

Fondamenti di Informatica

Accademia di Belle Arti di Verona

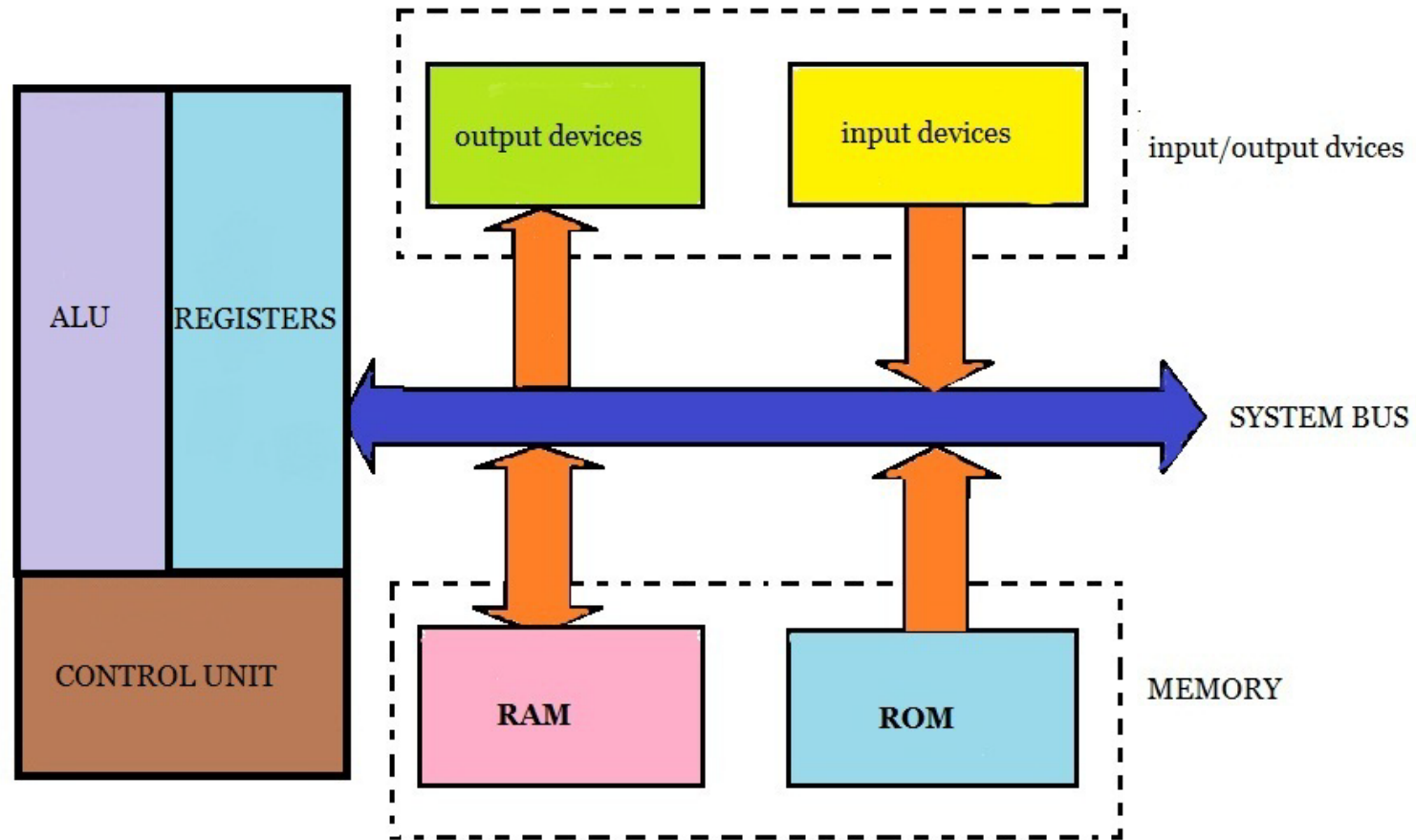
Università degli Studi di Verona

A.A. 2018-2019

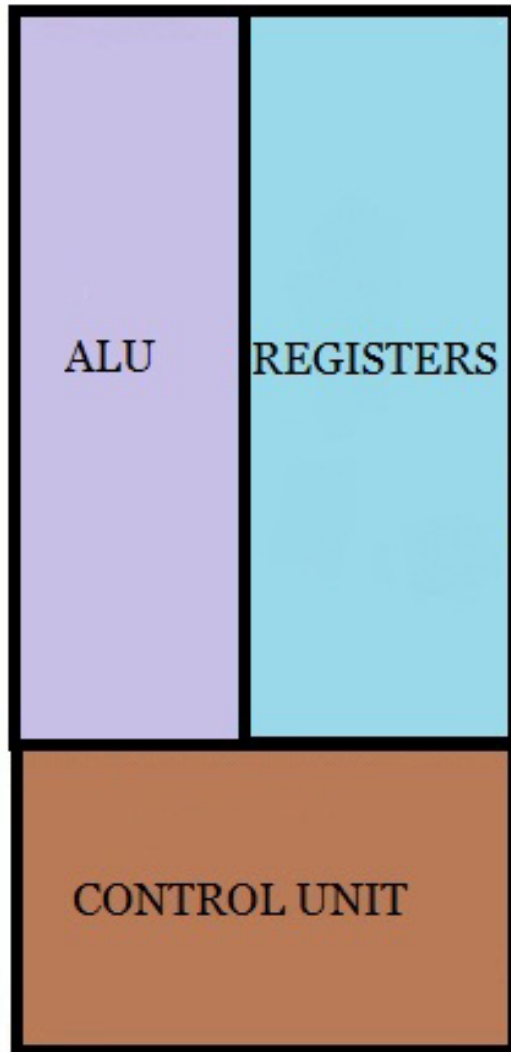
Docente - Vincenzo Giannotti

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.



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**».

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	(negazione)	
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

In informatica 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

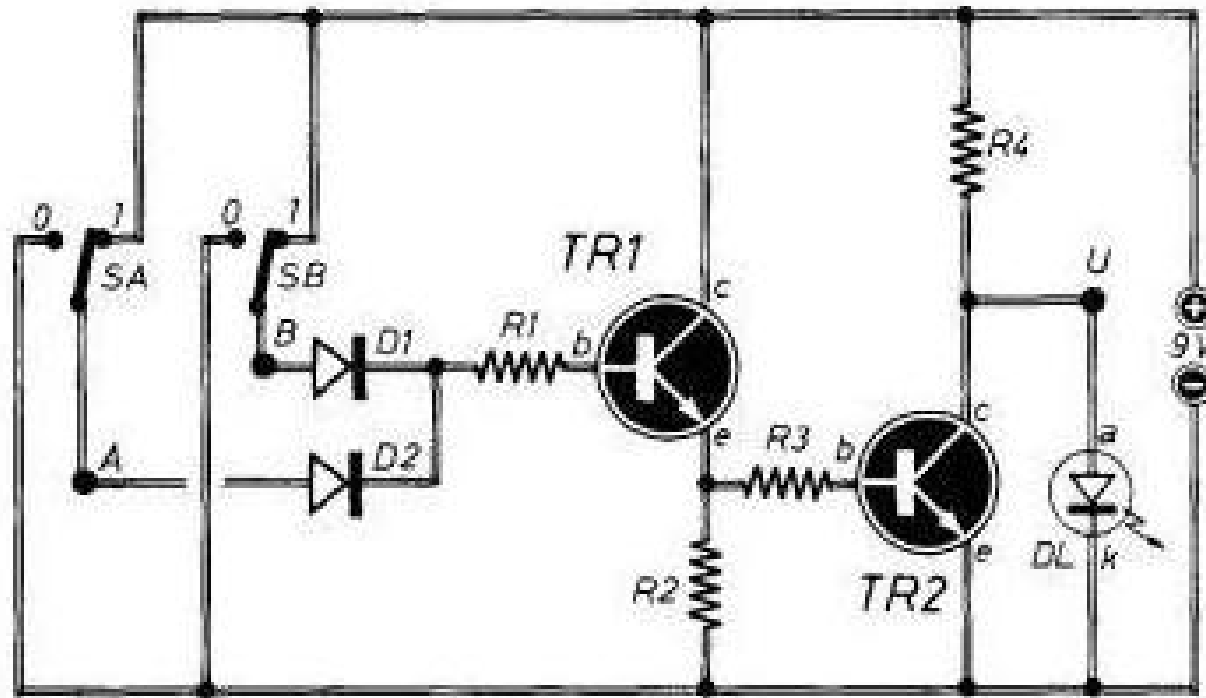
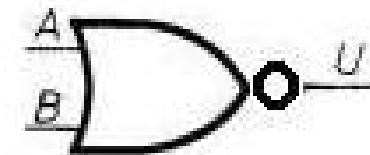


Tabella verita " NOR "

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

Figura tratta da
https://www.microst.it/Tutorial/funzioni_logiche_6.html



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;
 -

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, 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).

La CPU - Registri

I Registri sono delle piccola 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;

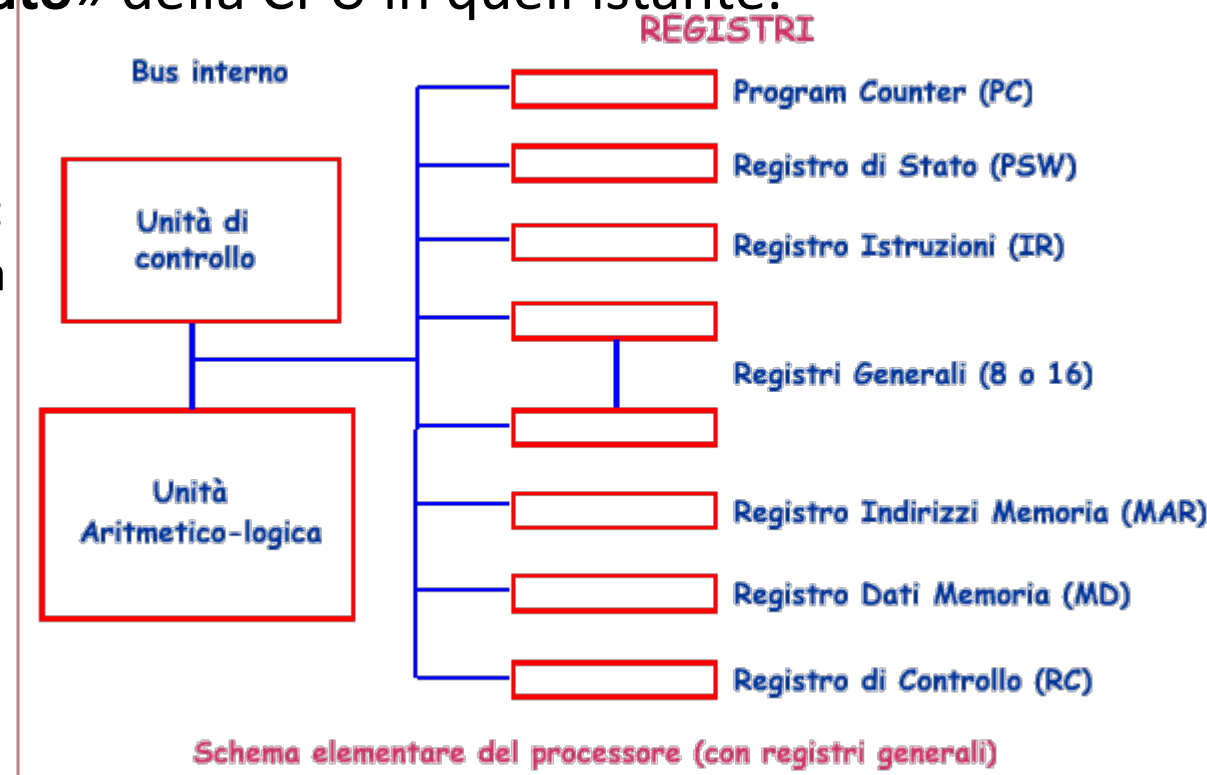
instruction register (IR): contiene l'istruzione da eseguire;

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 che memorizza particolari condizioni;

control register (RC) è il registro per i segnali di controllo (ready, wait, write, read,...).



La CPU – Unità di Controllo

L'ultima componente della CPU che funge da vero e proprio «cervello» del nostro computer è l'Unità di Controllo (UC).

Cosa fa l'Unità di Controllo?

Essa effettua un controllo su tutti gli altri componenti e inoltre 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.

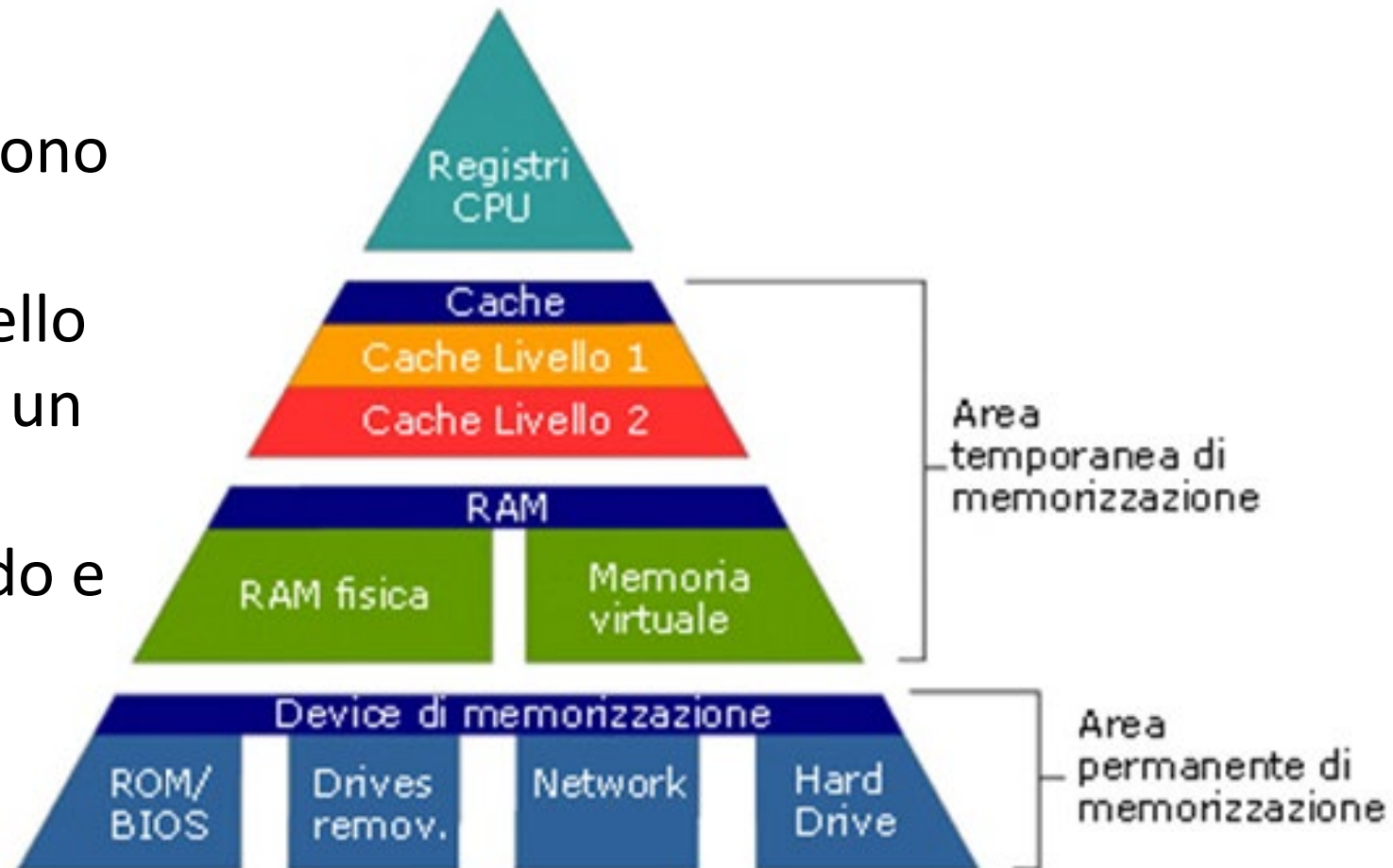
E produce in Output:

1. Segnali di controllo e di trasferimento dati

Tutte le Memorie del Computer

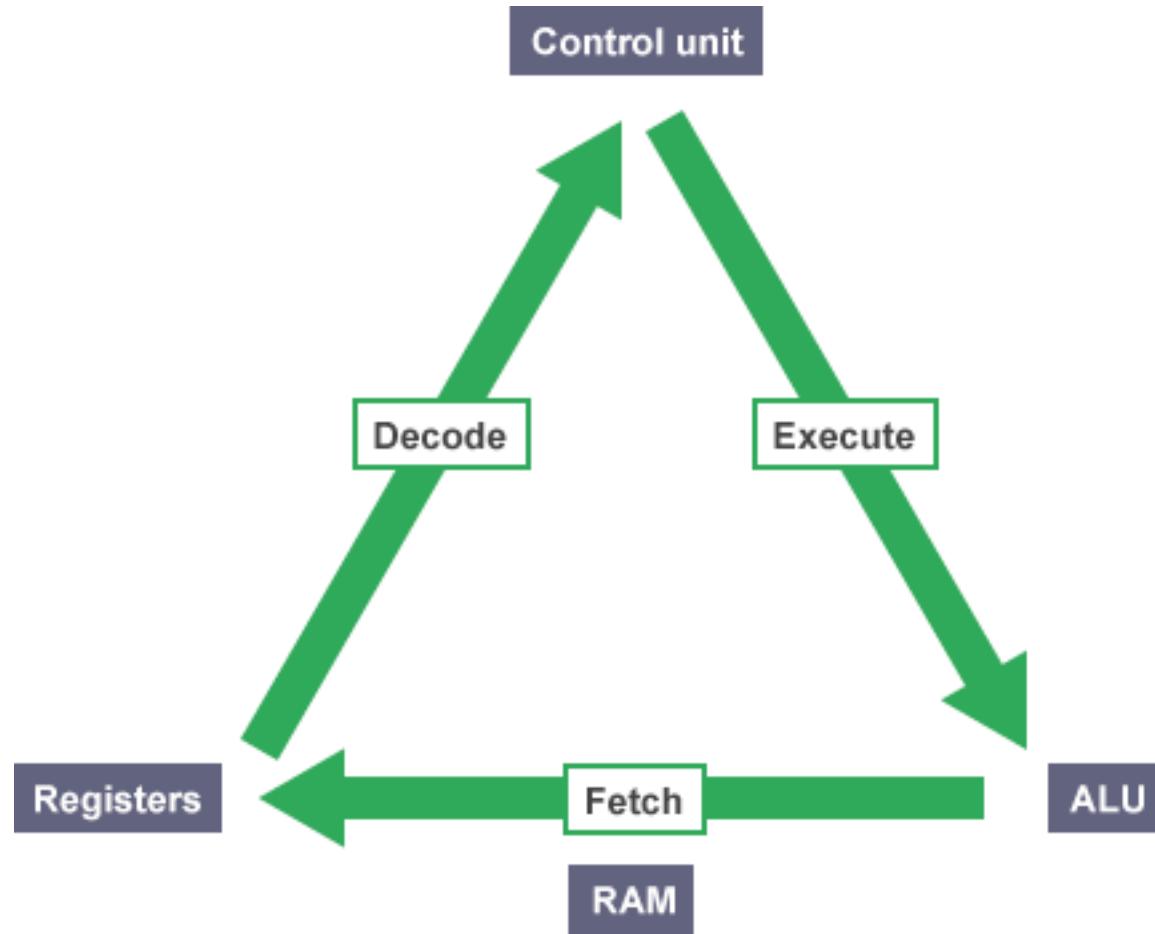
CPU e Memoria lavorano in simbiosi, infatti:

- I dati e i programmi risiedono in Memoria centrale
- Il compito della CPU è quello di eseguire le istruzioni di un programma presente in memoria centrale leggendo e scrivendo i dati, pure in memoria centrale.



Come funziona la CPU

Vediamo ora cosa fa una CPU quando esegue una istruzione.



Come funziona la CPU

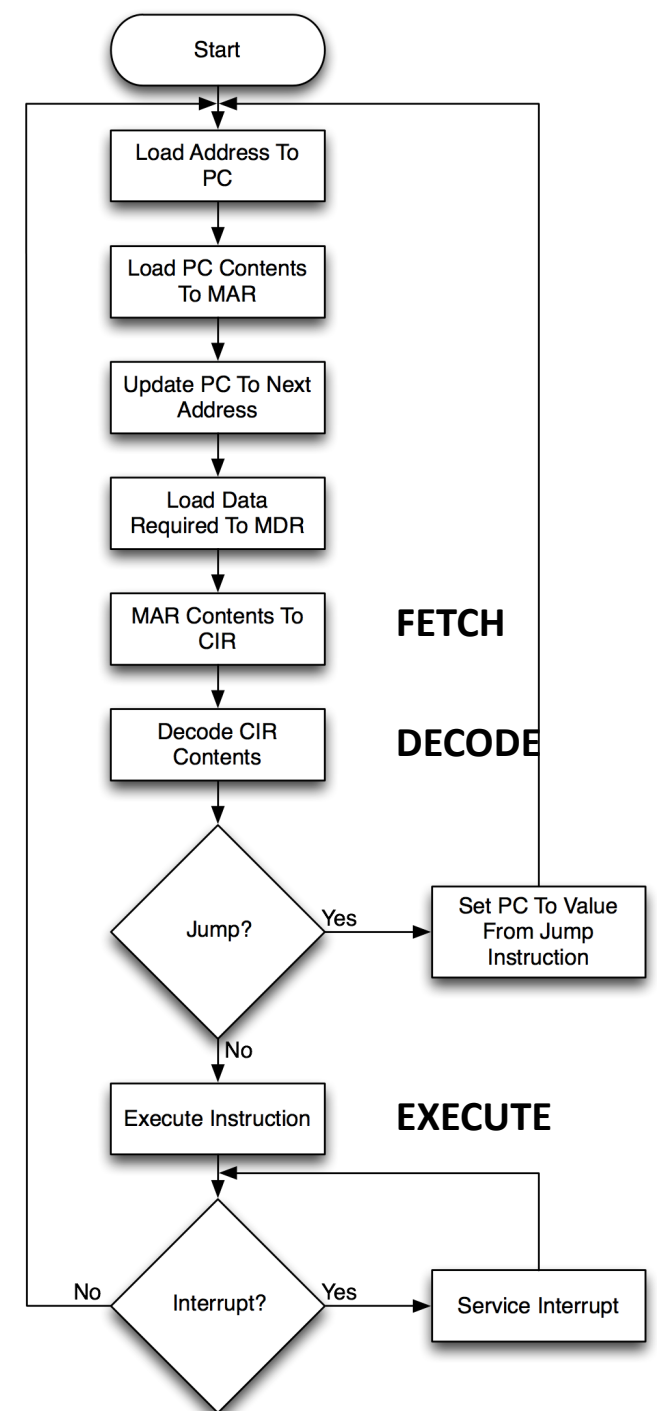
Il ciclo di lavoro di un processore è solitamente suddiviso in quattro fasi:

1. acquisizione dell'istruzione (**fetch**): il processore preleva l'istruzione dalla memoria, all'indirizzo specificato dal Program Counter
2. decodifica (**decode**): viene decodificato il operation code dell'istruzione; questo fornisce informazioni anche su quali dati prelevare in memoria (gli operandi)
3. esecuzione (**execute**): l'istruzione viene eseguita
4. riscrittura (**writeback**): si ritorna all'inizio del ciclo.

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

Come funziona la CPU

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 torna allo Start (o si finisce) completando il «**ciclo macchina**».



Come funziona la CPU

La CPU quindi 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 etc..) e che ora risiedono in memoria centrale e generando dei risultati (a monitor, verso i dispositivi audio etc..).

Per esempio, quando scriviamo una lettera con un programma di videoscrittura possiamo controllare a **Monitor** il risultato e alla fine, una volta che abbiamo terminato, archiviare definitivamente il risultato (la lettera in questo caso) in una memoria di massa in quello che viene detto 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.

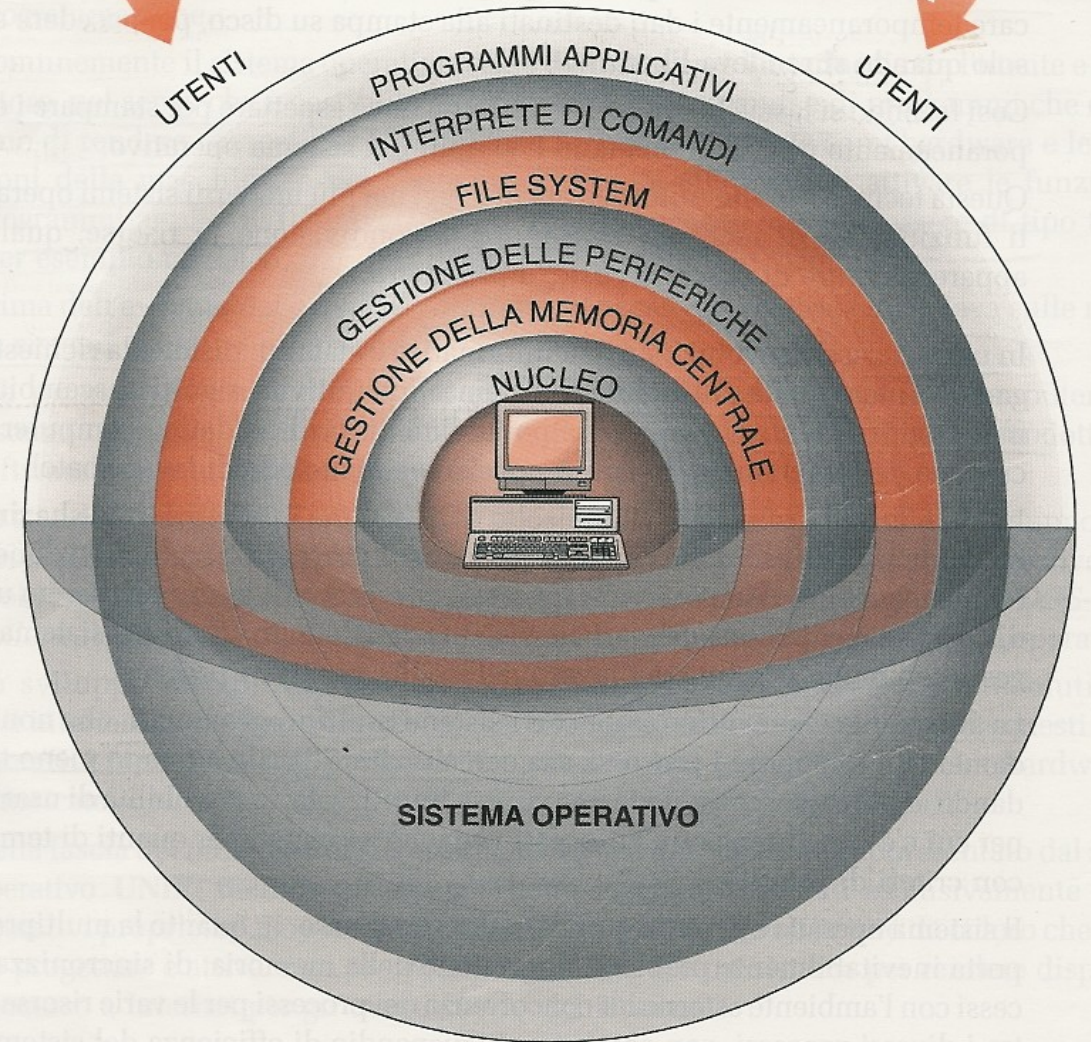
Il Sistema Operativo

Ma cosa è che rende trasparente all'utilizzatore tutte queste operazioni?

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

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.

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.

Mac OS is a series of graphical user interface-based operating systems developed by Apple Inc. for their Macintosh



iOS (previously iPhone OS) is a mobile operating system developed and distributed by Apple Inc. Originally unveiled in 2007 for the iPhone, it has been extended to support other Apple devices such as the iPod Touch



Linux is a Unix-like computer operating system assembled under the model of free and open source software development and distribution.



Android is a Linux-based operating system designed primarily for touchscreen mobile devices such as smartphones and tablet



Microsoft Windows is a series of graphical interface operating systems developed, marketed, and sold by Microsoft.



BSD/OS had a reputation for reliability in server roles; the renowned Unix programmer and author W. Richard Stevens used it for his own personal web server for this reason.



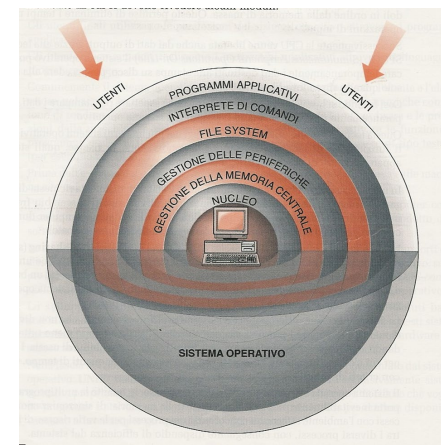
Il Sistema Operativo

- Un Sistema Operativo ha come fondamentale compito quello di coordinare, sincronizzare, gestire ed eseguire dei «**processi**» ossia dei compiti, delle mansioni (**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

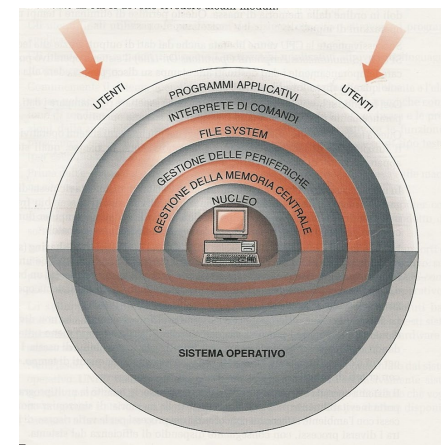
Gestione del Hardware



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:

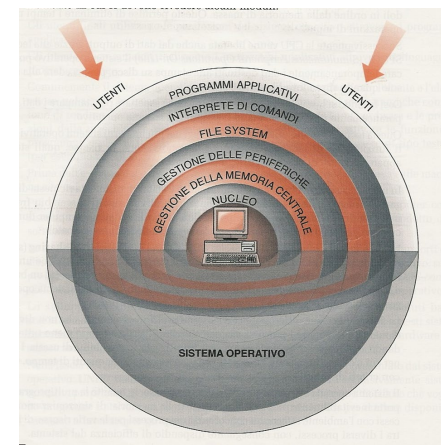
- L'esecuzione di un programma applicativo attraverso la creazione di un processo in cui il Kernel:
 1. assegna al programma lo spazio di memoria e le risorse richieste
 2. stabilisce la priorità del processo rispetto ad altri processi nei sistemi multi-tasking (in grado di eseguire più programmi contemporaneamente)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

La gestione del Hardware



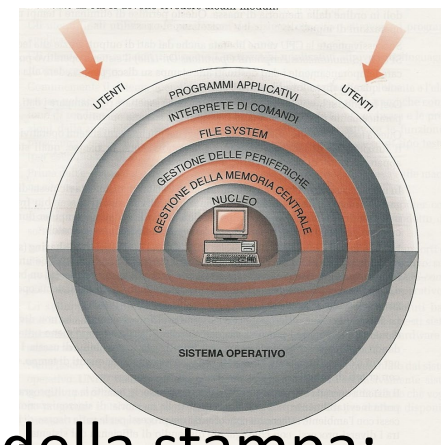
- Il Kernel del sistema operativo è responsabile della gestione di tutta la memoria di sistema utilizzata dai programmi. Questo assicura che un programma non interferisca con la memoria già in uso da un altro programma. Poiché il tempo viene condiviso dai programmi, ognuno di essi deve avere un accesso indipendente alla memoria.
- 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.

Gestione dei Processi

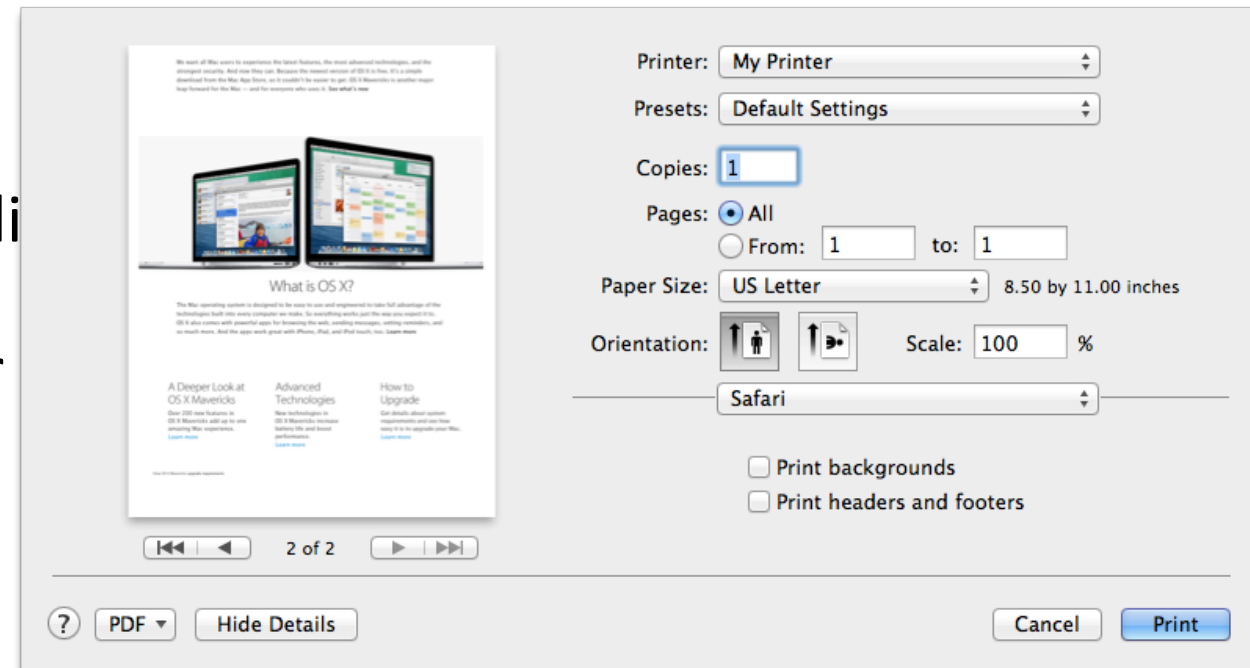


- Questa viene effettuata dallo **Scheduler** il quale scandisce il tempo di esecuzione dei vari processi e assicura che ciascuno di essi venga eseguito per il tempo richiesto.
- Normalmente lo scheduler gestisce anche lo stato dei processi e può sospenderne l'esecuzione nel caso questi siano in attesa 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.

lo Spooler di stampa

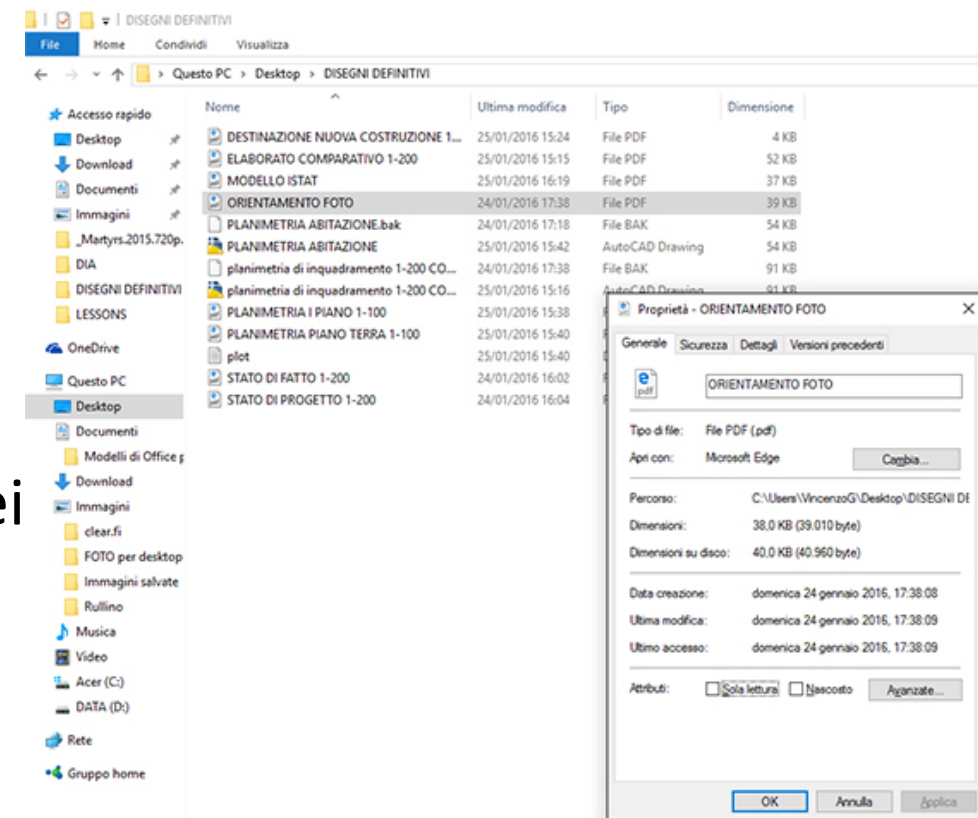
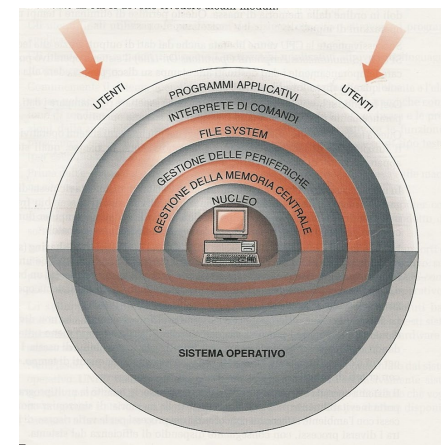


- Nel SO è anche integrato un programma che si comporta da gestore della stampa: lo Spooler. Esso riceve in ingresso i file da stampare dai programmi utilizzati dall'utente e li accoda mandandoli in stampa uno alla volta verso la stampante.
- Questo permette ai programmatori di applicativi di non doversi 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.



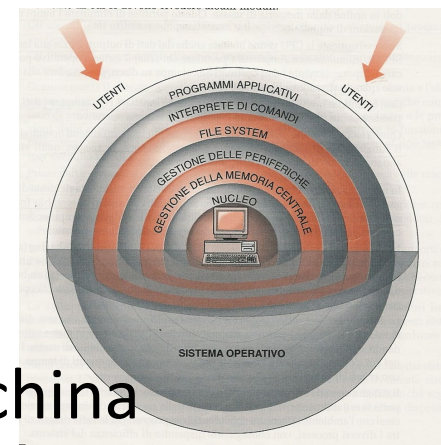
Gestione di File e Cartelle

- Tra i diversi compiti del Sistema Operativo c'è anche quello di gestire file e cartelle o meglio del 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.



La Shell

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



OS X Mountain Lion

Windows 10



La Shell



- 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.

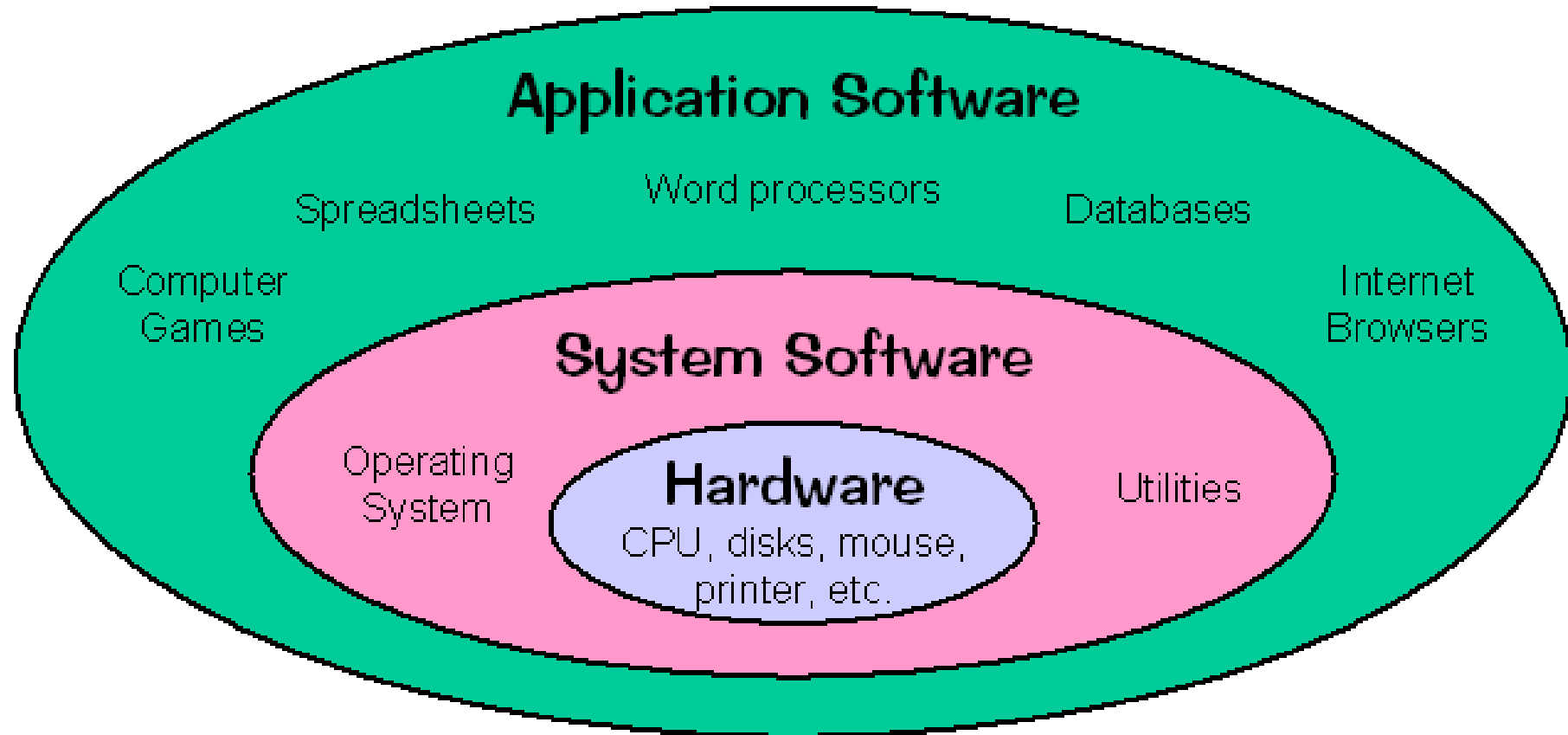
```
ca. Prompt dei comandi
Microsoft Windows [Versione 10.0.14393]
(c) 2016 Microsoft Corporation. Tutti i diritti sono riservati.

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, attività 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 se.

Software Open Source

- **Fedora** è un Sistema operativo basato su Kernel Linux.
- **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.
- **Debian** è un sistema operativo multi-architettura per computer, composto interamente da software libero.
- **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.



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 del 2008 dal Gruppo Standish afferma che l'adozione di modelli di software open-source ha portato a un risparmio di circa 60 miliardi di dollari all'anno per i consumatori.

Linguaggi di Programmazione

Tutta l'informatica ruota intorno all'attività della «programmazione» nella quale le idee e le esigenze vengono tradotte in algoritmi e in programmi da fornire al computer. Per realizzare un programma serve un linguaggio di programmazione.



Prossimo Capitolo – La codifica dell'Informazione Numerica

Nel prossimo capitolo faremo una incursione all'interno della CPU per vedere con maggior dettaglio come sia possibile che con i soli due simboli 1 e 0, un computer possa compiere tutte le meraviglie a cui siamo ormai abituati e che ci sembrano così naturali.