



ALMA MATER STUDIORUM
UNIVERSITÀ DI BOLOGNA

Fondamenti di Informatica T

Hardware & Software

Questioni di Scala

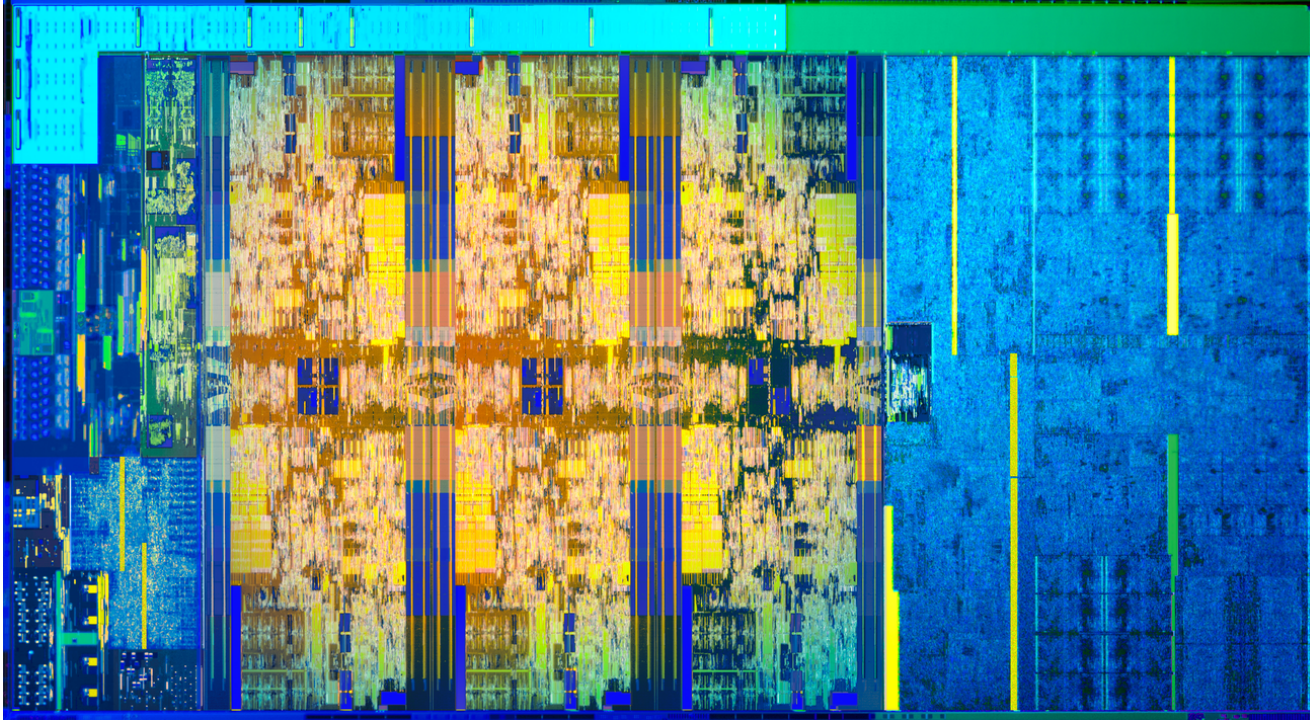


"A single car has about 30,000 parts, counting every part down to the smallest screws"

(Toyota, children's question room 😊)



Questioni di Scala



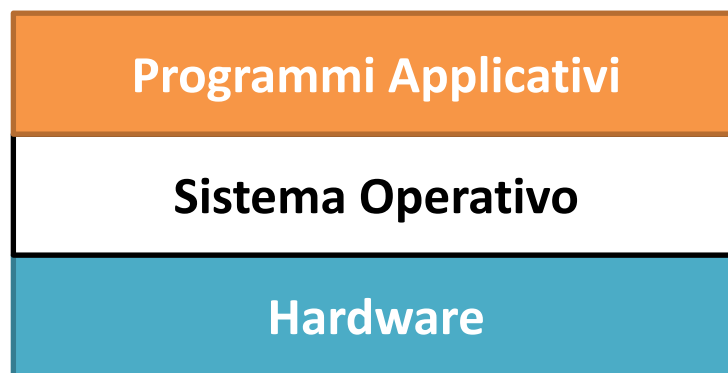
*"A single Sky, Kaby or Coffee Lake CPU core and its caches has approximately **217 million** transistors"*

(anon. utente, dopo un calcolo grossolano)



Livelli di Astrazione

Come gestire tale complessità? Si opera per livelli di astrazione:

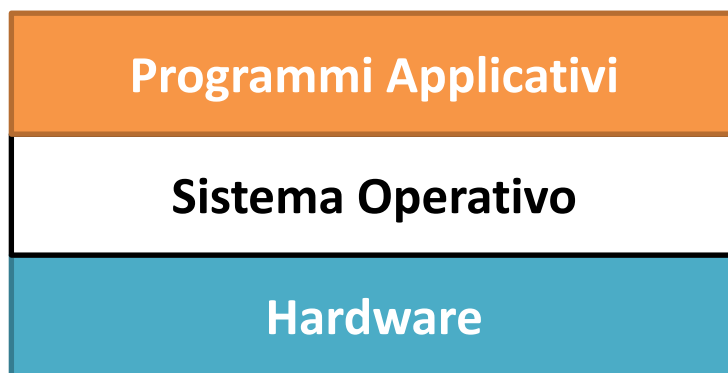


- L'**hardware** è l'elaboratore (e.g. Macbook Pro 2018)
- Il **Sistema Operativo** è un insieme di programmi per la gestione dell'elaboratore (e.g. Windows, OSX, Linux/Unix...)
- I **programmi applicativi** sono quelli che usiamo di solito (e.g. MS Word, Esplora Risorse, Music Player, IDE...)



Livelli di Astrazione

Come gestire tale complessità? Si opera per livelli di astrazione:

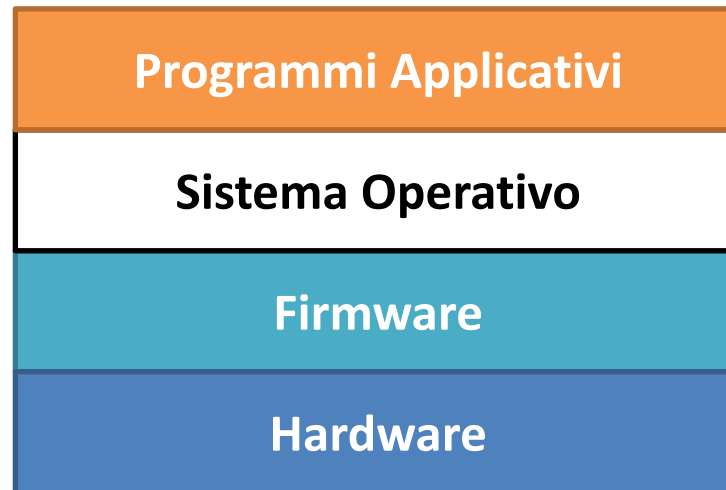


- L'hardware è la vera e propria **macchina fisica**
- Il Sistema Operativo realizza su di esso una **macchina virtuale**
 - È la stessa idea usata dai linguaggi di alto livello!
- I programmi applicativi si **avvalgono delle funzionalità del SO**



Livelli di Astrazione

Lo stesso approccio è utilizzato all'interno di ogni livello



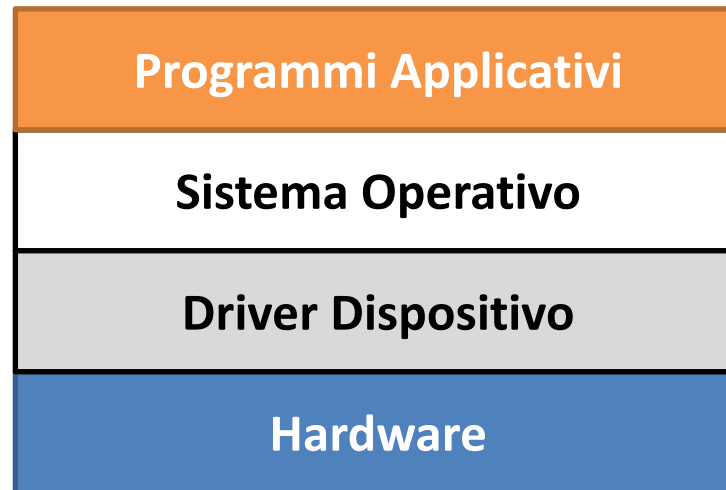
Qualche esempio:

- Il **firmware** è codice di bassissimo livello per la gestione dei dispositivi fisici (e.g. controllore termico)



Livelli di Astrazione

Lo stesso approccio è utilizzato all'interno di ogni livello



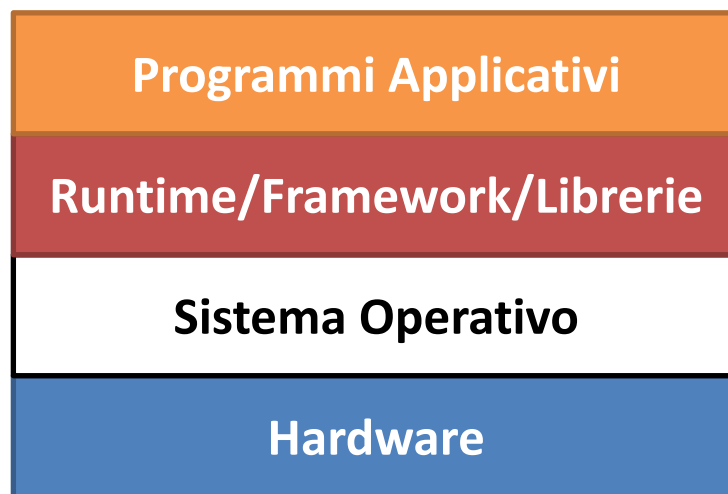
Qualche esempio:

- I **driver di dispositivo** sono programmi che permettono al SO di comunicare con i dispositivi



Livelli di Astrazione

Lo stesso approccio è utilizzato all'interno di ogni livello



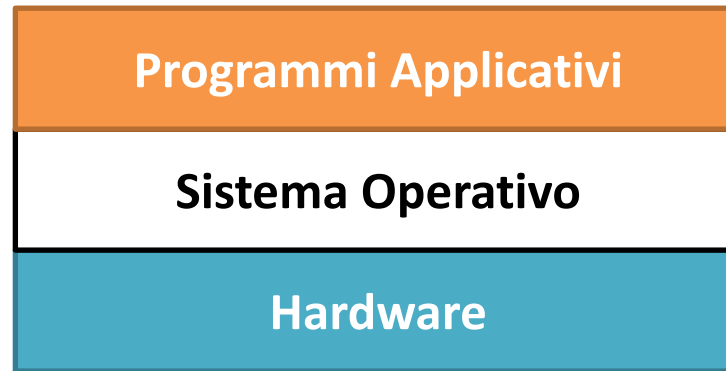
Qualche esempio:

- Molti SW applicativi si appoggiano su **librerie e software intermedio** più o meno complesso (cosiddetti framework)
 - E.g. i programmi C hanno bisogno di librerie, collettivamente note come "runtime C"



Livelli di Astrazione

In questa lezione discuteremo due dei livelli presentati



- Presenteremo brevemente l'architettura di un elaboratore
- Discuteremo alcune delle funzionalità del sistema operativo





ALMA MATER STUDIORUM
UNIVERSITÀ DI BOLOGNA

Fondamenti di Informatica T

Architettura dell'elaboratore

Macchina di Von Neumann

Un elaboratore elettronico:

- È organizzato in unità funzionali
- L'architettura è ispirata alla "Macchine di Von Neumann"
- Si tratta di un modello di elaboratore degli anni '40

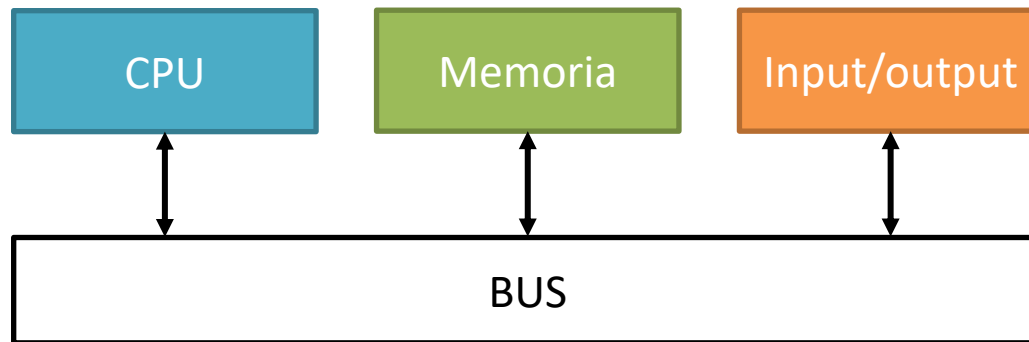


John Von Neumann



Macchina di Von Neumann

Una rappresentazione per la Macchina di Von Neumann:

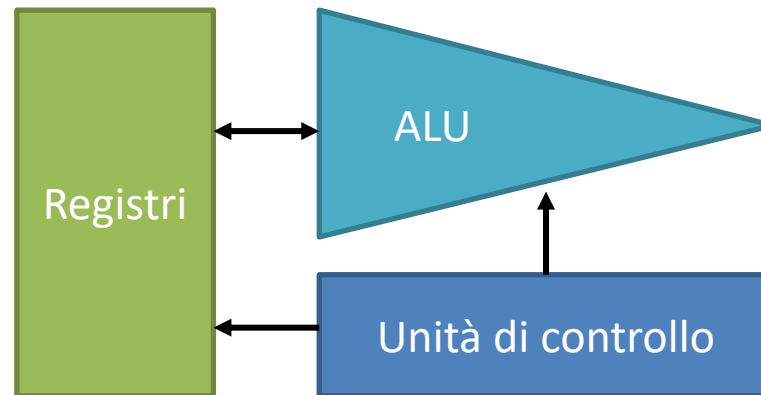


Quattro tipologie di unità funzionale:

- Processore (Central Processing Unit – CPU)
- Memoria Centrale (RAM & ROM)
- Unità di Input/Output (I/O)
- Bus di sistema (per la comunicazione tra componenti)

Central Processing Unit

Il processore è il "cervello" dell'elaboratore

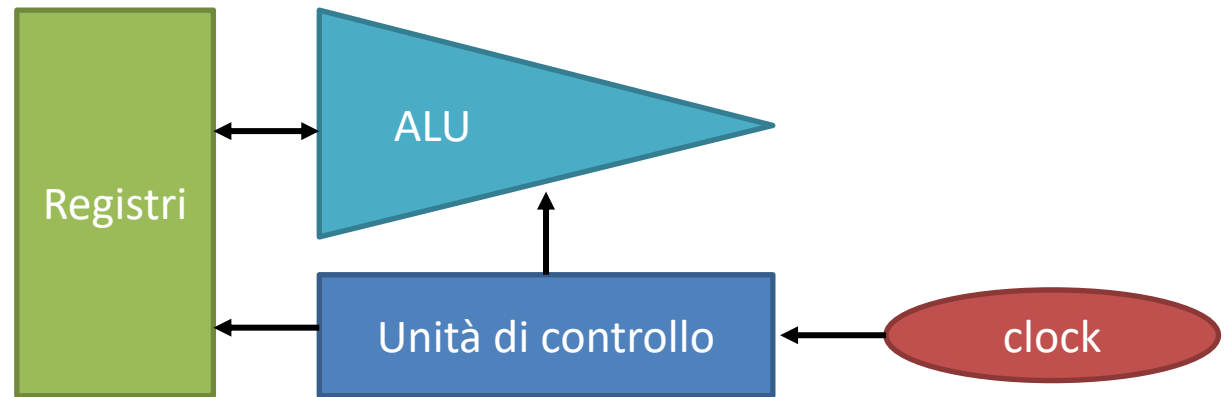


- Permette l'esecuzione di istruzioni
 - Calcoli, gestione del flusso di esecuzione (salti condizionali)...
- Ha tre sotto-componenti fondamentali
 - Unità Logico-Aritmetica (ALU)
 - Registri
 - Unità di controllo

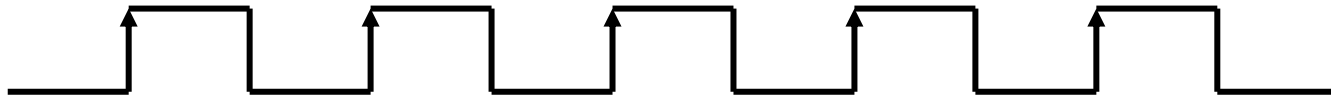


Central Processing Unit: Clock

Ogni operazione della CPU è regolata dal **clock**



- Il clock è un segnale elettrico periodico (onda quadra)

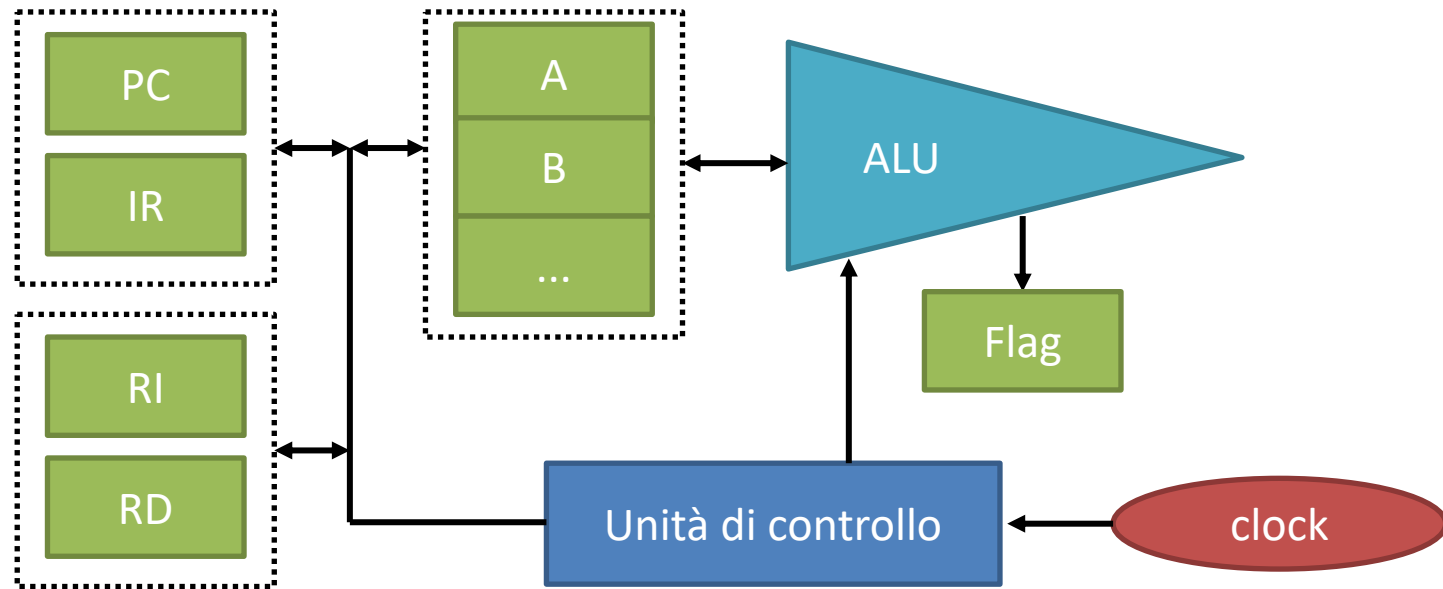


- Ogni elaborazione sulla CPU è sincronizzata dal clock
- Determina (in parte) la velocità di elaborazione
 - E.g. 2.1GHz, 3.1GHz, ecc.



Central Processing Unit: Registri

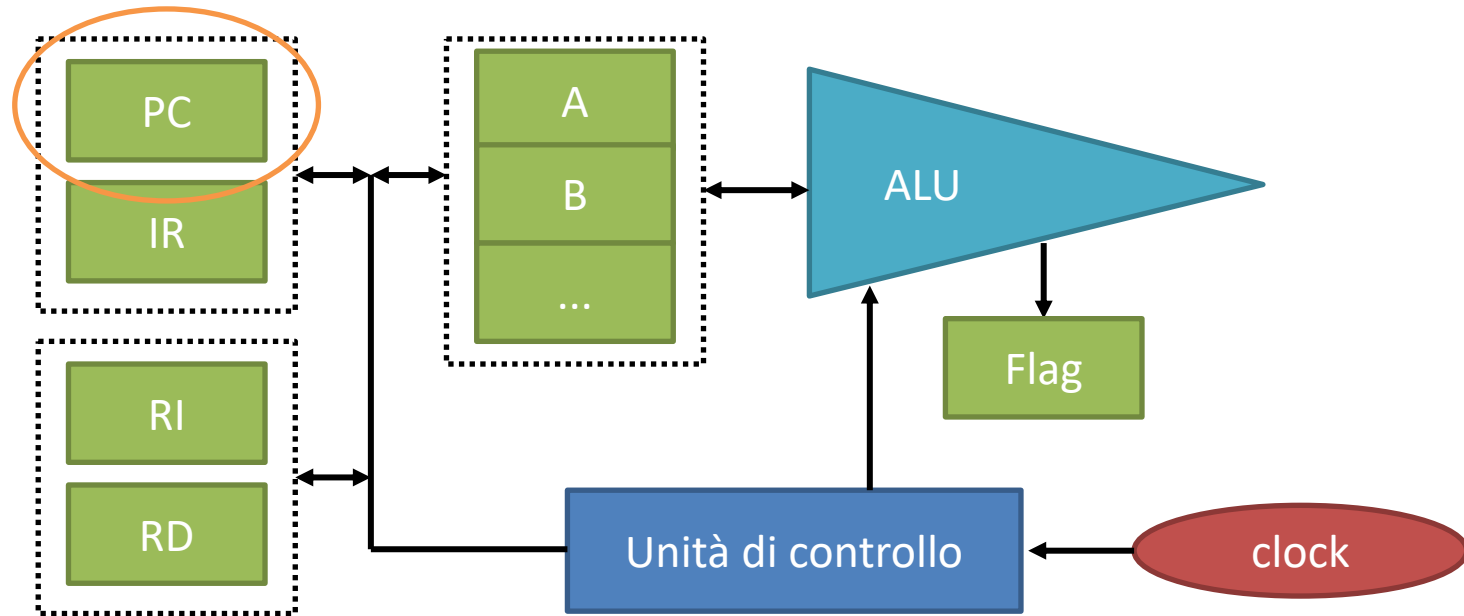
I **registri** sono locazioni di memoria interne al processore



- L'accesso ai registri è molto veloce (e.g. un ciclo di clock)
- Possono memorizzare dati, indirizzi, istruzioni

Central Processing Unit: Registri

I **registri** sono locazioni di memoria interne al processore



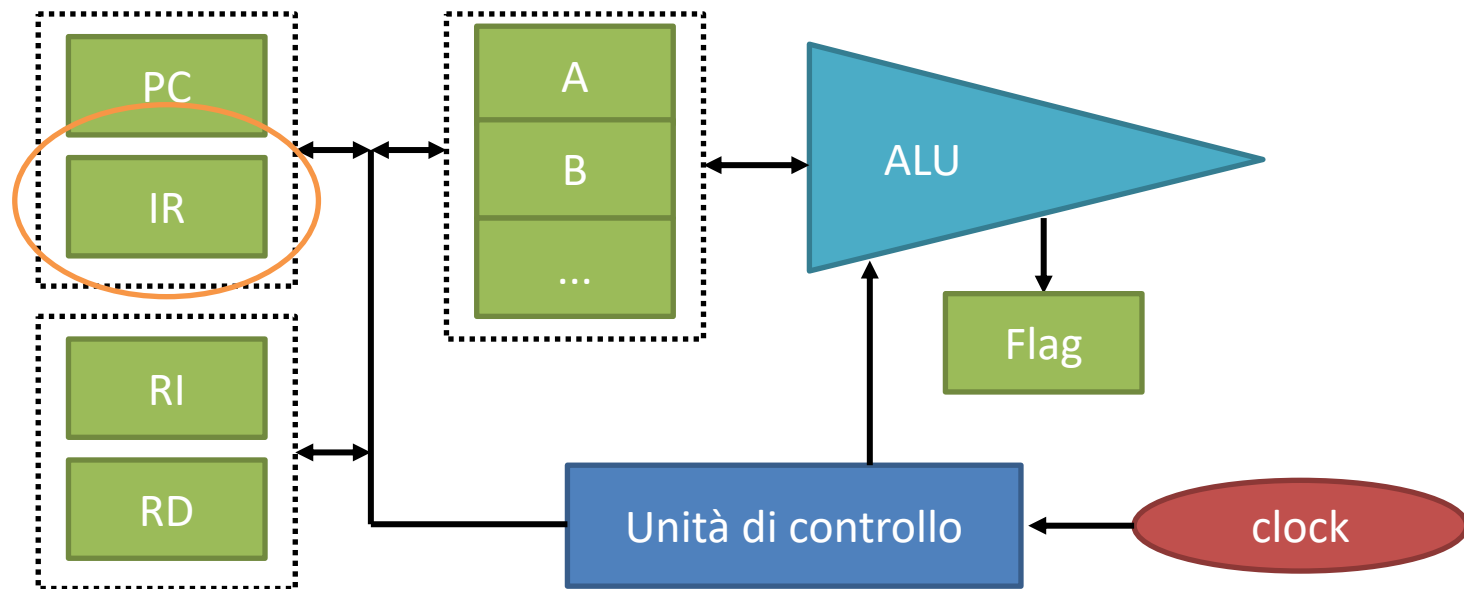
Program Counter:

- Contiene l'indirizzo della cella di memoria con la prossima istruzione da eseguire



Central Processing Unit: Registri

I **registri** sono locazioni di memoria interne al processore



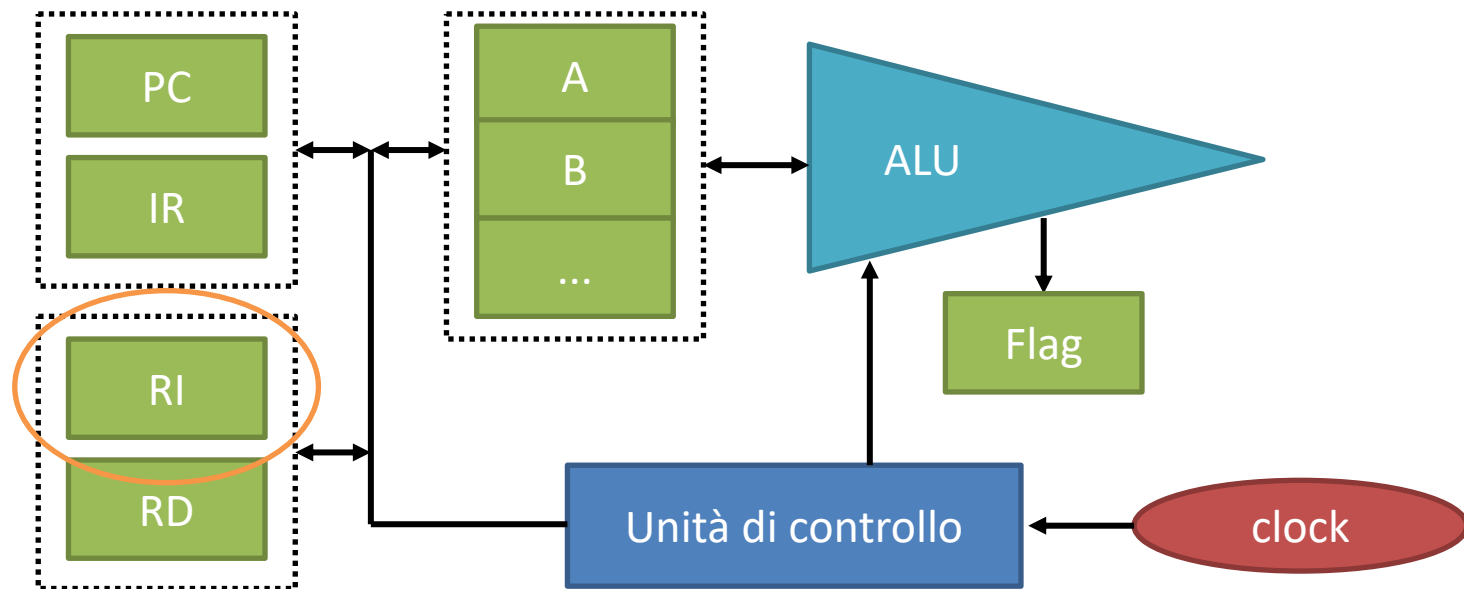
Instruction Register:

- Contiene l'istruzione da eseguire al ciclo corrente
- Viene riempito con l'istruzione indirizzata da PC



Central Processing Unit: Registri

I **registri** sono locazioni di memoria interne al processore



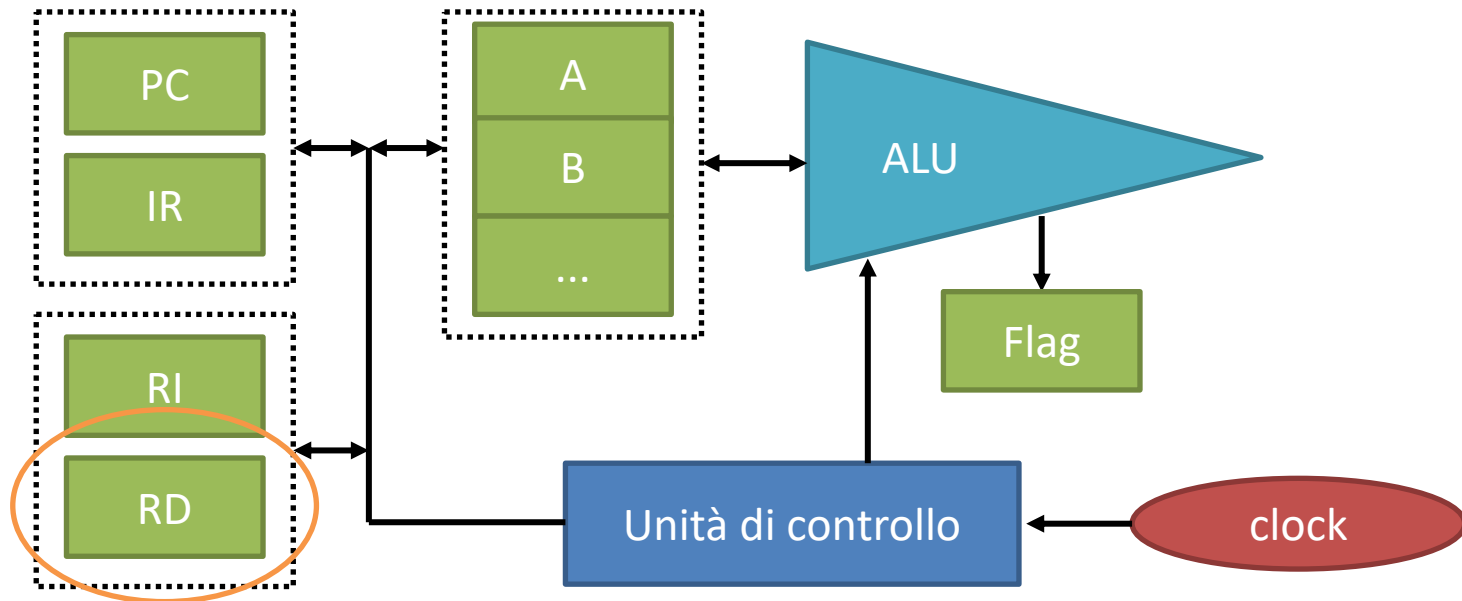
Registro Indirizzi:

- Contiene l'indirizzo della cella di memoria da utilizzare per un trasferimento di dati (da/verso la memoria)



Central Processing Unit: Registri

I **registri** sono locazioni di memoria interne al processore



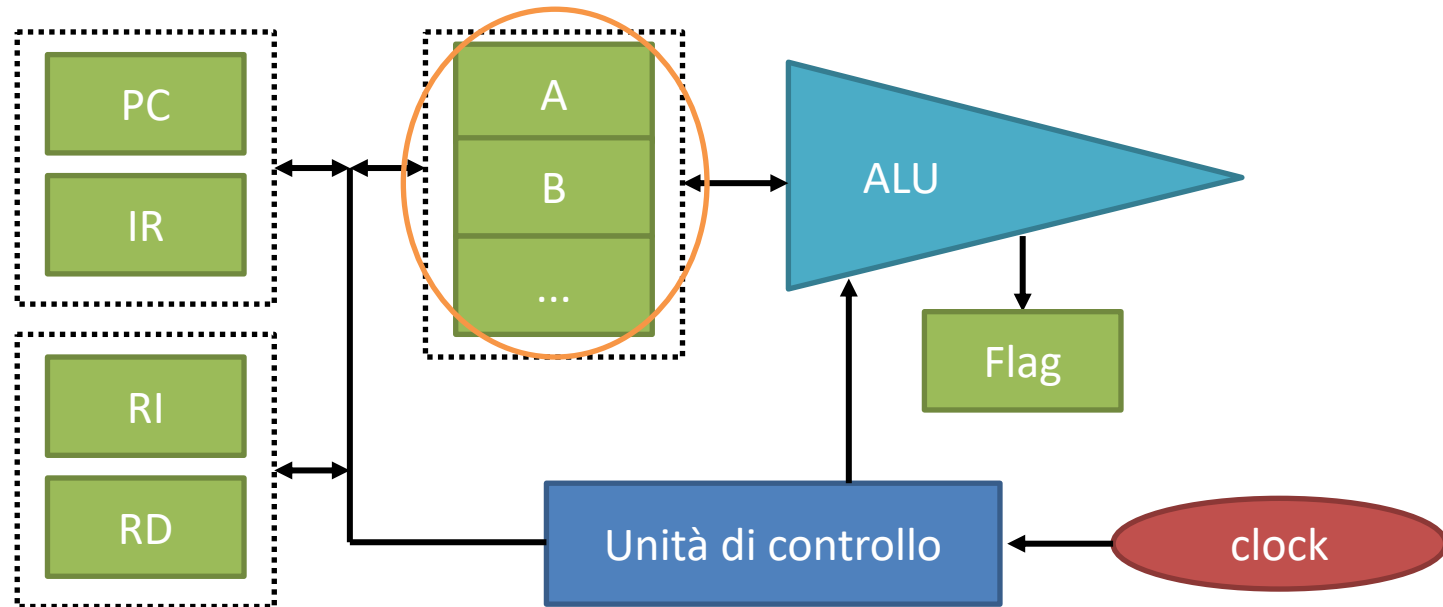
Registro Dati:

- Contiene il dato da trasferire da/verso la memoria



Central Processing Unit: Registri

I **registri** sono locazioni di memoria interne al processore

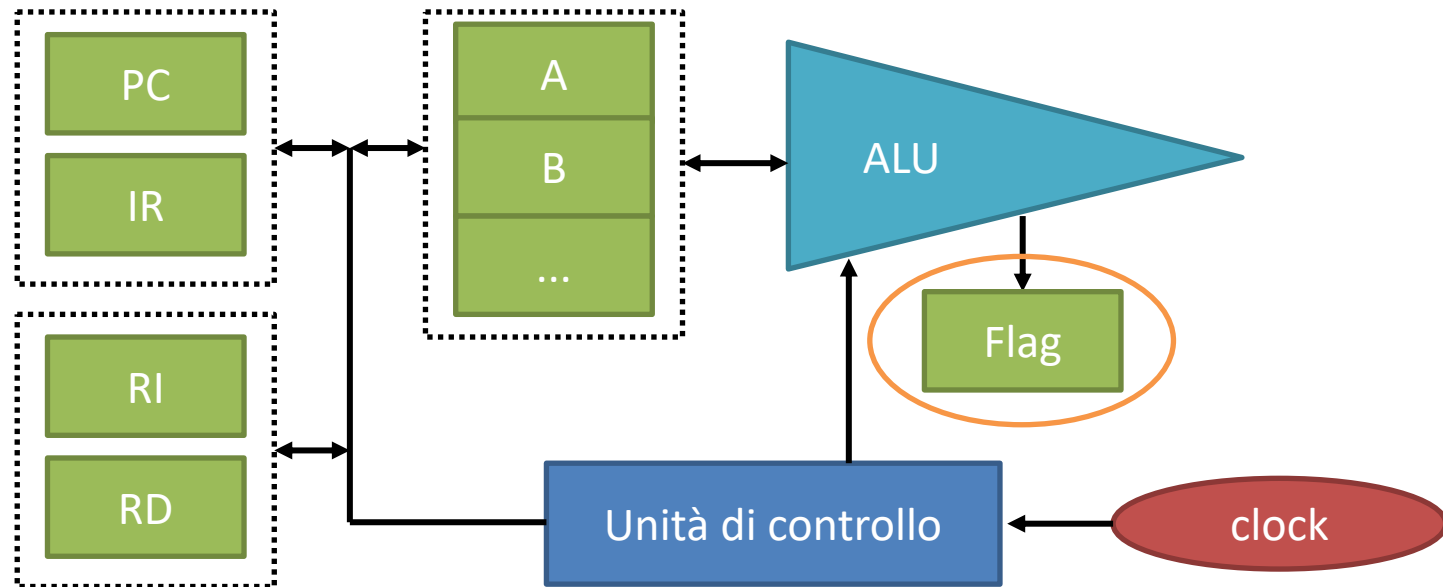


Registri di Uso Generale:

- Tipicamente usati per contenere operandi e risultati (in alcuni casi anche indirizzi – e.g. salti condizionali)

Central Processing Unit: Registri

I **registri** sono locazioni di memoria interne al processore



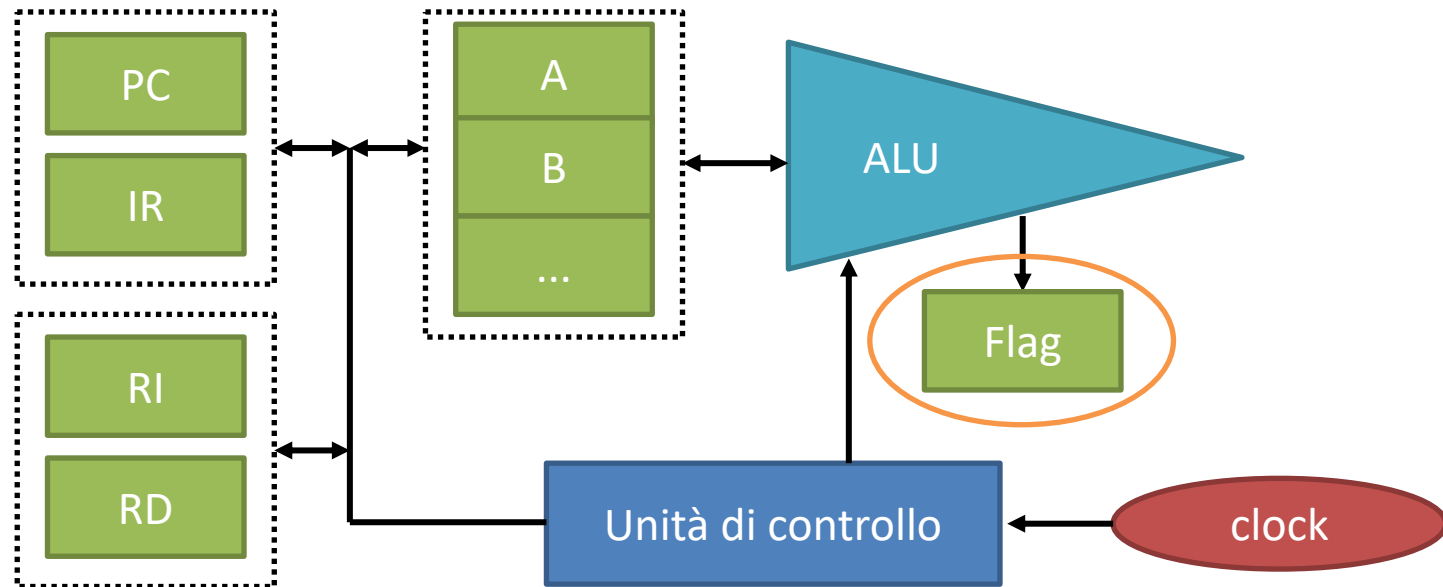
Registro dei Flag:

- Segnala proprietà del risultato dell'ultima operazione
- E.g. riporto, zero, divisione per zero...



Central Processing Unit: ALU

L'**unità aritmetico-logica** (Arithmetic Logic Unit – ALU):

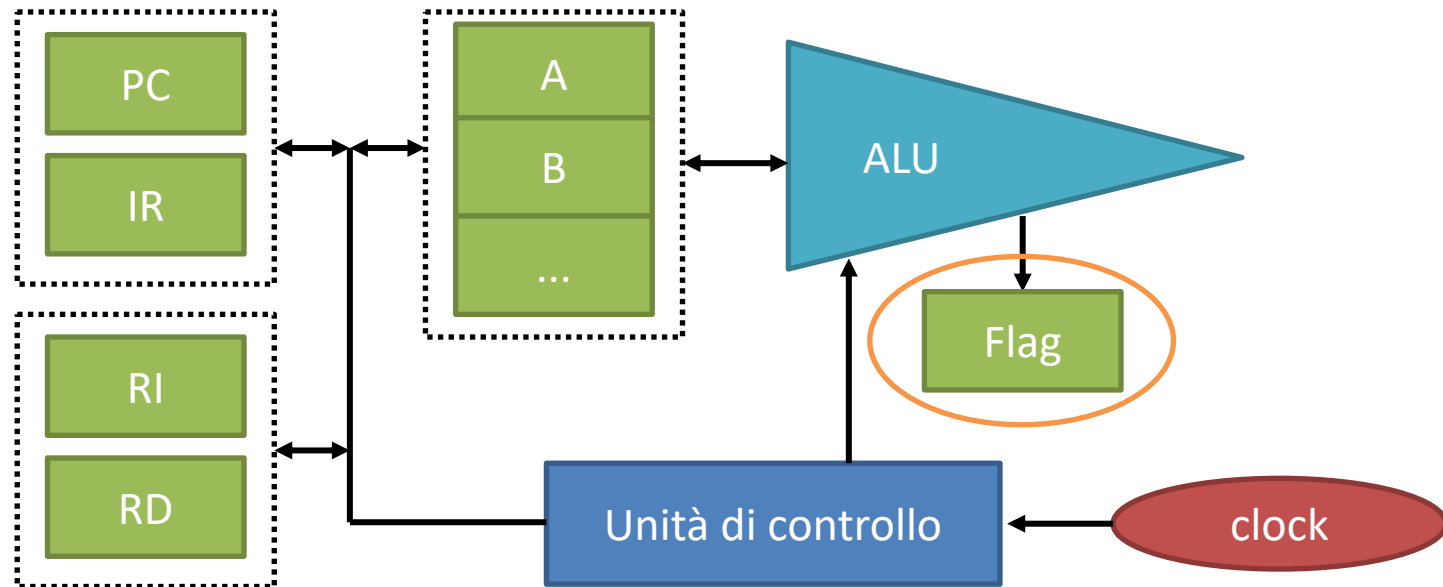


- Esegue operazioni aritmetiche e logiche
- È una sorta di (complessa) calcolatrice



Central Processing Unit: ALU

L'**unità aritmetico-logica** (Arithmetic Logic Unit – ALU):

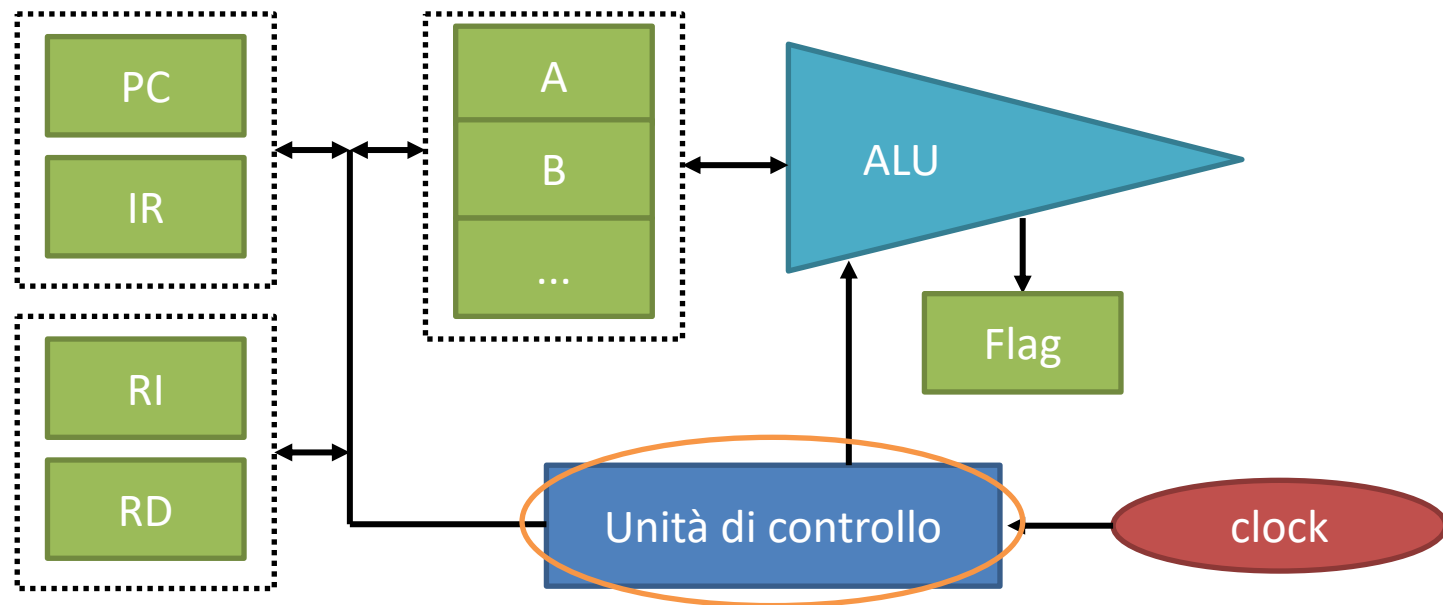


- Esegue operazioni aritmetiche e logiche
 - Pensatela come una (complessa) calcolatrice
- L'operazione da eseguire è determinata da IR



Central Processing Unit: ALU

L'**unità di controllo** orchestra il comportamento della CPU

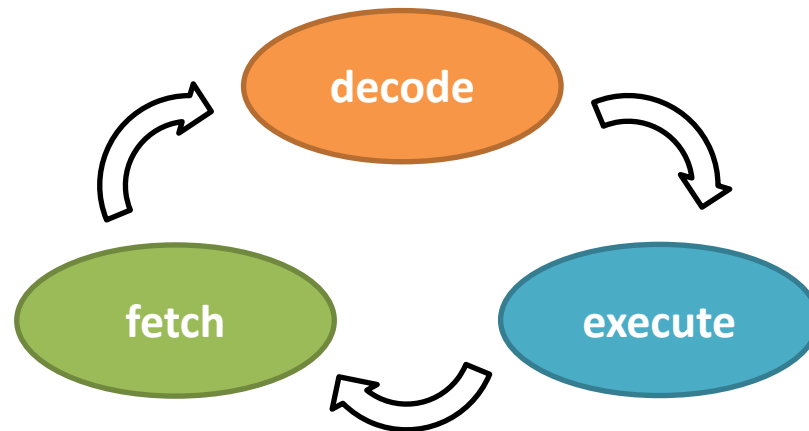


- Se l'ALU è una calcolatrice ed i registri sono memorie...
- ...L'unità di controllo è chi sta usando la calcolatrice

Central Processing Unit: Unità di Controllo

L'**unità di controllo** orchestra il comportamento della CPU

In pratica, ripete in continuazione il ciclo fetch/decode/execute:

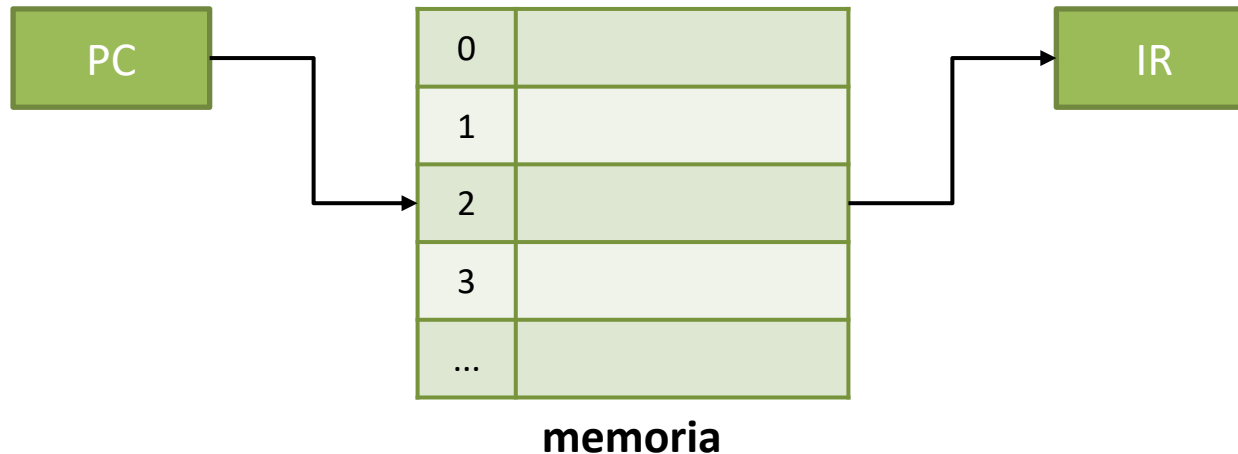


- **Fetch** (prelievo): si preleva la prossima istruzione da eseguire
- **Decode** (decodifica): si configura l'ALU per l'esecuzione
- **Execute** (esecuzione): l'operazione viene eseguita



Central Processing Unit: Fetch, Decode, Execute

Durante la fase "fetch":

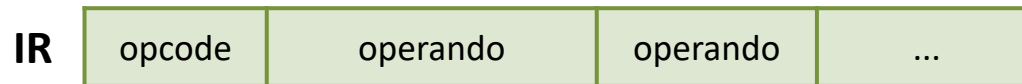


- Si usa il contenuto del PC per selezionare una cella di memoria
- Il contenuto della cella viene trasmesso tramite bus
- ...E viene copiato nell'Instruction Register



Central Processing Unit: Fetch, Decode, Execute

Durante la fase "decode":

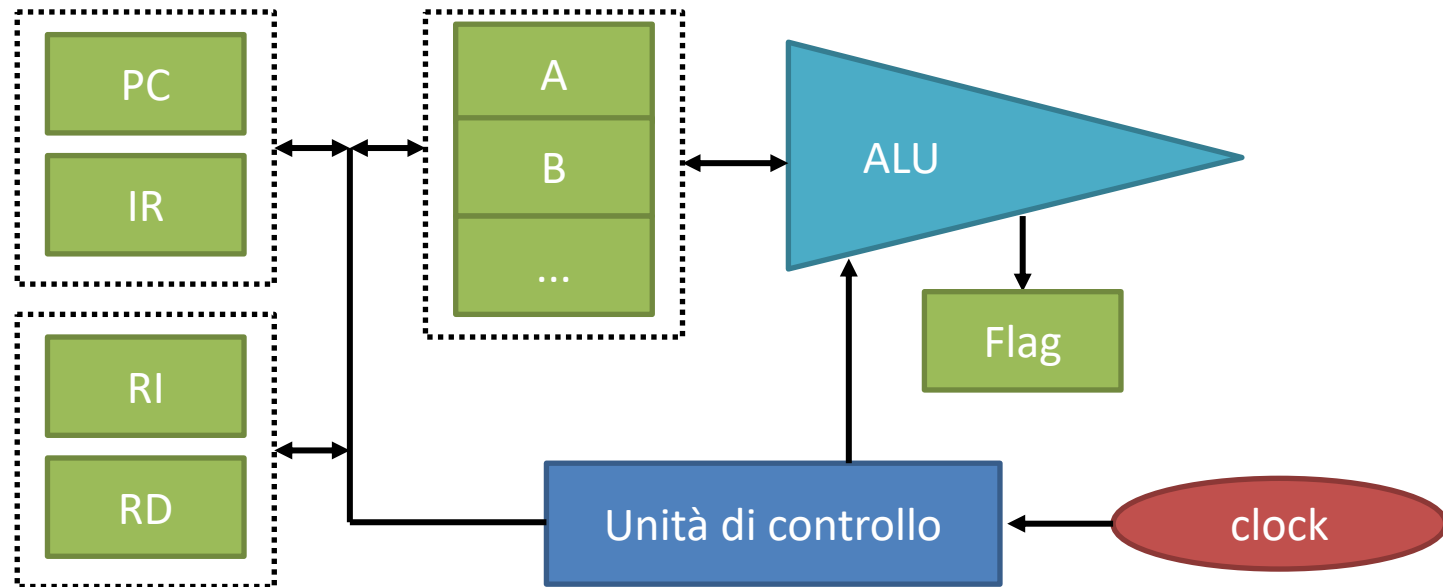


- Si analizza il contenuto dell'Instruction Register
- Si individua il tipo dell'istruzione (opcode)
 - E.g. somma, prodotto, incremento, salto...
- In base al tipo dell'operazione, si individuano gli operandi
- Si programma l'ALU di conseguenza



Central Processing Unit: Fetch, Decode, Execute

Durante la fase "execute":

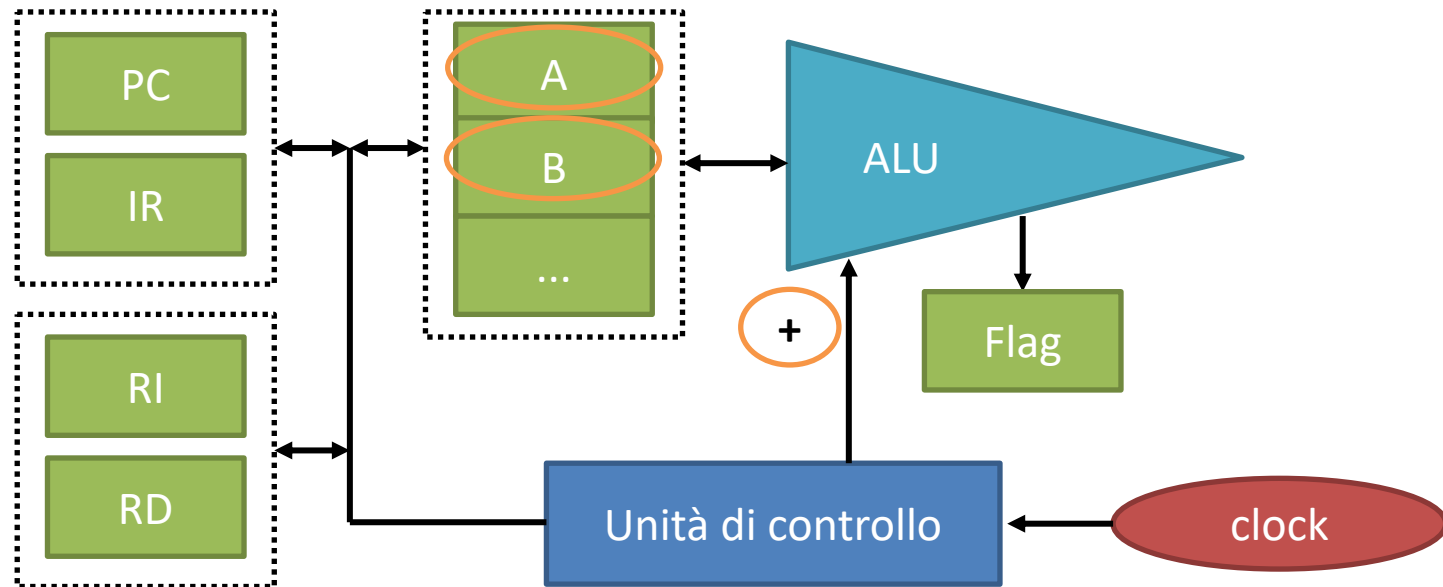


- Si esegue l'istruzione e si memorizza il risultato in registro
- Per le istruzioni di salto il risultato va nel PC
 - In questo modo altera la successiva fase "fetch"



Central Processing Unit: Un Esempio

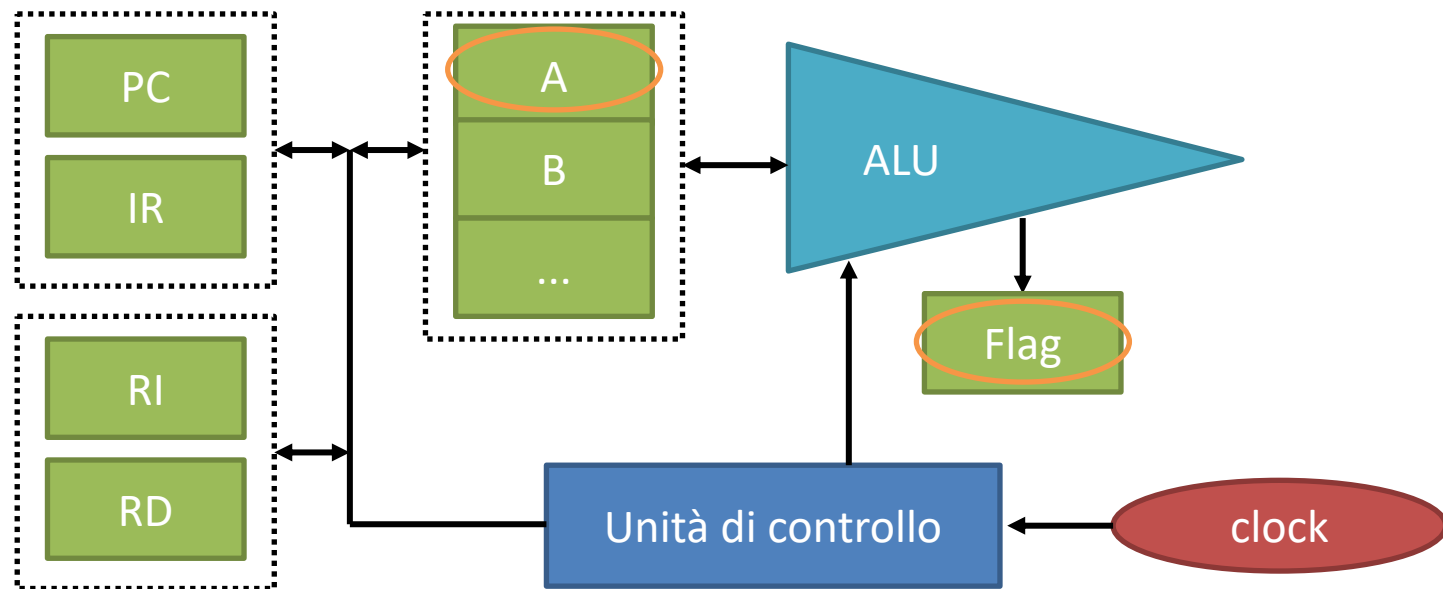
Un esempio di esecuzione:



- Gli operandi vengono caricati nei registri A e B
- All'ALU viene inviato un segnale che specifica il tipo di operazione da effettuare

Central Processing Unit: Un Esempio

Un esempio di esecuzione:



- Nel registro A viene caricato il risultato
- Nel registro Flag si indicano determinate proprietà del risultato (e.g. riporto, zero, segno...)



Memoria Centrale

La memoria centrale mantiene informazioni in formato binario:

0	
1	
2	
3	
...	

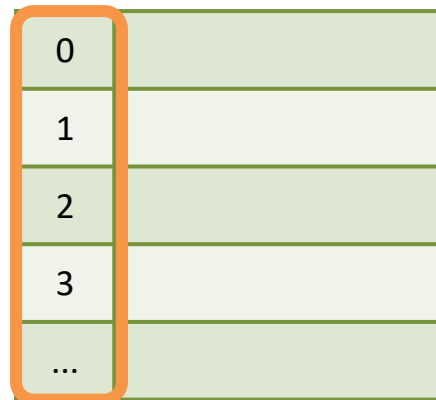
È costituita da una sequenza di **celle numerate**

- Ogni cella contiene un byte
- Le celle possono contenere dati o istruzioni
- È (generalmente) volatile, i.e. contenuto perso a PC spento
- È veloce (tempo di accesso ~10 ns)



Memoria Centrale

La memoria centrale mantiene informazioni in formato binario:



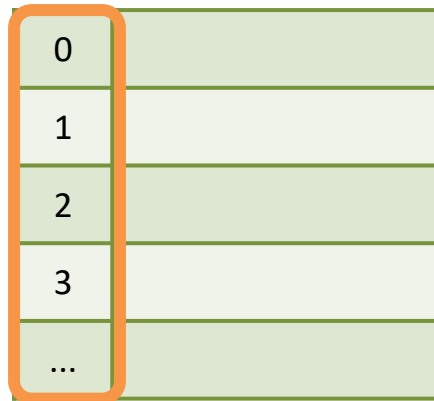
I numeri delle celle si chiamano indirizzi

- Si usano per selezionare la cella cui accedere
- I puntatori in C contengono esattamente questi indirizzi!



Memoria Centrale

La memoria centrale mantiene informazioni in formato binario:



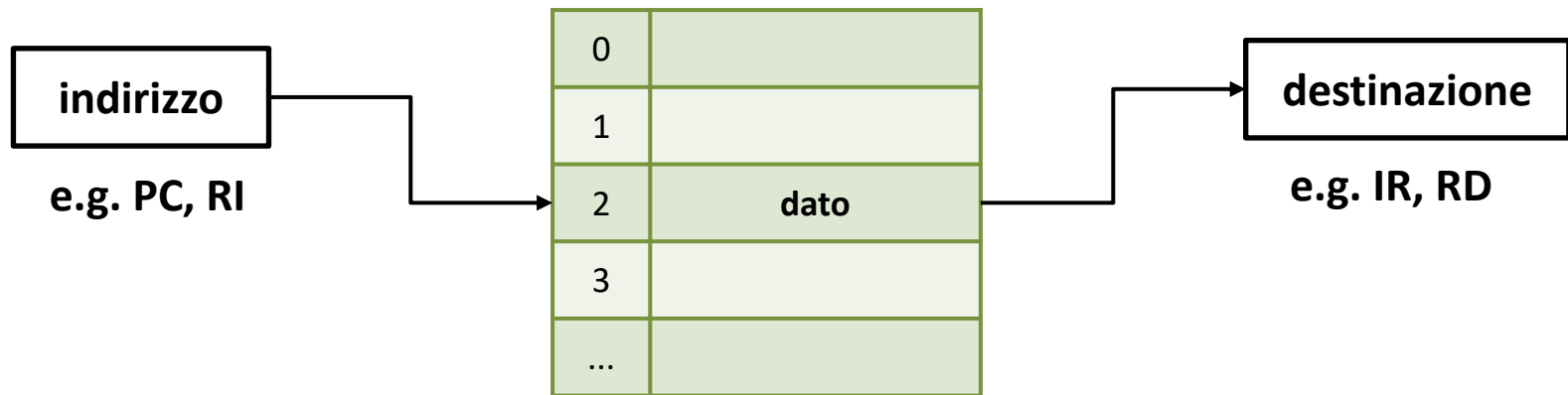
Lo spazio indirizzabile dipende dalla dimensione di un indirizzo

- Una CPU si dice "a N bit" se usa N bit per gli indirizzi:
- CPU a 16 bit: 2^{16} celle indirizzabili (64KiB)
- CPU a 32 bit: 2^{32} celle indirizzabili (4GiB)
- CPU a 64 bit: 2^{64} cella indirizzabili (16GiB)



Memoria Centrale: Lettura e Scrittura

Sulla memoria si possono fare operazioni di lettura e scrittura:



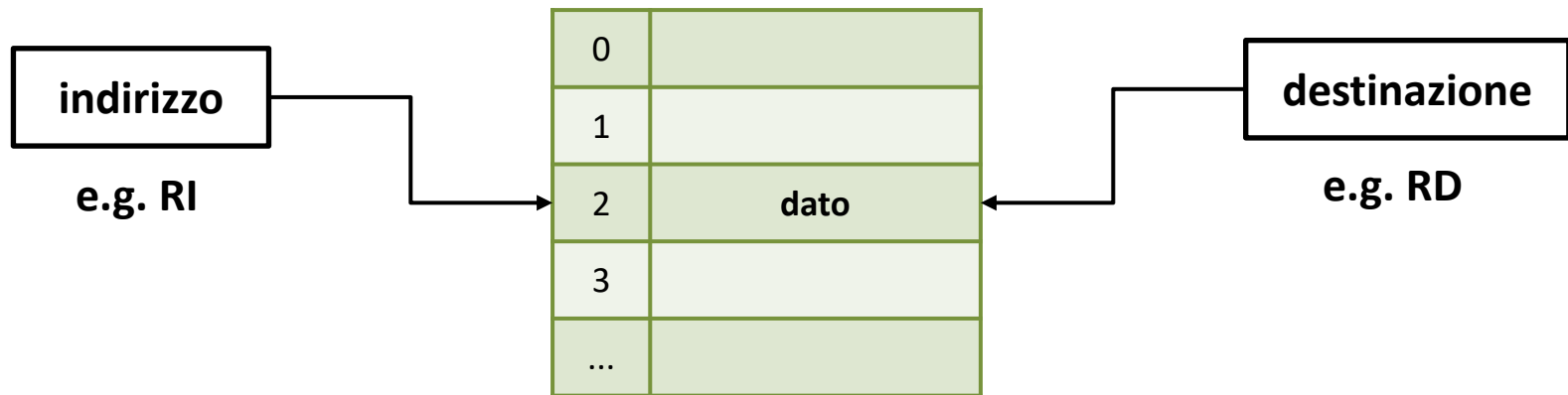
Durante una lettura

- Si seleziona una cella inviando un indirizzo sul bus
- Il dato contenuto alla cella viene trasmetto attraverso il bus...
- ...Fino ad una locazione di destinazione (e.g. un registro)



Memoria Centrale: Lettura e Scrittura

Sulla memoria si possono fare operazioni di lettura e scrittura:



Durante una **scrittura**

- Si seleziona una cella inviando un indirizzo sul bus
- Il dato contenuto in una sorgente (e.g. un registro)
- ...Viene inviato tramite il bus alla cella di memoria



Memoria Centrale: Dispositivi Fisici

Le memoria centrale è con realizzata con diversi **dispositivi fisici**

- **RAM** (Random Access Memory):
 - Accesso in lettura e scrittura
 - Volatile (contenuto perso a PC spento)
 - Costituisce la maggior parte della memoria centrale
 - Accesso non-sequenziale (anche detto "accesso casuale")
- **ROM** (Read Only Memory):
 - Accesso in sola lettura (il suo contenuto è immutabile)
 - Persistente
 - Utilizzata per dati e programmi necessari per l'avvio della macchina (BIOS, EFI...)



Memoria Centrale: Dispositivi Fisici

Le memoria centrale è con realizzata con diversi **dispositivi fisici**

- **PROM/EPROM/EEPROM:**
 - Varianti della ROM che possono essere modificate
 - Rimpiazzano largamente la ROM su macchine moderne
 - Tipicamente il firmware è ospitato su ROM/EPROM

Nota:

- La memoria centrale non include (e.g.) il disco rigido!



Memoria Centrale: Gerarchia delle Memorie

Problema: La RAM è veloce, ma non quanto un processore!

- Tempo di accesso in RAM ~ 10 ns
- 1 ciclo di clock a 3.1GHz: 0.32 ns

Conseguenza: la CPU può perdere tempo aspettando i dati

Soluzione: Inserire una **memoria cache** tra processore e RAM

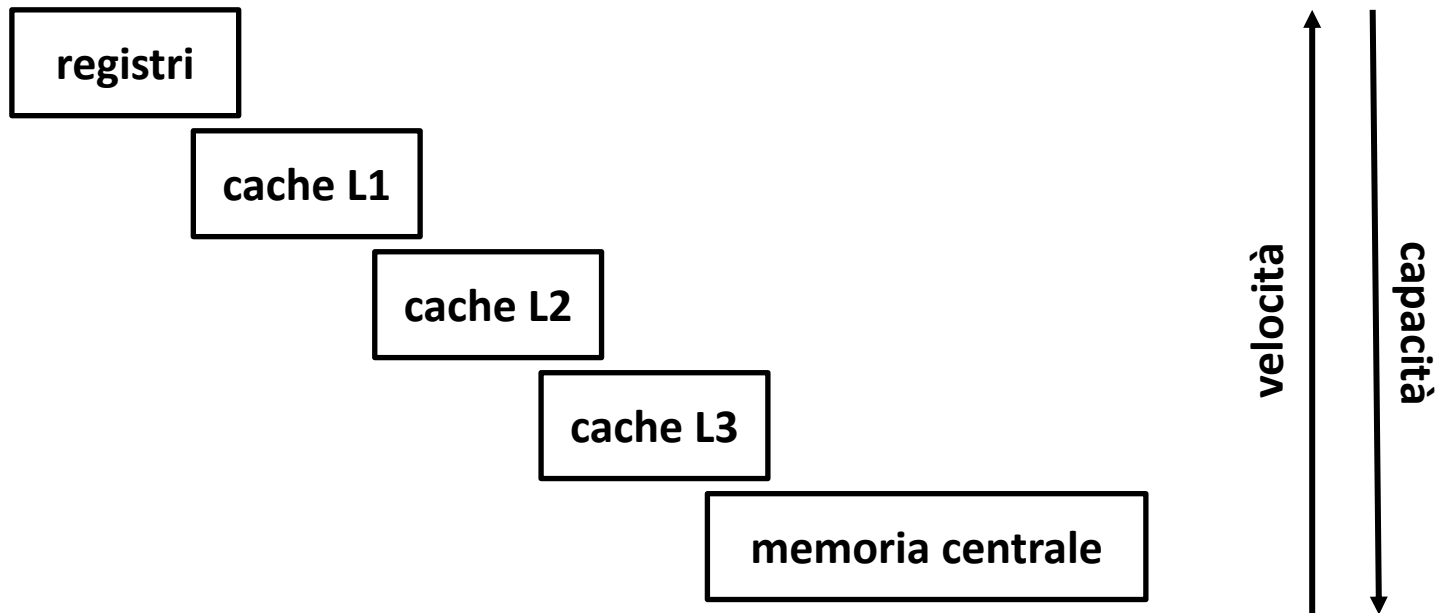
- Tale memoria è dedicata a dati ed istruzioni di uso frequente
- Al primo utilizzo di un dato/istruzione, lo si carica in cache
- Agli accessi successivi, si usa la copia in cache

Quando la cache è piena si rimpiazza il contenuto più vecchio



Memoria Centrale: Gerarchia delle Memorie

In pratica, si usano **diversi livelli di cache**, e.g.



E.g. per Intel Coffee Lake:

- L1: 32KiB (dati) + 32KiB (istruzioni) per core
- L2: 256KiB per core
- L3: 2MiB per core



Unità di Input/Output

Le unità di input/output...

...Includono tutti gli altri dispositivi collegati ad un elaboratore

- Monitor
- Tastiera
- Mouse
- Trackpad
- Touchscreen
- Stampante
- Fotocamera
- Interfacce di rete (wifi, ethernet)
- Memoria di massa
- ...



Unità di Input/Output

Le unità di input/output...

...Comunicano con l'elaboratore via **protocolli standard**. E.g.:

- USB-A
- USB-C
- PCI-express
- SATA

Ogni protocollo definisce:

- Il mezzo fisico (e.g. cavo, connettore...)
- Il formato per lo scambio di dati
- Il processo di scambio di dati



Unità di Input/Output: Memoria di Massa

La memoria di massa:

- Memorizza grandi quantità di informazioni
- È persistente (il contenuto si mantiene a PC spento)
- È molto più lenta della memoria centrale

La memoria di massa è realizzata da alcuni dispositivi di I/O:

- Dischi rigidi
- SSD (Solid State Drive)
- Penne USB
- CD/DVD
- ...

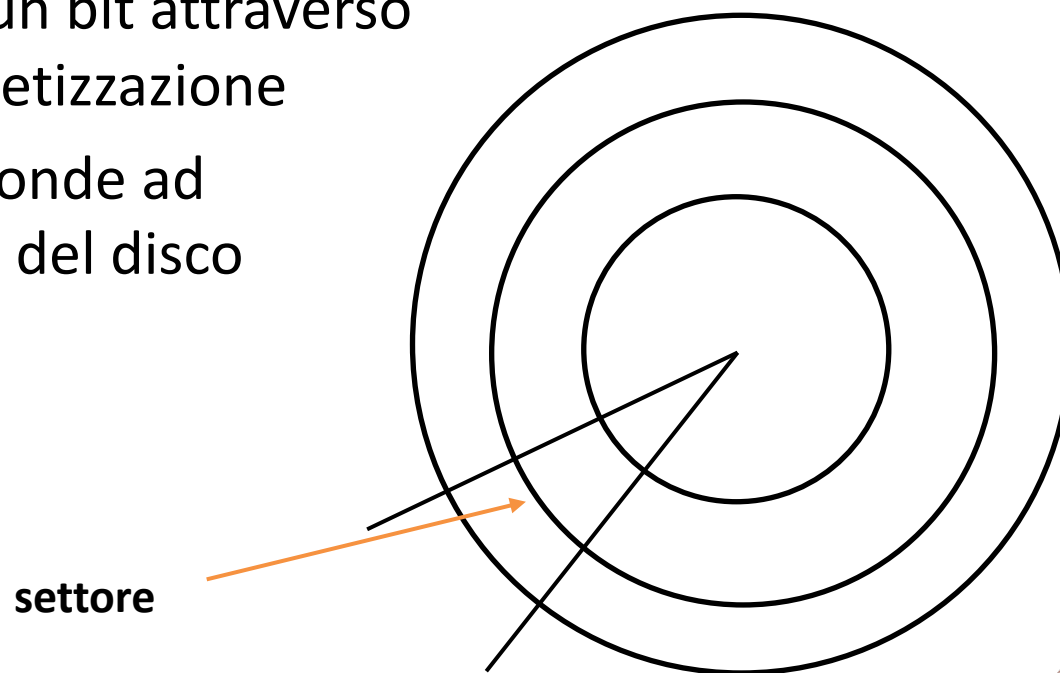


Unità di Input/Output: Memoria di Massa

Vediamo più nel dettaglio alcune tecnologie

I **dischi magnetici** includono gli hard-disk ed i vecchi floppy disk

- L'area del dispositivo è divisa in micro-zone
- Ogni zona codifica un bit attraverso il suo stato di magnetizzazione
- Ogni zona è corrisponde ad un settore circolare del disco



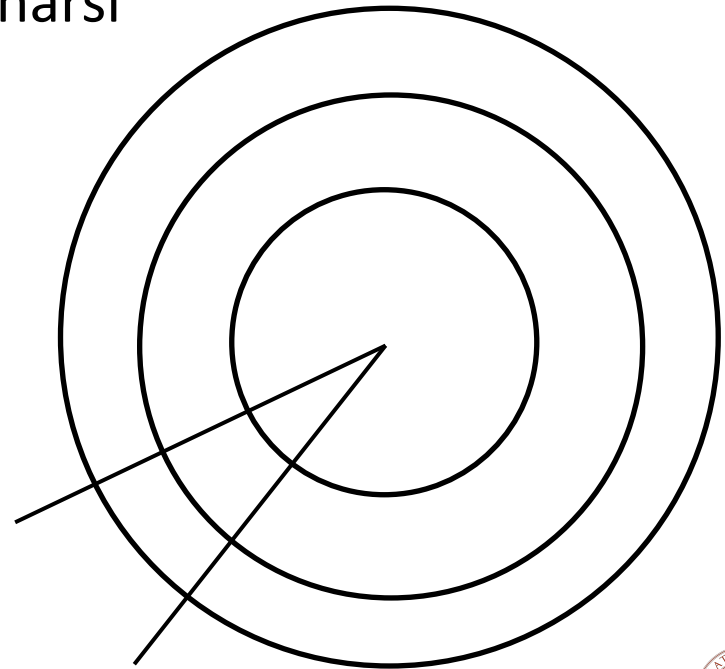
Unità di Input/Output: Memoria di Massa

Vediamo più nel dettaglio alcune tecnologie

Nei dischi magnetici:

- La lettura/scrittura è effettuata da una testina
- La testina può avvicinarsi/allontanarsi rispetto al centro (seek)
- Il disco è in rotazione (e.g. 7200 giri per minuto)
- Quindi il tempo (medio) per un trasferimento dati è:

$$T = T_{\text{seek}} + \frac{1}{2} T_{\text{rotazione}} + T_{r/w} * N_{\text{dati}}$$

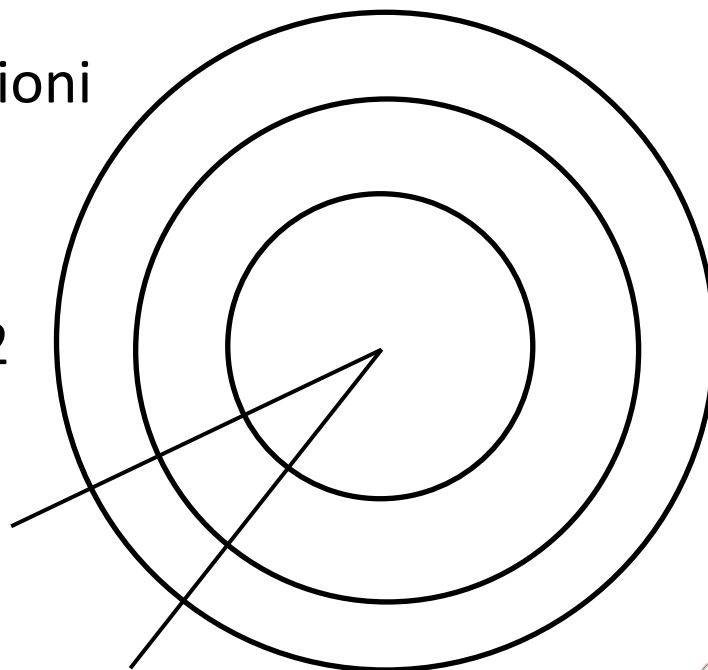


Unità di Input/Output: Memoria di Massa

Vediamo più nel dettaglio alcune tecnologie

Nei dischi magnetici:

- Il tempo di lettura scrittura dipende anche dal tempo di seek e dalla velocità di rotazione
- Per questo si misurano le prestazioni in termini di transfer rate
- Tipicamente: 100-200MB/s
 - Per confronto, una RAM DDR2 arriva a 4300 MB/s
- La capacità va da 1 a 16 TB
 - Per confronto, la RAM va da 512MB a 64GB



Unità di Input/Output: Memoria di Massa

Vediamo più nel dettaglio alcune tecnologie

Le **memorie flash** includono le chiavette USB e i SSD

- I bit sono realizzati mediante circuiti integrati (transistor)
- Non hanno parti meccaniche
- Transfer rate asimmetrica:
 - SSD, lettura: 500 MB/s
 - SSD, scrittura: 200 MB/s
 - Chiavette USB spesso molto più lente
- Capacità tipica: da 64GB a 8TB



Unità di Input/Output: Memoria di Massa

Vediamo più nel dettaglio alcune tecnologie

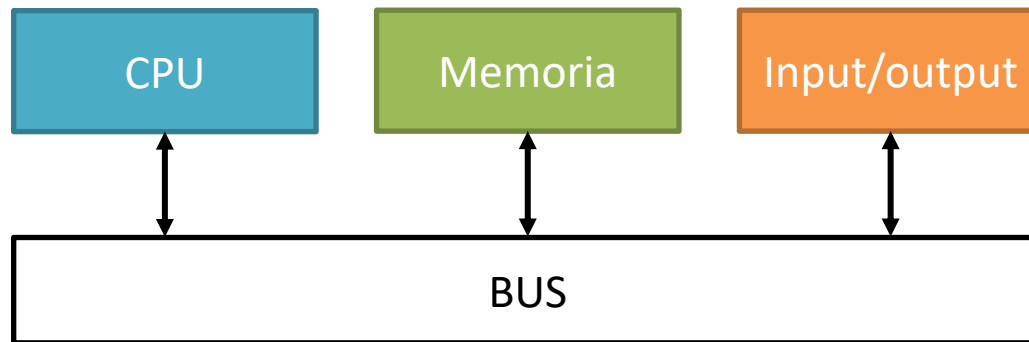
I **dischi ottici** includono CD e DVD

- Organizzati in modo simile ai dischi magnetici
- I bit sono memorizzati alterando la riflettività di micro-aree
- Transfer rate
 - 1X: 150 KB/s per i CD (multipli fino a 52X)
 - 1X: 1321 MB/s per i DVV (multipli fino a 21X)
- Capacità:
 - 640 MB per i CD
 - 4.7-8.5 GB per i DVD



Bus di Sistema

Il **bus di sistema** connette le unità funzionali

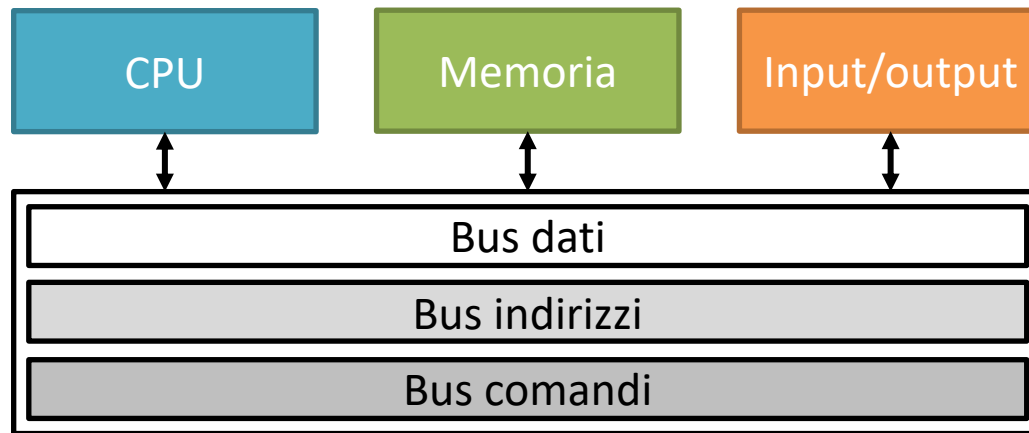


Durante una trasmissione dati collega due unità alla volta:

- Una unità trasmette
- L'altra unità riceve

Bus di Sistema

Il **bus di sistema** connette le unità funzionali



È in realtà costituito da un insieme di linee diverse:

- Bus dati
- Bus indirizzi (selezione di dispositivi e celle di memoria)
- Bus comandi (gestione del trasferimento)



Bus di Sistema

Il bus dati:

- È bidirezionale: trasmette sia verso che dalla memoria (o dispositivo di I/O)

Il bus indirizzi:

- Nella macchina di Von Neumann è unidirezionale
- È sempre la CPU a trasmettere indirizzi alla memoria (o dispositivo di I/O)

Il bus dei comandi:

- Trasmette i segnali per coordinare il trasferimento
- E.g. read/write/ack (ricevuto)



Oltre la Macchina di Von Neumann

I calcolatori moderni deviano dalla macchine di Von Neumann

Alcune delle differenze più significative:

- Bus multipli
 - E.g. bus memoria + bus I/O
- Presenza di processore/acceleratori dedicati:
 - Graphical Processing Unit (GPU)
 - Calcolo neuromorfico
- Esecuzioni in pipeline
 - L'esecuzione è divisa in stadi. Ad ogni ciclo di clock istruzioni diverse occupano ciascuno degli stadi
- Architetture parallele
 - E.g. core multipli, sistemi multiprocessore





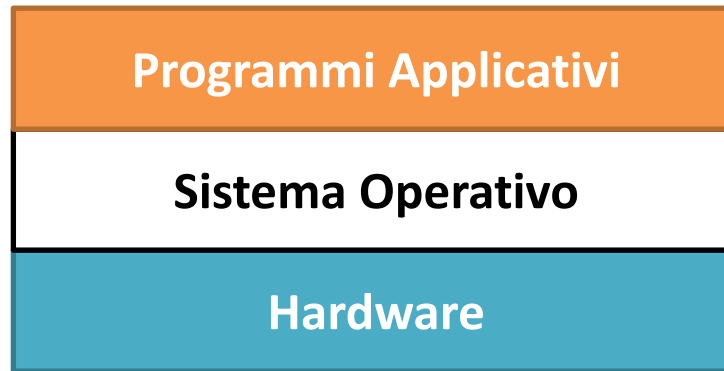
ALMA MATER STUDIORUM
UNIVERSITÀ DI BOLOGNA

Fondamenti di Informatica T

Sistema Operativo

Sistema Operativo

Il sistema operativo è un insieme di programmi e librerie:

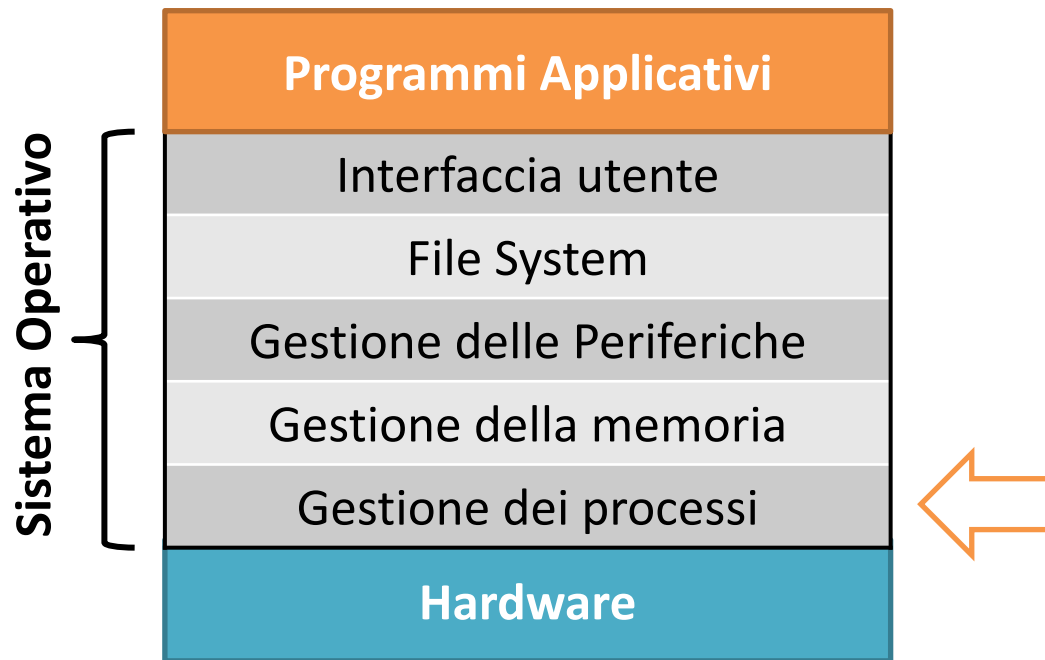


- Fa da interfaccia tra l'hardware ed i programmi applicativi
- ...Realizzando a tutti gli effetti una macchina virtuale
- Ci sono innumerevoli tipi di macchina fisica...
- ...Ma un numero relativamente ridotto di SO
 - E.g. Windows, OSX, Linux, iOS, Android...



Funzioni del Sistema Operativo

Il sistema operativo ha diverse funzioni:



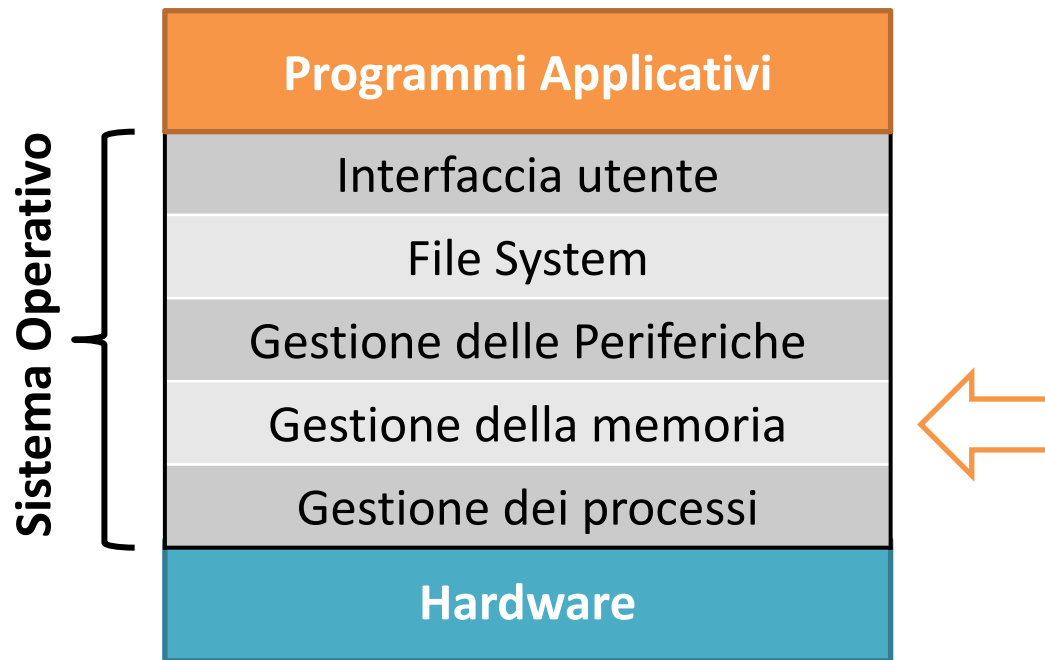
Gestione dei processi

- Gestisce l'unità di elaborazione (CPU)
- Permette (e.g.) l'esecuzione dei programmi



Funzioni del Sistema Operativo

Il sistema operativo ha diverse funzioni:



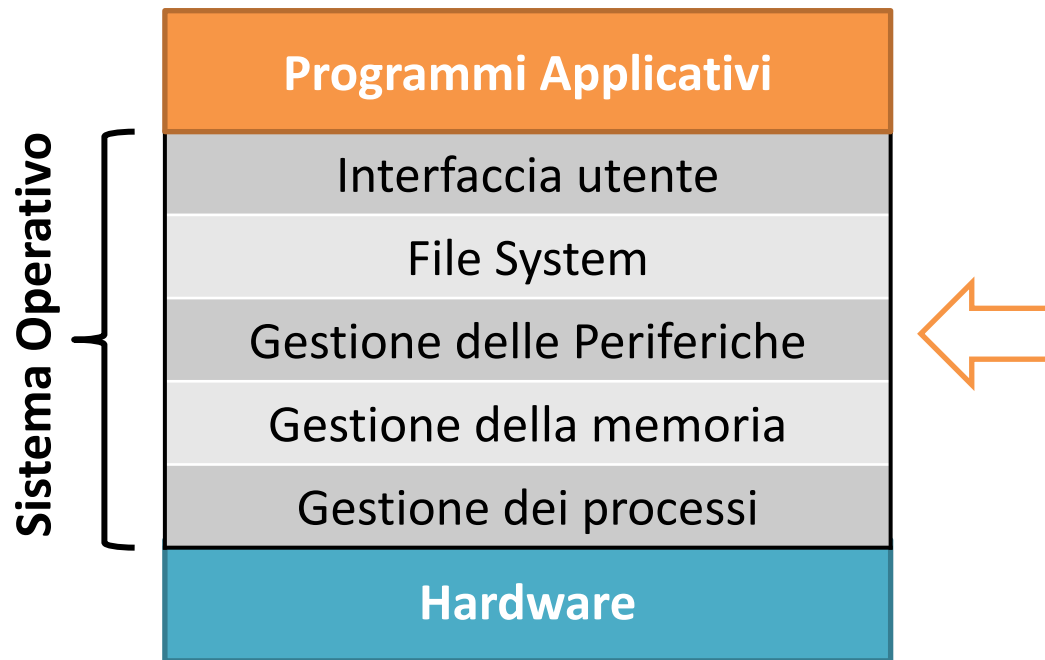
Gestione della memoria

- Gestisce la memoria centrale
- Permette (e.g.) l'allocazione di memoria ai programmi



Funzioni del Sistema Operativo

Il sistema operativo ha diverse funzioni:



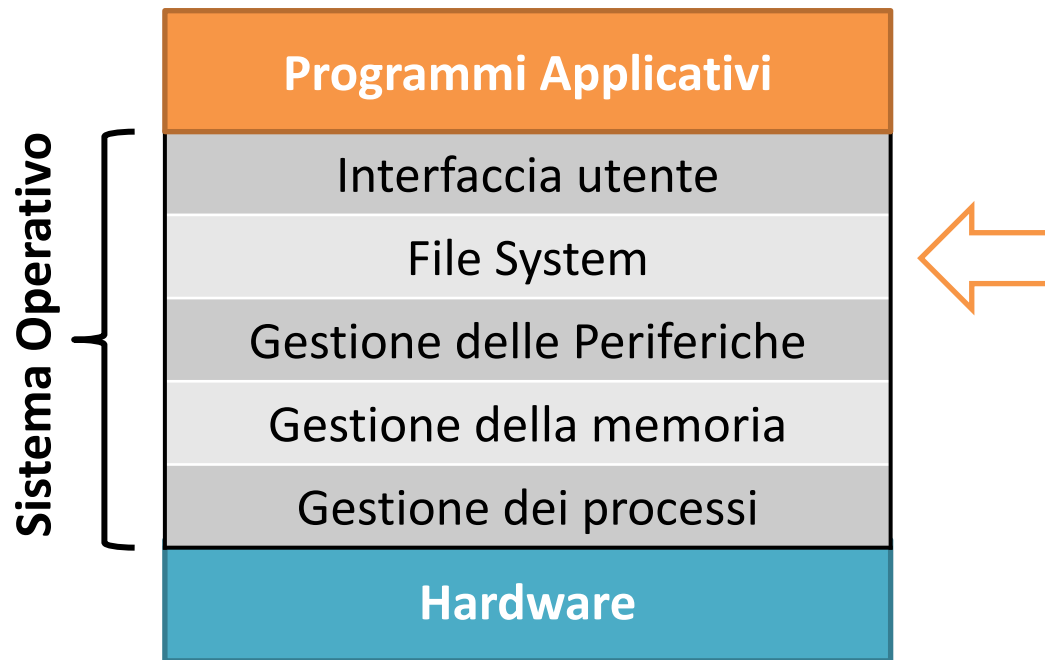
Gestione delle periferiche

- Gestisce le unità di input/output
- Permette (e.g.) ai programmi di accedervi



Funzioni del Sistema Operativo

Il sistema operativo ha diverse funzioni:



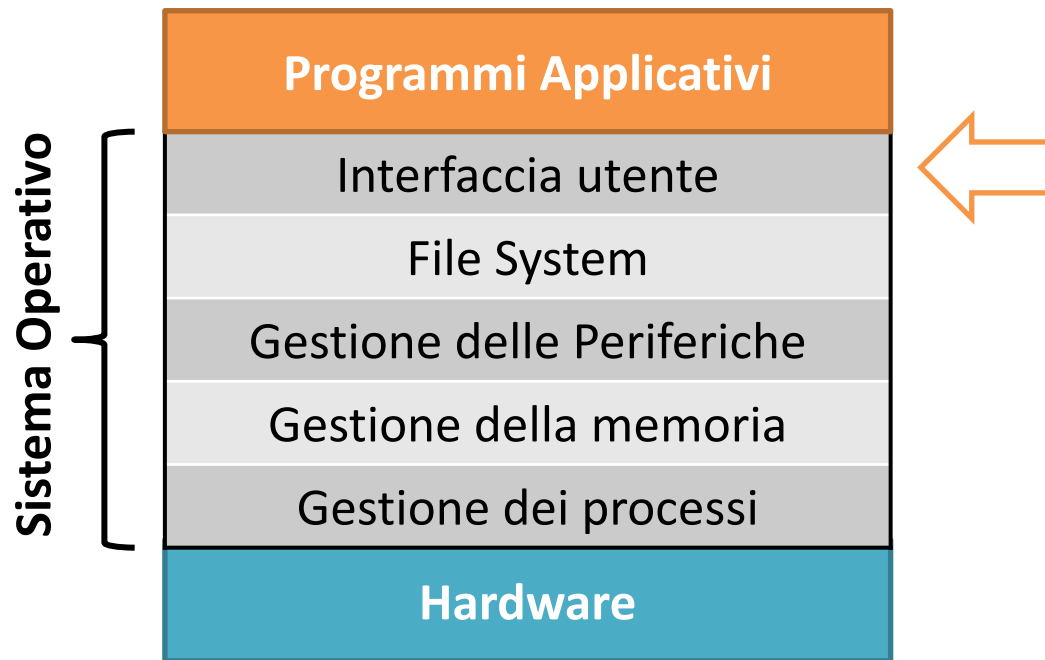
File System

- Gestisce dal punto di vista logico la memoria di massa
- Fornisce (e.g.) il concetto di file e directory (cartella)



Funzioni del Sistema Operativo

Il sistema operativo ha diverse funzioni:



Interfaccia utente

- Permette ad un utente di interagire con il SO
- Può essere grafica (GUI) o testuale (shell)



Funzioni del Sistema Operativo

Un esempio di interazione

Esecuzione del programma Unix "ls" da terminale:

ls

1. Recupero dell'input da tastiera (gestione I/O)
2. Interpretazione dell'input da tastiera (shell)
3. Recupero del codice di "ls" da disco (file system + gest. per.)
4. Caricamento in memoria di codice e dati (gest. memoria)
5. Avvio del programma (gest. processi)
6. Stampa dei risultati (shell + I/O)



Classificazione dei Sistemi Operativi

Si possono classificare i SO in base al **numero di utenti**:

- Mono-utente:
 - Un solo utente alla volta può usare il sistema
- Multi-utente:
 - Più utenti possono interagire con la macchina
 - Ogni utente ha l'illusione di avere un sistema dedicato

...Oppure in base al **numero di programmi in esecuzione**:

- Mono-programmato:
 - Un solo programma in esecuzione
- Multi-programmato:
 - Più programmi simultaneamente in esecuzione



Classificazione dei Sistemi Operativi

Esempi:

- Mono-utente, mono-programmato: MS DOS
- Mono-utente, multi-programmato: Win 95/98
- Multi-utente, multi-programmato:
 - Per PC: Win NT/2000, Win XP e seg., OS X, Linux/Unix
 - Per smartphone/tablet: iOS, Android...



Interfaccia Utente

L'interfaccia utente permette l'interazione con un utente

Ve ne sono due grandi categorie:

- Interfacce grafiche (Graphical User Interface – GUI)
 - Interazione per azionamento di elementi grafici
- Interfacce testuali (shell)
 - Interazione mediante comandi testuali

Spesso sono fornite entrambe!

- Windows: GUI + "prompt dei comandi"
- OS X: GUI + "terminale"
- Linux: GUI + "terminale"



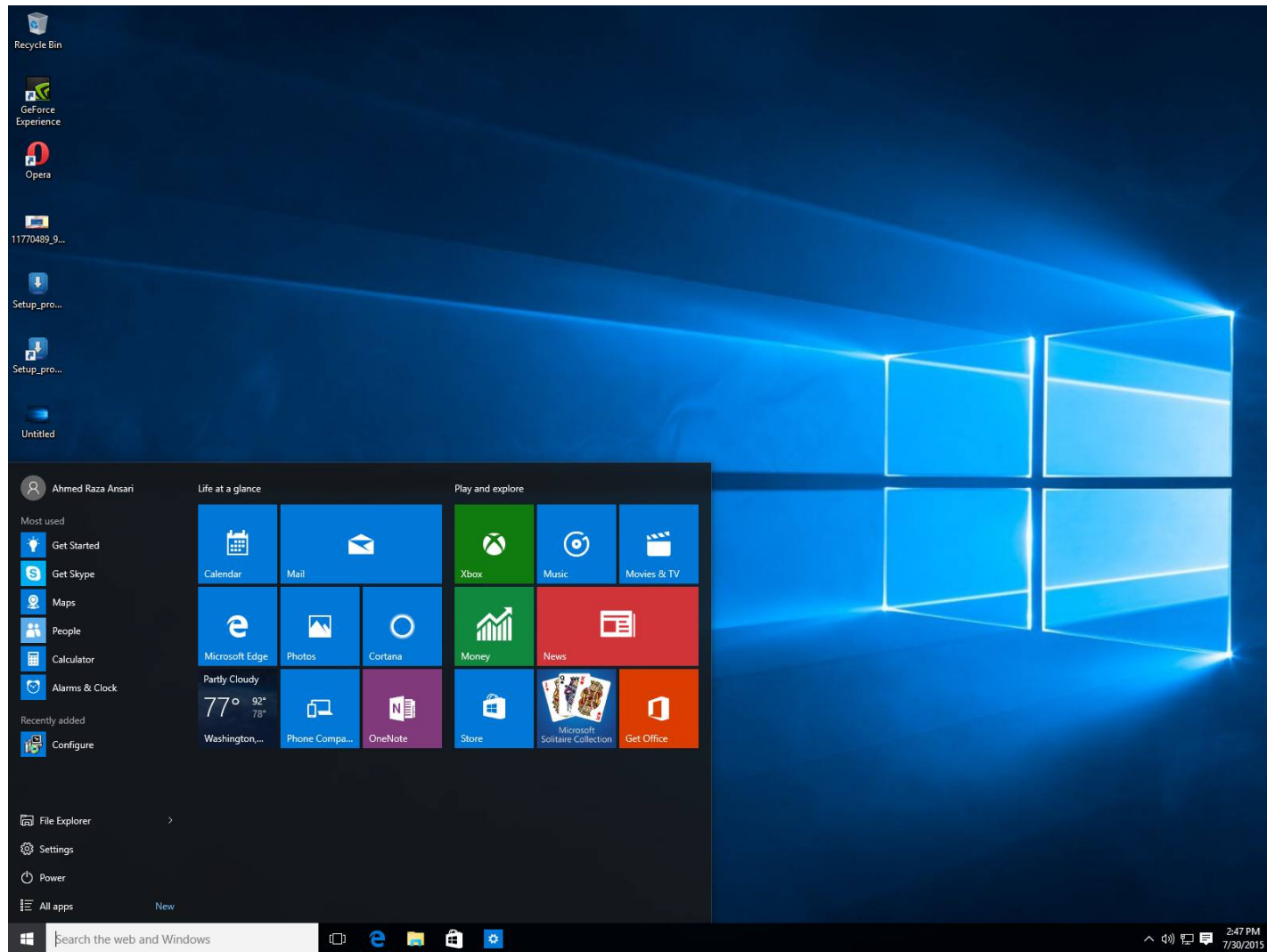
Interfaccia Utente: Interfacce Grafiche

Le interfacce grafiche:

- Fanno uso massiccio di metafore
 - E.g. desktop (scrivania), documenti, cartelle, cestino...
- Indicano le entità disponibili mediante elementi grafici
 - E.g. Pulsanti, icone, caselle
- Gestiscono l'interazione per azionamento dell'elemento grafico
 - E.g. click, click destro, touch
- Assegnano aree grafiche ai programmi
 - E.g. finestre



Interfaccia Utente: Interfacce Grafiche



Desktop di Windows 10



Interfaccia Utente: Interfacce Testuali

Le interfacce testuali:

- Derivano dai "terminali" dei supercomputer degli anni 60'



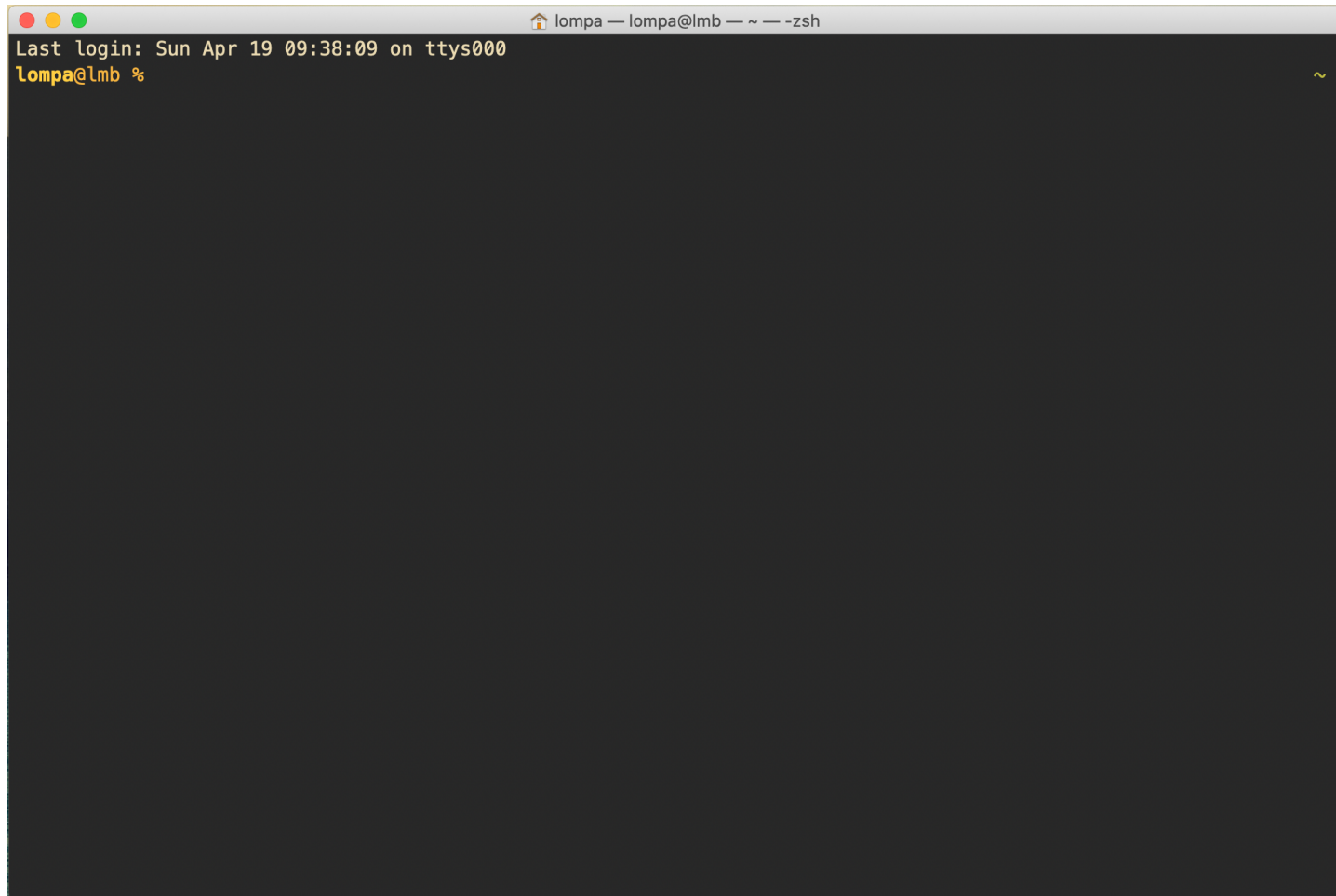
Interfaccia Utente: Interfacce Testuali

Le interfacce testuali:

- Derivano dai "terminali" dei supercomputer degli anni 60'
- Permettono l'interazione con il SO attraverso:
 - L'esecuzione di comandi e programmi
 - In alcuni casi: linguaggio di programmazione dedicato
- In questo modo permettono di:
 - Interagire con i file su disco
 - Interagire con le risorse di sistema
 - Preparare documenti, programmare, etc.
- I comandi sono digitati in corrispondenza di un "prompt"
 - Su Windows: "C:\>"
 - Su Linux/OSX: "nome@computer:~/"



Interfaccia Utente: Interfacce Grafiche



Emulatore di Terminale (con zshell) su OSX



Gestione dei Processi

Una delle funzione del SO è gestire i processi

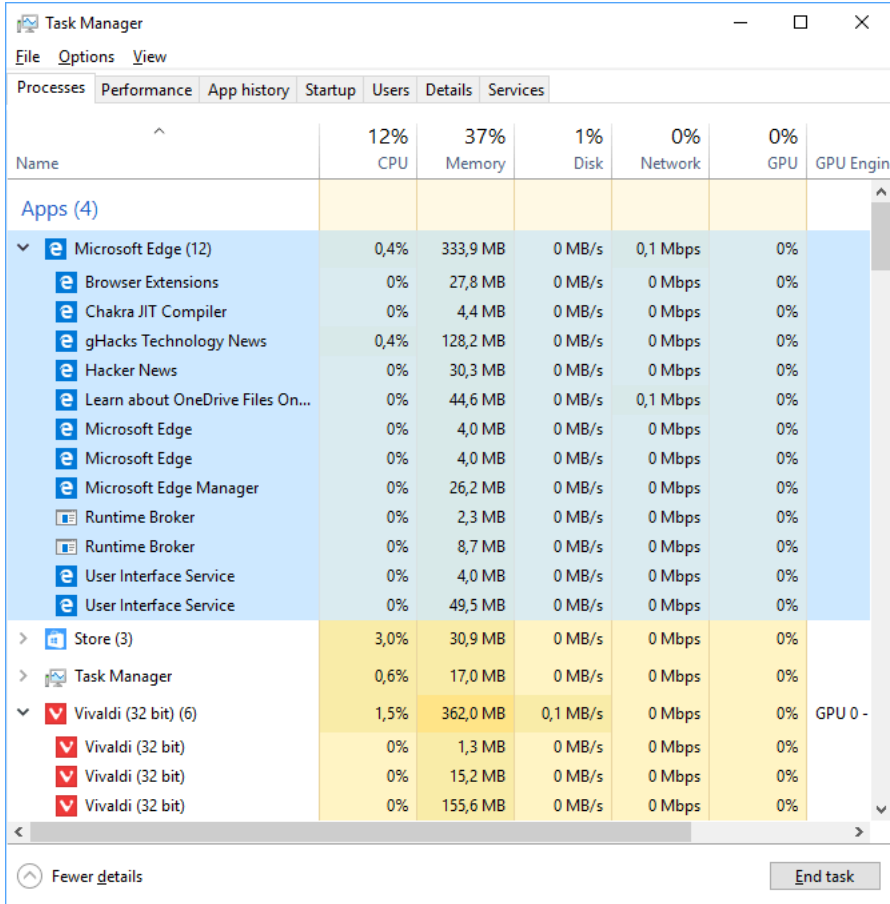
- Un processo è un programma in esecuzione
- La gestione dei processi include:
 - Avvio di un processo
 - Interruzione di un processo
 - Sospensione di un processo
 - Elenco dei processi in esecuzione
 - Condivisione della CPU tra processi
- Alcune funzionalità sono accessibili mediante interfaccia



Gestione dei Processi

Windows, GUI: Ctrl + Alt + Canc, poi "Task Manager"

- Fornisce l'elenco dei processi (non proprio...)
- Permette la loro interruzione forzata



Name	12% CPU	37% Memory	1% Disk	0% Network	0% GPU	GPU Engine
Apps (4)						
Microsoft Edge (12)	0,4%	333,9 MB	0 MB/s	0,1 Mbps	0%	
Browser Extensions	0%	27,8 MB	0 MB/s	0 Mbps	0%	
Chakra JIT Compiler	0%	4,4 MB	0 MB/s	0 Mbps	0%	
gHacks Technology News	0,4%	128,2 MB	0 MB/s	0 Mbps	0%	
Hacker News	0%	30,3 MB	0 MB/s	0 Mbps	0%	
Learn about OneDrive Files On...	0%	44,6 MB	0 MB/s	0,1 Mbps	0%	
Microsoft Edge	0%	4,0 MB	0 MB/s	0 Mbps	0%	
Microsoft Edge	0%	4,0 MB	0 MB/s	0 Mbps	0%	
Microsoft Edge Manager	0%	26,2 MB	0 MB/s	0 Mbps	0%	
Runtime Broker	0%	2,3 MB	0 MB/s	0 Mbps	0%	
Runtime Broker	0%	8,7 MB	0 MB/s	0 Mbps	0%	
User Interface Service	0%	4,0 MB	0 MB/s	0 Mbps	0%	
User Interface Service	0%	49,5 MB	0 MB/s	0 Mbps	0%	
Store (3)						
Store (3)	3,0%	30,9 MB	0 MB/s	0 Mbps	0%	
Task Manager						
Task Manager	0,6%	17,0 MB	0 MB/s	0 Mbps	0%	
Vivaldi (32 bit) (6)						
Vivaldi (32 bit)	1,5%	362,0 MB	0,1 MB/s	0 Mbps	0%	GPU 0 -
Vivaldi (32 bit)	0%	1,3 MB	0 MB/s	0 Mbps	0%	
Vivaldi (32 bit)	0%	15,2 MB	0 MB/s	0 Mbps	0%	
Vivaldi (32 bit)	0%	155,6 MB	0 MB/s	0 Mbps	0%	



Gestione dei Processi

OSX, GUI: Applicazione "Activity Monitor"

- Stesse funzionalità di Win

Activity Monitor (All Processes)

CPU

Memory

Energy

Disk

Network

Q

Search

Process Name	% CPU	CPU Time	Threads	Idle Wake-Ups	% GPU	GPU Time	PID	User
com.docker.hyperkit	10.1	11:49.55	19	229	0.0	0.00	823	lompa
WindowServer	9.6	7:01.99	10	33	0.0	7:58.87	257	_windowserver
Activity Monitor	6.5	4.29	5	3	0.0	0.00	2698	lompa
kernel_task	4.9	4:16.71	255	561	0.0	0.00	0	root
hidd	4.8	1:38.79	7	1	0.0	0.00	179	_hidd
screencapture	4.2	0.27	4	0	0.0	0.00	2701	lompa
AppleUserHIDDrivers	2.3	6.61	3	0	0.0	0.00	2310	_driverkit
Magnet	1.6	8.10	3	2	0.0	0.00	729	lompa
sysmond	1.0	1.00	5	0	0.0	0.00	357	root
Dropbox	0.8	2:33.68	142	28	0.0	0.00	727	lompa
launchservicesd	0.6	15.97	4	0	0.0	0.00	157	root
Backup and sync from Google	0.4	1:34.03	43	5	0.0	0.00	728	lompa
Dropbox Web Helper	0.4	6.52	10	62	0.0	54.60	828	lompa
launchd	0.4	18.42	4	0	0.0	0.00	1	root
suggestd	0.3	42.84	11	1	0.0	0.00	588	lompa
Screenshot	0.3	0.19	6	1	0.0	0.00	2702	lompa
Dropbox Web Helper	0.3	16.93	18	6	0.0	0.00	835	lompa
iconservicesagent	0.2	6.46	4	0	0.0	0.03	594	lompa
vpkit-bridge	0.2	16.16	16	29	0.0	0.00	813	lompa
com.apple.geod	0.1	2.69	5	0	0.0	0.00	601	lompa
powerd	0.1	6.85	3	0	0.0	0.00	134	root
Evernote Web Clipper (Safari)	0.1	1.16	8	1	0.0	0.00	2683	lompa
CalendarAgent	0.1	26.33	7	1	0.0	0.00	567	lompa
Safari	0.1	56.83	11	0	0.0	0.09	541	lompa
nextcloud	0.1	7.44	7	1	0.0	0.00	698	lompa
Spotlight	0.1	8.03	7	1	0.0	0.00	696	lompa
Alfred	0.1	25.05	3	7	0.0	0.00	730	lompa

System:2.17%

User:2.19%

Idle:95.64%

CPU LOAD

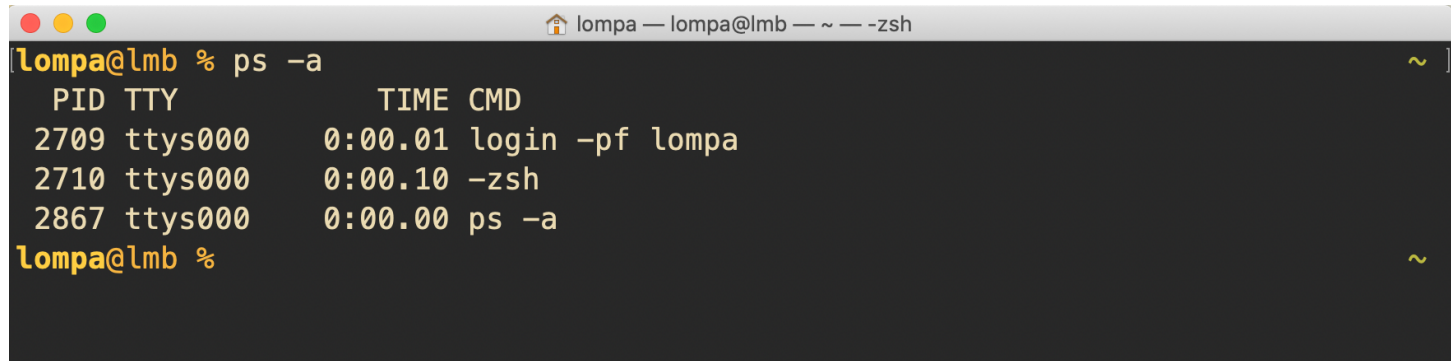
Threads:1,900

Processes:434



Gestione dei Processi

Linux/OSX, shell: comando "ps"



```
lomba@lmb % ps -a
  PID TTY          TIME CMD
 2709 ttys000    0:00.01 login -pf lomba
 2710 ttys000    0:00.10 -zsh
 2867 ttys000    0:00.00 ps -a
lomba@lmb %
```

- "ps" fornisce l'elenco dei processi
- Uso di argomenti da riga di comando per scegliere cosa visualizzare
- Altri comandi per interrompere, sospendere, etc.



Gestione della Memoria

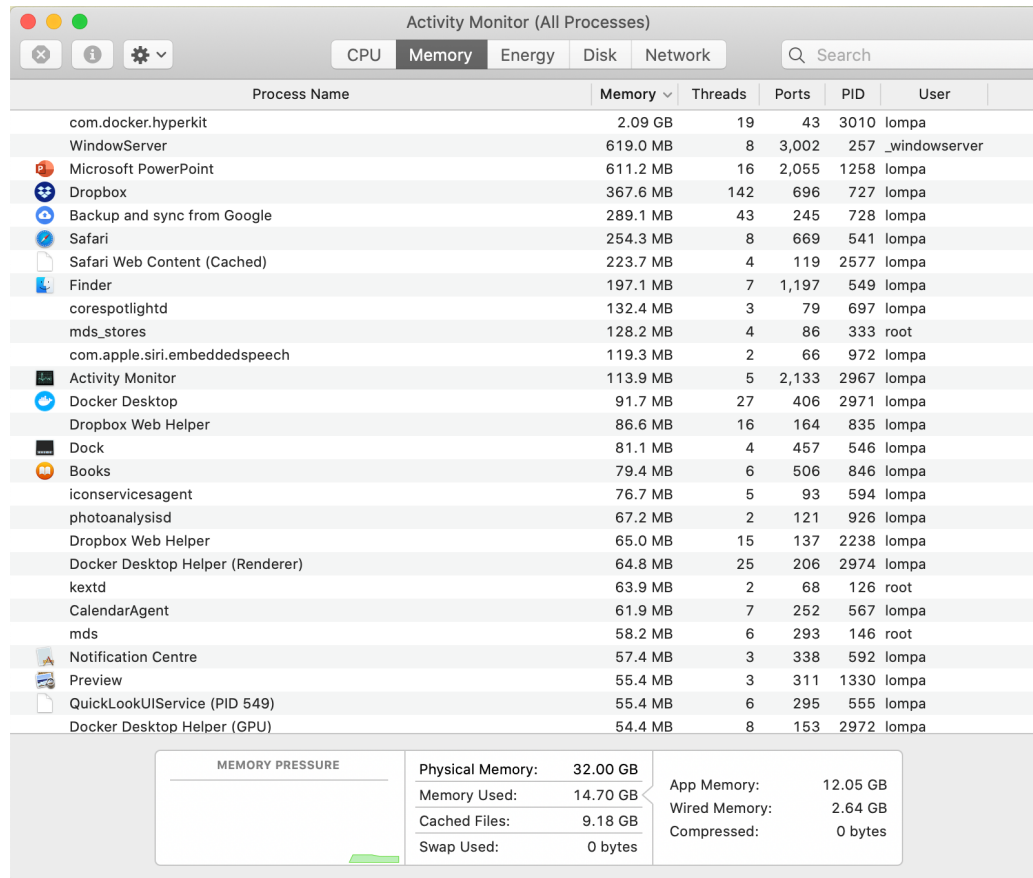
Il sistema operativo ha il compito di gestire la memoria centrale

- Alloca memoria ai programmi al momento dell'avvio
 - Sui sistemi moderni ogni programma ha accesso ad un'area di memoria distinta
 - È formata da una serie di "segmenti"
 - In caso di tentato accesso al suo esterno si ottiene un "segmentation fault"
- In caso di utilizzo di memoria eccessivo:
 - Il SO sposta su disco le aree destinate ai processi meno attivi
 - Si parla di "swap memory"
 - Si tratta di una operazione molto inefficiente



Gestione della Memoria

È possibile monitorare il consumo di memoria via GUI:



Activity Monitor (utilizzo di memoria) su OSX



Gestione delle Periferiche

Il sistema ha il compito di gestire le periferiche di I/O

- Fornisce ai programmi delle funzioni di accesso standardizzate
 - Si usa il termine **Application Programming Interface**...
 - ...Per ogni interfaccia SW standardizzata
- L'implementazione di tali funzioni dipende dalla periferica
 - Ci sono moltissimi tipi di periferica disponibili
 - Soluzione: per ogni periferica, un driver di dispositivo
 - Un "driver" è un programma/libreria che implementa l'API stabilita dal sistema operativo
 - I driver sono tipicamente sviluppati dal produttore della periferica



File System

Il File System è un componente del sistema operativo

- I driver rendono disponibile la memoria di massa come una unica sequenza di bit...
- ...Il File System vi permette l'accesso mediante il **concetto di file**

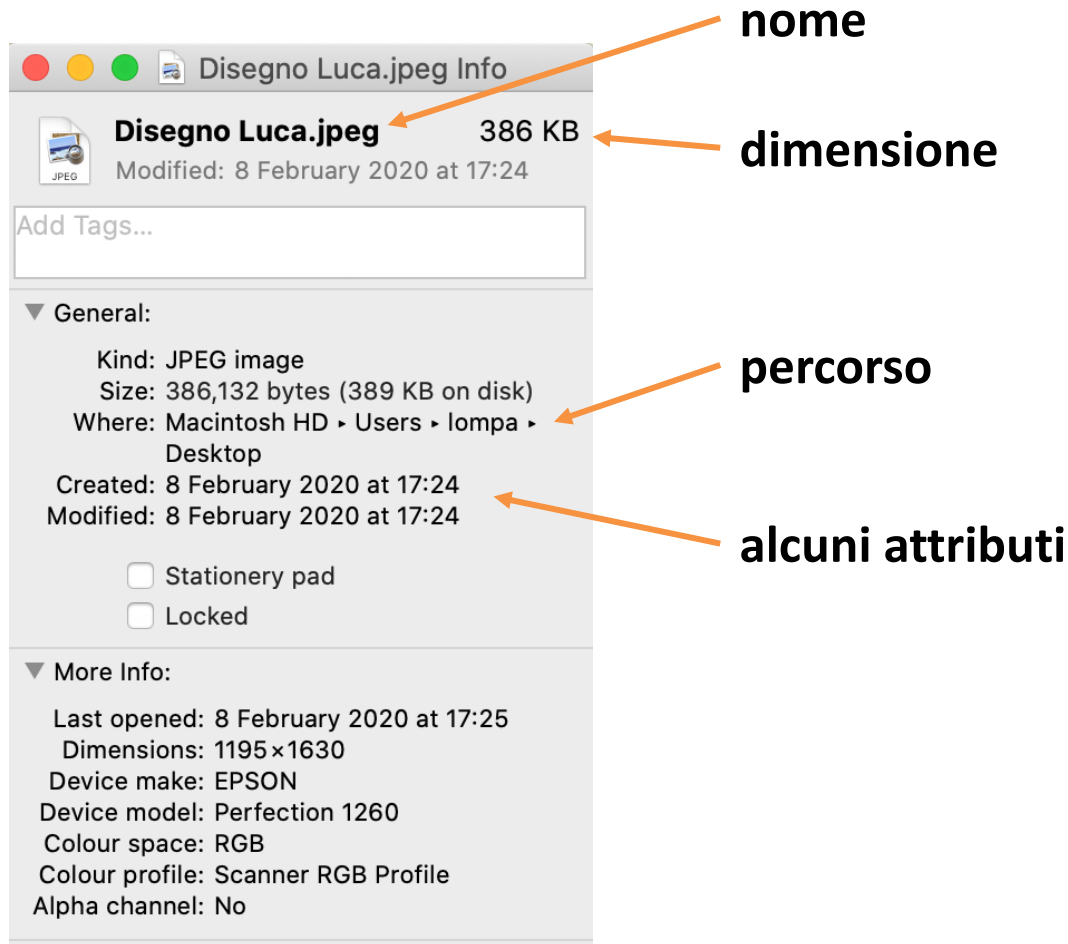
Un file è l'unità logica di memorizzazione delle informazioni:

- È un sequenza di byte
- Ha un nome
- ...Un percorso
- ...Una dimensione
- ...E altri attributi (e.g. data di creazione e modifica)



File System

Un esempio:



File System

Il **nome di un file** è nella forma: "nome.estensione"

Qualche esempio:

- tesi.docx
- budget.xlsx
- README.txt
- main.c

L'estensione è facoltativa:

- Indica al SO il tipo del contenuto nel file
- ...e di conseguenza quale programma si può usare per aprirlo
- Se rimossa o cambiata, non altera il contenuto



File System

Il file system:

- Fornisce l'astrazione di file
- Permette di operare sui file:
 - E.g. creazione, lettura, scrittura, spostamento...
- Fornisce una **organizzazione** per la memoria di massa

Si chiama **formattazione** l'operazione con cui il File System prepara un dispositivo di memoria di massa

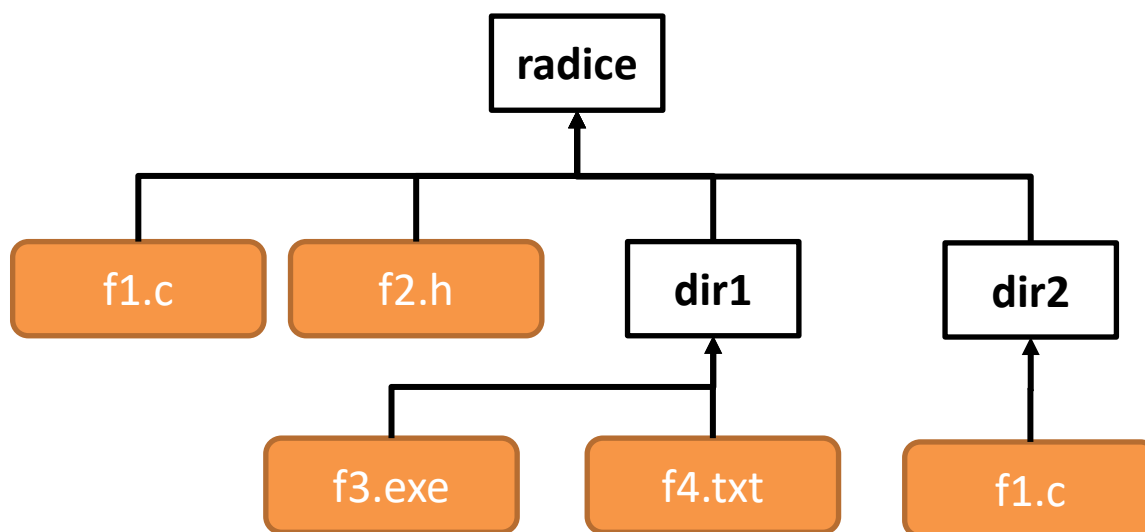
Si adotta tipicamente una organizzazione gerarchica

- Si basa sul concetto di **directory** (cartella)
- Una directory è un contenitore di file



File System: File, Directory, Percorsi

Una organizzazione gerarchica forma sempre un albero:



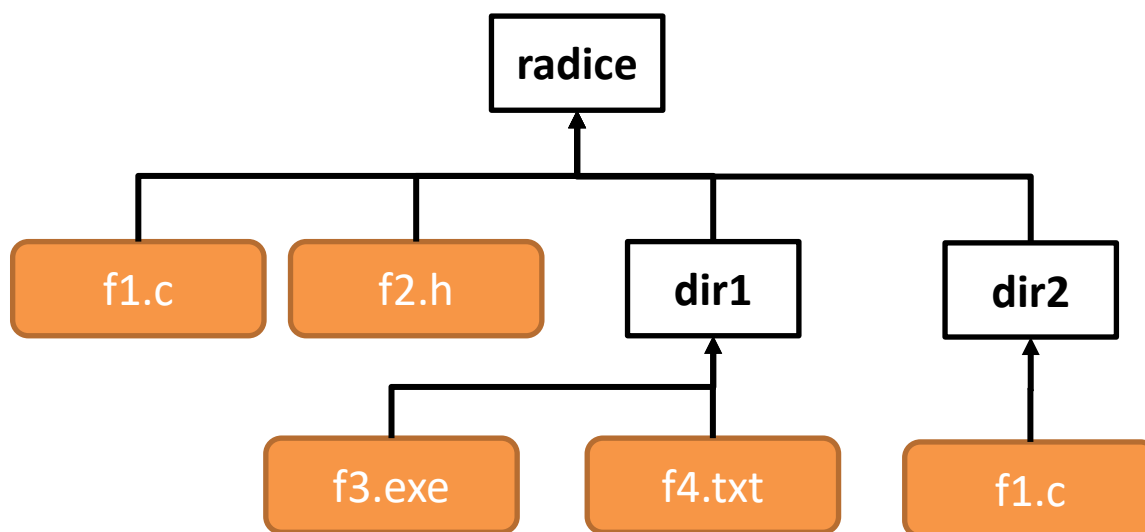
Le frecce denotano appartenenza

- Ogni file (in arancione) è sempre una foglia dell'albero
- Una cartella può essere una foglia o contenere altri file



File System: File, Directory, Percorsi

Una organizzazione gerarchica forma sempre un albero:



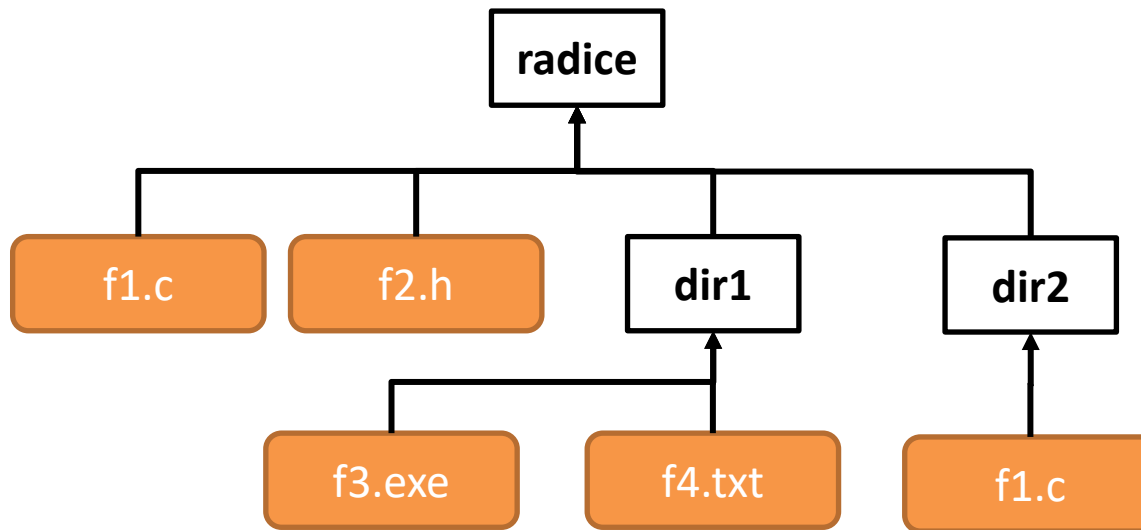
Esiste sempre una **directory radice**

- Su Windows: una cartella per "disco", e.g. "C:\", "D:\"...
- Su Linux/OSX: una sola radice, chiamata semplicemente "/"



File System: File, Directory, Percorsi

Ogni file è identificato dal suo **percorso assoluto**

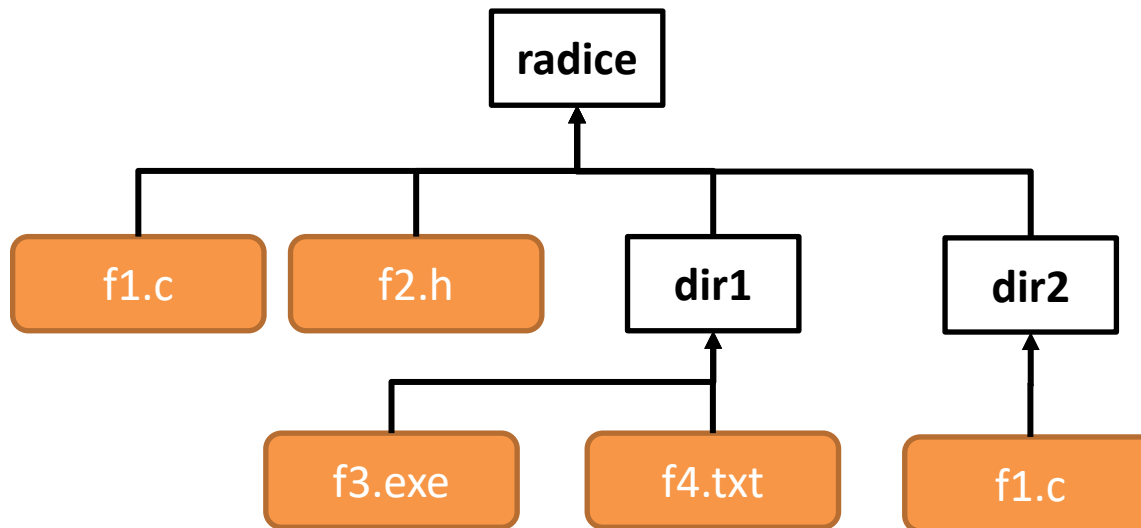


- Questo contiene il nome del file...
-preceduto dalla sequenza di directory...
- ...per raggiungerlo dalla radice



File System: File, Directory, Percorsi

Ogni file è identificato dal suo **percorso assoluto**



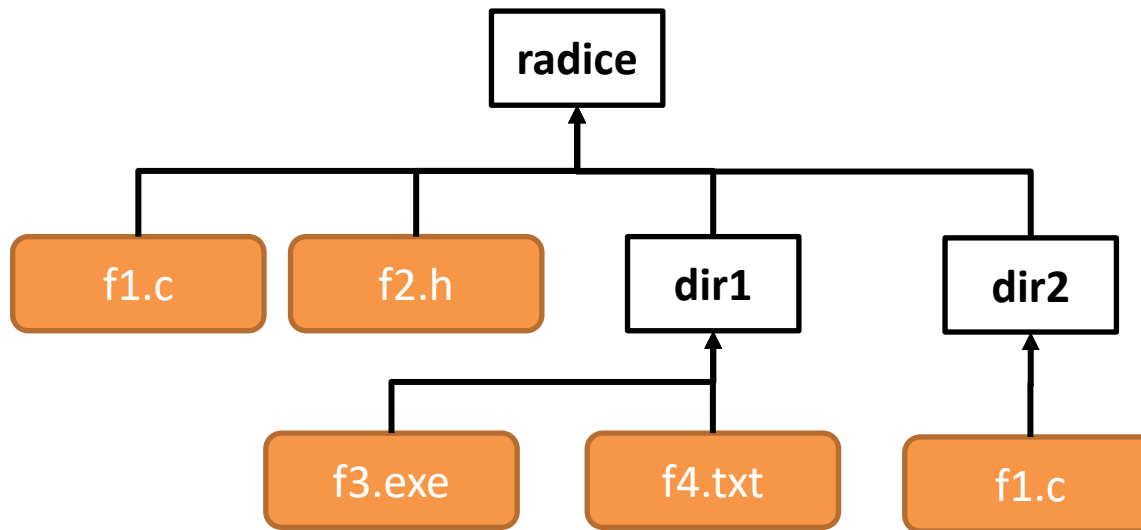
Esempi di percorso assoluto (su Win "\" separa le directory):

- C:\f1.c
- C:\dir1\f4.txt



File System: File, Directory, Percorsi

Ogni file è identificato dal suo **percorso assoluto**



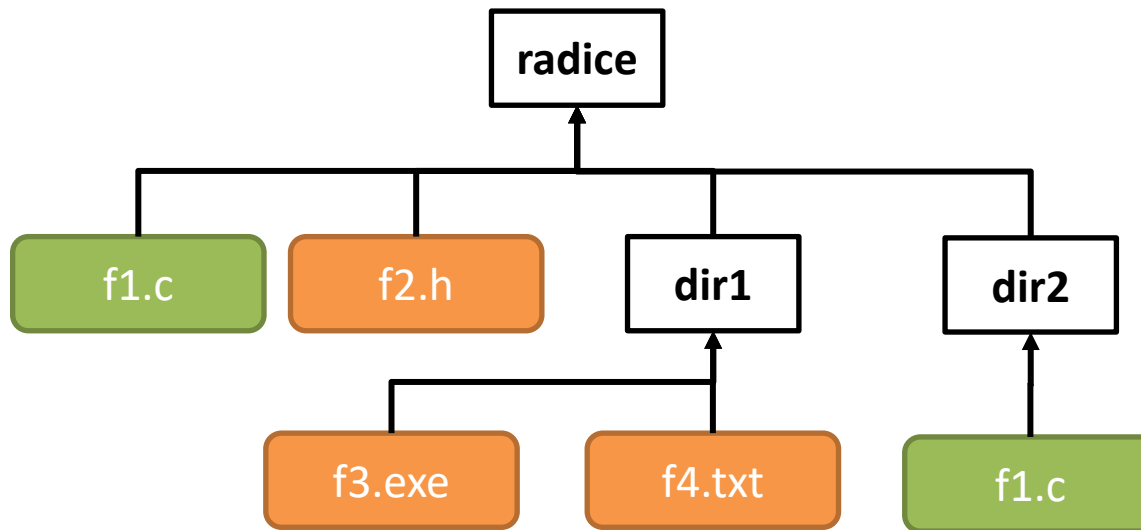
Esempi di percorso assoluto (su Linux/OSX "/" separa le directory):

- /f1.c
- /dir1/f4.txt



File System: File, Directory, Percorsi

Ogni file è identificato dal suo **percorso assoluto**

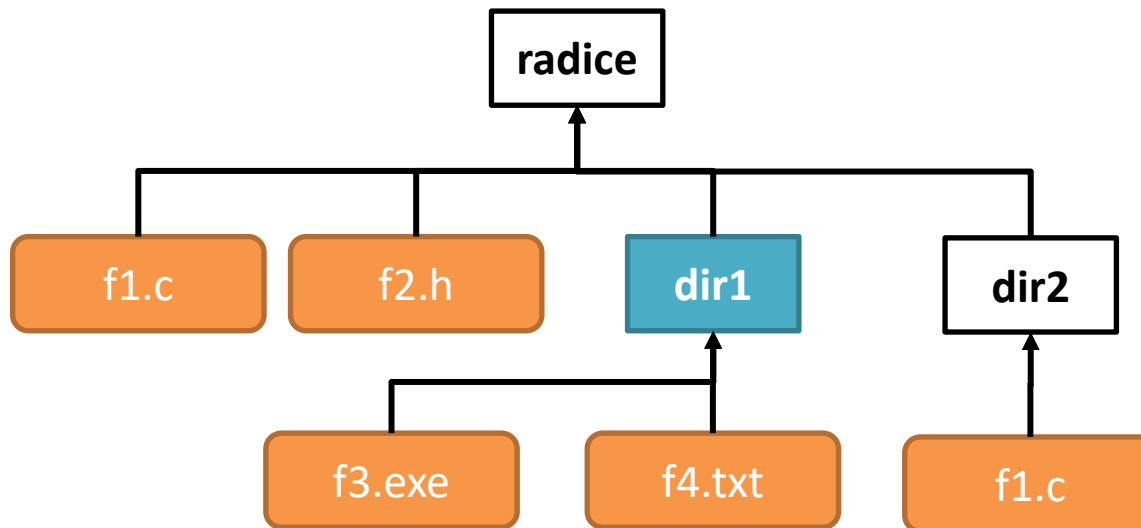


- Conseguenza: due file possono avere lo stesso nome...
- Ma solo se sono in **directory diverse**
- E.g. `"/f1.c"` e `"/dir2/f1.c"`



File System: File, Directory, Percorsi

Un programma è sempre associato ad una **directory corrente**



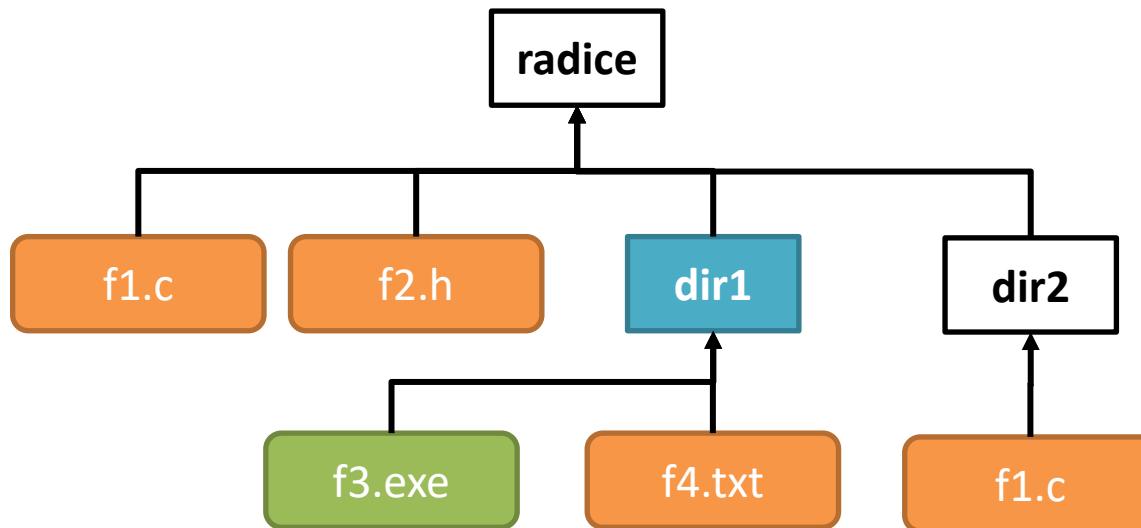
- È possibile riferirsi ad un file indicandone il percorso...
- ...rispetto alla directory corrente

In questo caso si parla di **percorso relativo**



File System: File, Directory, Percorsi

Un programma è sempre associato ad una **directory corrente**



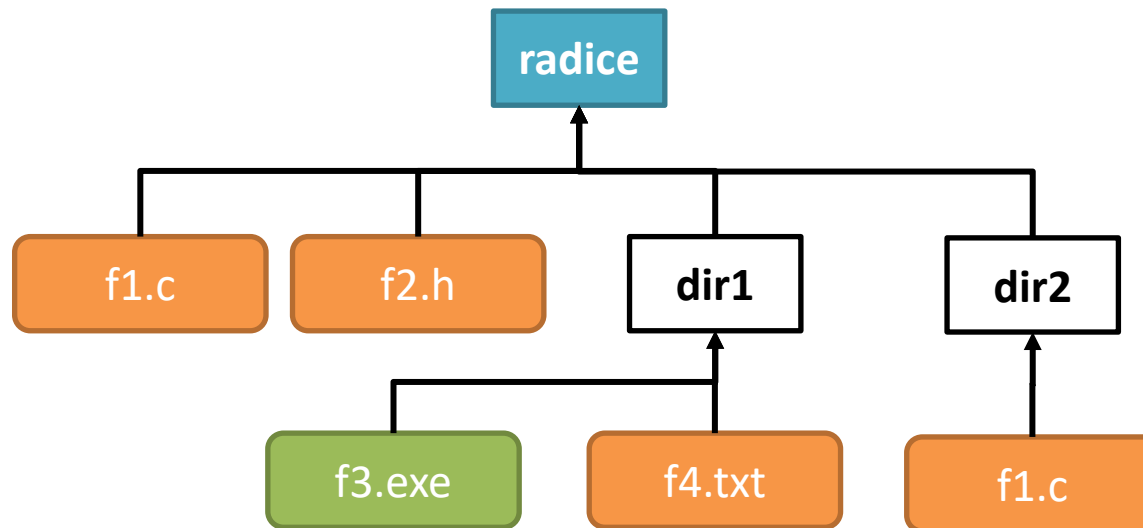
Esempio, per il file "f3.exe", da "dir1"

- Percorso **assoluto**: "C:\dir1\f3.exe" ("/dir1/f3.exe")
- Percorso **relativo**: "f3.exe"



File System: File, Directory, Percorsi

Un programma è sempre associato ad una **directory corrente**



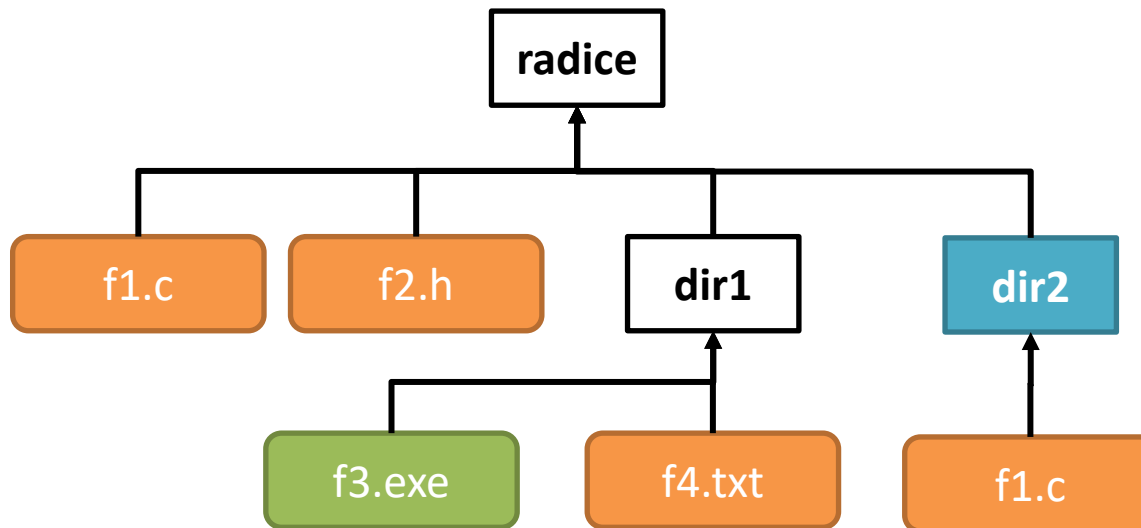
Esempio, per il file "f3.exe", dalla radice "dir1"

- Percorso **assoluto**: "C:\dir1\f3.exe" (" /dir1/f3.exe")
- Percorso **relativo**: "dir1\f3.exe" ("dir1/f3.exe")



File System: File, Directory, Percorsi

Un programma è sempre associato ad una **directory corrente**



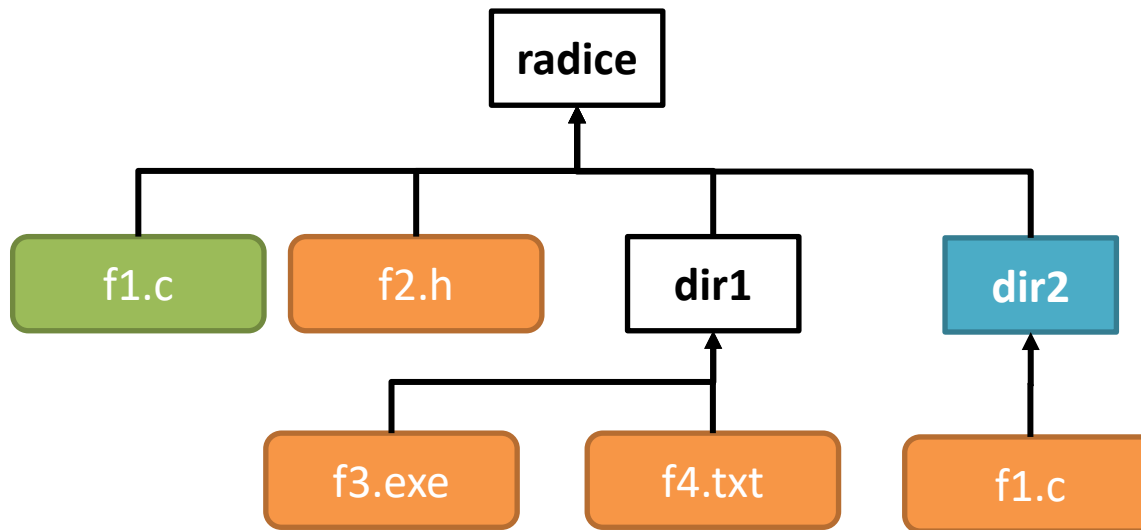
Si indica la directory "padre" con la notazione ".."

- "f3.exe" da "dir2": "..\dir1\f3.exe" ("../dir1/f3.exe")



File System: File, Directory, Percorsi

Un programma è sempre associato ad una **directory corrente**



Si indica la directory "padre" con la notazione ".."

- "f3.exe" da "dir2": "..\dir1\f3.exe" ("../dir1/f3.exe")
- "f1.c" sotto radice, da "dir2": "..\f1.c" ("../f1.c")



File System: File, Directory, Percorsi

Dall'interprete dei comandi:

Per conoscere il contenuto della directory corrente si usa:

- "dir" sotto Windows
- "ls" sotto Linux/OS X

Per conoscere la directory corrente si usa:

- "cd" sotto Windows
- "pwd" sotto Linux/OS X



File System: File, Directory, Percorsi

Dall'interprete dei comandi:

Per cambiare la cartella corrente si usa:

- "`cd <percorso>`" sia sotto Windows che sotto Linux/OS X
- Il percorso specificato può essere sia assoluto che relativo

Per avviare un file eseguibile nella directory corrente si usa

- "<nome del file>" sotto Windows
- "./<nome del file>" sotto Linux/OS X





ALMA MATER STUDIORUM
UNIVERSITÀ DI BOLOGNA

Fondamenti di Informatica T

Due Parole per Concludere

Due Parole per Concludere

“Se il settore dell’automobile si fosse sviluppato come l’industria informatica, oggi avremo veicoli che costano 25 dollari e fanno 500 Km con un litro”.

(Bill Gates)



Due Parole per Concludere

“Se le auto funzionassero come i software, si bloccherebbero due volte al giorno senza motivo e l’unica soluzione sarebbe reinstallare il motore”

(Dirigente General Motors)





ALMA MATER STUDIORUM
UNIVERSITÀ DI BOLOGNA

Fondamenti di Informatica T

Andrea Acquaviva <andrea.acquaviva@unibo.it>

Michele Lombardi <michele.lombardi2@unibo.it>

Andrea Borghesi <andrea.borghesi3@unibo.it>

Giuseppe Tagliavini <giuseppe.tagliavini@unibo.it>

Allegra De Filippo <allegra.defilippo@unibo.it>