



UNIVERSITÀ
DEGLI STUDI DELLA
TUSCIA

PROGRAMMAZIONE

LINUX:

FILE SYSTEM – SHELL E COMANDI PRINCIPALI

Prof.ssa Tiziana Castrignanò



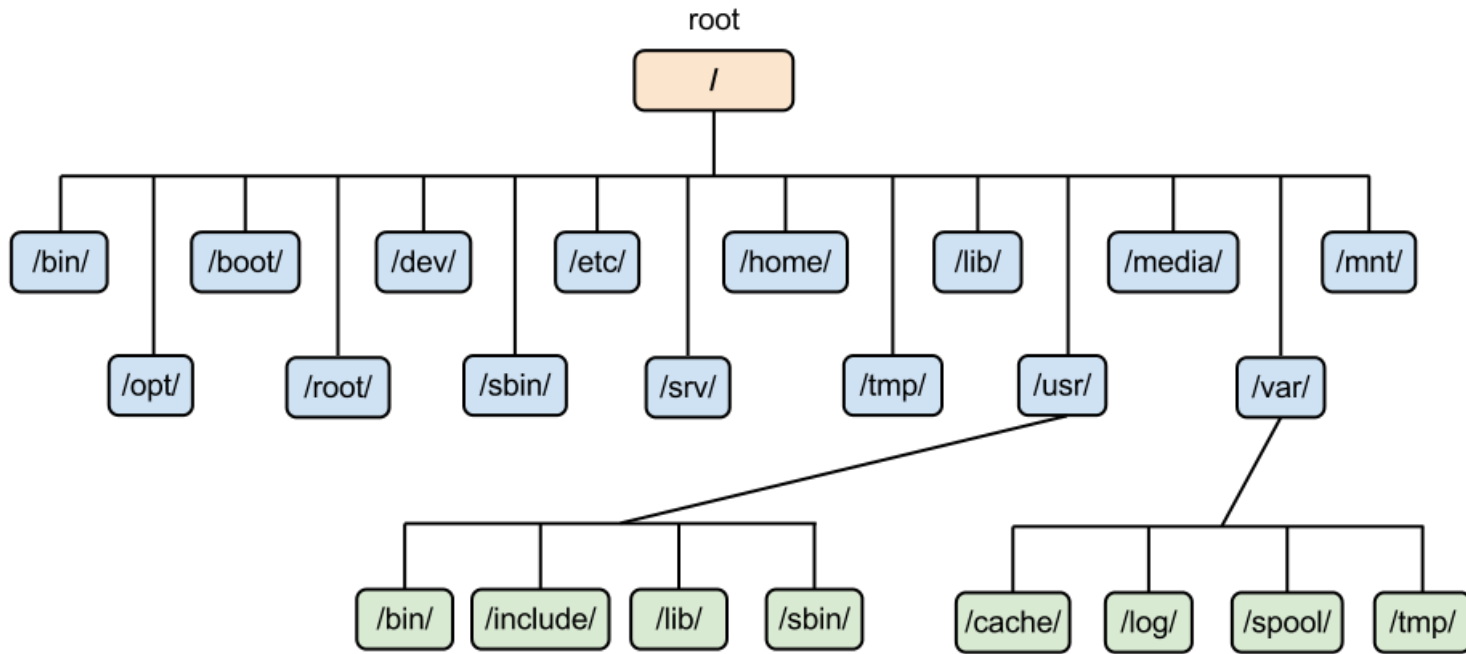
File system

LINUX

File System: struttura

- ❑ Il file system di Linux organizza i documenti digitali con una **struttura gerarchica** (o ad albero invertito)

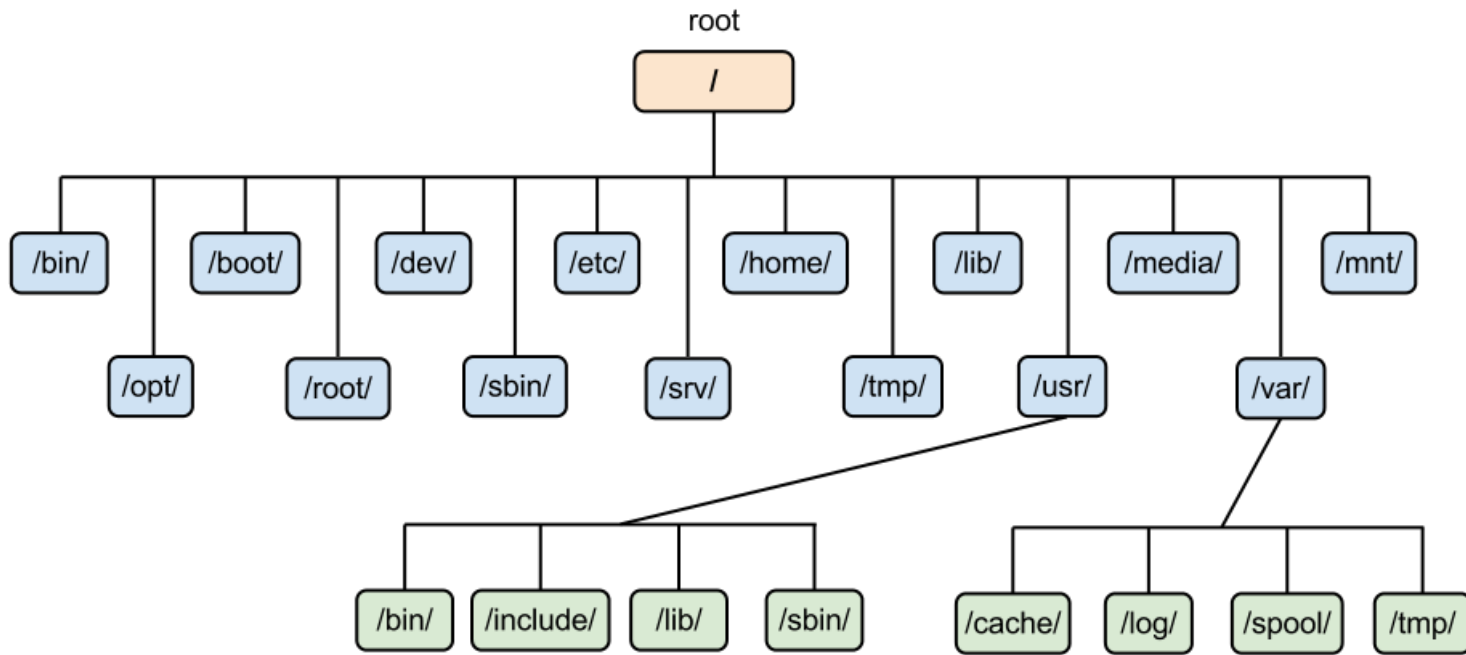
Nell'albero invertito, la radice si trova in alto e i rami delle directory e delle sottodirectory si estendono in basso la radice



LINUX

File System: albero gerarchico

- ❑ La directory **/** è la **radice dell'albero** (*root*) che si trova in cima alla gerarchia del file system
- ❑ Il carattere **/** è utilizzato anche come separatore di directory nel nome del file, ad esempio **/etc** è una sottodirectory della directory **/**



LINUX

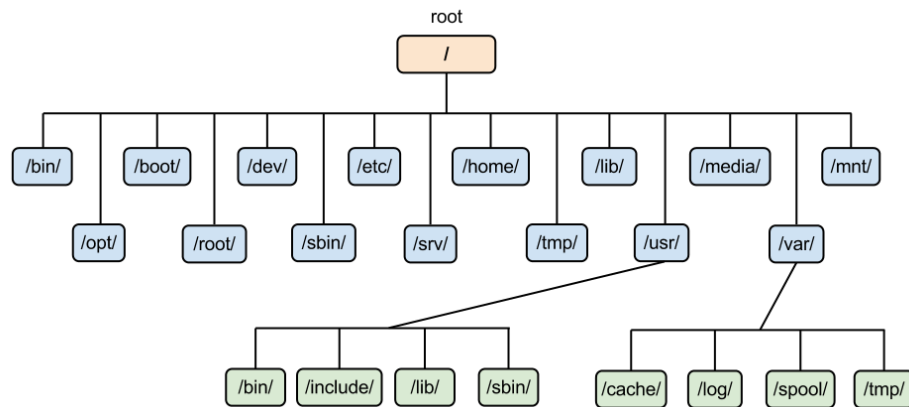
File System: percorso assoluto

- ❑ Per accedere ad un documento si può usare il percorso assoluto o quello relativo
- ❑ Nel **percorso assoluto** si specifica la posizione del documento a partire dalla radice

Supponendo che nella cartella bin di usr ci sia il documento franco.txt

Per accedere nella cartella dove si trova il programma si dovrà digitare nella shell dei comandi il comando di spostamento nella cartella (cd) e il percorso:

cd /usr/bin/



LINUX

File System: percorso relativo

- ❑ Per accedere ad un documento si può usare il percorso assoluto o quello relativo
- ❑ Nel **percorso relativo** si specifica la posizione del documento a partire dalla posizione corrente

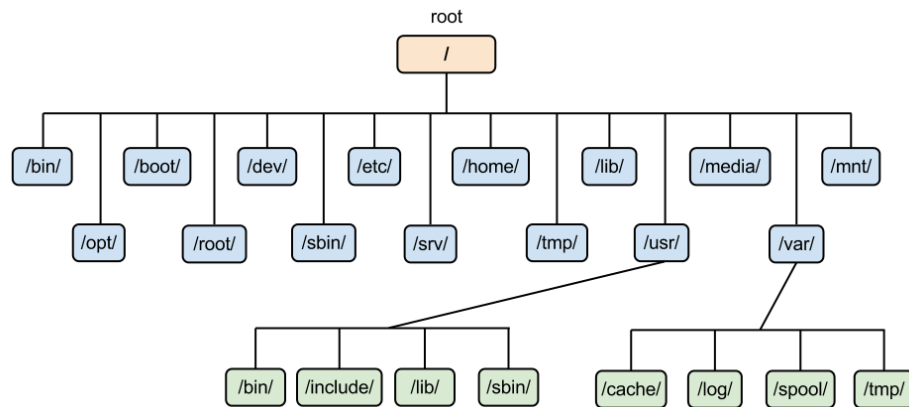
Supponendo che nella cartella bin di usr ci sia il documento franco.txt

Se ci si trova nella cartella

/usr

È sufficiente accedere alla cartella bin con il comando di spostamento:

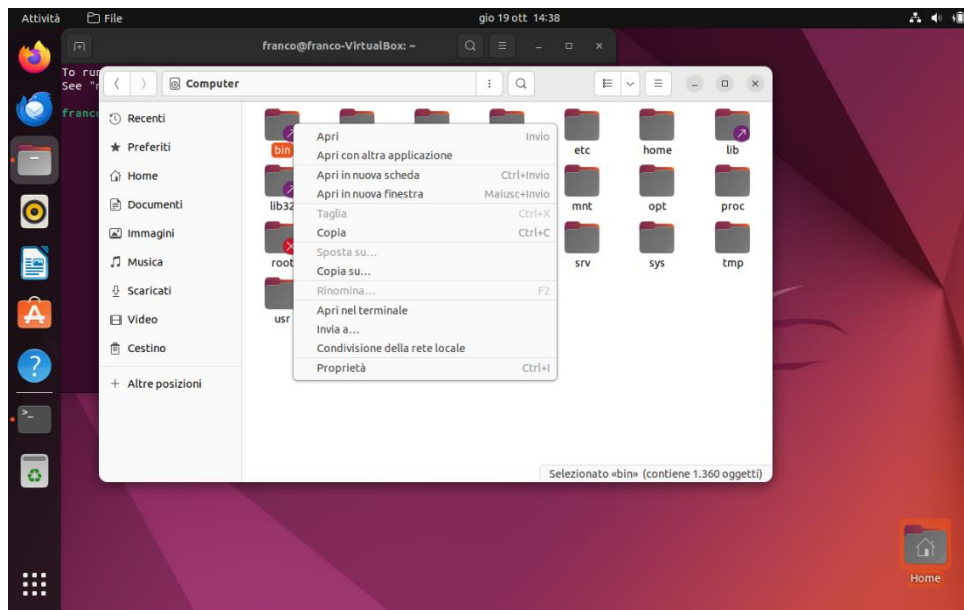
cd /bin



LINUX

File System: accesso alla root da Ubuntu

- ❑ Per vedere la **root** di amministratore da Ubuntu bisogna premere il comando **cd /**
- ❑ In alternativa è possibile andare su home e selezionare l'opzione APRI NEL TERMINALE o salire fino alla radice usando il terminale

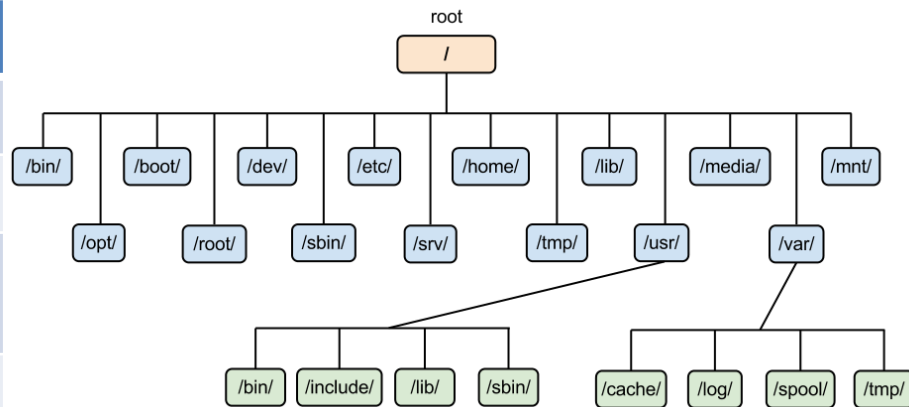


LINUX

File System: significato delle cartelle

❑ Le cartelle hanno il seguente significato:

Luogo	Significato
/home	Cartella degli utenti locali
/lib	Cartella con le librerie di sistema
/lib64	Cartella con le librerie di sistema (per sistemi a 64bit)
/bin	Cartella dei file binari e delle applicazioni binarie
/sbin	Cartella con programmi del sistema
/media	Cartella con i dispositivi removibili (es.: penne USB, HDD esterni)
/mnt	Cartella con i file system (esterni) montati

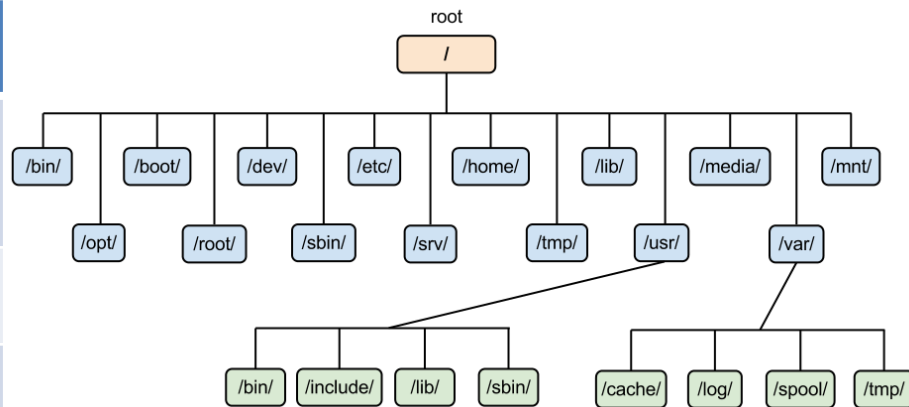


LINUX

File System: significato delle cartelle

❑ Le cartelle hanno il seguente significato:

Luogo	Significato
/boot	Cartella con programmi necessari per avviare il sistema operativo
/run	Cartella con file in esecuzione
/proc	Cartella con informazioni sui processi di sistema in esecuzione
/sys	Cartella dei file di sistema. È usata per visualizzare o configurare i device e i bus del sistema operativo

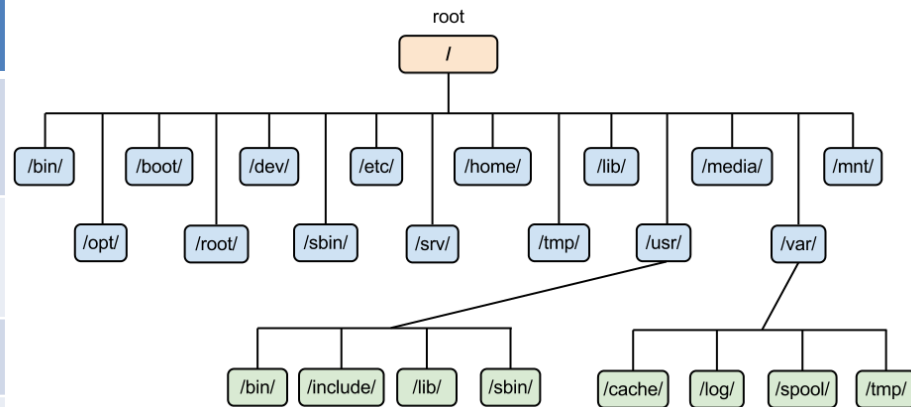


LINUX

File System: significato delle cartelle

❑ Le cartelle hanno il seguente significato:

Luogo	Significato
/etc	Cartella contenente i file di configurazione del sistema e gli script di avvio
/var	Cartella con file che cambiano dinamicamente (es.: database, file di cache)
/tmp	Cartella contenente file temporanei
/dev	Cartella nel quale sono presenti file per accedere ai dispositivi Esempio (per usare il CD) /dev/cdrom
/srv	Cartella che ha i dati serviti dal sistema (es. server web, ftp, ecc.).

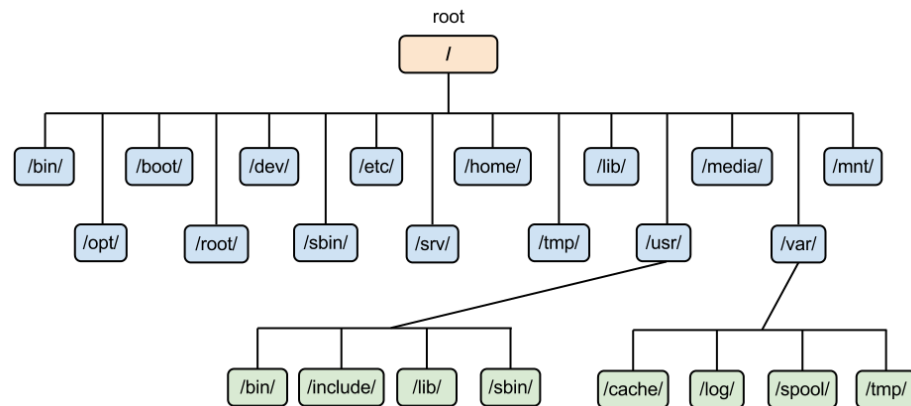


LINUX

File System: significato delle cartelle

❑ Le cartelle hanno il seguente significato:

Luogo	Significato
/usr	Cartella di un utente
/usr/bin	Cartella con gli eseguibili di un utente
/usr/sbin	Cartella con gli eseguibili del sistema
/usr/local	Cartella con applicativi generici
/cgroup	Cartella della gerarchia dei Control Groups



LINUX

File System: muoversi nelle cartelle da shell

- ❑ Nel sistema operativo Linux per **spostarsi tra le directory** si utilizza il comando CD ossia Change Directory.

cd percorso

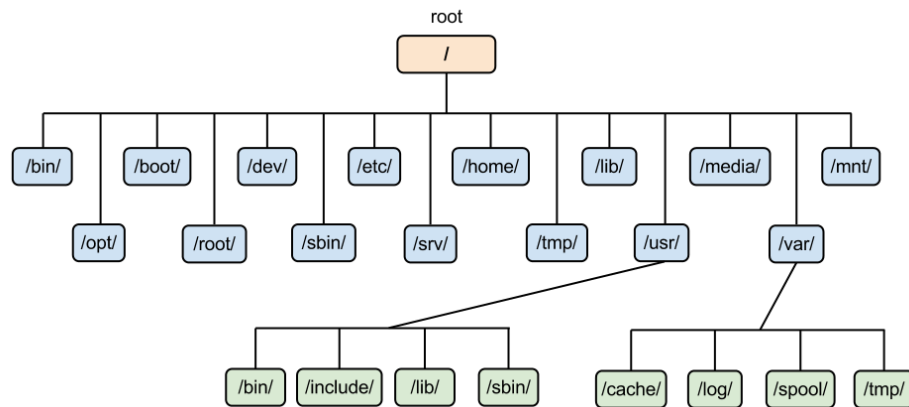
cd directory

- ❑ Ad esempio:

cd /usr/bin

- ❑ Il puntatore si sposta dalla directory attuale a quella di destinazione bin
Se l'utente si fosse trovato in /usr sarebbe bastato scrivere:

cd /bin



LINUX

File System: muoversi nelle cartelle da shell

- ❑ Per **tornare alla directory superiore** bisogna digitare CD seguito da uno spazio e due punti

`cd ..`

- ❑ In Linux un solo punto indica la **directory corrente** (non è molto utile usarlo tuttavia, a volte la sintassi ./ si può trovare nei programmi)

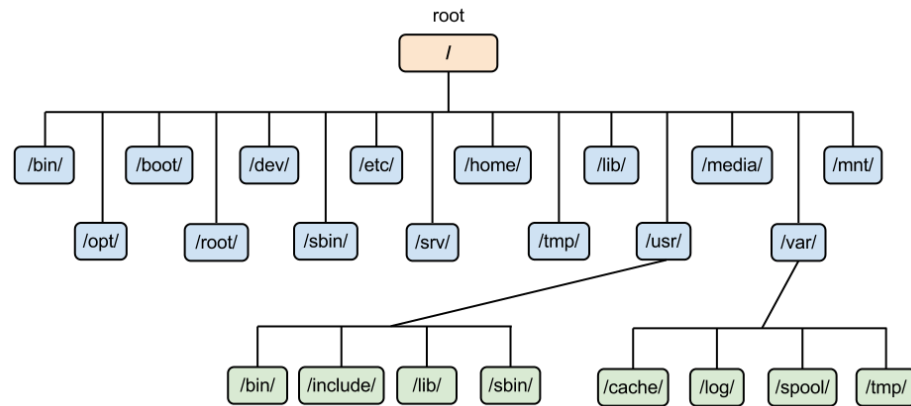
`cd .`

- ❑ Per **spostarsi alla directory root** bisogna digitare sulla linea comando il comando

`cd /`

- ❑ Un altro comando utile che porta automaticamente sulla directory precedente, anche se non è la directory madre è:

`cd -`



`cd /usr/lib`

// Sono in lib

`cd ..`

// sono in usr

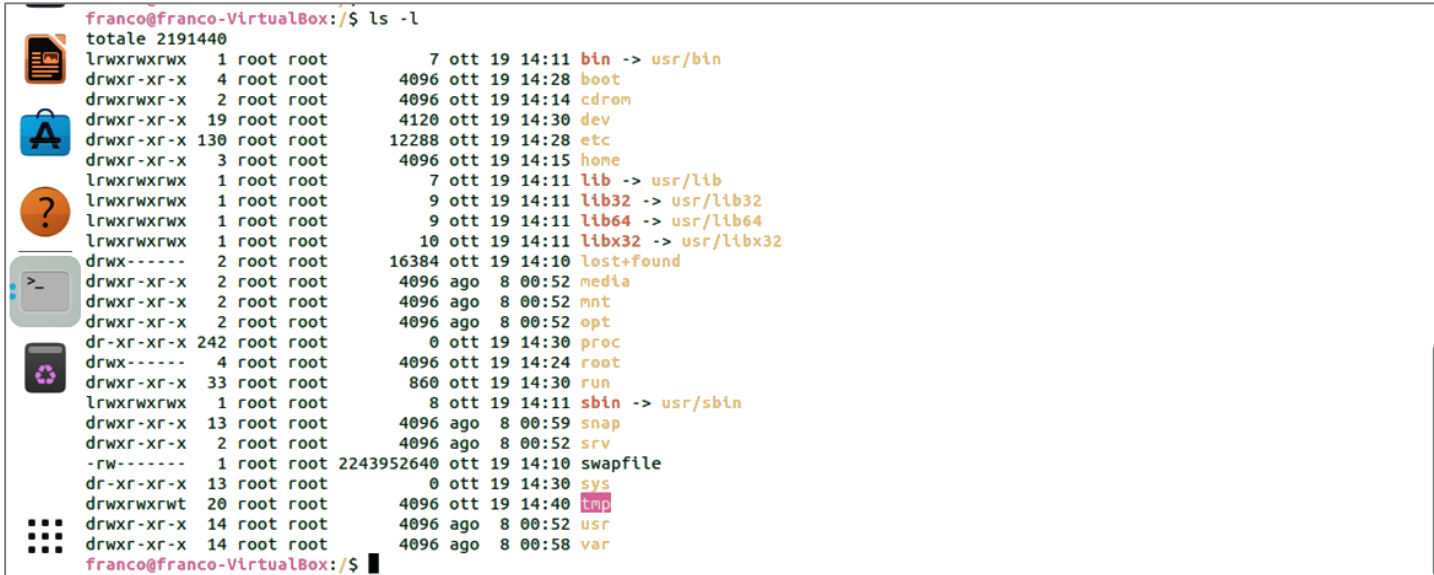
`cd /`

// sono in root

LINUX

File System: INODE

- ❑ Le informazioni sui file e sulle cartelle sono contenute in una tabella (il file descriptor) detta **INODE TABLE**
- ❑ Un INODE è una riga che ha metadati che specificano la dimensione del file, il tempo di creazione e di ultima modifica, nonché i permessi associati a file o cartelle



```
franco@franco-VirtualBox:/$ ls -l
totale 2191440
lrwxrwxrwx 1 root root 7 ott 19 14:11 bin -> usr/bin
drwxr-xr-x 4 root root 4096 ott 19 14:28 boot
drwxrwxr-x 2 root root 4096 ott 19 14:14 cdrom
drwxr-xr-x 19 root root 4120 ott 19 14:30 dev
drwxr-xr-x 130 root root 12288 ott 19 14:28 etc
drwxr-xr-x 3 root root 4096 ott 19 14:15 home
lrwxrwxrwx 1 root root 7 ott 19 14:11 lib -> usr/lib
lrwxrwxrwx 1 root root 9 ott 19 14:11 lib32 -> usr/lib32
lrwxrwxrwx 1 root root 9 ott 19 14:11 lib64 -> usr/lib64
lrwxrwxrwx 1 root root 10 ott 19 14:11 libx32 -> usr/libx32
drwx----- 2 root root 16384 ott 19 14:10 lost+found
drwxr-xr-x 2 root root 4096 ago 8 00:52 media
drwxr-xr-x 2 root root 4096 ago 8 00:52 mnt
drwxr-xr-x 2 root root 4096 ago 8 00:52 opt
dr-xr-xr-x 242 root root 0 ott 19 14:30 proc
drwx----- 4 root root 4096 ott 19 14:24 root
drwxr-xr-x 33 root root 860 ott 19 14:30 run
lrwxrwxrwx 1 root root 8 ott 19 14:11/sbin -> usr/sbin
drwxr-xr-x 13 root root 4096 ago 8 00:59 snap
drwxr-xr-x 2 root root 4096 ago 8 00:52 srv
-rw----- 1 root root 2243952640 ott 19 14:10 swapfile
dr-xr-xr-x 13 root root 0 ott 19 14:30 sys
drwxrwxrwt 20 root root 4096 ott 19 14:40 tmp
drwxr-xr-x 14 root root 4096 ago 8 00:52 usr
drwxr-xr-x 14 root root 4096 ago 8 00:58 var
franco@franco-VirtualBox:/$
```

File System: Permessi

- ❑ Linux ha per ogni file/cartella dei permessi: autorizzazioni concesse all'utente che permettono di fare o non fare alcune operazioni sul file
- ❑ Ogni file/cartella ha tre tipi di permessi:
 - ❖ **Lettura** (*r*, *read*). È il permesso che consente di vedere il contenuto. Nel caso delle directory il permesso *r* consente la lettura della directory.
 - ❖ **Scrittura** (*w*, *write*). È il permesso di scrivere o modificare il contenuto del file. Nel caso delle directory il permesso *w* consente di modificare il contenuto della directory.
 - ❖ **Esecuzione** (*x*, *eXecute*). È il permesso di eseguire il file. In genere, si tratta di un programma eseguibile. Nel caso delle directory il permesso *x* consente di accedere al contenuto dei file e dei metadati delle voci della directory.

A questi si aggiunge il simbolo del trattino (-) che indica la mancanza del permesso

LINUX

File System: Permessi

- ❑ Per **vedere i permessi su file/cartelle** è sufficiente usare il comando

ls -l

- ❑ La **stringa dei permessi** di un file è composta da 10 caratteri

```
franco@franco-VirtualBox:/$ ls -l
totale 2191440
lrwxrwxrwx  1 root root      7 ott 19 14:11 bin -> usr/bin
drwxr-xr-x  4 root root 4096 ott 19 14:28 boot
drwxrwxr-x  2 root root 4096 ott 19 14:14 cdrom
drwxr-xr-x 19 root root 4120 ott 19 14:30 dev
drwxr-xr-x 130 root root 12288 ott 19 14:28 etc
```

La prima posizione indica la tipologia del file :

file normale **-**
directory **d**
link **l**

I tre caratteri dalla seconda alla quarta posizione indicano i permessi dell'utente a cui appartiene il file (il proprietario, **owner**)

lettura **r**
scrittura **w**
esecuzione **x**

I tre caratteri dalla quinta alla settima posizione sono i permessi del gruppo associato al file (il gruppo, **group**)

lettura **r**
scrittura **w**
esecuzione **x**

I tre caratteri dall'ottava alla decima posizione sono i permessi degli altri utenti sul file (altri, **others**)

lettura **r**
scrittura **w**
esecuzione **x**



d rwx r-x r-x r-x



LINUX

File System: Gruppi

- ❑ In Linux esistono 3 profili: **owner**, il proprietario del file, applicazione o cartella; **group**, un gruppo di utenti; e **others** tutti gli altri utenti
- ❑ Sul Sistema Operativo Linux ogni utente è associato a uno o più gruppi. Un **gruppo** è un insieme di utenti che condividono gli stessi permessi su una serie di risorse, file e directory. Anche le risorse (file e directory) sono associate a gruppi
- ❑ I gruppi permettono di escludere gli utenti non autorizzati dall'utilizzo di una risorsa (es. programma, file, directory)

ESEMPIO

Si deve autorizzare la modifica del file "stipendi" soltanto agli utenti che appartengono al gruppo "rsu" (risorse umane).

Si va sul file "stipendi" e si assegna il gruppo "rsu" con il comando:

```
chgrp rsu stipendi.xls
```

Poi si modificano i permessi del file "stipendi", autorizzando la scrittura (w) soltanto al gruppo (g).

```
chmod 775 stipendi.xls
```

Ora il file è modificabile soltanto dagli utenti che appartengono al gruppo "rsu " e dal proprietario del file. Tutti gli altri utenti, invece, possono soltanto leggerlo (r) o eseguirlo (x)

LINUX

File System: modifica dei permessi

- ❑ Per cambiare i permessi di un file tramite il codice numerico, si utilizza il comando **chmod** (*change modality*)
- ❑ Al posto della stringa dei caratteri "rwxrwxrwx" si scrive un numero che ha il significato in **tabella dei permessi**

Esempio:

chmod 777 nomefile

Al file sono assegnati i permessi di lettura (r), scrittura (w) ed esecuzione (x) a tutti, all'utente, al gruppo e a tutti gli altri utenti (rwxrwxrwx).

Tabella dei permessi

Numero decimale	Numero binario	Stringa die permessi	Significato
0	000	---	Nessun permesso
1	001	--x	Solo esecuzione
2	010	-w-	Solo scrittura
3	011	-wx	Lettura e esecuzione
4	100	r--	Lettura
5	101	r-x	Lettura e esecuzione
6	110	rw-	Lettura e scrittura
7	111	rwx	Tutti i permessi

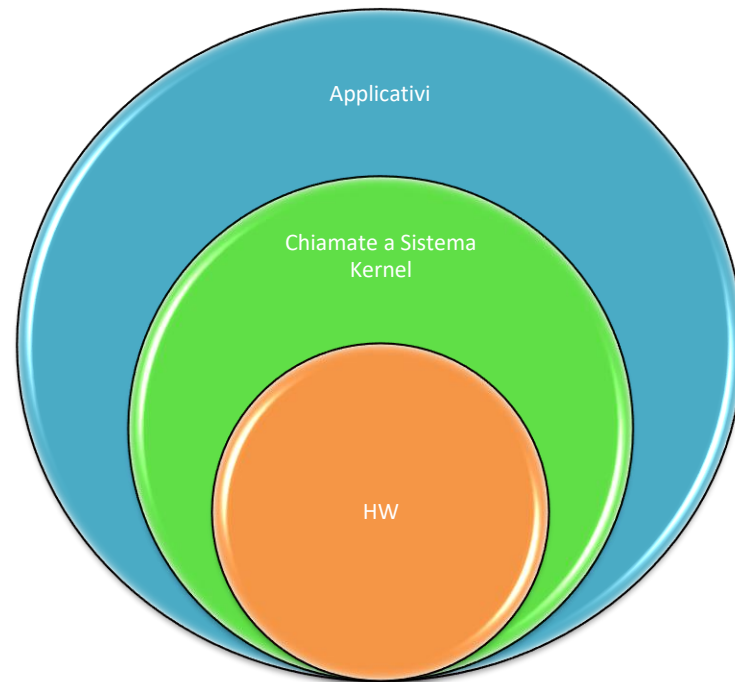


Shell e Comandi

Comandi: generalità

Definizione. Un **comando Linux** è un programma (o un'utilità) che è eseguito da un interprete dei comandi. L'interprete dei comandi in Linux è detto **bash** (o terminale)

- ❑ Il bash è un'interfaccia che accetta stringhe di testo e le elabora in istruzioni che devono essere eseguite dall'elaboratore
- ❑ Qualsiasi interfaccia grafica utente (GUI) è solo un'astrazione dei programmi a riga di comando. Ad esempio, quando si chiude una finestra cliccando sulla "X", si esegue un comando di terminazione del processo





LINUX

Comandi: generalità

- ❑ Un comando può essere corredato da un **argomento** (o parametro) cioè dei dati in ingresso. Nella maggior parte dei casi, l'argomento è il percorso di un file, ma può essere qualsiasi cosa si digiti nel terminale

Ad esempio il comando vi che attiva un editor di testo (simile al Blocco Note di windows) può usare come argomento il file che vuole essere visualizzato

vi divina_comedia.txt

- ❑ Un **flag** è un modo per passare delle opzioni al comando che si esegue. I flag si evocano usando i trattini (-) e i doppi trattini (--), mentre l'esecuzione degli argomenti dipende dall'ordine sono passati alla funzione

Ad esempio il comando ls che mostra i file e le cartelle può sfruttare il flag -l per avere informazioni degli attributi associati

li -l

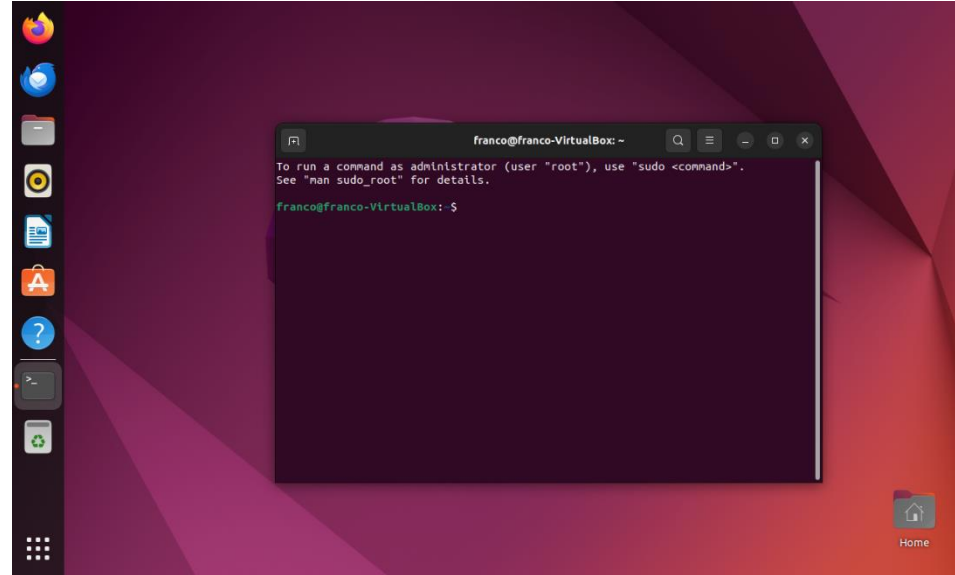
- ❑ È possibile avere **informazioni sui comandi (e relativi argomenti e flag)** usando il programma

help nome_comando

LINUX

Comandi: generalità

- ❑ Per **aprire la bash/terminale** bisogna premere la combinazione di tasti **Ctrl + Alt + T**
 - ❑ Se sull'emulatore Virtual Box non funziona, cercare "Terminale" nel pannello delle applicazioni



LINUX

Comandi: elenco dei file e cartelle (LS)

❑ Il comando **ls** (*list*) permette di elencare il contenuto di una cartella (quella corrente), inclusi i file e le altre directory annidate con i relativi metadati

❑ La sintassi usa dei flag e come argomento il nome del documento o della cartella:

ls [flag] file/directory

Flag options	Descrizione
-l	Informazioni dettagliate su file e cartelle
-a	Riporta file e cartelle (anche nascoste)
-t	Ordinamento di file/cartelle in relazione al tempo di ultima modifica in ordine decrescente
-r	Ordine di presentazione inverso
-S	Ordinamento di file/cartelle ordinate per dimensione (ordine crescente)
-R	Lista di file/cartelle in modo ricorsivo (incluse sottocartelle e relative contenuti)
-i	(inode) riporta l'indice numeric associato ad ogni file e cartella
-g	Raggruppa file/directorie per proprietario
-h	Riporta la dimensione usando una rappresentazione amichevole (e.s.: 1K, 3M, 2G).
-d	Lista delle cartelle

LINUX

Comandi: elenco dei file e cartelle (LS)

```
franco@franco-VirtualBox:/$ ls -l
```

```
totale 2191440
```

lrwxrwxrwx	1	root	root	7	ott 19 14:11	bin -> usr/bin
drwxr-xr-x	4	root	root	4096	ott 19 14:28	boot
drwxrwxr-x	2	root	root	4096	ott 19 14:14	cdrom
drwxr-xr-x	19	root	root	4120	ott 19 14:30	dev
drwxr-xr-x	130	root	root	12288	ott 19 14:28	etc
drwxr-xr-x	3	root	root	4096	ott 19 14:15	home
lrwxrwxrwx	1	root	root	7	ott 19 14:11	lib -> usr/lib
lrwxrwxrwx	1	root	root	9	ott 19 14:11	lib32 -> usr/lib32
lrwxrwxrwx	1	root	root	9	ott 19 14:11	lib64 -> usr/lib64
lrwxrwxrwx	1	root	root	10	ott 19 14:11	libx32 -> usr/libx32
drwx-----	2	root	root	16384	ott 19 14:10	lost+found
drwxr-xr-x	2	root	root	4096	ago 8 00:52	media
drwxr-xr-x	2	root	root	4096	ago 8 00:52	mnt
drwxr-xr-x	2	root	root	4096	ago 8 00:52	opt
dr-xr-xr-x	242	root	root	0	ott 19 14:30	proc
drwx-----	4	root	root	4096	ott 19 14:24	root
drwxr-xr-x	33	root	root	860	ott 19 14:30	run
lrwxrwxrwx	1	root	root	8	ott 19 14:11	sbin -> usr/sbin
drwxr-xr-x	13	root	root	4096	ago 8 00:59	snap
drwxr-xr-x	2	root	root	4096	ago 8 00:52	srv
-rw-----	1	root	root	2243952640	ott 19 14:10	swapfile
dr-xr-xr-x	13	root	root	0	ott 19 14:30	sys
drwxrwxrwt	20	root	root	4096	ott 19 14:40	tmp
drwxr-xr-x	14	root	root	4096	ago 8 00:52	usr
drwxr-xr-x	14	root	root	4096	ago 8 00:58	var

```
franco@franco-VirtualBox:/$
```

Campo 1 Permessi: 1 carattere specifica se si tratta di cartella, link o file 3 caratteri specificano la possibilità di lettura (r), scrittura (w), esecuzione (e) per utente e gruppi e altri.

Campo 2 Numero di collegamenti per il documento

Campo 3 Proprietario: il proprietario del file (nel caso lo username è 'franco')

Campo 4 Gruppo: Il gruppo proprietario del file (nel caso informaticidellatruscia').

Campo 5 Dimensione: specificata in byte

Campo 6 Data e orario di ultima modifica

Campo 7 Nome del documento digitale



LINUX

Comandi: cambio delle directory (CD)

☐ Il comando **cd** (*change directory*) consente di accedere ad una cartella

☐ La sintassi è:

cd [flag] directory (*i flag non sono rilevanti*)

☐ È possibile fornire il percorso assoluto della cartella:

`cd /home/franco/Letteratura/LetteraturaItaliana/`

☐ È possibile fornire il percorso relativo

Se mi trovo nella cartella franco è sufficiente scrivere `cd /Letteratura/LetteraturaItaliana/`

☐ Il comando `cd` ha delle "scorciatoie" molto utili:

☐ Vai alla cartella home

cd

☐ Sposta un livello sopra (si raggiunge la cartella superiore, se si è in una cartella di primo livello si raggiunge la cartella radice)

cd ..

☐ Torna alla directory precedente

cd -

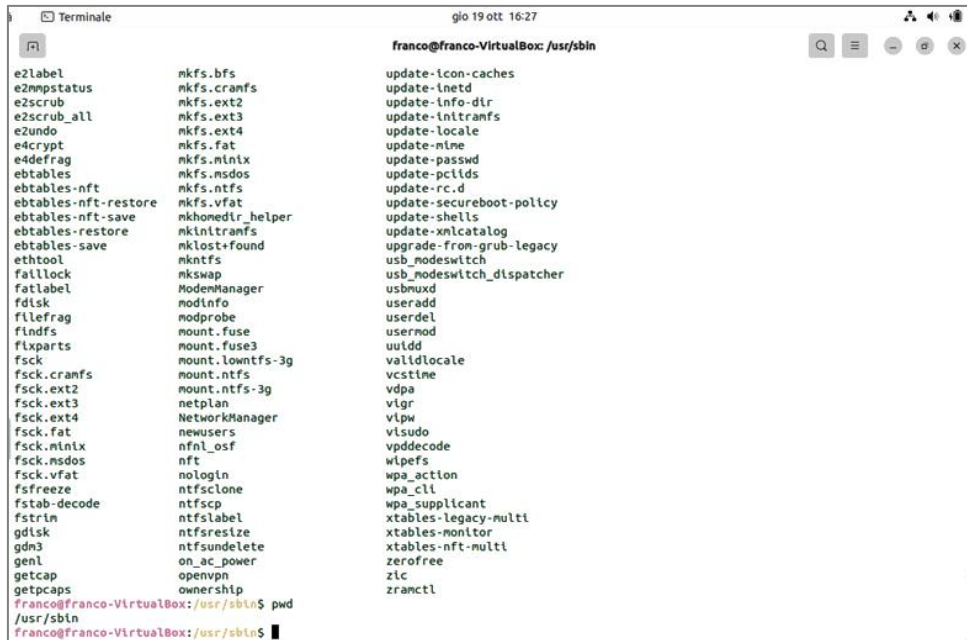
LINUX

Comandi: stampa del percorso attuale (PWD)

- ❑ Il comando **pwd** (*print working directory*) stampa il percorso assoluto della directory in cui l'utente si trova
- ❑ La sintassi è:

pwd [flag]

(i flag non sono rilevanti)



```
Terminale
gio 19 ott 16:27
franco@franco-VirtualBox: /usr/sbin

e2label          nkfs.bfs
e2mmpstatus      nkfs.cramfs
e2scrub          nkfs.ext2
e2scrub_all      nkfs.ext3
e2undo           nkfs.ext4
e4crypt          nkfs.fat
e4defrag         nkfs.minix
ebtables         nkfs.msdos
ebtables-nft     nkfs.ntfs
ebtables-nft-restore nkfs.vfat
ebtables-nft-save nkhnodedir_helper
ebtables-restore nklnitrans
ebtables-save    nklost+found
ethtool          nkntfs
faillock         nkswap
fatlabel         ModemManager
fdisk            modinfo
filefrag         modprobe
findfs           mount.fuse
fixparts         mount.fuse3
fsck             mount.lowntfs-3g
fsck.cramfs      mount.ntfs
fsck.ext2        mount.ntfs-3g
fsck.ext3        netplan
fsck.ext4        NetworkManager
fsck.fat         newusers
fsck.minix       nfnl_osf
fsck.msdos       nft
fsck.vfat        nologin
fsfreeze         ntfsclone
fstab-decode     ntfsck
fstrelin         ntfslabel
gdisk            ntfsresize
gdn3             ntfsundelete
genl             on_ac_power
getcap           openvpn
getpcaps         ownership
franco@franco-VirtualBox: /usr/sbin$ pwd
/usr/sbin
franco@franco-VirtualBox: /usr/sbin$
```

LINUX

Comandi: copia di un documento/cartella (CP)

```
franco@franco-VirtualBox: ~/LetteraturaItaliana

franco@franco-VirtualBox:~/Documenti$ ls
divina_comedia.txt
franco@franco-VirtualBox:~/Documenti$ cp divina_comedia.txt divina_comedia_dante.txt
franco@franco-VirtualBox:~/Documenti$ ls
divina_comedia_dante.txt divina_comedia.txt
franco@franco-VirtualBox:~/Documenti$ cd..
cd..: comando non trovato
franco@franco-VirtualBox:~/Documenti$ cd ..
franco@franco-VirtualBox:~$ cp -r Documenti/ LetteraturaItaliana/
franco@franco-VirtualBox:~$ ls
Documenti LetteraturaItaliana Musica Scaricati snap
Immagini Modelli Pubblici Scrivania Video
franco@franco-VirtualBox:~$ cd LetteraturaItaliana/
franco@franco-VirtualBox:~/LetteraturaItaliana$ ls
divina_comedia_dante.txt divina_comedia.txt
franco@franco-VirtualBox:~/LetteraturaItaliana$
```

LINUX

Comandi: creazione di una cartella (MKDIR)

- ❑ Il comando **mkdir** (*make directory*) crea una cartella

- ❑ La sintassi è:

mkdir [flag] Directory (le options non sono rilevanti)

Per esempio, per creare una cartella nella quale archiviare delle immagini, basta digitare:

mkdir immagini_libro/

- ❑ Per creare delle sottodirectory con un semplice comando, si può ricorrere al flag **-p** (*parent*) :

mkdir -p movies/2004/

Se la cartella già esiste il terminale segnala l'impossibilità di svolgere il comando

LINUX

Comandi: spostamento di file/cartelle (MV)

- ☐ Il comando **mv** (move) è utile per spostare (o rinominare) file e directory
- ☐ La sintassi è:

mv [flag] source_file_name(s) Destination_file_name

Flag/Opzioni	Significato
-i (interactive)	se nella cartella di destinazione c'è un file con lo stesso nome il processo di copia si interrompe e chiede all'utente se vuole sovrascrivere il contenuto
-b (backup)	se nella cartella di destinazione c'è un file con lo stesso nome questo ultimo è rinominato con una estensione differente (si aggiunge ~ alla fine dell'estensione)
-f (force)	si forza la copia
-p (preserve)	si salvano gli attributi/meta informazioni dei file copiati

LINUX

Comandi: spostamento di file/cartelle (MV)

- ❑ Per **spostare un file in una cartella** è sufficiente il comando:

mv file_sorgente cartella_destinazione/
es.: **mv** spoon_river.txt Letteratura/

- ❑ Per utilizzare i percorsi assoluti, bisogna usare:

mv /home/Documenti/paradiso_perduto.txt ./

dove ./ è la directory in cui si trova l'utente nel momento in cui emana il comando

- ❑ Il comando **mv** è utile anche per rinominare i file mantenendoli nella stessa directory:

mv old_name_file.txt new_name_file.txt

Es.: **mv** paradiso_perduto.txt il_paradiso_perduto.txt

LINUX

Comandi: rimozione file o cartelle (RM)

- ❑ Il comando **rm** (*remove file / remove directory*) è sato per rimuovere documenti digitali e cartelle.
- ❑ La sintassi è:

rm [OPTION] FILE/DIRECTORY

es.: **rm** divina_comedia.txt

Questo comando va usato con particolare attenzione perché è molto difficile (ma non impossibile) recuperare i file cancellati in questo modo

- ❑ Per **cancellare una directory vuota** si usa il flag ricorsivo (-r):

rm -r dir_to_remove/

- ❑ Per **rimuovere una directory con dei contenuti al suo interno**, è necessario utilizzare i flag di forzatura (-f, force) e quello ricorsivo (-r, recursive):

es.: **rm -rf** dir_with_content_to_remove/

Questo comando e i flag ricorsivo e forzatura vanno usati con particolare attenzione perché possono cancellare un'intera giornata di lavoro!

LINUX

Comandi: rimozione file definitiva (SHRED)

- ☐ Il comando **shred** cancella un file
- ☐ La sintassi è:

shred [flag] file

Se si vuole cancellare e rendere irrecuperabile un file si può usare il flag -u:

```
shred -u divina_commedia.txt
```

Flag	Significato
-n (iterations)	numero di volte che si opera una sovrascrittura con caratteri casuali sul file (se è omissso sono 3)
-u (remove)	si rimuove il file dopo la sovrascrittura
-v (verbose)	manda messaggi
-z (zero)	pone a zero tutti i byte del file dopo la fase di shredding
-f (force)	forza l'applicazione del comando anche su file protetti
-r (random-source=FILE)	Si usano i caratteri del file specificato per sovrascrivere il file

LINUX

Comandi: gestione cartelle

```
franco@franco-VirtualBox:~$ ls
Documenti  LetteraturaItaliana  Musica  Scaricati  snap
Immagini  Modelli              Pubblici  Scrivania  Video
franco@franco-VirtualBox:~$ mk Letteratura
mk: comando non trovato
franco@franco-VirtualBox:~$ mkdir Letteratura
franco@franco-VirtualBox:~$ ls
Documenti  Letteratura  Modelli  Pubblici  Scrivania  Video
Immagini  LetteraturaItaliana  Musica  Scaricati  snap
franco@franco-VirtualBox:~$ mv LetteraturaItaliana/ Letteratura/
franco@franco-VirtualBox:~$ ls
Documenti  Immagini  Letteratura  Modelli  Musica  Pubblici  Scaricati  Scrivania  snap  Video
franco@franco-VirtualBox:~$ cd Letteratura/
franco@franco-VirtualBox:~/Letteratura$ ls
LetteraturaItaliana
franco@franco-VirtualBox:~/Letteratura$ cd LetteraturaItaliana/
franco@franco-VirtualBox:~/Letteratura/LetteraturaItaliana$ ls
divina_comedia_dante.txt  divina_comedia.txt
franco@franco-VirtualBox:~/Letteratura/LetteraturaItaliana$
```

LINUX

Comandi: copia di un documento/cartella (CP)

- ☐ Il comando **cp** (*copy*) consente la copia di file (o di cartelle).
- ☐ La sintassi è:

cp [flag] Source Destination

cp [flag] Source Directory

cp [flag] Source-1 Source-2 Source-3 Source-n Directory

Flag/Opzioni	Significato
-i (interactive)	se nella cartella di destinazione c'è un file con lo stesso nome il processo di copia si interrompe e chiede all'utente se vuole sovrascrivere il contenuto
-b (backup)	se nella cartella di destinazione c'è un file con lo stesso nome questo ultimo è rinominato con una estensione differente (si aggiunge ~ alla fine dell'estensione)
-f (force)	si forza la copia
-r or -R (recursive)	si copia l'intera struttura della cartella (in altre parole, si includono anche le sotto cartelle)
-p (preserve)	si salvano gli attributi/meta informazioni dei file copiati

LINUX

Comandi: copia di un documento/cartella (CP)

- ❑ La **copia di un file** avviene con il comando:

cp *file_da_copiare nuovo_nome_del_file*

es.: **cp** file_da_copiare.tipo nuovo_file.tipo

- ❑ La **copia di una cartella** ha una sintassi analoga:

cp *cartella_da_copiare/ nuovo_nome_della_cartella/*

es.: **cp** Video/ Video_2023/

- ❑ È possibile **copiare l'intero contenuto** della cartella considerando anche i contenuti delle sottocartelle in essa presenti usando il flag ricorsivo **-r**:

cp -r *cartella_da_copiare/ nuovo_nome_della_cartella/*

es.: **cp -r** Documenti/ Documento_2023/

- ❑ Interessante l'uso della wildcard ***** per copiare file di un tipo:

```
>ls
```

```
a.txt b.txt c.txt d.txt e.txt Folder1
```

```
#Dapprima la cartella Folder1 è vuota
```

```
> cp *.txt Folder1
```

```
> cd Folder1
```

```
>ls
```

```
a.txt b.txt c.txt d.txt e.txt
```

LINUX

Comandi: manuale dei comandi (MAN)

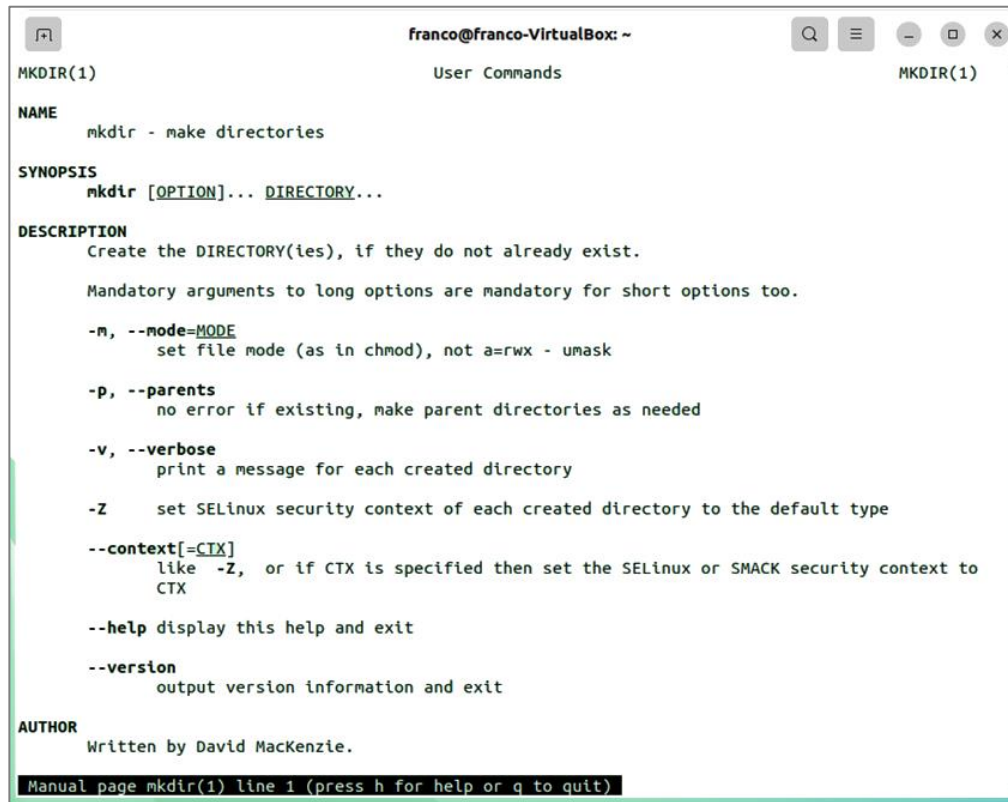
- ❑ Il comando **man** (*manual*) visualizza la pagina di manuale di qualsiasi altro comando (purché ne abbia una)
- ❑ La sintassi è:

man Comand_name

es.: **man** mkdir

Si può fare riferimento alla pagina del manuale man:

man man



```
franco@franco-VirtualBox: ~  
MKDIR(1)  
User Commands  
MKDIR(1)  
  
NAME  
    mkdir - make directories  
  
SYNOPSIS  
    mkdir [OPTION]... DIRECTORY...  
  
DESCRIPTION  
    Create the DIRECTORY(ies), if they do not already exist.  
  
    Mandatory arguments to long options are mandatory for short options too.  
  
    -m, --mode=MODE  
        set file mode (as in chmod), not a=rwx - umask  
  
    -p, --parents  
        no error if existing, make parent directories as needed  
  
    -v, --verbose  
        print a message for each created directory  
  
    -Z      set SELinux security context of each created directory to the default type  
  
    --context[=CTX]  
        like -Z, or if CTX is specified then set the SELinux or SMACK security context to CTX  
  
    --help display this help and exit  
  
    --version  
        output version information and exit  
  
AUTHOR  
    Written by David MacKenzie.  
  
Manual page mkdir(1) line 1 (press h for help or q to quit)
```



LINUX

Comandi: gestione Gruppi

- ❑ Sul Sistema Operativo Unix/Linux ogni utente è associato a uno o più **gruppi**.
- ❑ I gruppi sono liste di utenti che condividono le stesse autorizzazioni e permessi di lettura e scrittura dei files/directory ed esecuzione dei programmi.
- ❑ Per **creare** un nuovo gruppo si usa **groupadd** (comando impiegato dai soli amministratori del sistema).
- ❑ La sintassi è:

groupadd [flag] nomegruppo

Un esempio:

Si vuole creare un nuovo gruppo "LaCriccaDellaTuscia".

Bisogna accedere alla riga comandi come super-user e digitare :

groupadd LaCriccaDellaTuscia



LINUX

Comandi: gestione Gruppi

- ❑ Per **modificare il nome di un gruppo** esistente si usa il comando **groupmod** (comando impiegato dai soli amministratori di sistema)

La sintassi è:

groupmod -n [nuovo nome gruppo] [nome gruppo]

- ❑ Per visualizzare i permessi associati alla propria utenza si usa il comando

groups

- ❑ Per **eliminare un gruppo** sul sistema operativo Linux si utilizza il comando **groupdel**.

La sintassi è:

groupdel [nome gruppo]

*Il comando **groupdel** può essere utilizzato soltanto dagli amministratori di sistema (superuser)*



LINUX

Comandi: gestione Gruppi (aggiunta utenti)

- ❑ Per **aggiungere un utente ad un gruppo** si utilizza il comando **usermod** con il parametro -G (anche in questo caso, per usare il comando è necessario essere amministratore del sistema).

La sintassi è:

```
usermod -G [gruppo] [nomeutente]
```

- ❑ Un utente può essere aggiunto a più gruppi (i gruppi nella lista devono essere separati tra loro da una virgola).

```
usermod -G [gruppo1,gruppo2,...] [nomeutente]
```

- ❑ Un esempio

Si vuole associare l'utente "pippo" ai gruppi "sales" e "ced".

```
usermod -G sales,ced pippo
```

Se si vuole aggiungere alla lista dei gruppi dell'utente "pippo" anche il gruppo "andrea".

```
usermod -aG andrea pippo
```

- ❑ Attenzione. La nuova lista dei gruppi sostituisce quella precedente. Quindi, i gruppi non elencati sono esclusi (se si vuole aggiungere i gruppi a quelli già presenti si deve usare il comando **usermod -aG [gruppi] [nomeutente]**).



LINUX

Comandi: gestione Gruppi (gestione file)

- ❑ Per **cambiare il gruppo associato a un file oppure a una directory** si utilizza il comando **chgrp** (change group). Per modificare il gruppo di un oggetto potrebbe essere necessario avere i permessi di amministratore o super user.

La sintassi è:

chgrp [nome gruppo] [nome file/directory]

Un esempio:

La directory "prova" è assegnata al gruppo "dip" si deve associarla al gruppo "ced" :

chgrp ced prova

Dopo aver eseguito il comando, la directory è stata associata al nuovo gruppo "ced" ma non è più, invece, associata al vecchio gruppo "dip".

- ❑ La stessa procedura consente di cambiare i gruppi a qualsiasi file



LINUX

Comandi: modifica dei permessi (CHMOD)

- ❑ Il comando **chmod** (*change modality*) permette di cambiare il permesso di un file/cartella
- ❑ I permessi di base di un file possono essere:
 - ❑ r (lettura)
 - ❑ w (scrittura)
 - ❑ x (esecuzione)
- ❑ La sintassi nella sua forma più comune è:
chmod [number] file/directory

LINUX

Comandi: modifica della proprietà di accesso (TOUCH)

- ❑ Il comando **touch** permette di aggiornare i tempi di accesso e di modifica dei file specificati
- ❑ La sintassi è:

touch [flag] file

- ❑ Ad esempio, se un file è stato modificato l'ultima volta il 12 aprile, per cambiare la data di modifica di questo file impostando l'ora corrente, bisogna usare il flag **-m**:

touch -m old_file

Dopo il comando la data corrisponde a quella di oggi

- ❑ Tuttavia, nella maggior parte dei casi, il comando touch è usato per creare nuovi file vuoti:

touch new_file_name

```
franco@franco-VirtualBox:~$ ls -l
totale 40
drwxr-xr-x 2 franco franco 4096 ott 19 16:43 Documenti
drwxr-xr-x 2 franco franco 4096 ott 19 14:23 Immagini
drwxrwxr-x 3 franco franco 4096 ott 19 17:14 Letteratura
drwxr-xr-x 2 franco franco 4096 ott 19 14:23 Modelli
drwxr-xr-x 2 franco franco 4096 ott 19 14:23 Musica
drwxr-xr-x 2 franco franco 4096 ott 19 14:23 Pubblici
drwxr-xr-x 2 franco franco 4096 ott 19 14:23 Scaricati
drwxr-xr-x 2 franco franco 4096 ott 19 14:23 Scrivania
drwx----- 3 franco franco 4096 ott 19 14:24 snap
drwxr-xr-x 2 franco franco 4096 ott 19 14:23 Video
franco@franco-VirtualBox:~$ cd Letteratura/
franco@franco-VirtualBox:~/Letteratura$ cd LetteraturaItaliana/
franco@franco-VirtualBox:~/Letteratura/LetteraturaItaliana$ ls
divina_comedia_dante.txt divina_comedia.txt
franco@franco-VirtualBox:~/Letteratura/LetteraturaItaliana$ ls -l
totale 0
-rw-rw-r-- 1 franco franco 0 ott 19 16:44 divina_comedia_dante.txt
-rw-rw-r-- 1 franco franco 0 ott 19 16:44 divina_comedia.txt
franco@franco-VirtualBox:~/Letteratura/LetteraturaItaliana$ touch -m divina_comedia.txt
franco@franco-VirtualBox:~/Letteratura/LetteraturaItaliana$ ls -l
totale 0
-rw-rw-r-- 1 franco franco 0 ott 19 16:44 divina_comedia_dante.txt
-rw-rw-r-- 1 franco franco 0 ott 19 17:34 divina_comedia.txt
franco@franco-VirtualBox:~/Letteratura/LetteraturaItaliana$
```

LINUX

Comandi: attivazione eseguibile (./)

- ❑ La notazione ./ non è un comando ma permette di **attivare un file eseguibile** con qualsiasi interprete installato nel sistema, direttamente dal terminale (è una variante del doppio clic su una icona di un applicativo in un file manager con interfaccia grafica)
- ❑ Si può eseguire un file oggetto in C o uno script Python o un programma purché siano attivi i permessi di esecuzione (x)

FILE: SALUTI.c

```
#include <stdio.h>
```

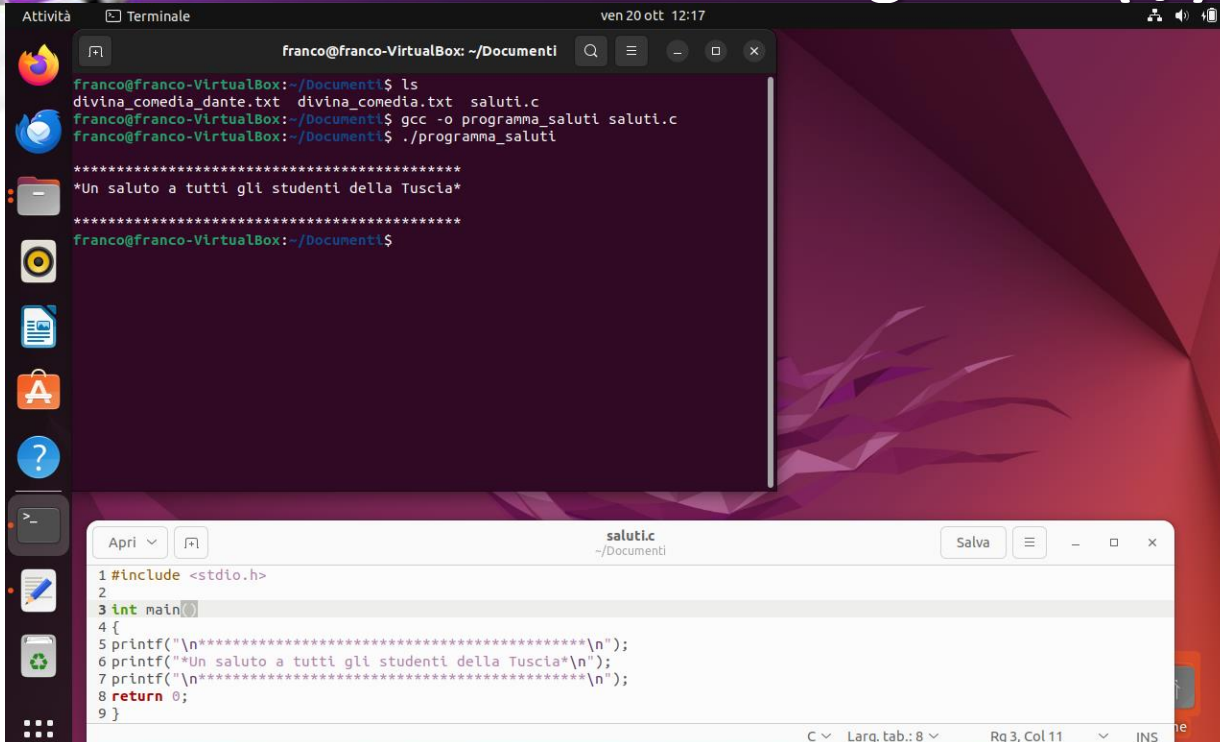
```
int main() {  
    printf("Ciao ragazzi e ragazze della Tuscia");  
    return 0;  
}
```

gcc -o programma_saluti SALUTI.c

./programma_saluti

LINUX

Comandi: attivazione eseguibile (./)



The screenshot shows a Linux desktop environment. On the left is a dock with various application icons. The main area contains two windows. The top window is a terminal titled 'Terminale' with the prompt 'franco@franco-VirtualBox: ~/Documenti'. It shows the execution of several commands: 'ls' to list files, 'gcc -o programma_saluti saluti.c' to compile a C program, and './programma_saluti' to run it. The output of the program is a message: '*Un saluto a tutti gli studenti della Tuscia*'. The bottom window is a code editor titled 'saluti.c' showing the source code of the program. The code includes the standard input/output header, defines a main function, and uses printf to output the greeting message.

```
franco@franco-VirtualBox: ~/Documenti
franco@franco-VirtualBox:~/Documenti$ ls
divina_comedia_dante.txt  divina_comedia.txt  saluti.c
franco@franco-VirtualBox:~/Documenti$ gcc -o programma_saluti saluti.c
franco@franco-VirtualBox:~/Documenti$ ./programma_saluti

*****
*Un saluto a tutti gli studenti della Tuscia*
*****
franco@franco-VirtualBox:~/Documenti$
```

```
#include <stdio.h>
int main()
{
    printf("\n*****\n");
    printf("Un saluto a tutti gli studenti della Tuscia*\n");
    printf("\n*****\n");
    return 0;
}
```

il pacchetto GCC potrebbe non essere installato di default
(bisogna installarlo con il comando **sudo apt install gcc**)



LINUX

Comandi: super utente (SUDO)

- ❑ Il comando **sudo** (*superuser do*) permette di agire come superuser o utente root quando si esegue un comando specifico. È il modo in cui Linux protegge se stesso e impedisce agli utenti di modificare accidentalmente il filesystem della macchina o di installare pacchetti inappropriati

- ❑ La sintassi è:

sudo -V | -h | -l | -v | -k | -K | -s | [-H] [-P] [-S] [-b] | [-p prompt] [-c class|-] [-a auth_type] [-r role] [-t type] [-u username|uid] command

Quando il comando sudo viene usato senza indicare lo username, utilizza di default il super user cioè l'amministratore del sistema.

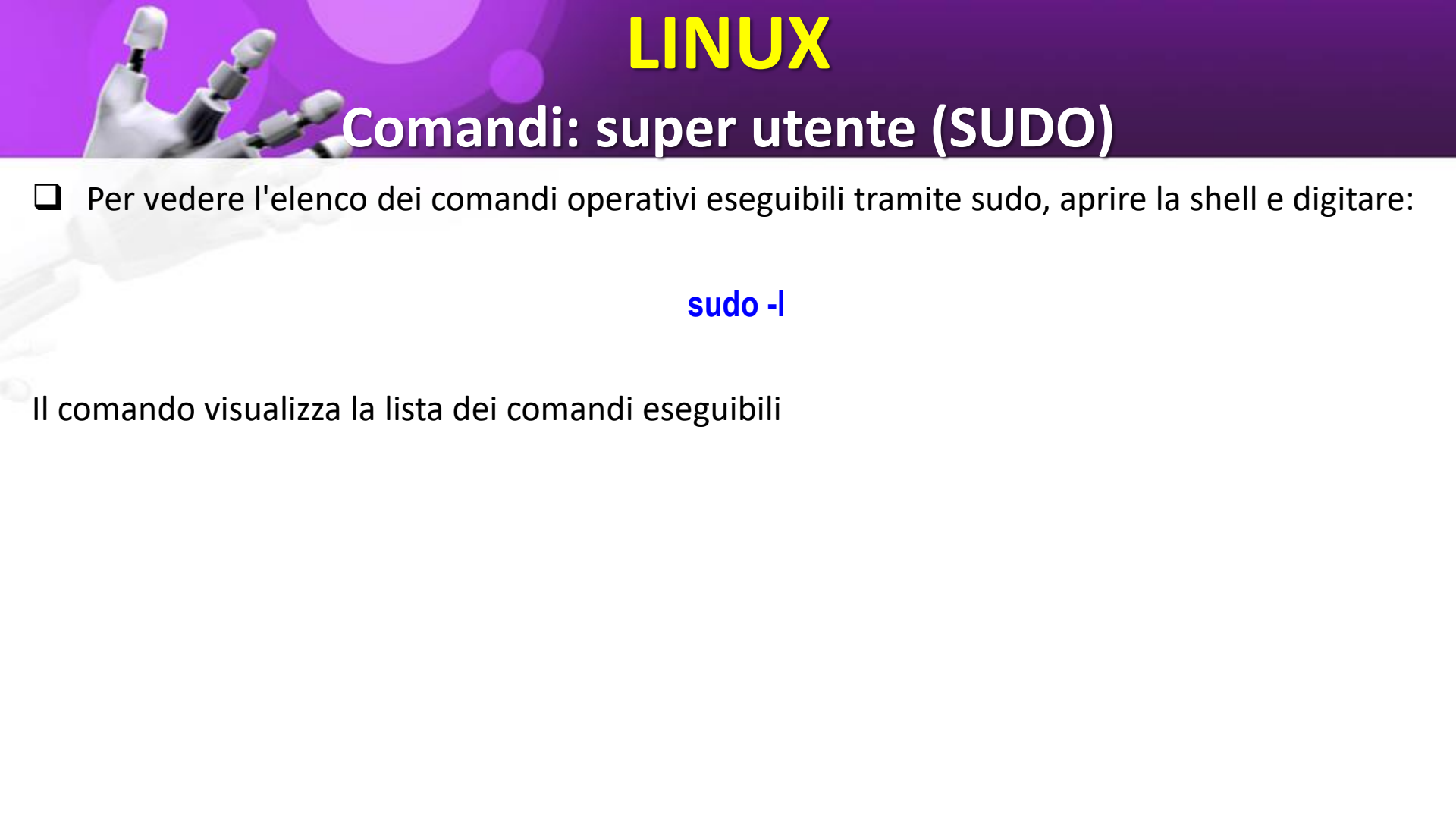
- ❑ Sudo è di solito utilizzato per installare software :

sudo apt install gimp

- ❑ Sudo è usato anche per modificare file al di fuori della home directory dell'utente:

sudo cd /root/

- ❑ Per eseguire il comando bisogna inserire la password di amministratore



LINUX

Comandi: super utente (SUDO)

- ❑ Per vedere l'elenco dei comandi operativi eseguibili tramite sudo, aprire la shell e digitare:

`sudo -l`

Il comando visualizza la lista dei comandi eseguibili



LINUX

Comandi: APT - YUM - PACMAN

- ❑ I comandi **APT**, **YUM** e **PACMAN** sono gestori per installare, aggiornare e rimuovere gli applicativi
- ❑ Negli esempi al lato si può vedere come installare GIMP, un software libero e open source per l'elaborazione delle immagini digitali, sulle diverse versioni di UNIX

Debian (Ubuntu, Linux Mint)

```
sudo apt install gimp
```

Red Hat (Fedora, CentOS)

```
sudo yum install gimp
```

Arch (Manjaro, Arco Linux)

```
sudo pacman -S gimp
```

LINUX

Comandi: gestione delle risorse (HTOP)

- Il comando **htop** permette la visualizzatore di processi in maniera interattiva (equivalente al Gestore delle Risorse di Windows)

- La sintassi è:

htop [-dChusv]

-d (delay): ritarda l'aggiornamento dei dati ogni dieci secondi

-C --no-color --no-colour: mostra l'interfaccia con i colori

-h (help): resituisce i messaggi

-u --user=USERNAME: mostra solo i processi dell'utente specificato

- È utile per gestire le risorse dell'elaboratore direttamente dal terminale

```
0E|          | 3E|          |          | 9E|          | 3
1E|          | 4E|          |          | 10E|         | 1
2E|          | 5E|          |          | 11E|         | 1
Mem|          |          |          |          |          |
Sup|          |          |          |          |          |

Tasks: 334, 780 thr: 2 running
Load average: 1.73 1.33 1.19
Uptime: 34:03:59

PID USER      PRI NI  VIRT  RES   SHW-S CPU%MEMS  TIME: Command
1123 daniel    20   0 390M 7540 21976 5 2.8 0.5 17:04.77 /usr/bin/python /usr/bin/qtile start
726      20   0 6283M 75536 30020 5 1.3 0.5 18:37:14 /usr/lib/korg:0 -seat seat0 -auth /run/lightdm/root:/0 -nolisten tcp vt7 -novtswitch
1239 daniel    9 -11 1003M 14408 9940 5 1.3 0.1 19:49:15 /usr/bin/pulseaudio --daemonize=no --log-target=journal
12657 daniel 20   0 41.8G 1275M 1800 5 1.3 0.8 6:56:41 /usr/lib/brave/brave --type=renderer --field-trial-handle=1383049695608654963,1660138126615002718,131072
380163 daniel 20   0 529M 86128 40580 5 1.3 0.5 8:52:86 alcritty
1399 daniel    20   0 398M 7540 21976 5 0.7 0.5 0:27:02 /usr/bin/python /usr/bin/qtile start
4051 daniel    20   0 1856M 5568 1728 5 0.7 3.5 16:47:11 /usr/lib/brave/brave
4096 daniel    20   0 1339M 437M 131M 5 0.7 2.7 18:05:02 /usr/lib/brave/brave --type=gpu-process --field-trial-handle=1383049695608654963,1660138126615002718,1310
4301 daniel    20   0 398M 110M 68656 5 0.7 0.7 11:26:87 /usr/lib/brave/brave --type=utility --utility-sub-type=network.mojom.NetworkService --field-trial-handle=1
4134 daniel    20   0 1339M 437M 131M 5 0.7 2.7 17:41:35 /usr/lib/brave/brave --type=gpu-process --field-trial-handle=1383049695608654963,1660138126615002718,1310
4383 daniel    20   0 37.3G 150M 90520 5 0.7 0.9 6:39:22 /usr/lib/brave/brave --type=renderer --field-trial-handle=1383049695608654963,1660138126615002718,131072
12651 daniel 20   0 41.8G 1275M 1800 5 0.7 0.8 6:56:41 /usr/lib/brave/brave --type=renderer --field-trial-handle=1383049695608654963,1660138126615002718,131072
386997 daniel 20   0 37.3G 192M 94428 5 0.7 1.2 5:14:14 /usr/lib/brave/brave --type=renderer --field-trial-handle=1383049695608654963,1660138126615002718,131072
387087 daniel 20   0 37.3G 192M 94428 5 0.7 1.2 5:19:35 /usr/lib/brave/brave --type=renderer --field-trial-handle=1383049695608654963,1660138126615002718,131072
387219 daniel 20   0 37.4G 192M 94856 5 0.7 1.0 5:55:13 /usr/lib/brave/brave --type=renderer --field-trial-handle=1383049695608654963,1660138126615002718,131072
900280 daniel 20   0 37.6G 424M 307M 5 0.7 2.7 1:06:45 /usr/lib/brave/brave --type=renderer --field-trial-handle=1383049695608654963,1660138126615002718,131072
962613 daniel 20   0 41.5G 213M 103M 5 0.7 1.3 0:17:04 /usr/lib/brave/brave --type=renderer --field-trial-handle=1383049695608654963,1660138126615002718,131072
963864 daniel 20   0 343M 58668 32556 5 0.7 0.4 0:14:15 /usr/lib/brave/brave --type=utility --utility-sub-type=media.mojom.CdmService --field-trial-handle=1383049
1021985 daniel 20   0 37.3G 127M 97028 5 0.7 0.8 0:00:73 /usr/lib/brave/brave --type=renderer --field-trial-handle=1383049695608654963,1660138126615002718,131072
1022083 daniel 20   0 13864 7992 3590 8 0.7 0.0 0:00:41 htop
1      0 107M 11084 8228 5 0.0 0.1 0:00:80 /sbin/init
399     20 8164 4784 1648 5 0.0 0.0 0:03:77 /usr/bin/havedd -w 1024 -v 1 --foreground
400     20 59944 25640 24660 5 0.0 0.2 0:00:30 /usr/lib/systemd/systemd-journald
401     20 11828 9720 6776 5 0.0 0.1 0:00:36 /usr/lib/systemd/systemd-udev
590     20 12720 5388 4620 5 0.0 0.0 0:00:08 avahi-daemon: running [danielanjanu.local]
591     20 9280 2692 2344 5 0.0 0.0 0:00:00 /usr/bin/crond
592     20 13840 6720 5016 5 0.0 0.0 0:00:46 /usr/bin/dbus-daemon --system --address=systemd --no-fork --nopidfile --systemd-activation --syslog-only
593     20 470M 19968 16900 5 0.0 0.1 0:00:86 /usr/bin/NetworkManager --no-daemon
595     20 2914M 23612 18368 5 0.0 0.1 0:00:12 /usr/lib/polkit-1/polkitd --no-debug
601     20 175M 8184 6900 5 0.0 0.0 0:00:07 /usr/lib/systemd/systemd-logind
602     20 14716 6828 6032 5 0.0 0.0 0:00:07 /usr/lib/systemd/systemd-machined
613     20 12448 688 0 5 0.0 0.0 0:00:00 avahi-daemon: chttp helper
701     20 2914M 23612 18368 5 0.0 0.1 0:00:00 /usr/lib/polkit-1/polkitd --no-debug
Failed to set up a virtual console, because:
Failed to start the vt terminal: No such file or directory
```

Questo pacchetto potrebbe non essere installato di default
(bisogna installarlo con il comando **sudo apt install htop**)

LINUX

Comandi: analisi dei processi (PS)

- ❑ Il comando **ps** (*process status*) mostra i processi attivi
- ❑ La sintassi è :

ps

- ❑ Stampa informazioni utili come ID del processo, TTY (TeleTYpewriter), tempo e nome del comando (è una forma arcaica di HTOP)

```
franco@franco-VirtualBox: ~/Documenti
franco@franco-VirtualBox:~/Documenti$ ps
  PID TTY          TIME CMD
 3708 pts/0    00:00:00 bash
 3921 pts/0    00:00:00 ps
franco@franco-VirtualBox:~/Documenti$
```

LINUX

Comandi: terminazione forzata processo (KILL)

- ❑ Il comando **kill** (killing) elimina un processo
- ❑ La sintassi è:

kill [signal] PID

- ❖ PID indica l'identificatore del processo (process ID) che deve essere terminato
- ❖ [signal] indica il segnale. I più importanti sono SIGTERM (è il segnale preimpostato) e SIGKILL (vanno scritti in maiuscolo): servono per far terminare un programma che non risponde più correttamente
 - ❖ A posto di SIGKILL si può mettere il numero 9

```
franco@franco-VirtualBox: ~  
franco@franco-VirtualBox:~$ ps  
  PID TTY          TIME CMD  
 2793 pts/0    00:00:00 bash  
 2805 pts/0    00:00:00 ps  
franco@franco-VirtualBox:~$ kill -9 2793
```

Comandi: stampa del percorso di un programma (WICH)

- ❑ Il comando **which** restituisce il percorso completo di un programma. Se non lo si trova si restituisce una messaggio
- ❑ La sintassi è:

which program

- ❑ Ad esempio, si può verificare il percorso del programma GCC :

```
which gcc
```

```
# /usr/bin/gcc
```

Comandi: descrizione programma (WHATIS)

- ❑ Il comando **whatis** stampa una singola riga di descrizione di un applicativo
- ❑ La sintassi è:

whatis [flag] program_name

ESEMPIO

```
whatis python
```

```
# python (1) - an interpreted,  
interactive, object-oriented  
programming language
```

```
whatis whatis
```

```
# whatis (1) - display one-line  
manual page descriptions
```

LINUX

Comandi: compressione file (ZIP)

- ❑ Il comando **zip** permette di archiviare il contenuto di una cartella specificata dal path o dal percorso relativo
- ❑ La sintassi è:

zip [flag] file_name.zip files_names

Flag /Opzioni	Significato	Sintassi
-d (Remove files from the archive)	Permette di rimuovere file da un archivio	zip -d [file_name.zip] [files_name]
-u (Update files in the archive)	Aggiunge o aggiorna file presenti in un archivio	zip -u [file_name.zip] [files_name]
-m (Move files into the archive)	aggiunge file in archivio sovrapponendo quelli esistenti con ugual nome	zip -m [file_name.zip] [files_name]
-r (Recursively zip a directory)	include file e sottodirectory della directory che si vuole comprimere	zip -r [file_name.zip] [directory_name]

Questo pacchetto potrebbe non essere installato di default
(bisogna installarlo con il comando **sudo apt install zip**)

Comandi: compressione file (ZIP)

- ❑ Il comando **zip** permette di archiviare il contenuto di una cartella specificata dal path o dal percorso relativo
- ❑ La sintassi è:

zip -r archive_file.zip directory

zip -r nome_file.zip cartella1 cartella2

zip -r archive_file.zip path

- ❑ Ad esempio se mi trovo nella cartella Letteratura posso comprimere i file della sotto-cartella LetteraturaItaliana con il comando:

zip -r testi.zip LetteraturaItaliana/

Il file testi.zip è nella cartella Letteratura



LINUX

Comandi: decompressione file (UNZIP)

- ❑ Il comando **unzip** permette di estrarre il contenuto di un file .zip (un documento di archivio) dal terminale
- ❑ La sintassi è:

```
unzip nome_file.zip
```

Ad esempio per estrarre il contenuto di un file di archivio di tipo zip pieno di immagini:

```
unzip images.zip
```

È possibile anche mettere il contenuto in una cartella con la sintassi:

```
unzip file.zip -d destination_folder
```

Convien usare il flag **-r** per garantire che tutte le sottocartelle siano estratte:

```
unzip file.zip -rd destination_folder
```

Questo pacchetto potrebbe non essere installato di default
(bisogna installarlo con il comando **sudo apt install unzip**)

LINUX

Comandi: compressione/decompressione file (TAR)

- ☐ Il comando "tar" viene utilizzato per creare un documento di archivio ed estrarre i file contenuti in un documenti di archivio. I file di archivio possono essere compressi o no.
- ☐ La sintassi è:

tar [options] file_name.tar file/directory

Flag di Opzioni	Significato
-c	crea un archivio dove inserire file o directory
-x	estrae i file da un archivio compresso
-f	specifica il nome dell'archivio da creare o estrarre
-t	visualizza la lista di file/cartelle in un archivio
-u	aggiunge file in un archivio
-r	aggiunge file direttamente nelle sotto cartelle esistenti
-v	mostra messaggi mentre archivia o decompime
-A	concatena più file di archivio tar in un singolo
-z	usa la compressione GZIP quando si crea un file tar (il risultato è un file con estensione .tar.gz)
-J	usa la compressione BZIP2 quando si crea un file tar (il risultato è un file con estensione .tar.bz2)



LINUX

Comandi: compressione/decompressione file (TAR)

- ❑ Esempio per comprimere un gruppo di file con estensione *.c presenti in una cartella

tar cvf cartella_compressa_soli_file_c.tar *.c

'-c': crea un nuovo archivio

'-v': mostra i file archiviati e mostra segnalazioni varie

'-f': specifica il nome del documento di archiviazione da creare

- ❑ Esempio per estrarre il contenuto di un documento di archivio:

tar xvf cartella_compressa _soli_file_c.tar

'-x': estraie i file da un archivio

'-v': mostra i file estratti e mostra segnalazioni varie

'-f': specifica il nome del documento di archivio

LINUX

Comandi: stampa messaggi (ECHO)

- ❑ Il comando **echo** visualizza un testo definito nel terminale:

- ❑ La sintassi è:

echo message/variable_system

echo " Carissimi studenti della Tuscia"

- ❑ L'impiego principale è quello di stampare le variabili usate nell'ambiente di lavoro :

echo "Ciao \$USER"

Ciao Franco



```
franco@franco-VirtualBox: ~  
franco@franco-VirtualBox:~$ echo "Ciao"  
Ciao  
franco@franco-VirtualBox:~$ echo "Ciao $USER"  
Ciao franco  
franco@franco-VirtualBox:~$
```

LINUX

Comandi: visualizzazione concatenazione file (CAT)

- ❑ Il comando **cat** (*concatenation*) consente di creare, visualizzare e concatenare file direttamente dal terminale

- ❑ La sintassi è:

cat file1 file2 ...

- ❑ È usato principalmente per vedere l'anteprima di un file senza aprire un editor di testo grafico:

cat divina_comedia.txt

```
franco@franco-VirtualBox:~/Documenti$ cat divina_comedia.txt
```

```
Nel mezzo del cammin di nostra vita  
mi ritrovai per una selva oscura,  
ché la diritta via era smarrita.
```

```
Ahi quanto a dir qual era è cosa dura  
esta selva selvaggia e aspra e forte  
che nel pensier rinova la paura!
```

```
Tant' è amara che poco è più morte;  
ma per trattar del ben ch'i' vi trovai,  
dirò de l'altre cose ch'i' v'ho scorte.
```

```
Io non so ben ridir com' i' v'intrai,  
tant' era pien di sonno a quel punto  
che la verace via abbandonai.
```

```
Ma poi ch'i' fui al piè d'un colle giunto,  
là dove terminava quella valle  
che m'avea di paura il cor compunto,
```

```
guardai in alto e vidi le sue spalle  
vestite già de' raggi del pianeta  
che mena dritto altrui per ogni calle.
```

```
Allor fu la paura un poco queta,  
che nel lago del cor m'era durata  
la notte ch'i' passai con tanta pietà.
```

```
E come quei che con lena affannata,  
uscito fuor del pelago a la riva,  
si volge a l'acqua perigliosa e guata,
```

```
così l'animo mio, ch'ancor fuggiva,  
si volse a retro a rimirar lo passo  
che non lasciò già mai persona viva.
```

LINUX

Comandi: visualizzazione file testuale (HEAD-TAIL)

- ❑ Il comando **head** riporta su terminale le prime 10 righe di un file di testo
- ❑ La sintassi è

head [flag] filef

- ❑ È possibile impostare qualsiasi numero di righe che si desidera visualizzare con il flag -n:

head -n 5 long.txt

- ❑ Il comando **tail** è complementare al comando head. Stampa le ultime dieci righe del contenuto di un file (anche in questo caso con il flag n si può estendere il numero di righe)

```
franco@franco-VirtualBox:~/Documenti$ head divina_comedia.txt
```

```
Nel mezzo del cammin di nostra vita  
mi ritrovai per una selva oscura,  
ché la diritta via era smarrita.
```

```
Ahi quanto a dir qual era è cosa dura  
esta selva selvaggia e aspra e forte  
che nel pensier rinova la paura!
```

```
franco@franco-VirtualBox:~/Documenti$ tail divina_comedia.txt
```

```
E io a lui: «Poeta, io ti ricieggio  
per quello Dio che tu non conoscesti,  
a ciò ch'io fugga questo male e peggio,
```

```
che tu mi meni là dov' or dicesti,  
sì ch'io veggia la porta di san Pietro  
e color cui tu fai cotanto mesti».  
eADV
```

```
Allor si mosse, e io li tenni dietro.
```

LINUX

Comandi: statistiche sul contenuto di un file (WC)

- ❑ Il comando **wc** (*word count*) restituisce il numero di parole in un file di testo
- ❑ La sintassi è:

wc [option] file

```
Franco@franco-VirtualBox:~/Documenti$ wc divina_comedia.txt
186  955 5127 divina_comedia.txt
Franco@franco-VirtualBox:~/Documenti$
```

186 linee
955 parole
5127 caratteri

Flag	Significato
-l	stampa il numero di linee
-w	stampa il numero di parole
-m	stampa il numero di caratteri in un file testuale
-L	Stampa il numero di caratteri della linea più lunga



LINUX

Comandi: ricerca file e cartelle(FIND)

- ❑ Il comando **find** cerca i file in una directory sulla base di un'espressione regolare
- ❑ La sintassi è:
find [where to start searching from] [expression determines what to find] [-options] [what to find]

LINUX

Comandi: ricerca file e cartelle (FIND)

- ❑ Per **cercare un file specificando un nome nella cartella corrente**

Ad esempio ricercare: file *divina_comedia.txt* nella directory corrente bisogna scrivere:

```
find ./ -name "divina_comedia.txt"
# ./divina_comedia.txt
```

- ❑ Per **cercare un file specificando un nome in una cartella**

Ad esempio ricercare: file *divina_comedia.txt* in una specifica cartella bisogna scrivere:

```
find /Letteratura/LetteraturaItaliana -name "divina_comedia.txt"
# ./divina_comedia.txt
```

- ❑ Per **cercare i file che finiscono con un'estensione .r** (R Language) nella cartella corrente, si può usare il seguente comando:

```
find ./ -type f -name "*.r"
# ./get_keys.r
# ./github_automation.r
# ./binarysearch.r
```

LINUX

Comandi: ricerca file e cartelle (FIND)

- ❑ Per **cercare un file o una sotto cartella vuota** nella cartella corrente

```
find ./ -empty
```

```
# ./divina_commedia.txt
```

- ❑ Per **cercare un testo all'interno di molteplici file specificati da una estensione**

```
find /Letteratura/LetteraturaItaliana -name "*.txt" -exec grep 'vita' {} \;
```

```
# nel mezzo del cammin di nostra vita
```

LINUX

Comandi: ricerca di parole chiavi (GREP)

- ☐ Il comando **grep** cerca una chiave - cioè una parola, una stringa di caratteri o una espressione regolare - in un file testuale (e le stampa o la evidenzia)
- ☐ La sintassi è:

grep [flag] pattern [files]

Flag	Significato
-c	mostra il numero delle linee in cui si trova la chiave
-h	mostra la linea in cui si trova la chiave (ma non mostra il nome del file)
-i	svolge una ricerca senza considerare la differenza tra lettere maiuscole e minuscole e ne mostra il risultato
-l	mostra la lista dei file che hanno la chiave
-n	mostra le linee e il numero di posizione
-v	mostra le linee che non soddisfano il criterio di ricerca
-E	uso dell'espressione regolare
-w	ricerca l'intera parola
-o	Mostra i risultati su linee diverse
-A n	mostra le linee cercate e le n successive (esiste anche la versione -B n che mostra le linee cercate e le n precedenti e -C n che mostra le linee cercate e le n successive e precedenti)

LINUX

Comandi: ricerca di parole chiavi (GREP)

- ❑ Ricerca di una stringa senza valutare maiuscole e minuscole:

grep -i "ViTa" divina_comedia.txt

```
franco@franco-VirtualBox:~/Documenti$ grep -i "ViTa" divina_comedia.txt
Nel mezzo del cammin di nostra vita
franco@franco-VirtualBox:~/Documenti$
```

- ❑ Ricerca del numero di occorrenze i un carattere:

grep -c "e" divina_comedia.txt

```
franco@franco-VirtualBox:~/Documenti$ grep -c "e" divina_comedia.txt
135
franco@franco-VirtualBox:~/Documenti$
```

- ❑ Ricerca dei file in cui si trova la parola chiave cercata (si cerca nella cartella corrente):

grep -c "e" *

```
franco@franco-VirtualBox:~/Documenti$ grep -l "e" *
divina_comedia.txt
programma_saluti
saluti.c
```

LINUX

Comandi: espressione regolare

- ❑ Un'**espressione regolare** (*regular expression*) è una sequenza di simboli che identifica un insieme di stringhe.
- ❑ Per creare un'espressione regolare occorre utilizzare una sintassi specifica, ossia caratteri speciali e regole di costruzione

Caratteri speciali	Significato
.	indica qualsiasi carattere (escluso un accapo)
*	indica zero o più occorrenze (di un carattere o di un gruppo di caratteri)
?	indica zero o una occorrenza (di un carattere o di un gruppo di caratteri)
{ }	le parentesi graffe, che indicano il numero esatto, o minimo, o massimo, o l'intervallo di occorrenze (di un carattere o di un gruppo di caratteri)
+	indica una o più occorrenze (di un carattere o di un gruppo di caratteri)
^	indica l'inizio della stringa (o, se all'interno di una classe di caratteri, la negazione della stessa)
\$	indica la fine della stringa
	indica l'operatore OR
()	le parentesi tonde, destinate a contenere una sottostringa
[]	le parentesi quadre, destinate a contenere una 'classe' di caratteri

LINUX

Comandi: ricerca di parole chiavi (GREP)

- ❑ Ricerca di una stringa usando l'espressione regolare:

grep -E "*ere" divina_comedia.txt

```
franco@franco-VirtualBox:~/Documenti$ grep -E '*ere' divina_comedia.txt
sì che pareva che l'aere ne tremesse.
«Miserere di me», gridai a lui,
«A te convien tenere altro viaggio»,
franco@franco-VirtualBox:~/Documenti$
```

Comandi: espressione regolare

- ❑ Si riporta una **espressione regolare** che corrisponde a qualsiasi **numero di telefono cellulare** nella forma: +nn-nnn-nnnnnnn:

$^{+}[0-9]\{2\}-[0-9]\{3\}-[0-9]\{7\}$

- ❑ Si riporta una **espressione regolare** che corrisponde ad un **indirizzo di posta elettronica**:

$^{([a-z0-9_.-])+@((([a-z0-9_.-])+.[a-z]\{2,6\})$}$

- ❑ Si riporta un'**espressione regolare** che analizza zero o più **occorrenze di un carattere** in un testo:

t^*lbar

Accetta tlbar, tolbar, toolbar, toolbar t98lbar

- ❑ Si riporta un'espressione regolare che analizza almeno una o più occorrenze di un carattere in un testo:

$t+lbar$

Accetta tolbar, toolbar, toolbar t98lbar NON accetta tlbar



vim

- ❑ Il comando **vim** attiva un editor di testo gratuito e open source per terminali (in uso già dagli anni '90).
- ❑ Permette di modificare file di testo (non è immediato ed è molto macchinoso)
- ❑ Per avviare il programma, basta digitare:



LINUX

Comandi: cambio password utenti (PASSWD)

☐ Il comando **passwd** permette di cambiare le password degli account utente

☐ La sintassi è:

passwd [flag] username

☐ Per la modifica bisogna prima inserire la password corrente, poi la nuova password e la conferma

Flag	Significato
-d (delete)	rende l'account privo di password
-e (expire)	costringe l'utente a reimpostare la password
-i (inactive)	segue un numero che indica i giorni dopo i quali la password e l'account scadono (es.: passwd -i 3 user1)
-l (lock)	blocca la password di un account
-u (unlock)	sblocca la password di un account
-k (keep-tokens)	recupera la password dopo che è scaduta
-S (status)	mostra lo stato della password attraverso sette campi: user1 P 12/22/2018 0 99999 7 3 Il primo campo è la login dell' utente; il secondo indica se l'account ha una password bloccata (L) non ha password (NP) o ha una password in uso (P); il terzo campo esprime la data dell'ultimo cambio di password; il quarto campo indica la durata (in giorni) minima e massima della password, il periodo entro cui sarà mandato un messaggio per indicare il cambio e il periodo di inattività

LINUX

Comandi: prelievo documenti dalla rete (WGET)

- ❑ Il comando **wget** (World Wide Web get) recupera documenti digitali da internet.
- ❑ Dispone di una delle maggiori collezioni di flag
- ❑ Per scaricare un file Python da una repo GitHub:

wget https://raw.githubusercontent.com/DaniDiazTech/Object-Oriented-Programming-in-Python/main/object_oriented_programming/cookies.py

LINUX

Comandi: lettura file system periferiche (MOUNT)

- ❑ Il comando **mount** è utilizzato per montare (collegare) il filesystem di una periferica sul filesystem di Linux (sotto la radice in "/") e renderlo accessibile all'utente
- ❑ La sintassi è:

mount -t [type] [device] [dir]

Il comando indica al kernel di allegare il file system trovato su [device] alla directory [dir].
L'opzione -t [tipo] è facoltativa e descrive il tipo di file system (EXT3, EXT4, BTRFS, XFS, HPFS, VFAT, ecc.)

- ❑ Se la directory di destinazione è omessa, monta i file system elencati nel file /etc/fstab.

flag	Significato
-l	Elenca tutti i file system montati
-h	Visualizza le opzioni per il comando
-a	Monta tutti i dispositivi descritti in /etc/fstab
-t	tipo di file system utilizzato dal dispositivo
-r	modalità di sola lettura montata

Comandi: lettura file system periferiche (MOUNT)

- ❑ ESEMPIO montaggio del file system di una penna USB
- ❑ Creazione di una cartella in cui posizionare il file system
`mkdir /media/usb-drive`
- ❑ Ricercare il file system della penna USB
`fdisk -l` (ad esempio la penna USB è `/dev/sdb1`)
- ❑ Installazione del file system
`sudo mount [identifier] /media/usb-drive`
quindi `sudo mount /dev/sdb1 /media/usb-drive`

Comandi: distacco file system periferiche (UNMOUNT)

- ❑ È possibile utilizzare il comando **umount** per staccare i file system montati

- ❑ La sintassi è

umount [dir]

oppure

umount [device]

- ❑ Esempio

`umount /dev/sdb1`

Si deve essere sicuri che non ci siano file della periferica aperti in uso, altrimenti il comando non funziona e bisogna ricorrere a `fuser [dir]` per vedere i file aperti

- ❑ Esempio

`fuser -m /media/usb-drive`

Comandi: lista dei comandi (HISTORY)

- ❑ Il comando **history** visualizza una lista numerata con i comandi usati in precedenza:
- ❑ La sintassi è:

history

LINUX

Comandi: informazioni sul sistema operativo (NEOFETCH)

- ❑ Il comando Neofetch visualizza informazioni sul sistema operativo in uso (la versione del kernel, la shell e l'hardware) accanto a un logo ASCII della distribuzione Linux:

- ❑ La sintassi è

neofetch

- ❑ Un comando analogo è

uname -a

```
~/Documents/linux-commands via v3.9.6
> neofetch

daniel@danielmanjaro
-----
OS: Manjaro Linux x86_64
Kernel: 5.4.138-1-MANJARO
Uptime: 7 hours, 36 mins
Packages: 1312 (pacman)
Shell: fish 3.3.1
Resolution: 1366x768, 1920x1080
DE: qtile
WM: LG3D
Theme: Matcha-dark-azul [GTK2/3]
Icons: Papyrus-Dark [GTK2/3]
Terminal: alacritty
CPU: AMD Ryzen 5 3600 (12) @ 3.600GHz
GPU: NVIDIA GeForce GT 710
Memory: 3709MiB / 15988MiB
```



LINUX

Comandi: chiusura terminale (EXIT) e elaboratore (SHUTDOWN)

- ☐ Il comando **exit** termina una sessione di shell e, nella maggior parte dei casi, chiudere automaticamente il terminale in uso

- ☐ La sintassi è:

`exit`

- ☐ Il comando **shutdown** permette di spegnere l'elaboratore. Ma può anche essere utilizzato per fermarla e riavviarla

- ☐ La sintassi è:

`shutdown[OPTIONS] [TIME] [MESSAGE]`

- ☐ Per lo spegnimento immediato (l'impostazione predefinita è un minuto) si usa il parametro now:

`shutdown now`

- ☐ Nel caso si voglia programmare lo spegnimento si può specificare l'orario (in un formato di 24 ore):

`shutdown 20:40`

- ☐ Per annullare una precedente chiamata di shutdown, si deve usare il flag -c:

`shutdown -c`

8 MARZO

Oggi la mimosa...
solo se è quella
<https://www.facebook.com/pages/Mafalda/182385161863013>
crema e cioccolato.



<https://www.facebook.com/pages/Mafalda/182385161863013>



Fine