

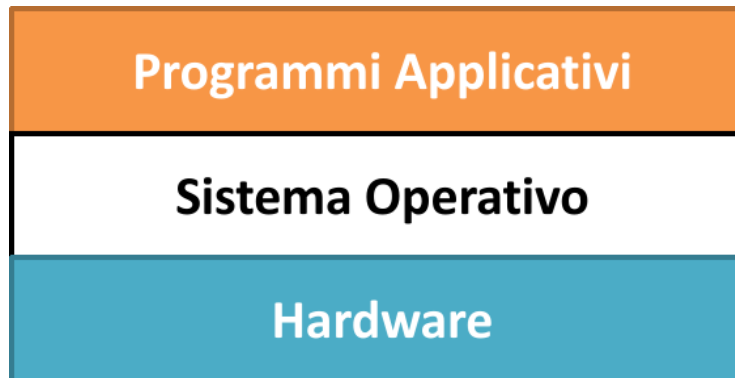
# Sistema Operativo

---



# Il 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**

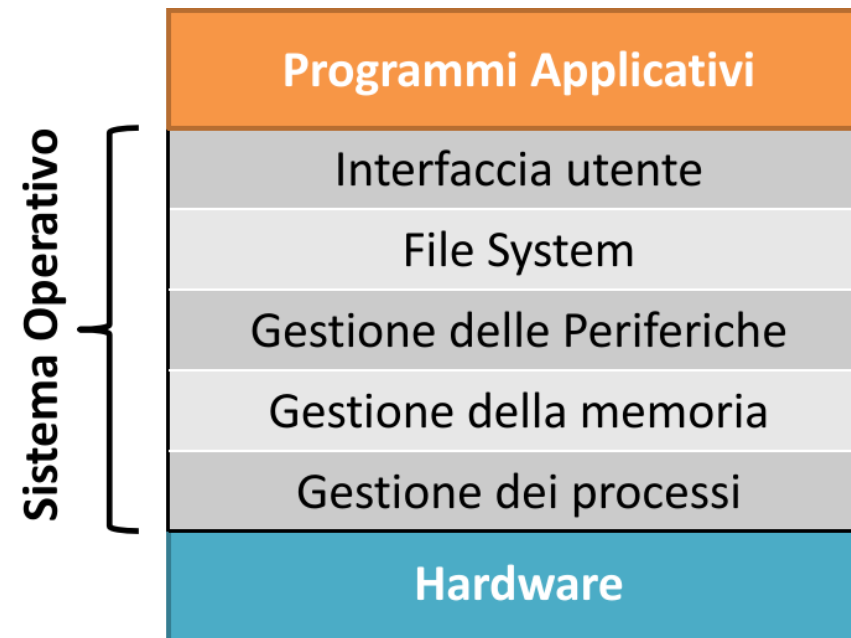
## In altre parole:

- Ci sono innumerevoli tipi di macchina fisica...
- ...Ma un numero relativamente ridotto di SO
  - E.g. Windows, OSX, Linux, iOS, Android...



# Il Sistema Operativo

Il sistema operativo ha diverse funzioni



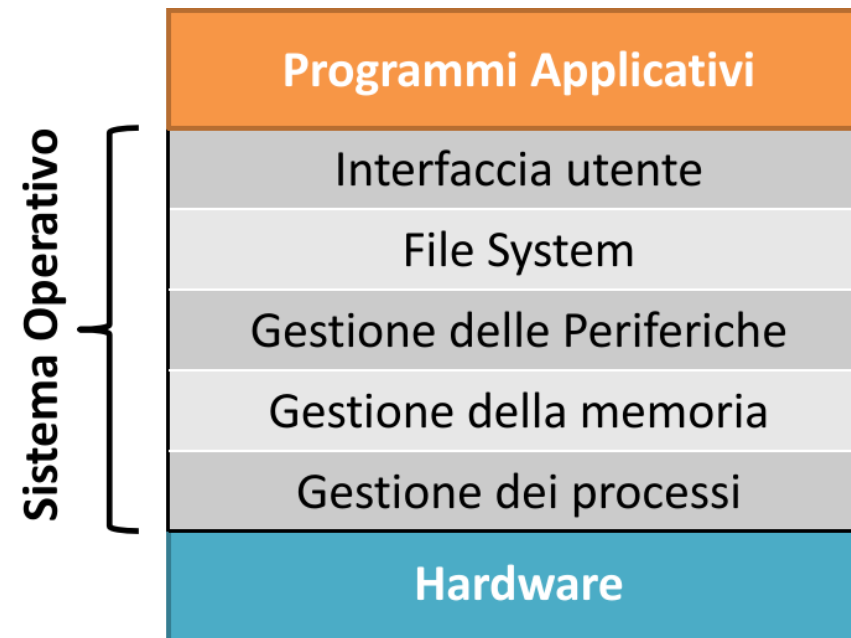
## Gestione dei processi

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



# Il Sistema Operativo

Il sistema operativo ha diverse funzioni



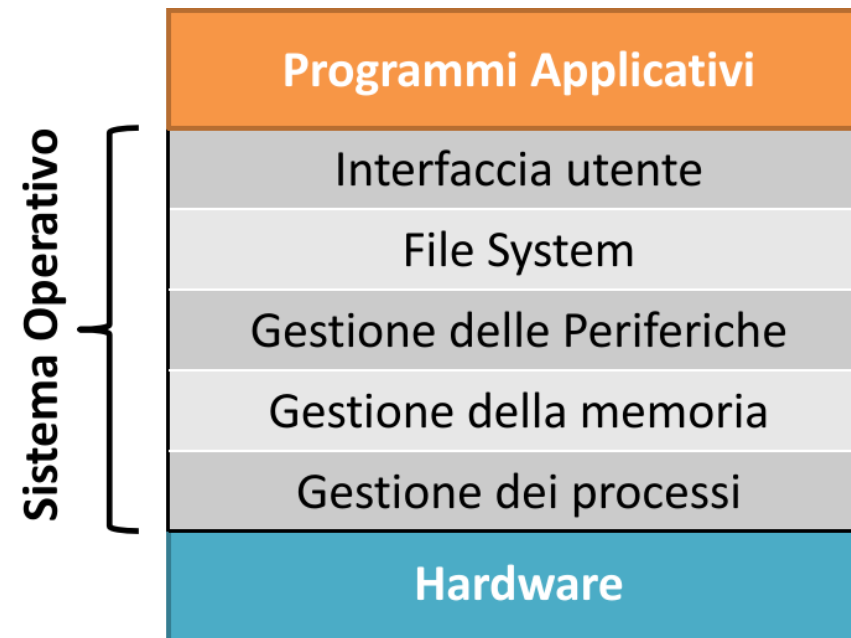
## Gestione della memoria

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



# Il Sistema Operativo

Il sistema operativo ha diverse funzioni



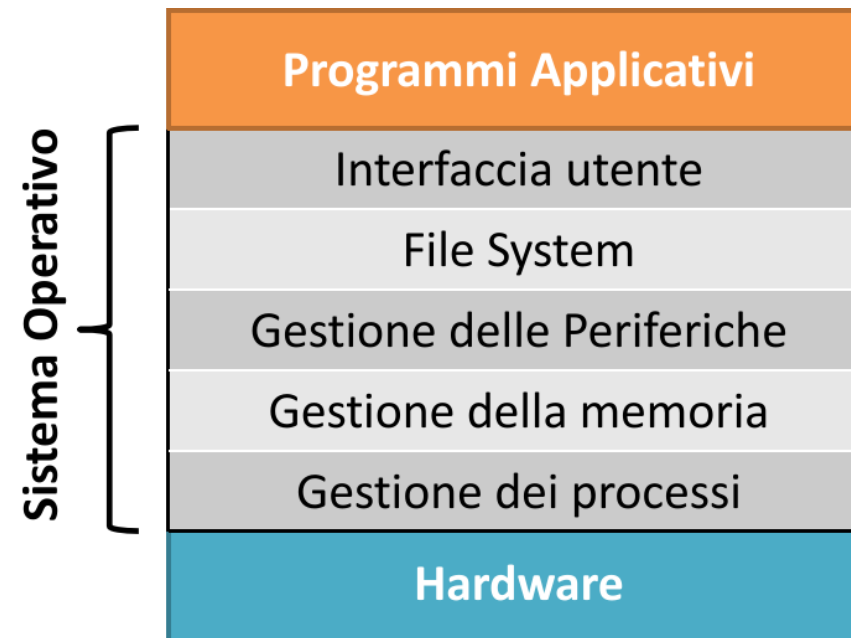
## Gestione delle periferiche

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



# Il Sistema Operativo

Il sistema operativo ha diverse funzioni



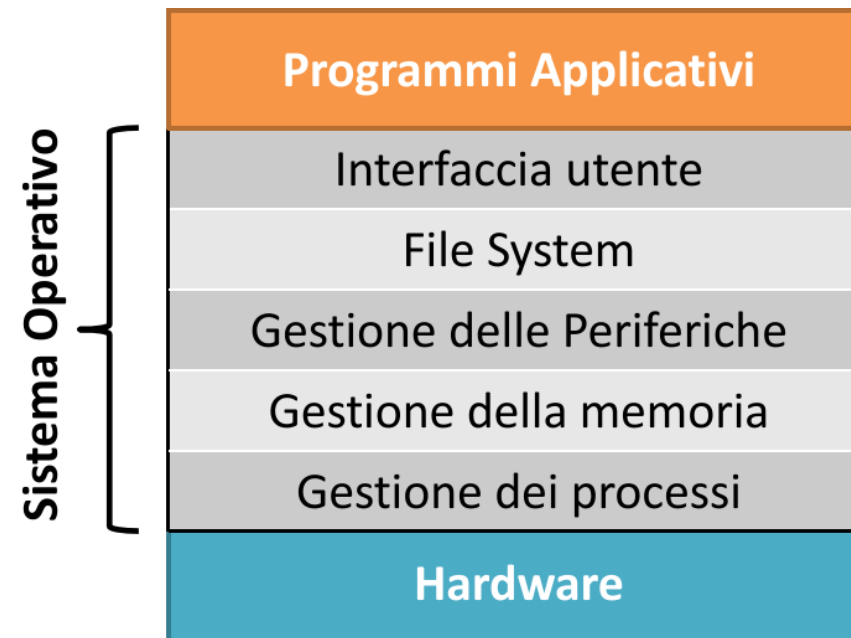
## File System

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



# Il 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)



# Interfaccia Utente e Terminale

---





# 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 + PowerShell
- OS X: GUI + "terminale"
- Linux: "terminale" + GUI



# Interfaccia Utente

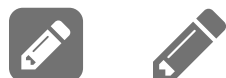
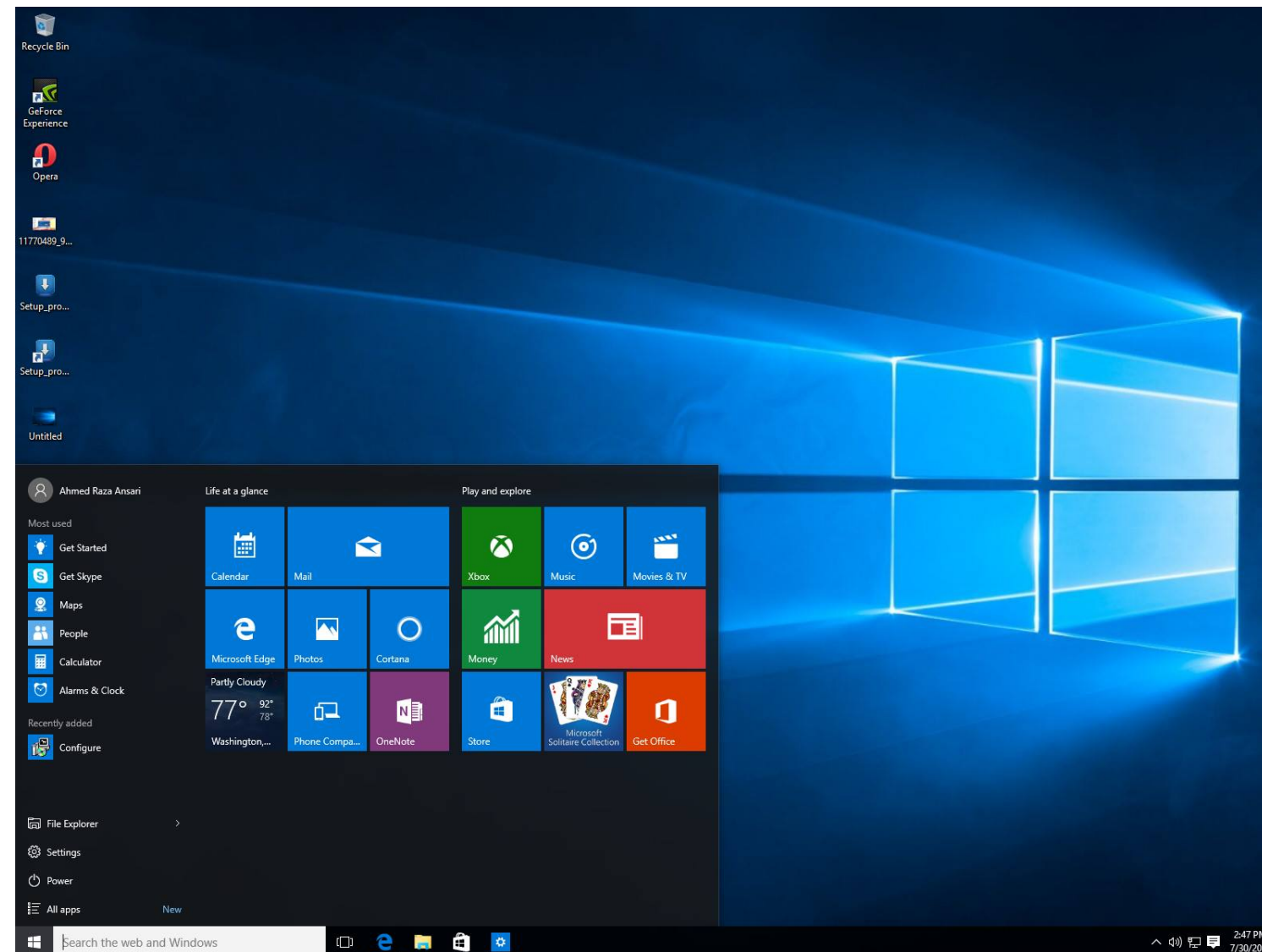
## Le interfacce grafiche:

- 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

## Esempio: l'interfaccia grafica di Win10



# Interfaccia Utente

Le interfacce testuali derivano dai supercomputer degli anni '60



# Interfaccia Utente

## Le interfacce testuali:

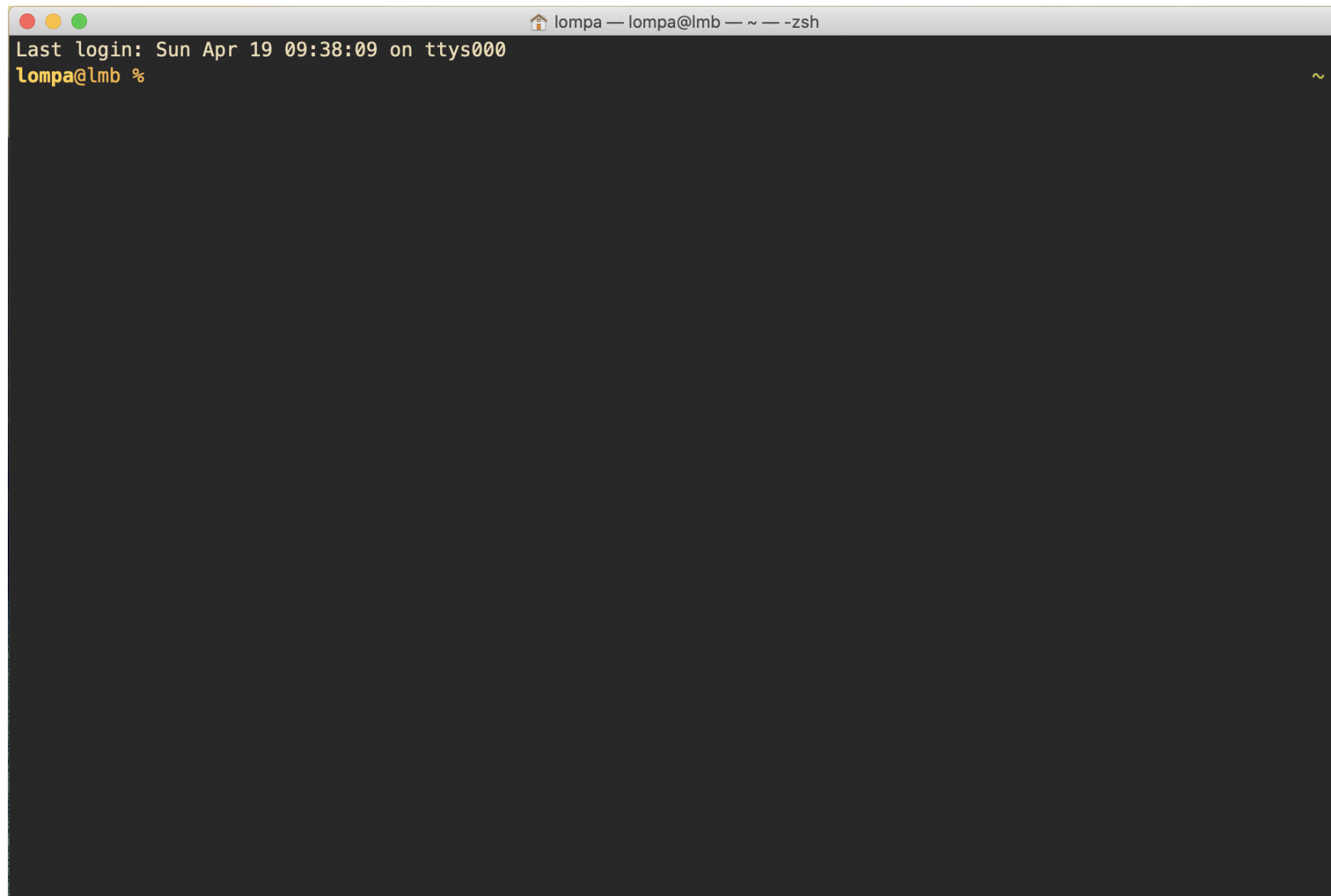
- Si chiamano anche (emulatori di) **terminali**
- Permettono l'interazione con il SO attraverso:
  - L'esecuzione di comandi e programmi
  - ...Che vengono digitati ed eseguiti (di solito premendo "invio")
- In questo modo permettono di:
  - Interagire con i file su disco
  - Interagire con le risorse di sistema
  - Preparare documenti, programmare, etc.

**Nella seconda parte della lezione di oggi impareremo ad usare un terminale su Windows**



# Interfaccia Utente

## Esempio: emulatore di terminale su OSX

A screenshot of a macOS terminal window. The title bar at the top shows three colored window control buttons (red, yellow, green) on the left and a title string 'lampa — lampa@lmb — ~ — -zsh' on the right. The terminal content area has a dark background. The first line of text is 'Last login: Sun Apr 19 09:38:09 on ttys000' in a light gray font. The second line shows the prompt 'lampa@lmb %' in a yellow font, followed by a cursor. A small yellow tilde icon is visible in the top right corner of the terminal area.

```
lampa — lampa@lmb — ~ — -zsh
Last login: Sun Apr 19 09:38:09 on ttys000
lampa@lmb %
```



# Esercizio: Aprire un Terminale

## Vediamo adesso come aprire un terminale (per Andacoda)

Il terminale è un normale programma, che potete avviare come tutti gli altri:

### ■ Su Windows:

- Cliccate sul menu "Avvio" (o premete il tasto Win)
- Cliccate sulla search box e cercate "anaconda powershell"
- Cliccate sull'entry "Anaconda Powershell Prompt" appena la vedete

### ■ Su OS X:

- Aprite spotlight
- Cercate "terminale"
- Cliccate l'entry "terminale" appena la vedete

### ■ Su Linux: immagino che sappiate già come si apre un terminale ;-)

**In alternativa, potete cercarlo nella lista di tutte le applicazioni installate**



# Gestione dei Processi

---





# Gestione dei Processi

## Una delle funzione del SO è gestire i processi

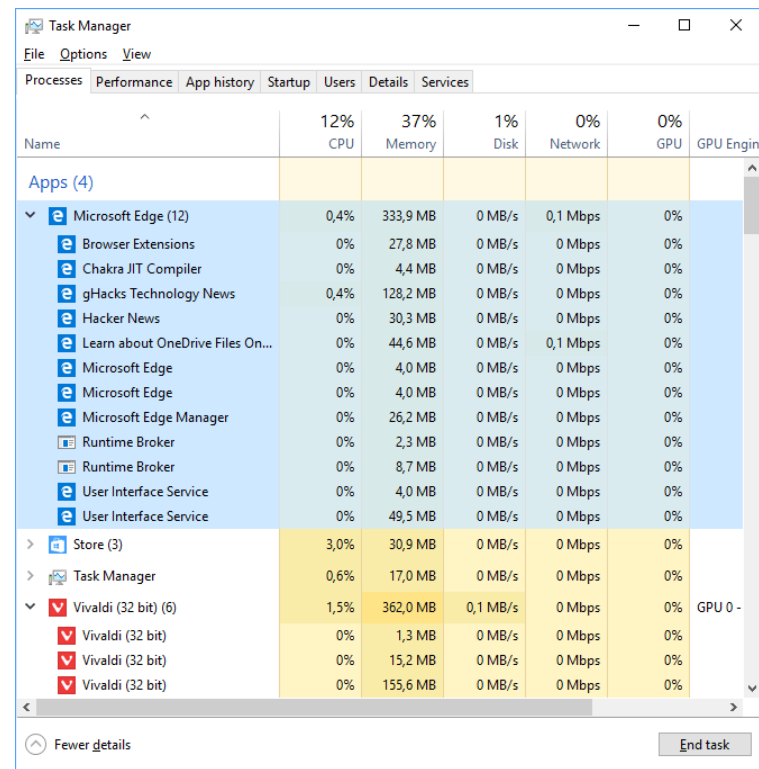
- Un processo è un programma in esecuzione
- La gestione dei processi include:
  - Avvio di un processo (esecuzione di un programma)
  - Interruzione di un processo
  - Sospensione di un processo
  - Elenco dei processi in esecuzione
  - Condivisione della CPU tra processi
- Alcune funzionalità sono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



# Esercizio: Gestione Processi

Su Win potete accedere ad una "lista di processi" come segue:

- Premete Ctrl+Alt+Canc
- Quindi selezionate "Gestione Attività"\*\*



Task Manager						
File Options View						
Processes Performance App history Startup Users Details Services						
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)	3,0%	30,9 MB	0 MB/s	0 Mbps	0%	
Task Manager	0,6%	17,0 MB	0 MB/s	0 Mbps	0%	
Vivaldi (32 bit) (6)	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%	
Fewer details End task						

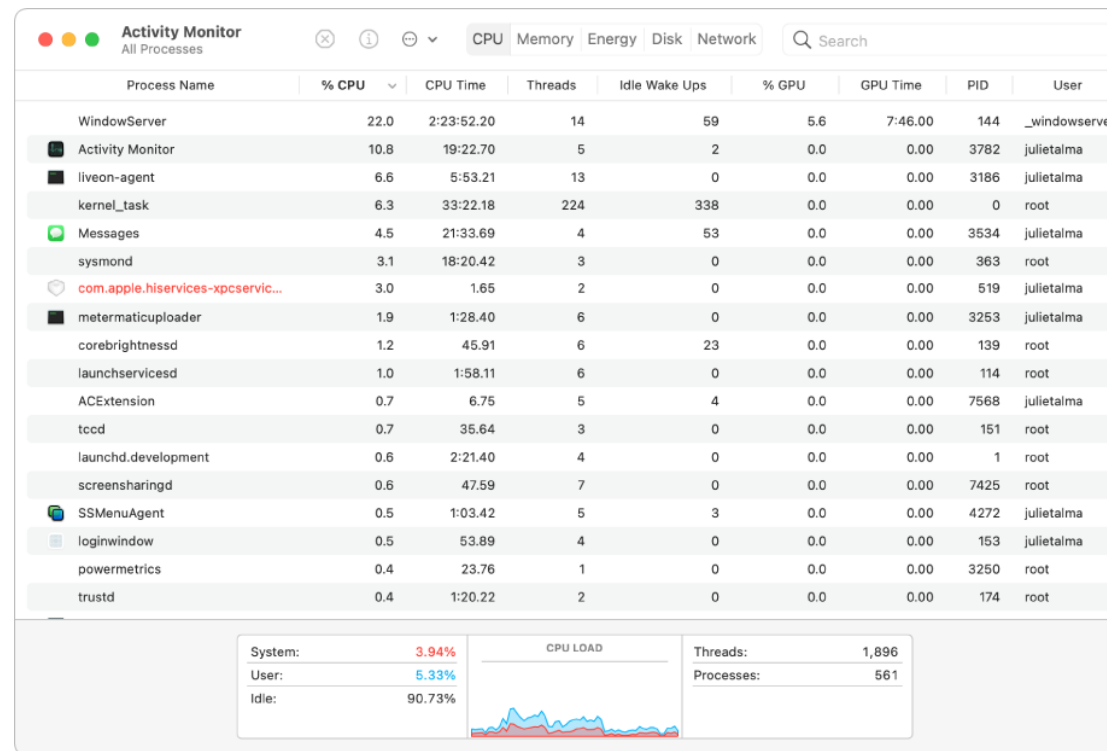
Selezionando un processo, Windows vi propone un pulsante per interromperlo



# Esercizio: Gestione Processi

Su OSX potete accedere ad una "lista di processi" come segue:

- Aprite spotlight
- Quindi cercate (ed avviate) "Monitor Attività"



Selezionando un processo, OSX vi propone un pulsante per interromperlo



# Esercizio: Gestione Processi

Da terminale, potete fare lo stesso come segue

- Su **Windows**:

- Avviate un terminale
- Digitate il comando "`gps`", poi premete invio

- Su **OSX** (e Linux):

- Avviate un terminale
- Digitate il comando "`ps`"

L'output dei due comandi è piuttosto diverso:

- Windows stampa per default tutti i processi in esecuzione
- OSX/Linux solo quelli che avete avviato dal terminale corrente
  - Se li volete tutti, dovete usare "`ps -A`"



# Gestione delle Memoria

---



# Gestione della Memoria

**Il sistema operativo ha il compito di gestire la memoria centrale**

- **Alloca memoria** (RAM) ai programmi al momento dell'avvio
  - Si tratta del loro "spazio di lavoro"
  - ...Senza il quale non possono funzionare
- In caso di utilizzo di memoria eccessivo
  - Il SO sposta su disco le aree destinate ai processi meno attivi
  - Si parla di "**swap memory**"
  - ...E si tratta di una operazione molto inefficiente



# Unità di Misura per la Memoria

## Le informazioni sono rappresentate come sequenze di 0/1

- Una cifra digitale, i.e. 0 o 1, si chiama **bit** (Binary digT)
- 8 bit formano un **Byte**

Dopodiché si usano i prefissi del Sistema Internazionale, con una particolarità:

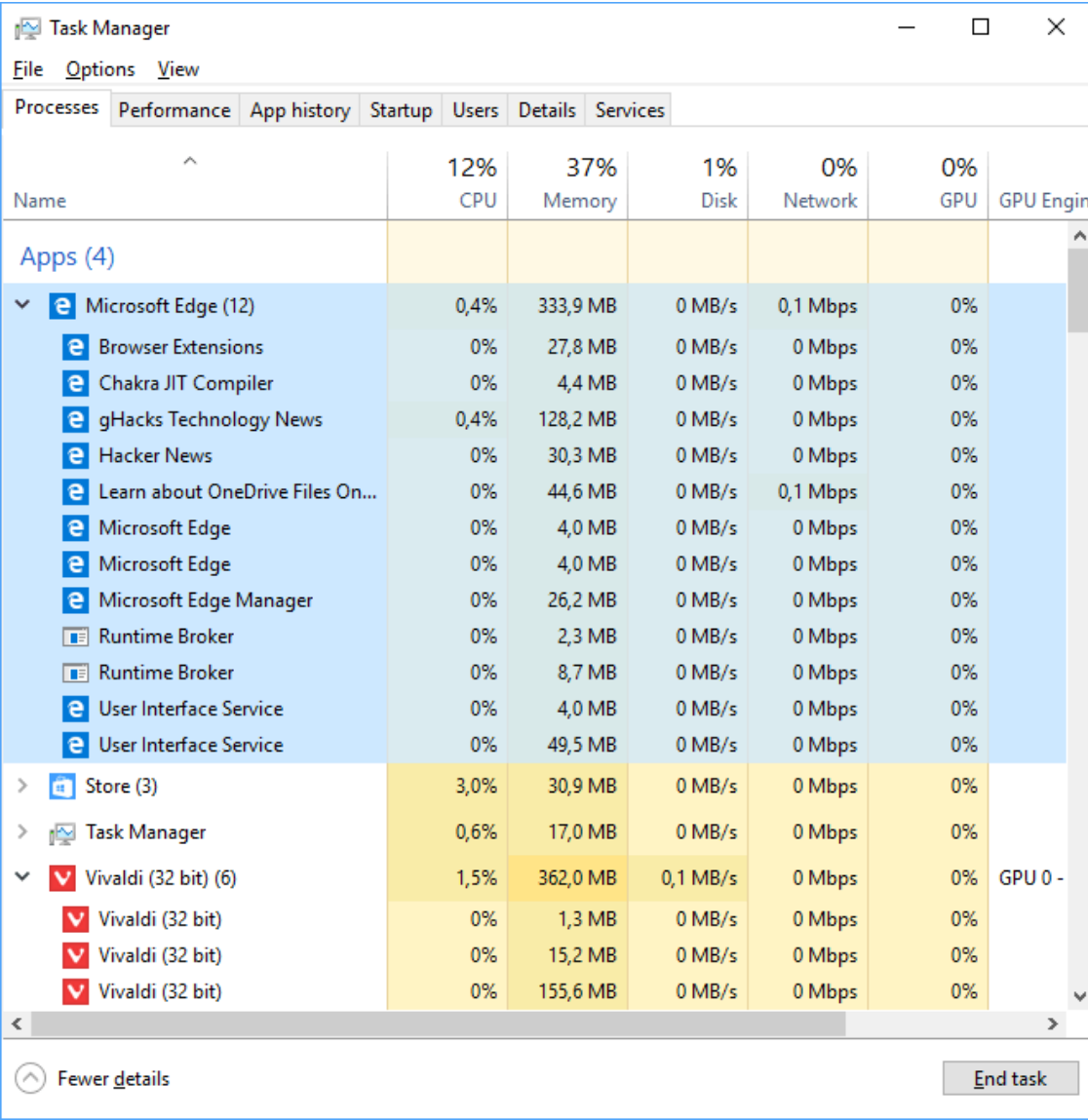
- Se si usa la scrittura kB, MB, GB, etc.
  - Si legge **kilobyte**, **megabyte**, **gigabyte**, etc.
  - ...E si intende  $10^3$ ,  $10^6$ ,  $10^9$  Byte
- Se si usa la scrittura KiB, MiB, GiB, etc.
  - Si legge **kibibyte**, **mebibyte**, **gibibyte**, etc.
  - ...E si intende  $10^{10}$  (1024),  $10^{20}$  ( $1024^2$ ),  $10^{30}$  ( $1024^3$ ) Byte



# Esercizio: Gestione della Memoria

## Aprire di nuovo "Gestione Attività" (o "Monitor Attività")

Su windows, una colonna visualizza la quantità di memoria utilizzata



The screenshot shows the Windows Task Manager window with the 'Performance' tab selected. The 'Memory' column is highlighted in yellow, indicating it is the selected resource. The table lists various applications and their memory usage.

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)	3,0%	30,9 MB	0 MB/s	0 Mbps	0%	
Task Manager	0,6%	17,0 MB	0 MB/s	0 Mbps	0%	
Vivaldi (32 bit) (6)	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%	

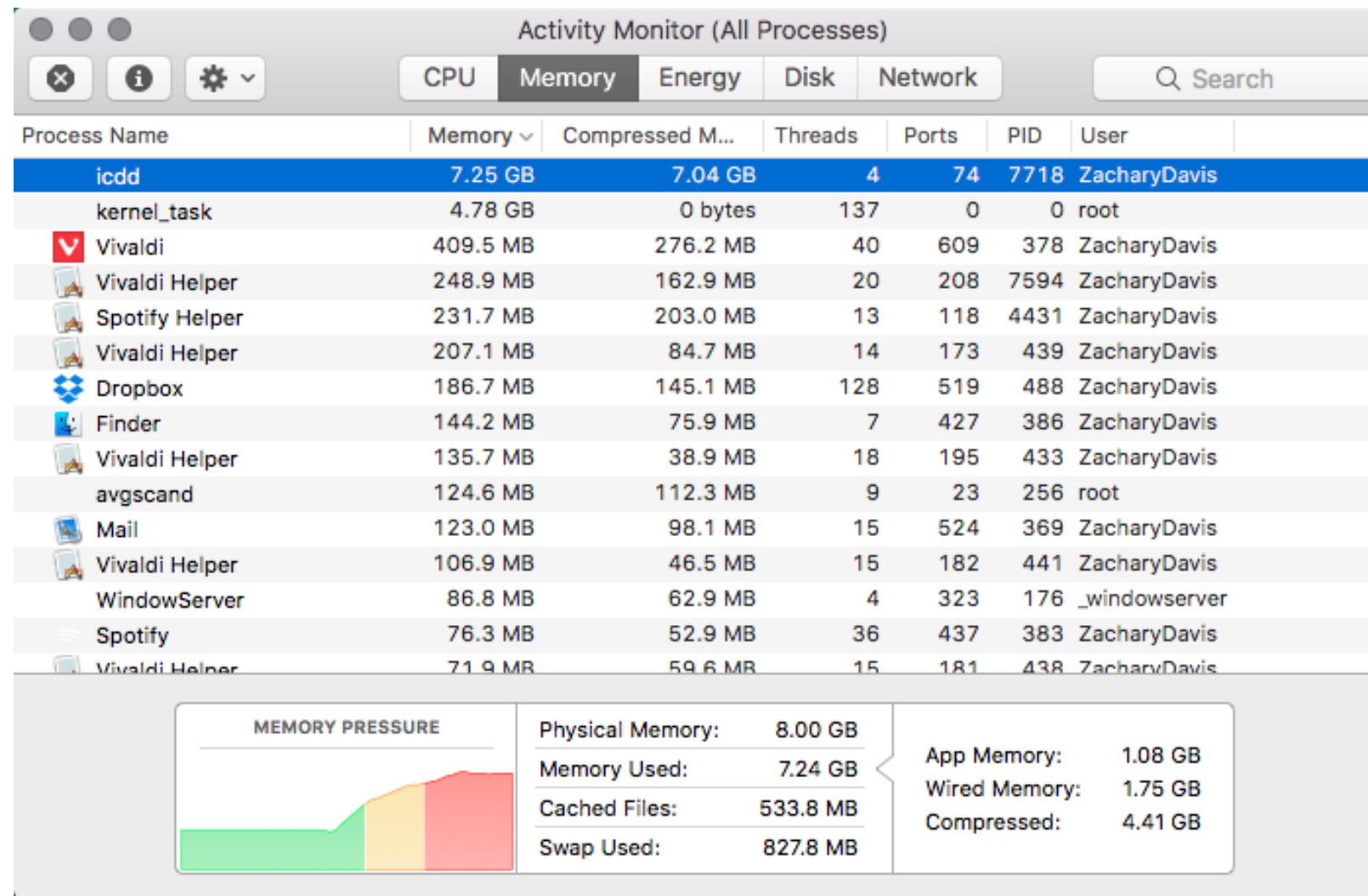




# Esercizio: Gestione della Memoria

Su OSX, aprite di nuovo "Monitor Attività"

Una pagina è dedicata a visualizzare informazioni sull'utilizzo di memoria



# Esercizio: Gestione della Memoria

**Da terminale, potete visualizzare la memoria disponibile:**

- Sia su PowerShell che su OSX/Linux:
  - Digitate il comando `free -m` e premete invio

## Qualche informazione:

- `free` è il nome del comando (o programma) che volete eseguire
- Premere "invio" richiede al terminale di eseguire il comando
- `-m` è un **argomento da riga di comando**
  - Indica al comando/programma di comportarsi in modo leggermente diverso
  - In questo caso, visualizza i valori in mebibyte
- Molti comandi supportano l'argomento `--help`
  - ...Che visualizza una breve guida sull'utilizzo del comando



# Gestione Periferiche e File System

---



# 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 (API)...
  - ...Che nascono l'esatto modello di hardware collegato
- 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
- ...Ma la trattano come una unica sequenza di bit!
- Il File System ne fornisce una **organizzazione**

Tipicamente, questa è basata sulle astrazioni di **file** e **directory**

- "Formattare" un dispositivo
- ...Significa predisporre questa organizzazione

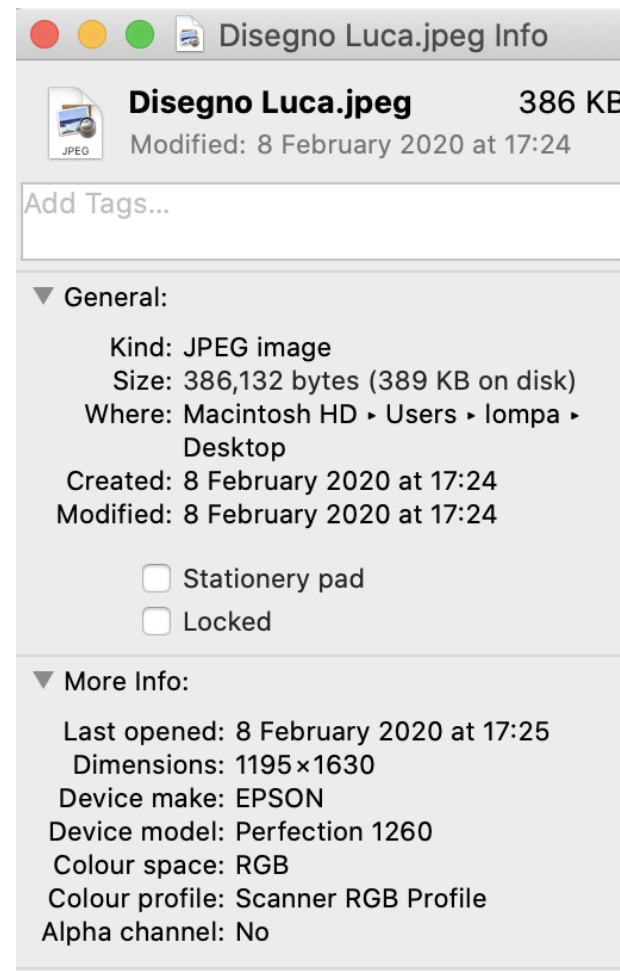
## Un **file** è l'unità logica di memorizzazione delle informazioni:

- È una sequenza di byte
- Ha un nome
- ...E una serie di proprietà aggiuntive



# File System

## Un esempio: finestra di ispezione file su OSX



# 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:**

- Alcuni SO la nascondono per default (e.g. Windows)
- 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 permette di effettuare operazioni sui file**

- Creazione
- Lettura/scrittura
- Spostamento...

**I file sono tipicamente organizzati in **directory****

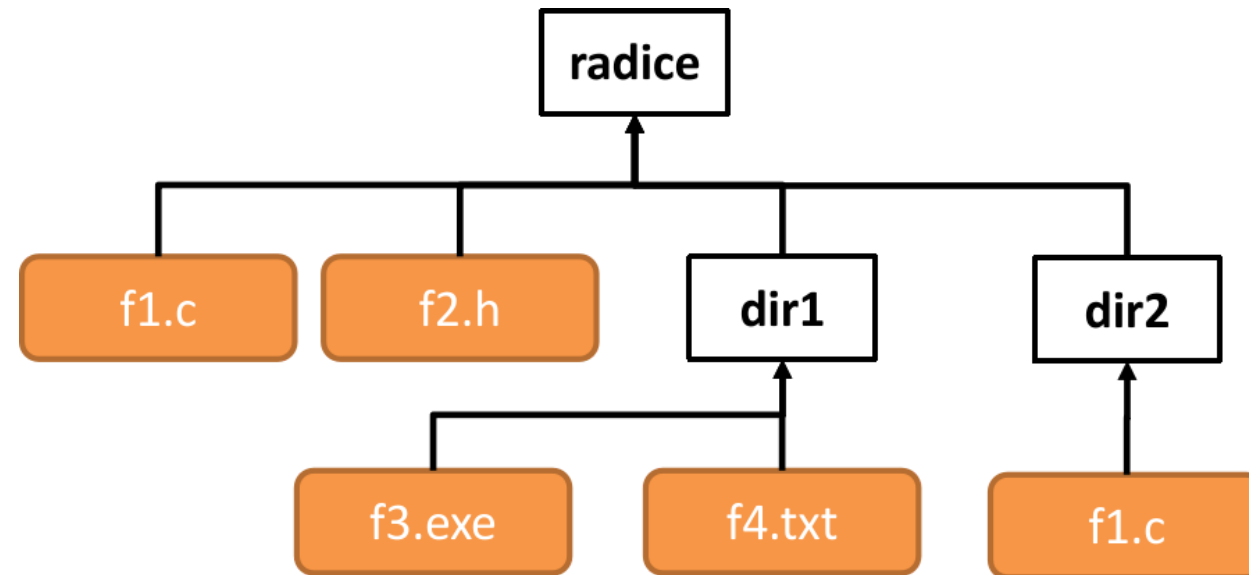
- Una directory è un contenitore di file (o altre directory)
  - Sono le "cartelle" di Windows/OSX
- I file sono quindi organizzati in modo gerarchico
  - La relazione è di quella di appartenenza (contenitore/contenuto)





# File System

Una organizzazione gerarchica forma sempre un albero:



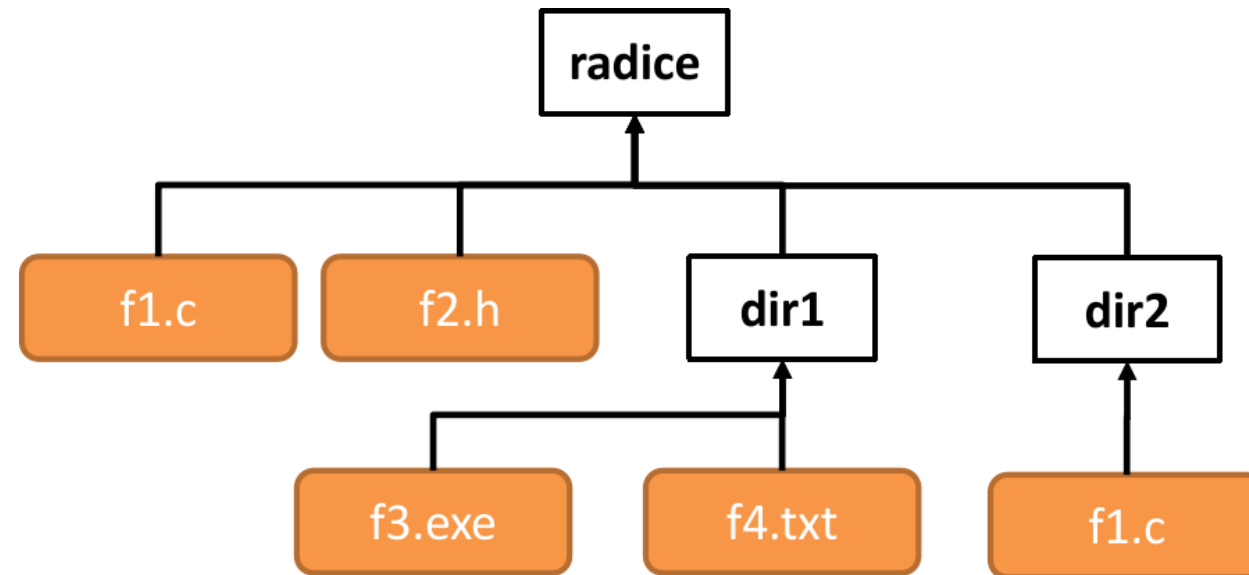
Le frecce denotano appartenenza

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



# File System

Una organizzazione gerarchica forma sempre un albero:



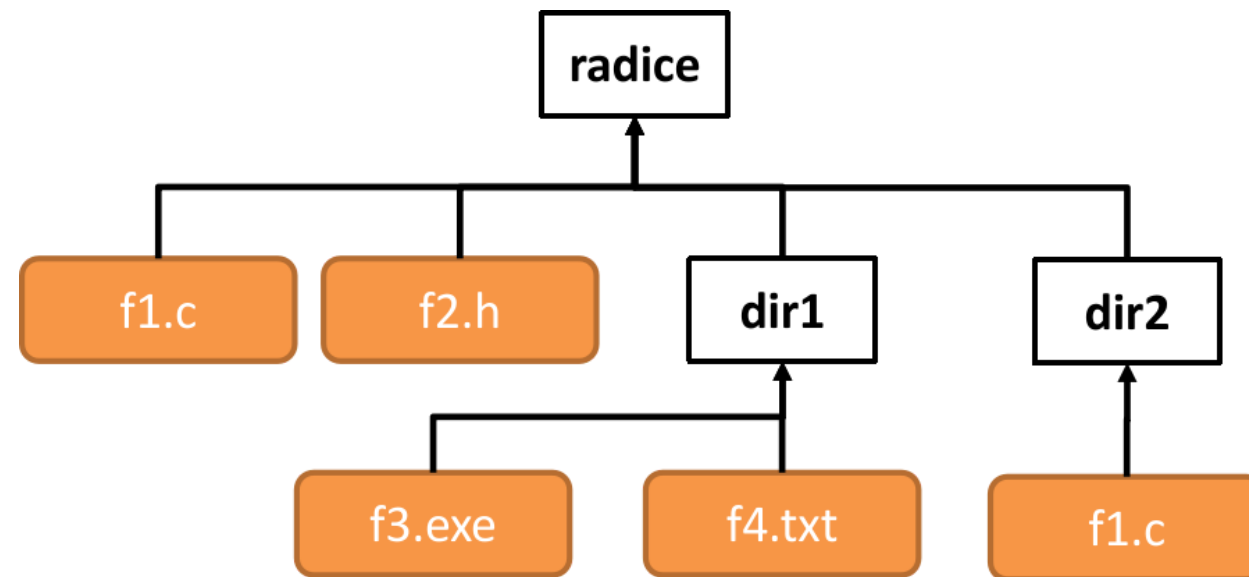
Le frecce denotano appartenenza

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



# File System

Una organizzazione gerarchica forma sempre un albero:



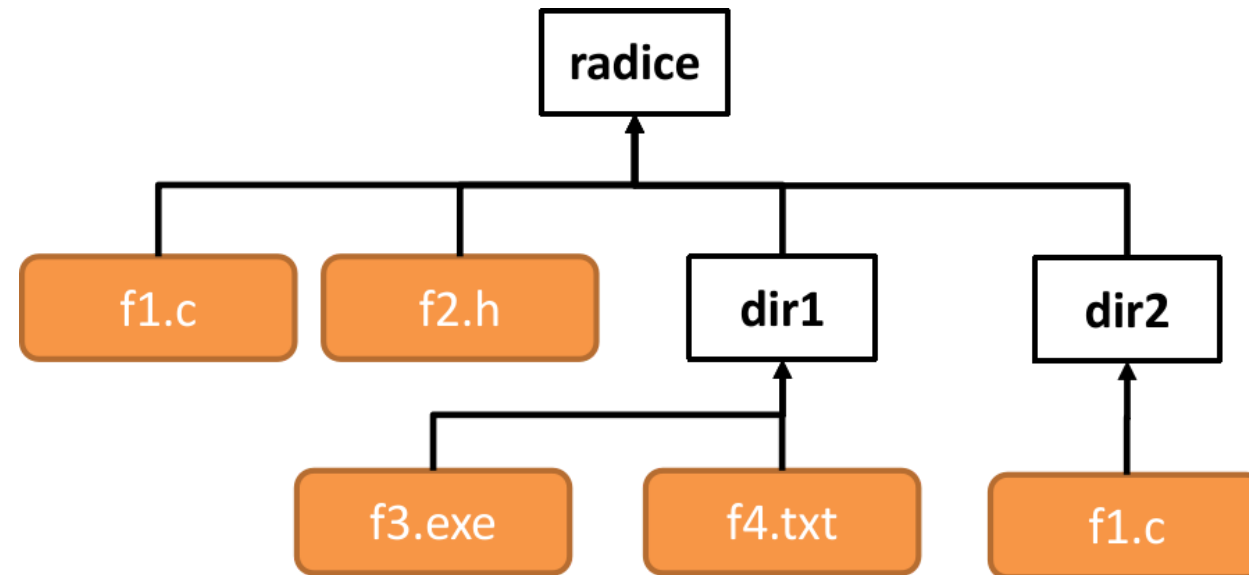
Esiste sempre una **directory radice**

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



# File System

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

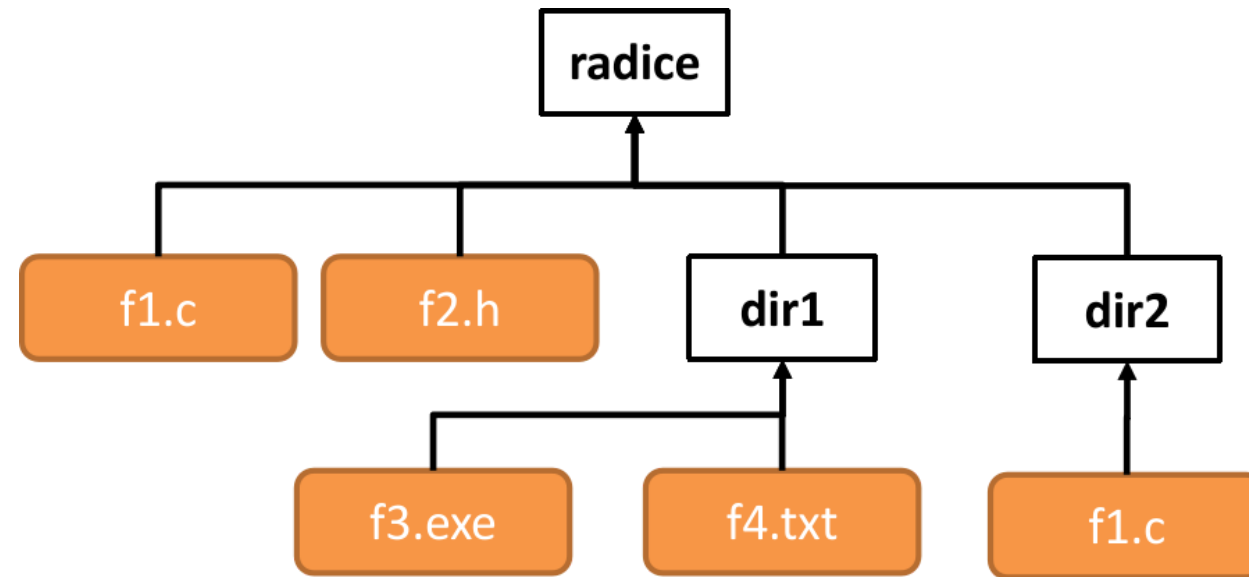


- Questo contiene il nome del file...
- ....preceduto dalla sequenza di directory...
- ...per raggiungerlo dalla radice



# File System

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



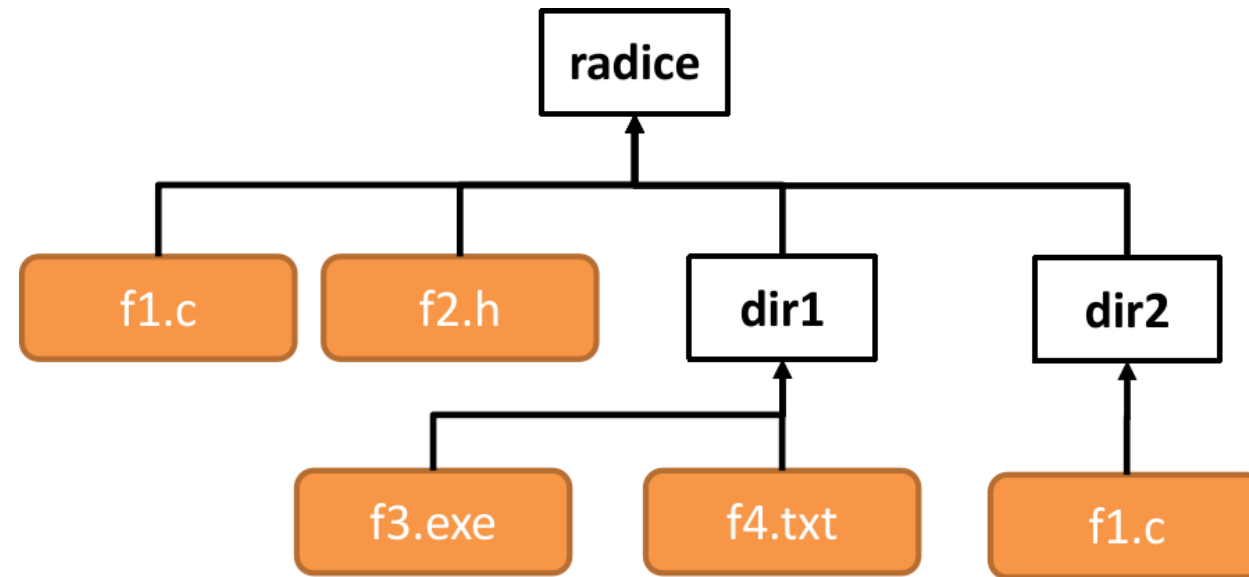
Esempi di percorso assoluto su Win ("" separa le directory):

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



# File System

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



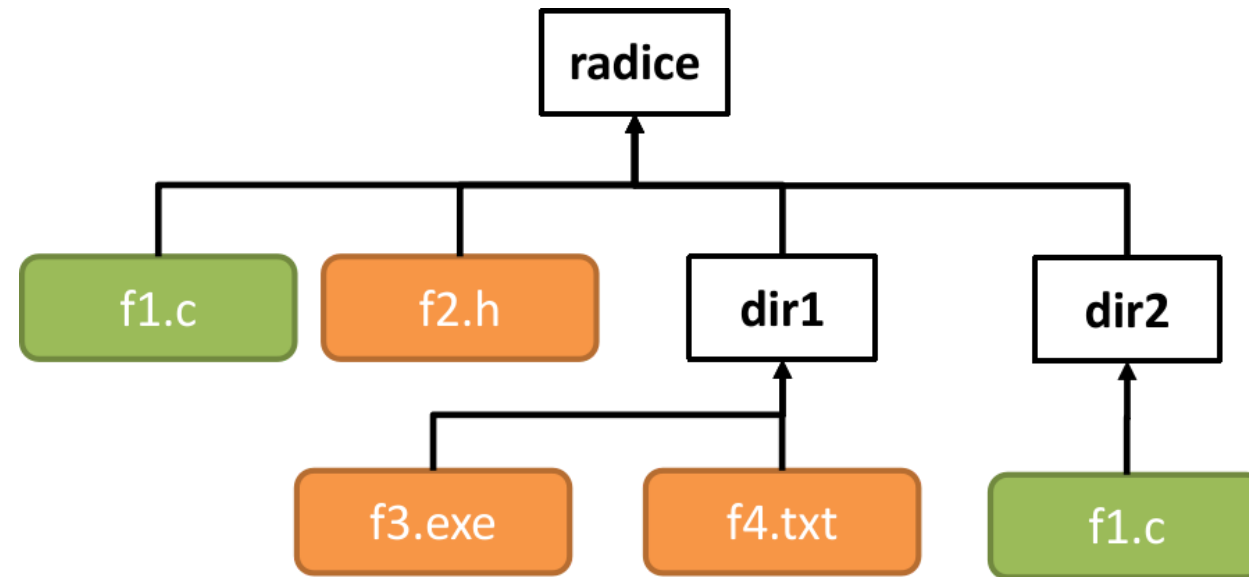
Esempi di percorso assoluto su OSX/Linux ("/" separa le directory):

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



# File System

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

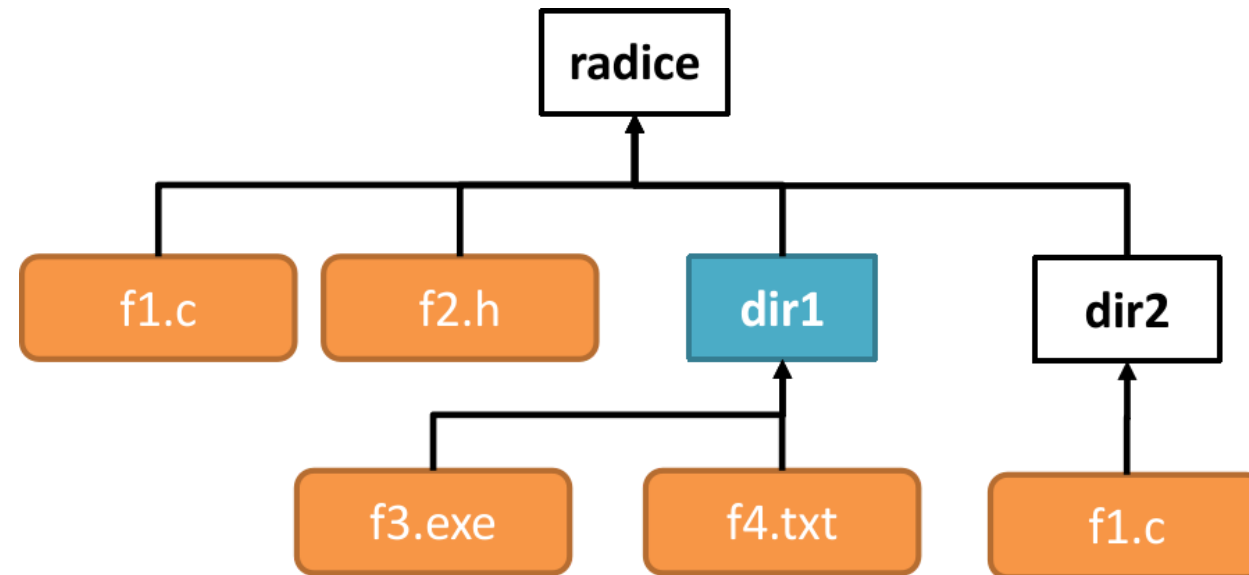


- Conseguenza: due file possono avere lo stesso nome...
- Ma solo se sono in directory diverse
- E.g. `"/f1.c"` e `"/dir2/f1.c"`



# File System

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



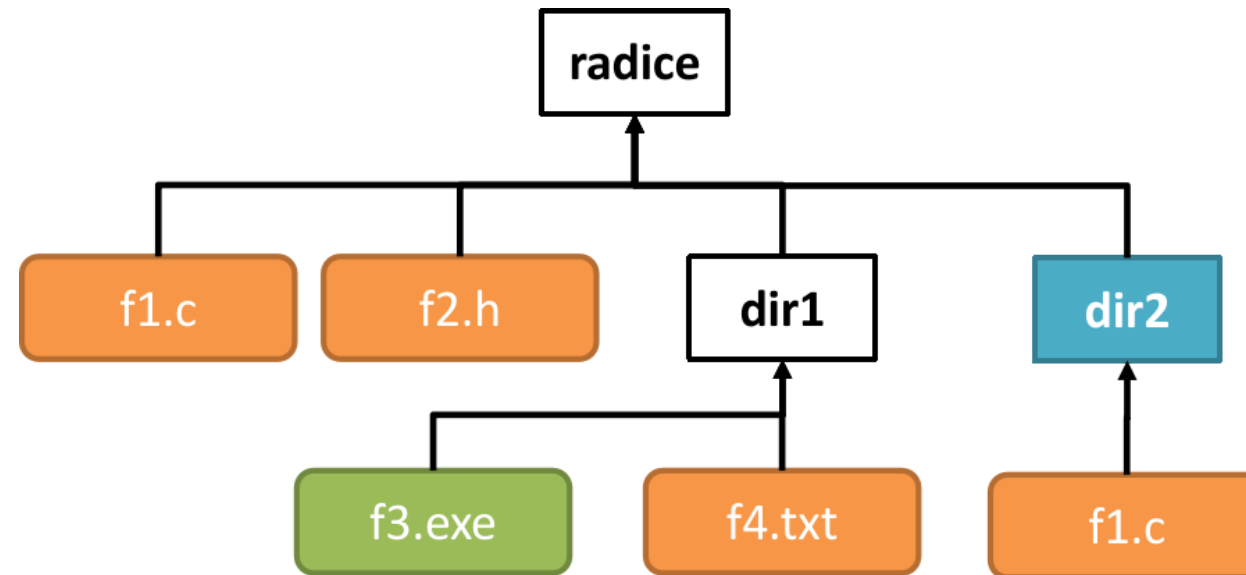
- È possibile riferirsi ad un file indicandone il percorso...
- ...rispetto alla directory corrente
- In questo caso si parla di **percorso relativo**





# File System

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

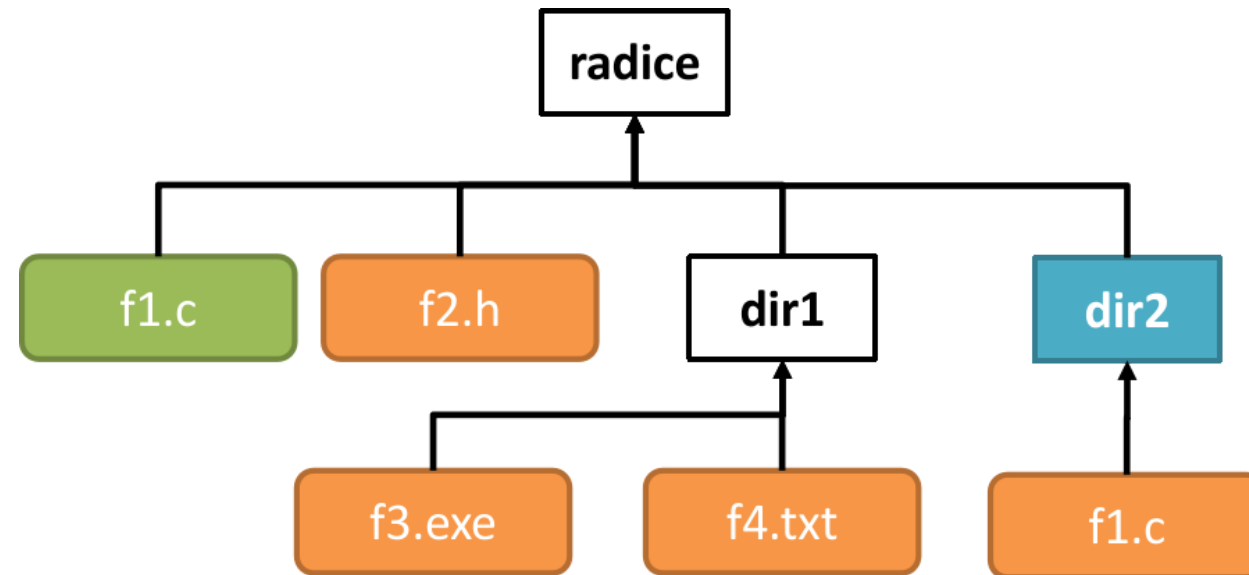


- Si indica la directory "padre" con la notazione ".."
- "f3.exe" da "dir2": "..\dir1\f3.exe" ("../dir1/f3.exe")



# File System

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



- Si indica la directory "padre" con la notazione ".."
- "f1.c" evidenziato, da "dir2": "..\f1.c" ("../f1.c")



## Esercizio: Directory Corrente

**Verifichiamo come un programma sia associato ad una directory corrente**

- Aprite un terminale
- Quindi eseguite il comando `"pwd"`

Il comando visualizza il percorso assoluto della directory corrente

**Di "default" (se non diversamente specificato) questa la vostra home**

- Su Windows (recente) ha un percorso del tipo `"c:\Users\<nome utente>"`
- Su OSX/Linux ha un percorso del tipo `"/home/<nome utente>"`



## Esercizio: Contenuto di una Cartella

**La vostra home è una cartella che contiene i vostri file personali. Tipicamente:**

- Documenti (una cartella)
- Download
- Desktop (è una cartella anche lui)!
- ...

**Provate a visualizzare il contenuto della cartella corrente**

Per farlo, eseguite il comando "`ls`"

- Dovreste vedere i file/cartelle indicati sopra



## Esercizio: Cambiare la Cartella Corrente

### È possibile cambiare la cartella corrente

Per farlo, potete usare il comando `"cd <percorso cartella>"`

- Il percorso può essere assoluto o relativo
- In particolare, se volete spostarvi in una cartella contenuta in quella corrente
- ...Allora basta eseguire `"cd <nome della cartella contenuta>"`
- Se a metà del girate premete "tab", il terminale cercherà di **completare il nome**

### Per tornare sui vostri passi potete usare:

- Il comando `"cd .."`, che vi sposta nella cartella contenente
- Il comando `"cd ~"`, che vi sposta nella cartella home
  - Se non sapete come scrivere il simbolo "~", [guardate qui](#)

**Provate a spostarvi nella cartella "Documenti", quindi tornare nella home**



# Esercizio: Raggiungere la Lezione di Oggi

## Proviamo con qualcosa di un po' più complicato

Sul vostro PC, preparate una cartella per il corso

- È una buona idea metterla da qualche parte in "Documenti"
  - ...O ovunque teniate le lezioni di solito
- Siete (chiaramente) liberi di non farlo
  - ...Però poi starà a voi ritrovare il materiale in mezzo al casino :-)

Scaricate quindi la lezione di oggi (in formato zip) dal sito del corso

- Spostate il file zip nella cartella che avete preparato

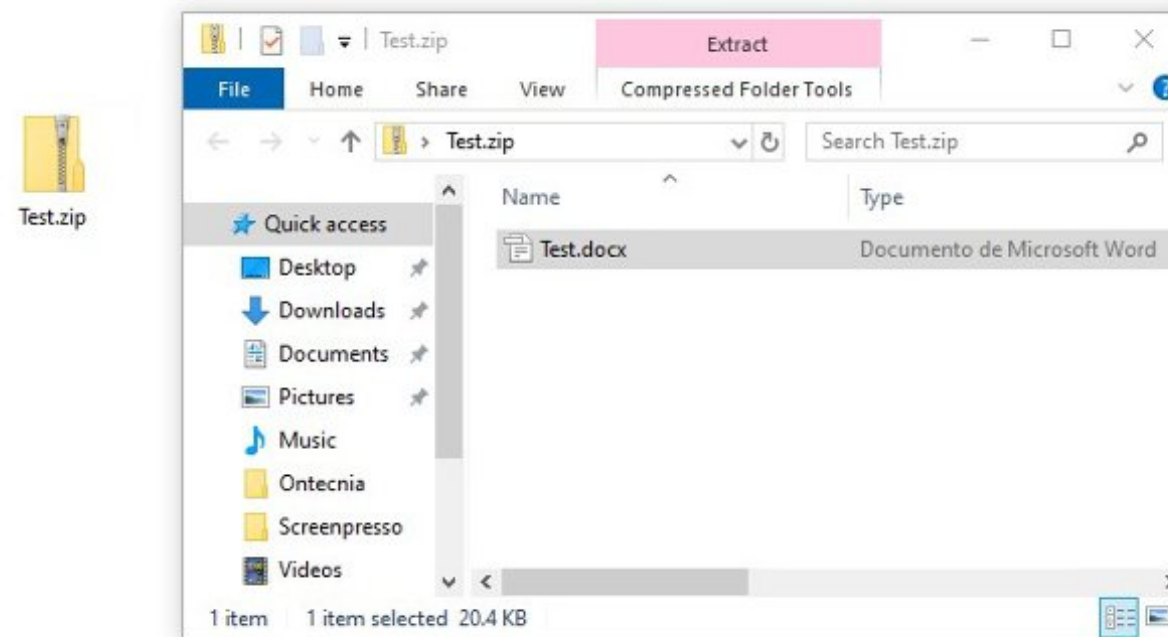


# Esercizio: Raggiungere la Lezione di Oggi

## Proviamo con qualcosa di un po' più complicato

Estraete l'archivio zip

- Su Win, facendo doppio click, può essere che si apra un visualizzatore di zip
- Ve ne accorgete se vedete una barra rosa tipo questa



- Se succede usate il click destro: ci sarà una voce per estrarre l'archivio



# Esercizio: Raggiungere la Lezione di Oggi

**Ora, provate a raggiungere la cartella appena estratta con il terminale**

- Aprite un terminale nel solito modo
- Usare "`cd`" (ed al bisogno "`cd ..`" e "`cd ~`") per raggiungerla
- Vi accorgete di essere nel posto giusto se "`ls`" visualizza qualcosa come:

```
"01. Architettura dell'Elaboratore.ipynb"
'02. Sistema Operativo.ipynb'
'03. (Esercizi) Utilizzo del Terminale.ipynb'
pdfs
LICENSE.md                      requirements.txt
README.md                      rise.css
assets
```

Notate come i nomi dei file vengano visualizzati con la loro estensione (".txt", ".md", etc.)





# Struttura di una Lezione

## Possiamo ora discutere la struttura della cartella

...Cioè di una lezione tipo

- I file `.pynb` sono le lezioni vere e proprie
  - Si tratta di *notebook* del sistema Jupyter
- La cartella `"pdfs"` contiene le slide in formato PDF
- La cartella `"assets"` contiene immagini ed altri file usati nelle lezioni
- Il file `"LICENSE.md"` contiene la licenza d'uso del materiale
- Il file `"README.md"` contiene alcune informazioni per accedere alla lezione
- Il file `"requirements.txt"` contiene una lista di pacchetti Python che useremo
- Il file `"rise.css"` contiene informazioni su come visualizzare le slide



## Esercizio: Visualizzare il Contenuto di un File

È possibile visualizzare il contenuto di un file **di testo** dal terminale

- Per farlo, potete usare il `"cat <percorso del file>"`
- Usandolo sul file `"README.md"` dovrete vedere qualcosa del genere:

```
# README
```

```
Questa è una delle lezioni dell'insegnamento Fondamenti di Informatica T, del corso di laurea in Ingegneria dell'Energia Elettrica dell'Università di Bologna.
```

```
...
```

Provate a fare lo stesso con gli altri file nella cartella



# Esercizio: Visualizzare il Contenuto di un File

## Il terminale può visualizzare solo testo

- Eseguendo " `cat` " su un file non testuale si ottengono strani risultati
- ...Il terminale si sforza di interpretare la sequenza di bit come testo

## Vediamolo in atto

- Spostatevi nella cartella " `assets` "
- Eseguite `cat` sul file " `Dante.docx` " (che è un documento MS Word)

Dovreste ottenere una sequenza di strani simboli



# Esercizio: Installare Pacchetti Anaconda

## Anaconda fornisce un gestore di pacchetti accessibile da terminale

...Ossia un gestore di estensioni del linguaggio Python

- È un comando che si chiama "conda"
- Eseguendolo con "conda --help" otterrete qualcosa di questo tipo:

```
usage: conda [-h] [-V] command ...
```

```
conda is a tool for managing and deploying applications, environments and packages.
```

```
Options:
```

```
...
```

```
install      Installs a list of packages into a specified conda
              environment.
```

```
list         List linked packages in a conda environment.
```

```
...
```

```
remove       Remove a list of packages from a specified conda environment.
```



# Esercizio: Installare Pacchetti Anaconda

**Anaconda fornisce un gestore di pacchetti accessibile da terminale**

- Con `conda list`
  - ...Potete visualizzare i pacchetti già installati
- Con `conda install <nome pacchetto>`
  - ...Potete installare nuovi pacchetti
- Con `conda remove <nome pacchetto>`
  - Potete rimuovere un pacchetto installato

**Provate a:**

- Visualizzare la lista dei pacchetti installati
- Installare il pacchetto "numpy", con `conda install numpy`
  - Se avete la versione base di Anaconda, numpy sarà già installato



# Esercizio: Installare Pacchetti Anaconda

## Assicuriamoci di avere tutti i pacchetti necessari per il corso

Installate tutti i pacchetti indicati nel file `requirements.txt`

■ Questi sono:

```
numpy  
scipy  
matplotlib  
jupyter
```

■ Potete farlo con `"conda install --file requirements.txt"`



## Esercizio: Avviare la Lezione

**Finalmente possiamo eseguire una lezione per la prima volta**

- Dalla cartella con la lezione (quella con i file in formato `.ipynb`)
- ...Eseguite il comando `jupyter notebook`

Se tutto va bene:

- Si aprirà automaticamente una finestra del vostro browser
- ...E presenterà il contenuto della cartella corrente
- Potete aprire uno qualunque dei notebook cliccando sul file corrispondente

**Per adesso ci fermiamo qui**

- Per terminare chiudete il browser
- Quindi tornate al terminale e premete "CTRL+c"
- Così facendo terminerete il processo jupyter

