cervello» del nostro computer è la Control Unit.

Cosa fa la Control Unit?

Essa svolge la delicata operazione di «interpretare» l'istruzione ricevuta e di impartire i conseguenti «ordini» a tutti gli organi coinvolti: ALU, Registri, Unità di Uscita etc.. affinché l'istruzione sia di fatto eseguita.

L'unità di controllo ad ogni clock riceve in Input:

1. il codice operativo dell'istruzione da eseguire;
2. i segnali di flag che determinano lo stato della CPU;
3. i segnali di controllo (interrupts, acknowledgments)

E produce in Output:

1. Segnali di controllo verso ALU, di trasferimento dati, di controllo verso altri dispositivi

# La CPU – Instruction set

- Come nelle lingue naturali, composte da fonemi, parole e frasi e così via, anche nel linguaggio dei computer, il **linguaggio macchina**, i simboli dell'alfabeto utilizzato (1 e 0) sono organizzati in «**parole**» e queste ultime possono essere utilizzate per comporre «**frasi**» secondo delle ben precise regole grammaticali.
  - Le frasi del linguaggio macchina sono dette «istruzioni»
  - Ogni istruzione ordina al processore di eseguire un'azione elementare che va a modificare lo stato interno del computer:
    - come la lettura di una locazione di memoria
    - l'esecuzione della somma dei valori contenuti in due registri
    - l'incremento di un contatore
    - Il salto a una successiva istruzione
- .....

# La CPU - Instruction Set

- Le **istruzioni** sono generalmente costituite da una parola iniziale detta **opcode** (**operation code**), che specifica l'operazione da eseguire, seguita da altre parole che specificano gli eventuali **operandi** (o dati) a cui l'azione deve essere applicata (per esempio «somma 10 e 5»).
- Il formato e la sintassi delle operazioni sono proprie di ciascun processore e sono contenute nel **Instruction Set** di quel dato processore, appunto.
- Per completezza diciamo che, mentre i processori di un tempo (scalari) potevano eseguire una sola operazione per volta, quelli attuali sono in grado di eseguire più operazioni simultanee sullo stesso processore (i.e. nei processori superscalari per il calcolo parallelo) o più operazioni su più processori integrati in un unico chip (multi core).

# Come funziona la CPU?

La CPU coordina tutte le operazioni svolte dal computer; queste possono essere:

## **Funzioni interne alla CPU, come:**

- Interpretare le istruzioni del programma
- Effettuare operazioni logiche/aritmetiche
- Generare i segnali di controllo per i componenti interni
- Verificare la correttezza dei dati
- Sincronizzare le funzioni del sistema

## **Funzioni esterne alla CPU**

- Operare sulla main memory (lettura e scrittura)
- Attivare componenti esterni
- Ricevere segnali da componenti esterni

Vediamo ora cosa fa una CPU quando esegue una istruzione (ciclo macchina).

# Il ciclo macchina

Il funzionamento della CPU si basa sul seguente ciclo di interpretazione del linguaggio macchina:

**FETCH dell'istruzione:** il processore preleva l'istruzione dalla memoria centrale, all'indirizzo specificato dal **Program Counter** (PC). Si incrementa il PC per predisporre per la successiva istruzione.

**DECODE dell'istruzione:** una volta che l'istruzione è stata prelevata, la Control Unit determina di quale istruzione si tratti, quale operazione debba essere eseguita, come ottenere gli operandi.

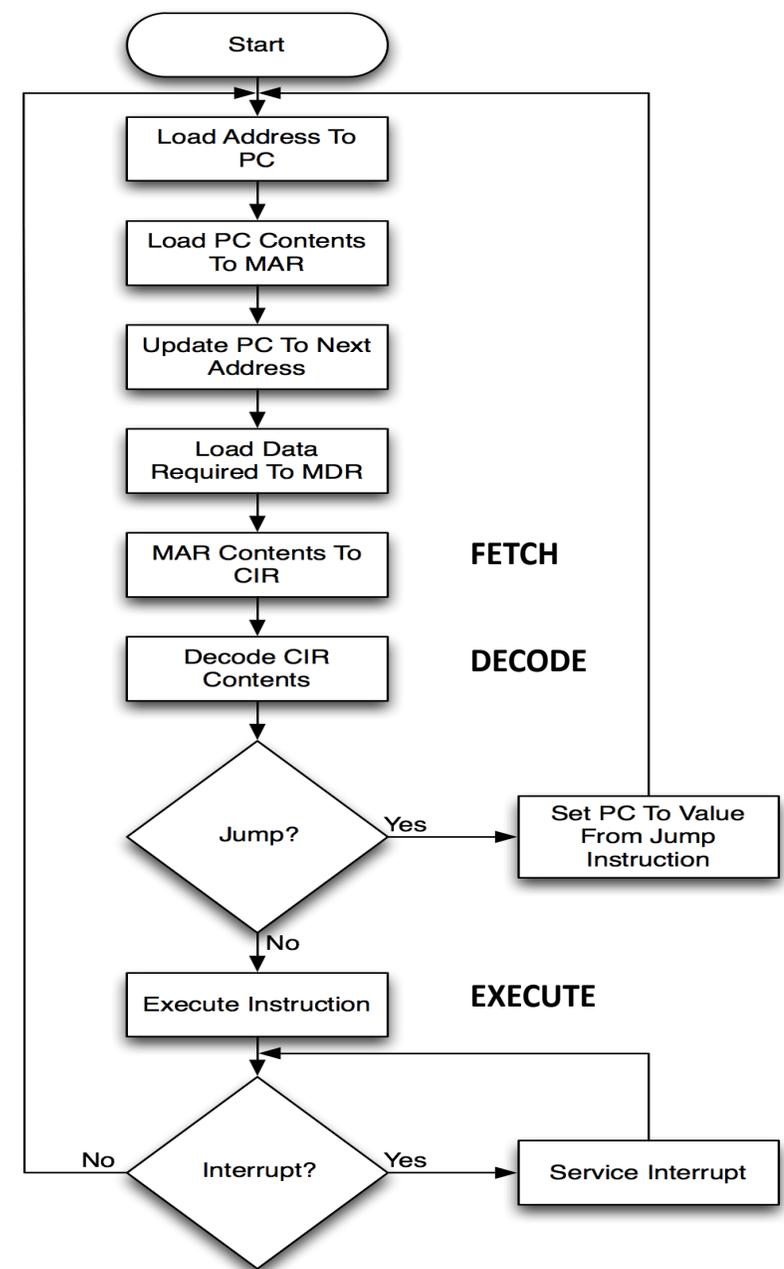
**EXECUTE dell'istruzione:** l'operazione viene eseguita. La Control Unit genera i segnali dovuti. Se l'istruzione è un salto condizionale viene aggiornato il PC il cui valore dipende pertanto dall'esito dell'istruzione.

**WRITEBACK:** si ritorna all'inizio del ciclo e ci si dispone per la successiva istruzione.

Il numero di cicli che il processore è in grado di svolgere in un secondo, ne determina la velocità.

# Esempio di ciclo macchina

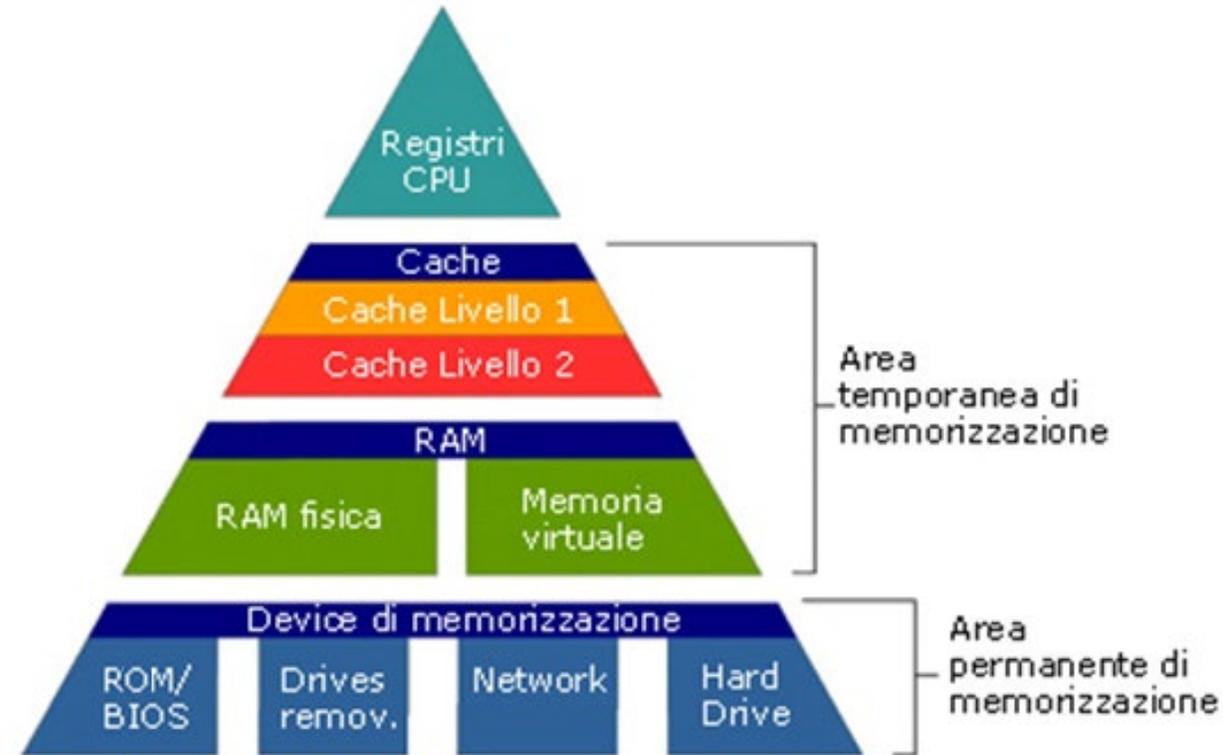
1. Si acquisisce l'indirizzo corrente dalla memoria - in base all'indirizzo del Program Counter (PC) - e lo si inserisce nel Registro degli Indirizzi di Memoria (MAR)
2. Si incrementa il PC per predisporre per la successiva istruzione
3. Se serve prelevare un dato dalla memoria questo viene caricato nell'apposito registro MDR. Questa eventuale operazione (sulla memoria) viene eseguita al ciclo successivo
4. Il contenuto di MAR va nel Registro delle Istruzioni (IR)
5. La Control Unit decodifica il tipo di istruzione contenuta in IR
6. Se l'istruzione è un «salto» si aggiorna PC e si torna allo Start
7. Altrimenti viene eseguita l'istruzione: la Control Unit genera una serie di segnali che realizzano l'istruzione
8. Si verifica l'esistenza di Interrupt ed eventualmente si cede il controllo
9. Si torna allo Start (o si finisce) completando il «**ciclo macchina**».



# La Memoria del Computer

Per poter lavorare la CPU ha sempre la necessità di interagire con la Memoria. Queste due componenti lavorano in simbiosi per un motivo molto semplice:

- Sia i dati, sia i programmi risiedono nella Memoria centrale (architettura Von Neumann)
- Principale compito della CPU è quello di eseguire le istruzioni di un programma presente in memoria centrale leggendo e scrivendo i dati, che pure risiedono in memoria centrale.



# Oltre la CPU

Come abbiamo visto, la CPU esegue una ad una le istruzioni affidatele dal programma utilizzando i dati che le sono stati forniti in ingresso (tramite la Tastiera, il mouse, il microfono, il disco fisso etc..) e che ora risiedono in memoria centrale. L'esecuzione delle istruzione comporta la generazione di risultati (che vanno al monitor, verso i dispositivi audio, le memorie esterne etc.).

Per esempio, quando scriviamo una lettera con un programma di videoscrittura possiamo controllare a **Monitor** il risultato di ciò che abbiamo scritto e, una volta che abbiamo terminato, archiviare definitivamente il risultato (la lettera in questo caso) in una memoria di massa - in un **File** - e poi se vogliamo anche inviarlo a una **Stampante**.

Tutte le operazioni elementari eseguite dalla CPU, che sottintendono quelle di più alto livello appena citate, non riusciamo a percepirle perché operiamo nei confronti del computer mediante una **serie di comandi** che rendono trasparente sia l'architettura del computer stesso, sia le modalità di esecuzione dei programmi.

# Il Sistema Operativo

# Il Sistema Operativo

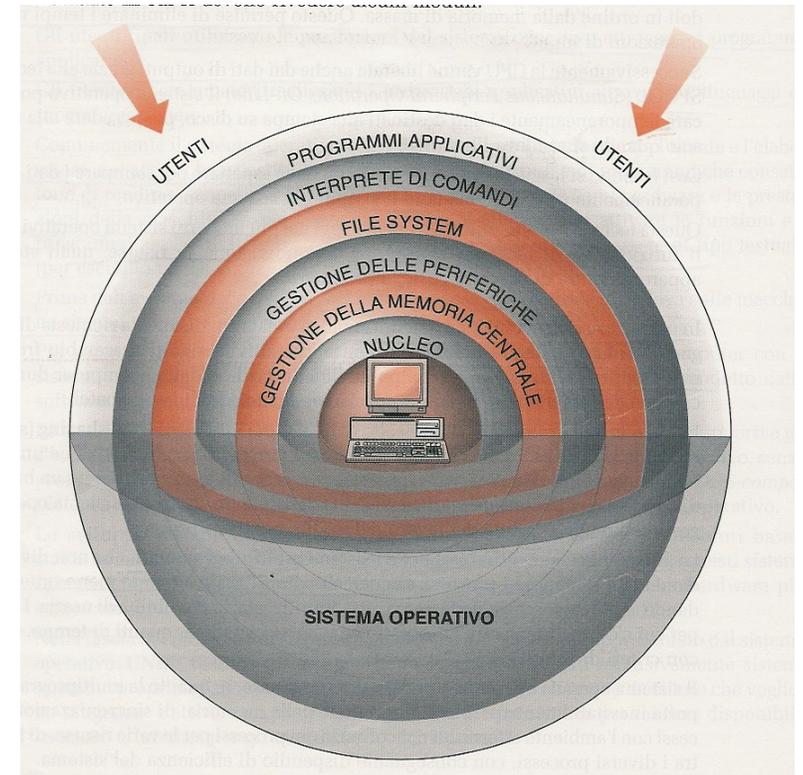
Cos'è dunque che rende trasparente all'utilizzatore tutte queste operazioni?

Cosa fa in modo che io possa disinteressarmi di come archiviare un file o come stampare un documento o come gestire l'attivazione di un dispositivo con priorità real time?

Si tratta del

«**SISTEMA OPERATIVO**»

una componente fondamentale del computer che però questa volta non è di tipo Hardware, ma di tipo Software.



# Il Sistema Operativo

Analogamente al cervello umano che col sistema nervoso governa il funzionamento delle attività del corpo, il **Sistema Operativo** supervisiona le attività della macchina in cui si trova e di tutte le sue componenti: la memoria di massa, la memoria centrale, il processore, le schede accessorie, il lettore cd, la stampante, il mouse, la tastiera etc.

Il Sistema Operativo permette all'utente di interfacciarsi facilmente al sistema hardware, evitandogli una interazione diretta che sarebbe alquanto difficoltosa anche per un tecnico.

# Il Sistema Operativo

Poiché un Sistema Operativo deve essere in grado di gestire, trattare e comunicare con la macchina in maniera trasparente ed indipendente da essa, è intuibile che più il software sarà realizzato in maniera coesa con l'hardware, maggiore sarà la sua specializzazione e maggiore sarà la sua efficienza.



# Il Sistema Operativo

- Un Sistema Operativo ha come fondamentale compito quello di coordinare, sincronizzare, gestire ed eseguire dei **processi** ossia delle sequenze di attività (**task**) che il processore deve portare a termine su richiesta dell'utente.
- Per realizzare questi task il Sistema Operativo ha bisogno di poter utilizzare delle **risorse**: sono risorse, per esempio, il processore stesso, le diverse memorie, centrali o di massa, il monitor, la tastiera e così via. Anche una operazione semplicissima come quella di premere un tasto della tastiera visualizzando il carattere corrispondente sul monitor, comporta l'esecuzione di diversi processi e l'allocazione di svariate risorse.

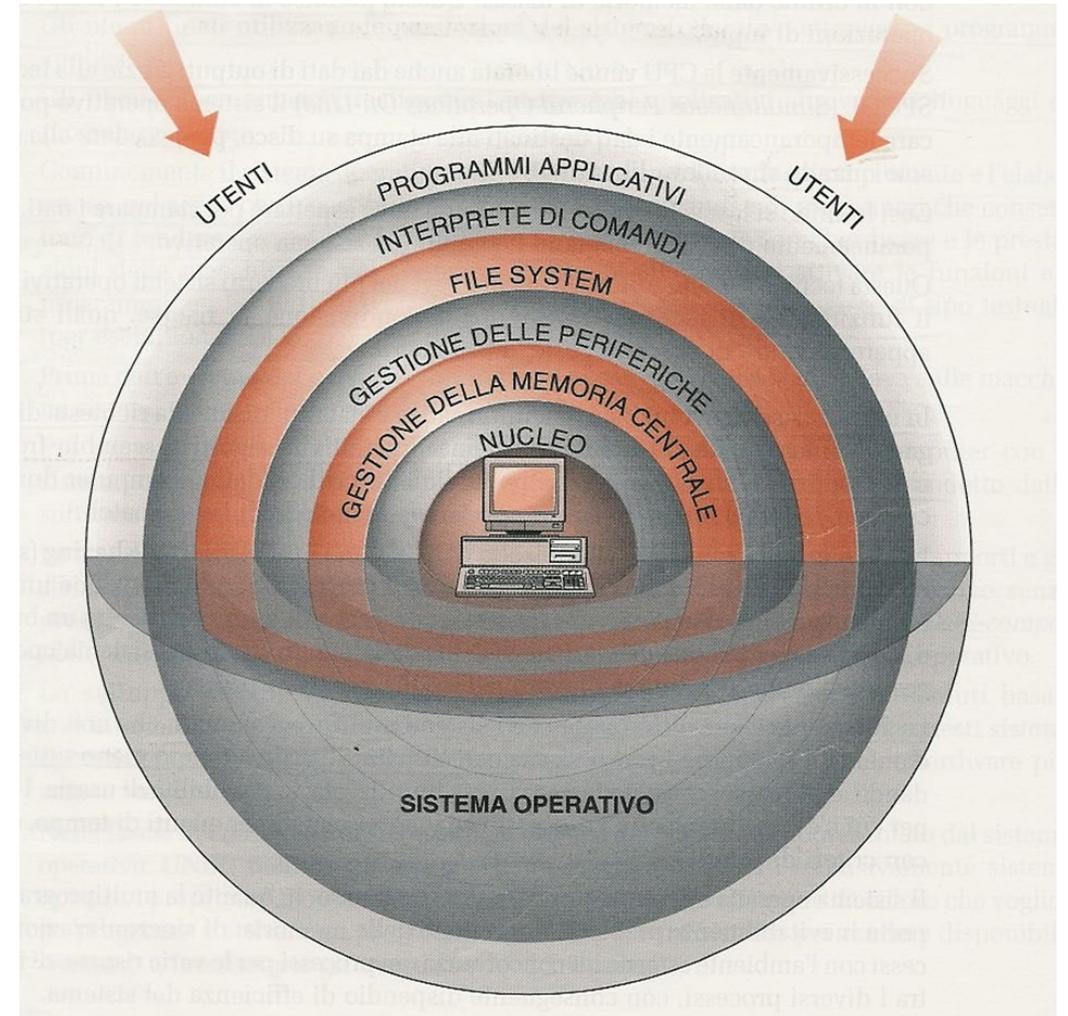


# Alcuni compiti del Sistema Operativo

# La gestione del Hardware

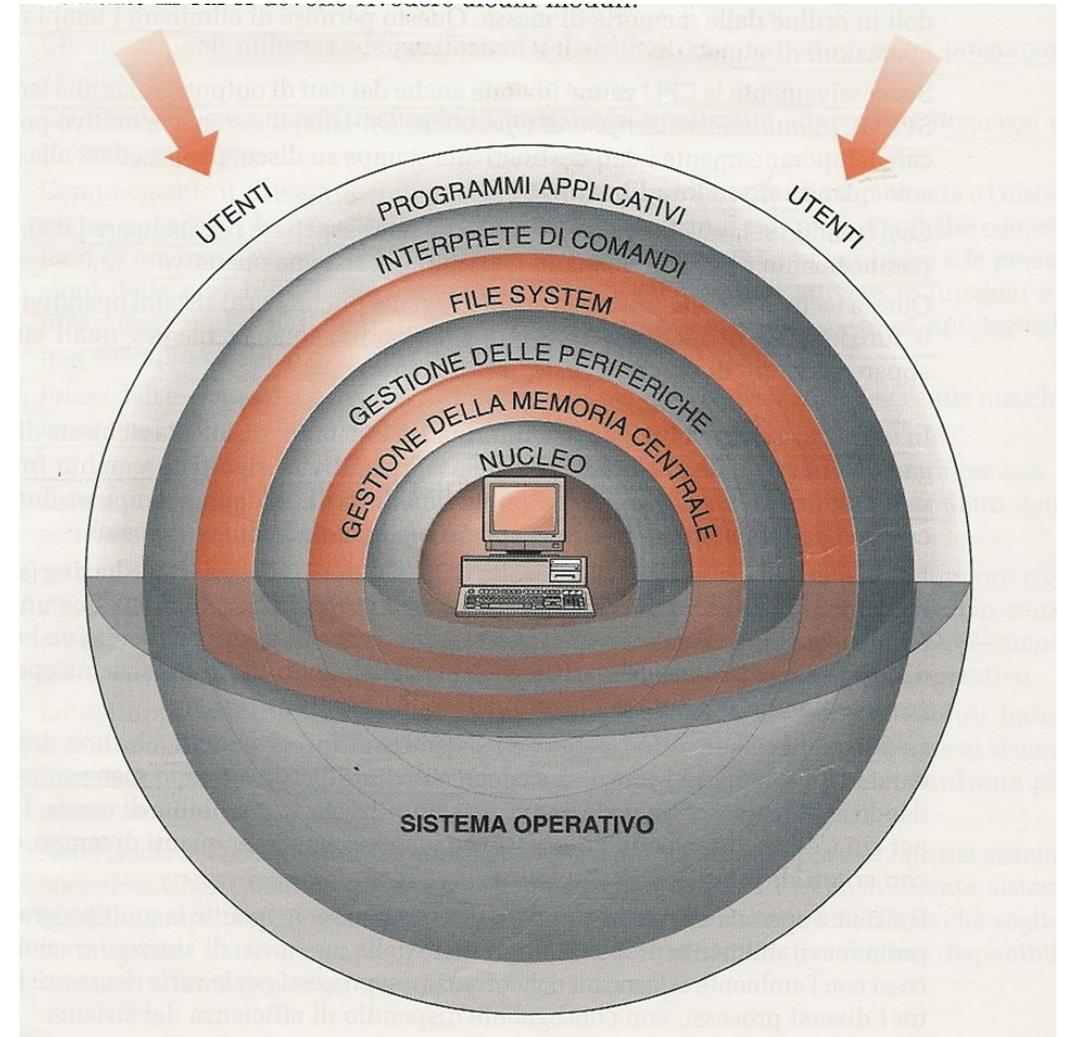
Insieme coi Driver delle periferiche e col Firmware, il **Kernel** fornisce il livello più elementare di controllo su tutti i dispositivi hardware del computer. Alcune delle funzioni del Kernel sono descritte nel seguito:

1. L'esecuzione di un programma applicativo attraverso la creazione di un processo in cui il Kernel:
  - assegna al programma lo spazio di memoria e le risorse richieste
  - stabilisce la priorità del processo rispetto ad altri processi nei sistemi multi-tasking (in grado di eseguire più programmi contemporaneamente)
  - carica il codice binario del programma nella memoria
  - avvia l'esecuzione del programma applicativo
  - interagisce con l'utente e coi dispositivi hardware.



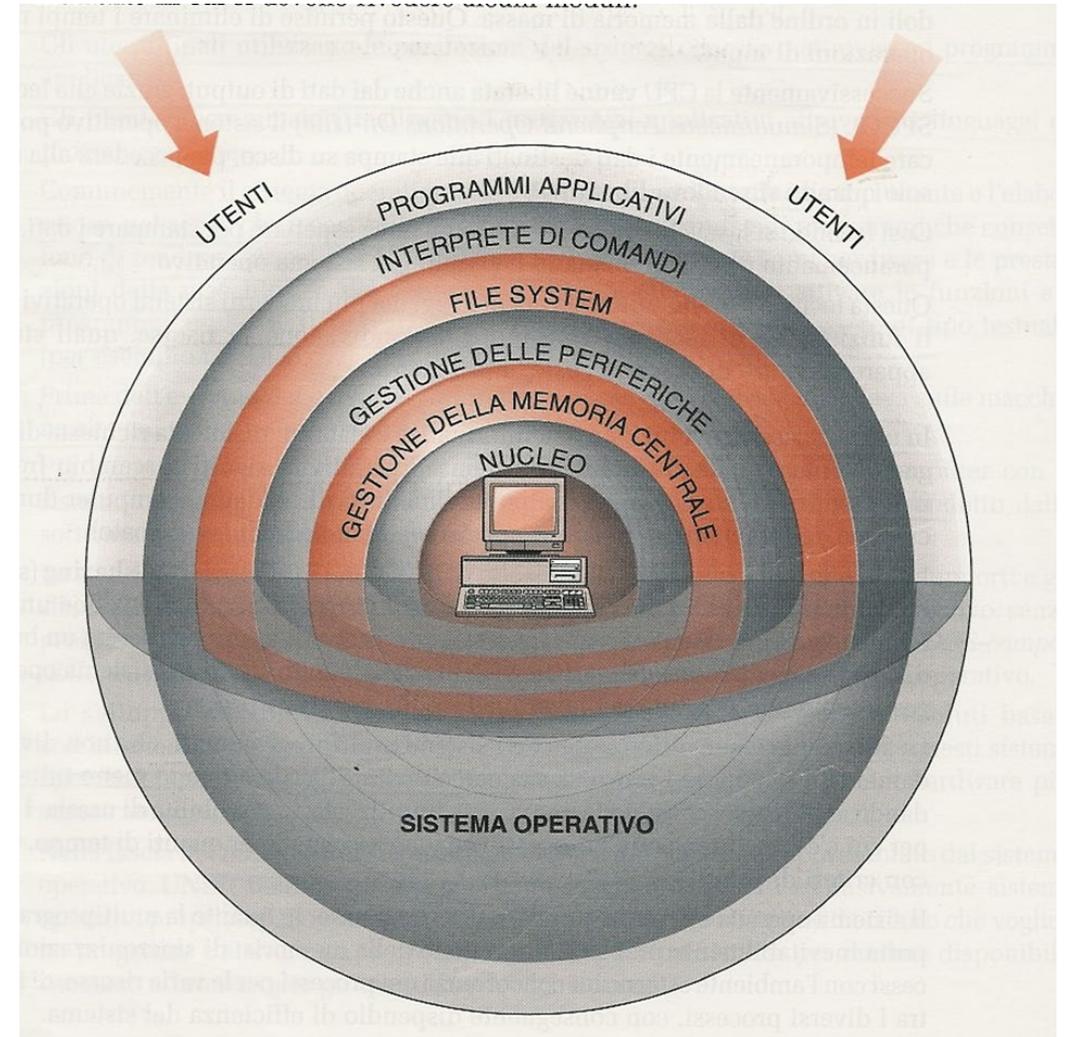
# La gestione del Hardware

2. Il Kernel del sistema operativo è responsabile della gestione di tutta la memoria di sistema utilizzata dai programmi. Questo assicura che un programma non interferisca con la memoria già in uso da un altro programma. Poiché il tempo viene condiviso dai programmi, ognuno di essi deve avere un accesso indipendente alla memoria.
3. Il Kernel può inoltre ottimizzare la **gestione delle memorie** attraverso delle tecniche di virtualizzazione; in tal modo esso può decidere quale memoria concedere ad un dato programma e può migliorarne l'impiego attribuendo delle porzioni condivise di memoria per l'esecuzione di più compiti.



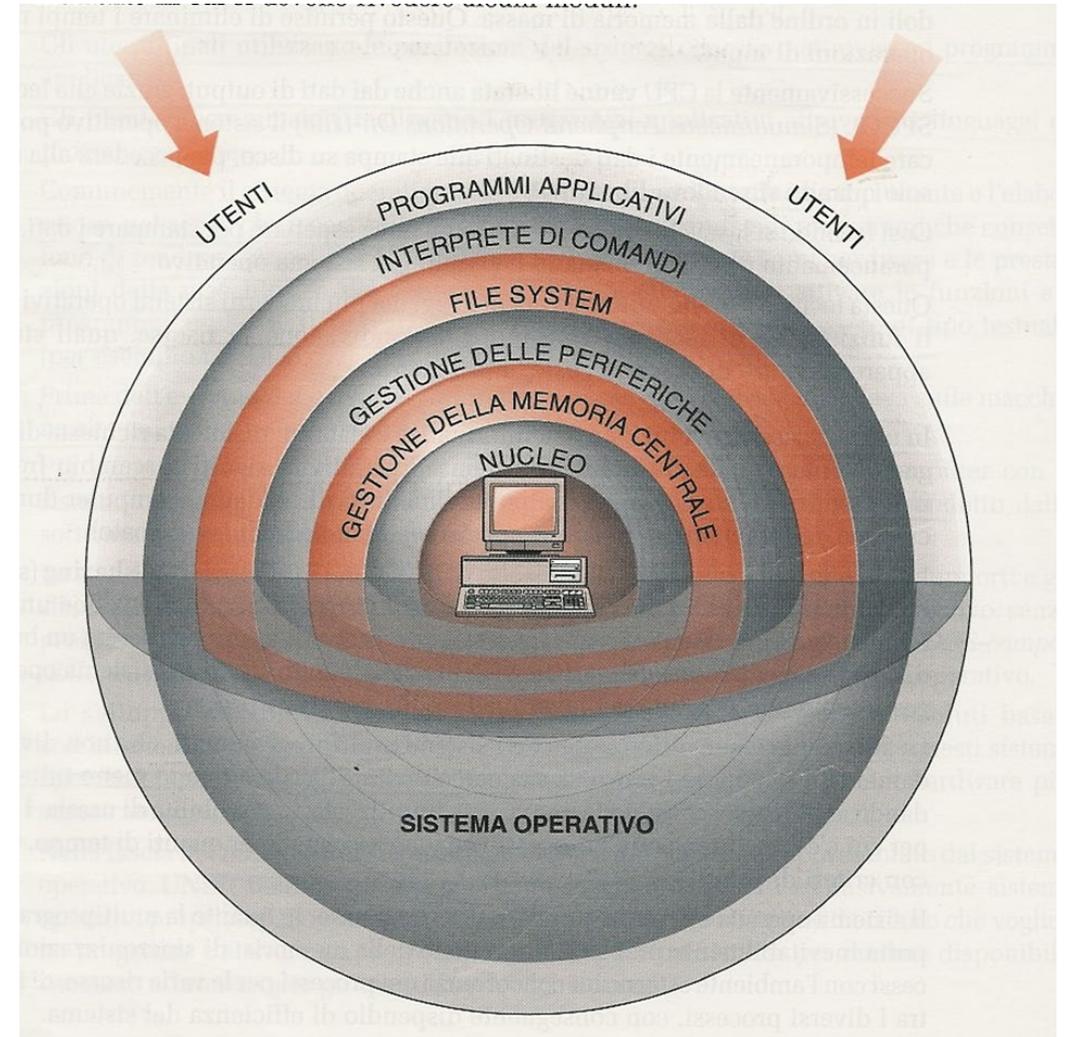
# La gestione dei Processi

- All'interno del Kernel può anche essere effettuata la gestione dei processi. Questa viene realizzata dallo **Scheduler**, un programma che scandisce il tempo di esecuzione dei vari processi e assicura che ciascuno di essi venga eseguito per il tempo richiesto.
- Normalmente lo scheduler gestisce anche lo stato dei processi e può sospendere l'esecuzione nel caso questi siano in attesa senza fare nulla, assegnando le risorse inutilizzate ad altri processi che ne hanno bisogno (esempio classico è la richiesta di dati da disco).
- Nei sistemi operativi realtime lo scheduler si occupa anche di garantire una *timeline*, cioè un tempo massimo di completamento per ciascun task in esecuzione, ed è notevolmente più complesso.



# La gestione della memoria centrale

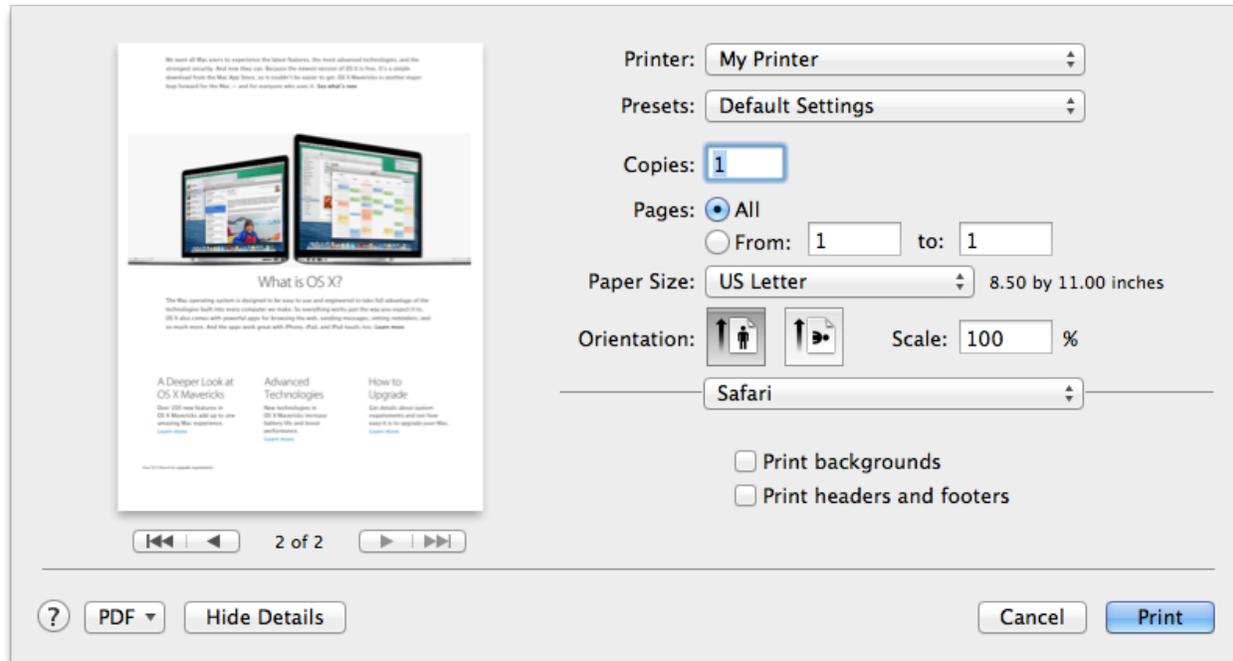
- A livello hw ogni sistema è equipaggiato con un unico spazio di memoria accessibile direttamente da CPU e dispositivi
- Nella gestione generale della Memoria Centrale i compiti del Sistema Operativo sono, in linea di massima:
  - allocare uno spazio di memoria a ciascun processo
  - deallocare la memoria di un processo concluso
  - proteggere il sistema separando gli spazi di indirizzi associati ai diversi processi in modo che non interferiscano ne tra loro ne col supervisore
  - realizzare i collegamenti tra gli indirizzi logici specificati dai processi e le corrispondenti locazioni nella memoria fisica che possono essere anche non contigue
  - gestire la memoria virtuale, ossia gli spazi di indirizzi logici di dimensioni superiori allo spazio fisico, da allocare sulla memoria di massa



Nome	Stato	1% CPU	24% Memoria	2% Disco	0% Rete	1% GPU	Motore GPU	Consumo elett...	Tendenza cons...
<b>Applicazioni (2)</b>									
> Gestione attività		0,2%	27,1 MB	0 MB/s	0 Mbps	0%		Molto basso	
> Microsoft PowerPoint		0%	119,4 MB	0 MB/s	0 Mbps	0%		Molto basso	
<b>Processi in background (97)</b>									
ACMON (32 bit)		0%	0,8 MB	0 MB/s	0 Mbps	0%		Molto basso	
> Adobe Acrobat Update Service ...		0%	0,7 MB	0 MB/s	0 Mbps	0%		Molto basso	
> Adobe Genuine Software Integri...		0%	2,2 MB	0 MB/s	0 Mbps	0%		Molto basso	
> Adobe Genuine Software Servic...		0%	1,4 MB	0 MB/s	0 Mbps	0%		Molto basso	
Adobe IPC Broker (32 bit)		0%	1,4 MB	0 MB/s	0 Mbps	0%		Molto basso	
> Adobe Update Service (32 bit)		0%	0,9 MB	0 MB/s	0 Mbps	0%		Molto basso	
> Antimalware Service Executable		0%	130,6 MB	0 MB/s	0 Mbps	0%		Molto basso	
AnyDesk (32 bit)		0%	13,8 MB	0 MB/s	0 Mbps	0%		Molto basso	
> AnyDesk (32 bit)		0%	14,2 MB	0 MB/s	0 Mbps	0%		Molto basso	
Application Frame Host		0%	11,0 MB	0 MB/s	0 Mbps	0%		Molto basso	
> Applicazione sottosistema spoo...		0%	4,9 MB	0 MB/s	0 Mbps	0%		Molto basso	
> ASLDR Service (32 bit)		0%	0,8 MB	0 MB/s	0 Mbps	0%		Molto basso	
ASUS GIFTBOX (32 bit)		0%	44,9 MB	0 MB/s	0 Mbps	0%		Molto basso	
ASUS GIFTBOX (32 bit)		0%	0,4 MB	0 MB/s	0 Mbps	0%		Molto basso	
ASUS GIFTBOX (32 bit)		0%	6,9 MB	0 MB/s	0 Mbps	0%		Molto basso	
ASUS GIFTBOX (32 bit)		0%	9,2 MB	0 MB/s	0 Mbps	0%		Molto basso	
ASUS Patch For Touch Panel		0%	0,4 MB	0 MB/s	0 Mbps	0%		Molto basso	
ATK Media (32 bit)		0%	0,9 MB	0 MB/s	0 Mbps	0%		Molto basso	

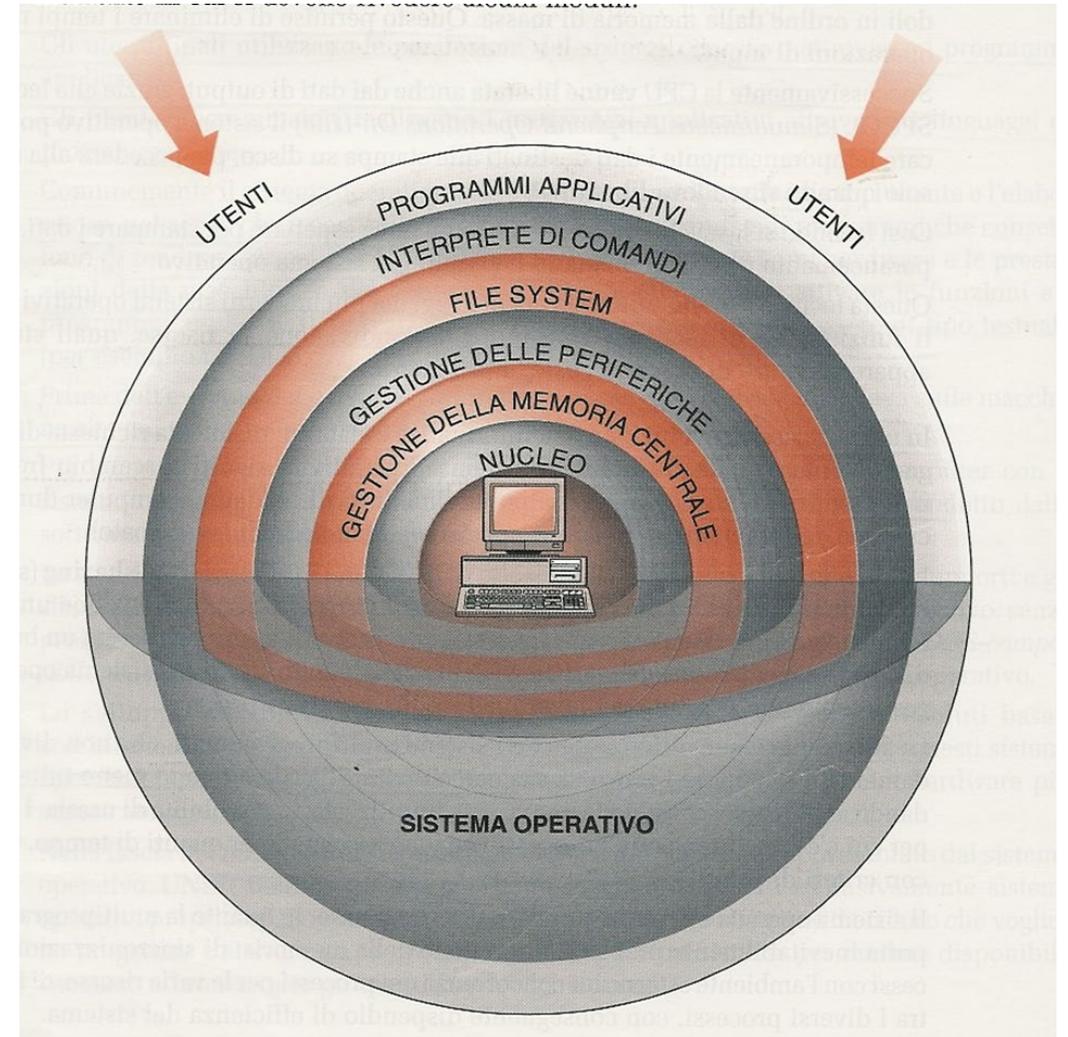
# La gestione delle Periferiche

- Altro compito fondamentale del SO è la gestione delle periferiche (stampante, mouse, tastiera, monitor etc.). Per realizzare questo compito il SO spesso utilizza del software aggiuntivo che ne diviene parte integrante e che dipende dallo specifico hardware di quel dispositivo. Questo software aggiuntivo è chiamato **Driver**.
- Talvolta, nella gestione della stampante, viene impiegato un cosiddetto **Spooler** di stampa. Questo è un componente del SO che riceve in ingresso i file da stampare dai programmi utilizzati dall'utente e li accoda mandandoli in stampa uno alla volta verso la stampante.
- Questo permette ai programmatori di applicativi di non doversi affatto preoccupare della gestione delle code di stampa e di eventuali conflitti sulla stampante che dovessero verificarsi per via di altri processi concomitanti in quanto lo spooler accetterà qualsiasi documento e farà sembrare la stampante sempre disponibile.



# La Gestione di File e Cartelle

- Tra i diversi compiti del Sistema Operativo c'è anche quello fondamentale di gestire file e cartelle: il cosiddetto **File System**.
- In questo caso il SO si occupa di operazioni quali: archiviare e proteggere le risorse, renderle disponibili agli utenti di una rete, gestire le modifiche a tali risorse.
- Questi strumenti includono per esempio la gestione delle cartelle condivise, la crittografia dei dati e così via. Quando una cartella è condivisa, il SO si preoccupa che gli utenti che vi accedono tramite la rete siano autorizzati e che più utenti non effettuino modifiche contemporanee.
- Il File System si occupa anche di come memorizzare i file all'interno del disco fisso, di come indicizzarne le varie parti e così via.



File Home Condividi Visualizza

Questo PC > Desktop > DISEGNI DEFINITIVI

Nome	Ultima modifica	Tipo	Dimensione
DESTINAZIONE NUOVA COSTRUZIONE 1...	25/01/2016 15:24	File PDF	4 KB
ELABORATO COMPARATIVO 1-200	25/01/2016 15:15	File PDF	52 KB
MODELLO ISTAT	25/01/2016 16:19	File PDF	37 KB
<b>ORIENTAMENTO FOTO</b>	24/01/2016 17:38	File PDF	39 KB
PLANIMETRIA ABITAZIONE.bak	24/01/2016 17:18	File BAK	54 KB
PLANIMETRIA ABITAZIONE	25/01/2016 15:42	AutoCAD Drawing	54 KB
planimetria di inquadramento 1-200 CO...	24/01/2016 17:38	File BAK	91 KB
planimetria di inquadramento 1-200 CO...	25/01/2016 15:16	AutoCAD Drawing	91 KB
PLANIMETRIA I PIANO 1-100	25/01/2016 15:38		
PLANIMETRIA PIANO TERRA 1-100	25/01/2016 15:40		
plot	25/01/2016 15:40		
STATO DI FATTO 1-200	24/01/2016 16:02		
STATO DI PROGETTO 1-200	24/01/2016 16:04		

Proprietà - ORIENTAMENTO FOTO

Generale Sicurezza Dettagli Versioni precedenti

ORIENTAMENTO FOTO

Tipo di file: File PDF (.pdf)

Apri con: Microsoft Edge [Cambia...](#)

Percorso: C:\Users\VincenzoG\Desktop\DISEGNI DE

Dimensioni: 38.0 KB (39.010 byte)

Dimensioni su disco: 40.0 KB (40.950 byte)

Data creazione: domenica 24 gennaio 2016, 17:38:08

Ultima modifica: domenica 24 gennaio 2016, 17:38:09

Ultimo accesso: domenica 24 gennaio 2016, 17:38:09

Attributi:  Solo lettura  Nascosto [Avanzate...](#)

OK Annulla Applica

# L'interprete di Comandi

La **Shell** (Interfaccia Utente) permette di interagire con la macchina attraverso un **interprete di comandi**.



OS X Mountain Lion

Windows 10



# L'interprete di Comandi

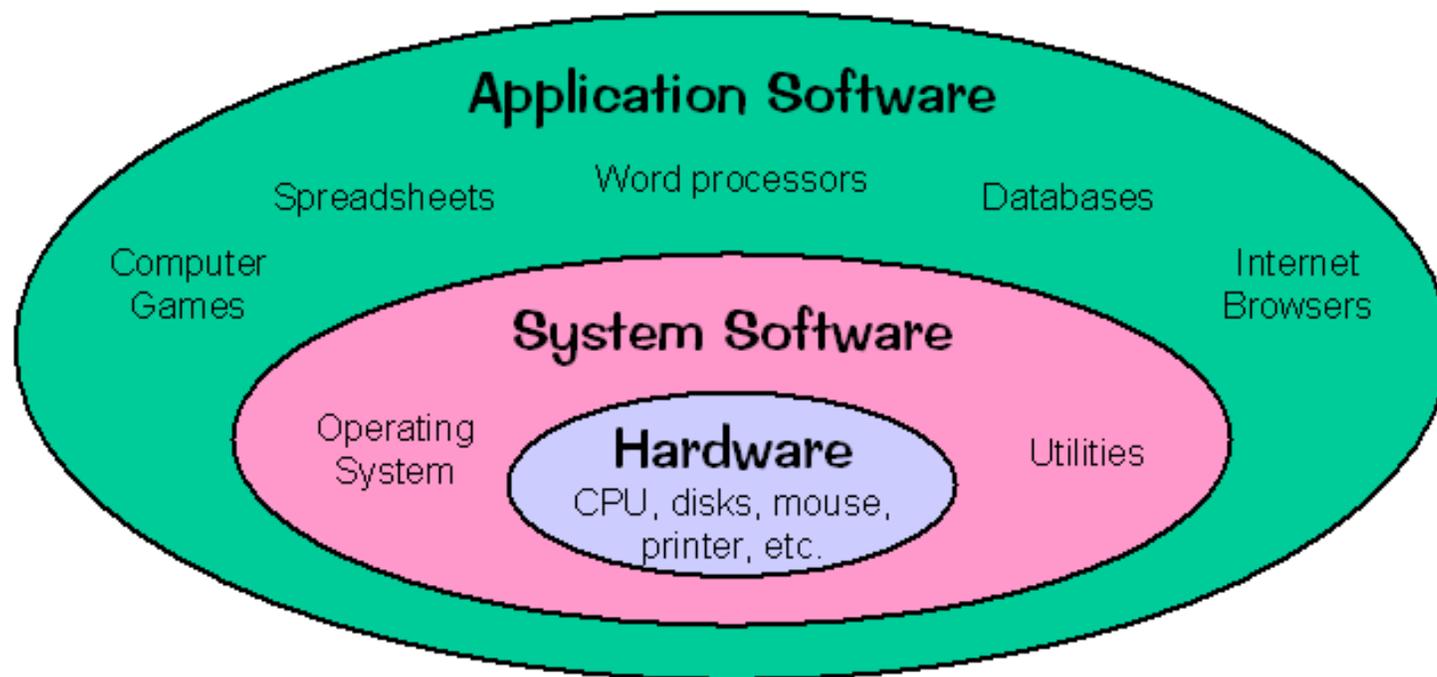
- In generale la **Shell** del sistema operativo può utilizzare una interfaccia di tipo testuale, a riga di comando, ovvero una interfaccia grafica (GUI= Graphical User Interface).
- È detta Shell, ossia guscio, perché rappresenta lo strato «visibile» intorno al sistema operativo.
- La progettazione di una Shell è condizionata soprattutto dal tipo di periferiche che saranno utilizzate col computer (tastiera, mouse, touchscreen etc.) che sono l'interfaccia diretta tra uomo e macchina.

```
Prompt dei comandi
Microsoft Windows [Versione 10.0.14393]
(c) 2016 Microsoft Corporation. Tutti i diritti sono riservati.

C:\Users\VincenzoG>dir
Il volume nell'unità C non ha etichetta.
Numero di serie del volume: CCA9-42C6

Directory di C:\Users\VincenzoG
13/03/2017  23:54    <DIR>          .
13/03/2017  23:54    <DIR>          ..
08/01/2016  17:20    <DIR>          .ITtrading
25/05/2016  18:59    <DIR>          .matplotlib
09/08/2016  10:14    <DIR>          .oracle_jre_usage
25/05/2016  18:58    <DIR>          .qgis2
16/01/2017  15:26    <DIR>          Contacts
14/03/2017  07:26    <DIR>          Creative Cloud Files
15/03/2017  09:18    <DIR>          Desktop
19/12/2016  16:30    <DIR>          dikeliteTmpdir
03/03/2017  17:20    <DIR>          Documents
15/03/2017  09:17    <DIR>          Downloads
23/09/2016  16:38    <DIR>          Dropbox
08/01/2016  15:58    <DIR>          EfiRedattore
16/01/2017  15:26    <DIR>          Favorites
16/01/2017  15:26    <DIR>          Links
16/01/2017  15:26    <DIR>          Music
15/03/2017  08:55    <DIR>          OneDrive
16/01/2017  15:26    <DIR>          Pictures
14/11/2016  09:04                1.068 pixinsight-license
14/03/2017  16:38                74 pslog.txt
```

Dal software  
di sistema al  
software  
applicativo



# Software applicativo

- Un programma applicativo (o applicazione o anche app) è un programma per computer progettato per eseguire un gruppo di funzioni coordinate o attività a beneficio degli utenti. Esempi di applicazioni sono: un word processor, un foglio di calcolo, un programma di contabilità, un browser web, un lettore multimediale, un simulatore di volo aeronautico, un programma di grafica, un programma di elaborazione video, un programma di calcolo strutturale, un programma GIS etc.

## **iMiBACT MUSEUM**

- Contiene schede dettagliate e informazioni su più di 500 musei statali italiani, percorsi storico-artistici, consigli utili un filtro di ricerca che permette di trovare cosa c'è da vedere intorno a sé.



# Software applicativo

- A differenza del sistema operativo e delle sue componenti (la maggior parte delle quali si trovano preinstallate al momento dell'acquisto), una applicazione non è considerata necessaria al funzionamento del computer per cui, genericamente, si può parlare di:
  - software di base, ossia tutte le istruzioni fondamentali di cui un computer ha bisogno per funzionare e per far funzionare il software applicativo
  - software applicativo, che fornisce all'utente finale determinate funzionalità specialistiche e di utilità, nei più svariati settori: scientifico, ufficio, business, gioco etc.
- Il software applicativo può essere sviluppato in maniera dedicata per un dato sistema operativo, oppure essere multiplatforma se contiene delle estensioni che ne consentono l'utilizzazione su più sistemi operativi.

# Software Open Source

- Si parla di software open-source (OSS) quando il codice sorgente viene reso disponibile con una licenza in cui il titolare del copyright prevede il diritto di studiare, modificare e distribuire il software a chiunque e per qualsiasi scopo.  
Il software open-source può essere sviluppato in maniera aperta e collaborativa da parte di community che si costituiscono allo scopo.

Lo sviluppo di software open-source, o sviluppo collaborativo da più fonti indipendenti, genera un ambito di progetto sempre più diversificato e in grado di essere sostenuto nel lungo termine. Un rapporto - datato eppure significativo - del 2008 dal Gruppo Standish, affermava che l'adozione di modelli di software open-source aveva portato a un risparmio di circa 60 miliardi di dollari all'anno per i consumatori.

# Software Open Source

- **Fedora** è un Sistema operativo basato su Kernel Linux per PC.
- **Blender** è una 3D creation suite che supporta una grande varietà di operazioni di elaborazione grafica: modeling, animation, rendering, motion tracking, video editing e altro.
- **MySQL** è un potente DBMS relazionale utilizzato anche da alcuni dei content management system di maggior successo come WordPress, Joomla, Drupal.
- **Apache** HTTP Server, o più comunemente Apache, è la piattaforma server Web modulare più diffusa, in grado di operare su una grande varietà di sistemi operativi, tra cui UNIX/Linux, Microsoft Windows e OpenVMS.
- **WordPress** è una piattaforma software di «Personal Publishing» e di Content Management System (CMS) per la creazione e distribuzione di siti Internet.



# Linguaggi di Programmazione

- Tutta l'informatica ruota intorno all'attività della «programmazione» nella quale le idee e le esigenze vengono tradotte in algoritmi e quindi in programmi da implementare nei computer.
- Per realizzare un programma serve un linguaggio di programmazione.
- La compilazione è il processo per cui il programma, scritto in un linguaggio di programmazione ad alto livello (codice sorgente), viene tradotto in un codice eseguibile per mezzo di un altro programma detto appunto compilatore.
- L'interpretazione, è invece una sorta di traduzione passo passo di ciascuna istruzione del programma, in codice eseguibile .



```
REM -----  
REM elevamento a potenza X elevato alla Y  
REM X e Y sono due numeri naturali.  
REM -----  
REM  
INPUT "Inserisci la base "; X  
INPUT "Inserisci l'esponente "; Y  
LET Z = 1  
  FOR I = 1 TO Y  
    LET Z = Z * X  
  NEXT  
PRINT X; "^"; Y; "="; Z  
END  
REM -----FINE PROGRAMMA-----
```

# Prossimo Capitolo – La codifica dell'Informazione Numerica

Nel prossimo capitolo vedremo come viene codificata l'informazione attraverso i soli due simboli a disposizione nel mondo digitale e come sia possibile rappresentare un numero o un'immagine all'interno di un computer.