

# Comandi Linux

---

... e altro su Linux ;)

# Cenni storici 1/2

---

- Linux è un sistema operativo free basato su API POSIX e UNIX.
- Nel 1985 Richard Stallman fondò la Free Software Foundation, il cui obiettivo era riassunto nell'acronimo ricorsivo GNU = Gnu's Not Unix.
- In realtà l'obiettivo era la scrittura di un intero Sistema Operativo, analogo a UNIX, con delle caratteristiche innovative e varie applicazioni di corredo, che fosse interamente free, negando così la natura commerciale di UNIX.

# Cenni storici 2/2

---

- Nel 1991 Linus Benedict Torvalds, studente del corso di Sistemi Operativi all'Università di Helsinki, aprì la strada allo sviluppo di LINUX, che ad oggi è forse il segno più tangibile della realizzazione del progetto GNU.

# Il kernel

---

- Il kernel è il cuore di Linux.
- In esso sono implementate le funzioni più importanti del OS...

# Il file-system

---

- In un sistema Linux, o più in generale Unix, l'utente non deve avere conoscenza delle caratteristiche della macchina, in quanto la gestione della macchina è riservata all'amministratore del sistema.
- In questo tipo di sistemi all'utente viene presentato un unico "dispositivo", come fosse un unico disco.

- 
- In realtà, non tutte le installazioni di Linux hanno lo stesso albero delle directory, in particolare ogni distribuzione può presentare delle differenze anche significative.

# Directory radice e /bin

---

- Il file system ha origine in `/`, detta **root** o directory radice.
- Si nota esplicitamente che `root` è anche la username dell'amministratore di sistema, la cui home directory è `/root`.
- **/bin** è la directory che contiene i programmi principali del sistema, quelli che devono essere disponibili subito all'accensione, per poter avviare il sistema.

# /home

---

- Questa directory contiene le 'aree locali' dei singoli utenti.
- Quando un utente si collega facendo login, egli non si trova nella directory principale del sistema (la root /) ma viene posizionato in genere in una directory "privata", diversa per ogni utente.
- Questa directory, detta home, in genere ha il nome dell'utente ed è una sottodirectory della directory home.



# /usr

---

- In questa directory risiede la maggior parte del sistema.
- In usr e nelle sue sottodirectory risiedono tutti i programmi installati, i file del manuale, la documentazione ed altro ancora.
- Una caratteristica di usr è che i suoi file in genere possono essere solo letti.

# /sbin e /usr/sbin

---

- Altre directory importanti sono **/sbin** e **/usr/sbin**;
- sono analoghe a /bin e /usr/bin, la s sta per Superuser ed è riferita al root.
- Queste directory, infatti, contengono soprattutto comandi usati per l'amministrazione di sistema.

# /usr/man

---

- La directory **/usr/man** contiene delle sottodirectory in cui si trovano dei file corrispondenti alle pagine di manuale in linea.

# /etc

---

- La directory etc è una delle più importanti del sistema perché contiene la quasi totalità dei file di configurazione del sistema, compresi quelli che servono per la fase di accensione.

# /lib

---

- Questa directory contiene le librerie di sistema, cioè quegli archivi di funzioni utilizzati da tutti.
- Troveremo in questa directory le librerie di funzioni utilizzate dal linguaggio C, richiamate da tutti i programmi compilati in questo linguaggio, ma anche varie librerie utilizzate da altri linguaggi o sistemi.

# /mnt e /proc

---

- **/mnt** è un'altra directory vuota, come la cdrom o la floppy, predisposta per poterci montare un qualunque dispositivo.
- Anche **/proc** è una directory usata per montare un filesystem. Questo filesystem è però un filesystem virtuale, che non corrisponde a spazio su nessun dispositivo, ma consente di accedere alle informazioni su tutto il sistema, come tipo di CPU, quantità di memoria, dispositivi disponibili e altro.

# /tmp e /var

---

- **/tmp**: in questa directory vengono immagazzinati i file temporanei durante le elaborazioni. Il contenuto di questa directory viene cancellato ad ogni accensione.
- La directory **/var** contiene, nelle sue sottodirectory, i file che registrano gli eventi del sistema (i cosiddetti log). Oltre ai log in questa directory troviamo le code di stampa ed in genere tutti i file di sistema che vengono modificati.

# Devices

---

- **/cdrom** è una directory vuota, alla quale l'amministratore di sistema potrà collegare (montare) un disco inserito nel lettore CD-Rom.
- Come la directory **/cdrom**, **/floppy** è una directory vuota, predisposta perché l'amministratore di sistema vi "monti" un floppy disk.
- **/dev** contiene dei file speciali che vengono usati per indicare al sistema i dispositivi.



# Dispositivi e file speciali

---

- Per riferirsi ai dispositivi (floppy, partizioni, etc.), Linux fa uso di **file speciali**.
- Tali file sono visibili e in genere si trovano nella directory /dev. Alcuni esempi sono:
  - **fd0**: corrisponde al floppy A
  - **fd1**: corrisponde al floppy B
  - **hda, hdb, hdc, hdd** sono i quattro possibili dischi fissi IDE.
  - **hda1, hda2, hda3, hda4** sono le quattro partizioni primarie del primo disco fisso IDE.

# Utenti

---

- Esistono differenze fra i vari utenti, definite dai permessi e dall'accesso ai file e comandi che un'utente può lanciare.
- E' convenzione che i semplici utenti possano scrivere, leggere e modificare file solo all'interno del loro ambiente (home) e lanciare semplici comandi che non influiscono sulla configurazione del sistema.

# Superutenti

---

- Per poter accedere completamente alle risorse del sistema bisogna accedere al sistema come superuser, detto root.
- L'utente root è l'amministratore del sistema con tutti i poteri che comporta questo ruolo, quindi è molto importante che l'accesso a root sia limitato solo al reale amministratore di sistema (sysadmin).

# Utenti e gruppi

---

- Prima di accedere al sistema occorre autenticarsi indicando un nome di utente conosciuto dal sistema.
- Gli utenti, a loro volta, appartengono a **gruppi**.
- Ogni utente deve appartenere almeno ad un gruppo, ma può appartenere a più gruppi.

# Autenticazione 1/2

---

- La prima cosa che si nota quando si avvia un sistema Linux, se si è abituati ad un sistema DOS o Windows, è il fatto che il sistema non è immediatamente disponibile, ma si presenta con una richiesta:

*<host> login:*

# Autenticazione 2/2

---

- Con questa richiesta il sistema ci chiede di autenticarci. Alla richiesta di login noi dovremo rispondere con un nome di utente noto al sistema. Una volta introdotto il nome, il sistema ci chiederà:

*password:*

# Diritti di accesso

---

- Il nome dell'utente ed il nome del gruppo di appartenenza (o meglio, l'identificativo dell'utente e l'identificativo del gruppo o **user id** e **group id**) sono alla base dello schema di protezione dei file di Linux.
- Per ogni file ed ogni directory abbiamo tre possibili diritti indipendenti:
  - diritto di lettura R
  - diritto di scrittura W
  - diritto di esecuzione X

# Lettura e scrittura

---

- Se su un file ho diritto di lettura, ovviamente lo posso leggere. Se ho diritto di lettura su di una directory posso listarne i file.
- Se ho diritto di scrittura su di un file posso modificarne il contenuto o cancellarlo. Se ho diritto di scrittura su di una directory posso creare e cancellare file in quella directory.



# Esecuzione

---

- Se ho diritto di esecuzione su di un file, posso eseguirlo. Naturalmente per eseguire un file esso deve essere un programma o uno script.
- Per programma si intende un file contenete istruzioni eseguibili dal processore, mentre per script si intende un file di testo che può essere eseguito da qualche interprete, come ad esempio l'interprete di comandi o shell.

# La shell

---

- La shell è un programma che gestisce la comunicazione fra utente e sistema operativo interpretando ed eseguendo i comandi dell'utente (la shell viene chiamata anche *command interpreter*).
- Su gran parte delle distribuzioni Linux è preimpostata di default la shell *bash*.

# I comandi fondamentali

---

- **ls** : consente di elencare i file in una directory. Se non si indica nessun parametro, elenca i soli nomi di tutti i file della directory corrente.
- Il parametro **-l** consente di vedere l'elenco in modo esteso. Oltre al nome vengono indicati i diritti, il proprietario, il gruppo la lunghezza e la data.

# ls: esempio 1/2

---

- Ad esempio, nella seguente riga di ls -l:  
drwxrw-r-- 1 nobody nogroup 625 Jan  
12 1998 source
  - La d indica che source è una directory.
  - Le lettere rwx indicano i diritti del proprietario
  - Le lettere rw- indicano i diritti del gruppo del file
  - Le lettere r-- indicano i diritti del resto del mondo

# man

---

- **man**

- consente di consultare il manuale in linea.
- Per poter leggere la pagina del manuale che parla di `ls`, ad esempio, si può usare il comando:

```
man ls
```

# pwd e cd

---

- **pwd**

- sta per print working directory, vale a dire, "stampa la directory di lavoro".

- **cd**

- sta per change directory; serve per cambiare la directory di lavoro corrente.

# Creare e cancellare directory

---

- **mkdir**

- make directory serve per creare una nuova directory.

- **rmdir**

- remove directory serve per cancellare una directory. Funziona solo con le directory vuote.

# Copiare file

---

## ■ cp

- copy serve per copiare file o gruppi di file.
- Richiede sempre una sorgente e una destinazione.
- Se non si indica la destinazione, viene dato un messaggio di errore.
- La sorgente può essere un unico nome di file o una lista.
- La destinazione può essere un nome di file (solo nel caso la sorgente indichi un unico file) oppure un nome di directory.



# Spostare e cancellare file

---

## ■ **mv**

- move serve per spostare o cambiare nome ad uno o più file. La sintassi è identica a quella del comando cp.

## ■ **rm**

- remove serve per cancellare uno o più file. Con l'opzione **-r** (ricorsivo) consente di cancellare anche le sottodirectory, compreso il loro contenuto.

# Utenti

---

## ■ **who**

- stampa il nome di tutti gli utenti attualmente connessi al sistema.
- Oltre al nome degli utenti ci indica su che terminale sono connessi e quando si sono collegati.

## ■ **whoami**

- stampa il nome con il quale ci siamo connessi al sistema.
- È utile se abbiamo la possibilità di connetterci al sistema con nomi diversi.

# Stato dei programmi

---

## ■ `ps`

- consente di visualizzare lo stato dei programmi presenti nel sistema.
- Lanciato senza parametri, visualizza i dati dei programmi associati con l'utente che ha lanciato `ps`.
- Se invece usiamo il parametro `-a`, il comando visualizza tutti i processi del sistema, anche quelli associati ad altri terminali o non associati a nessun terminale (i cosiddetti daemon).

# kill

---

## ■ **kill**

- serve per inviare messaggi ai processi.
- Il primo parametro di kill è il messaggio da inviare, i parametri seguenti sono i pid (identificatori di processo) dei processi che devono ricevere il messaggio.
- I messaggi che possono essere inviati sono vari. I più importanti sono SIGTERM e SIGKILL (vanno scritti in maiuscolo) che servono per far terminare un programma che non risponde più correttamente .

# Montare dispositivi

---

## ■ **mount**

- è un comando che può essere usato solo dall'amministratore di sistema (l'utente root).
- Questo comando serve per rendere disponibile il disco inserito in un dispositivo.
- Ad esempio, per utilizzare un dischetto inserito nel floppy A: (che in Linux si chiama /dev/fd0) e vederne il contenuto sotto la directory /floppy dovremo inviare il comando:

```
mount /dev/fd0 /floppy
```

# Smontare dispositivi

---

## ■ **umount**

- può sempre essere usato solo dall'amministratore di sistema.
- Serve per rimuovere un dispositivo.
- Umount richiede un unico parametro, che può essere il nome del dispositivo (ad esempio /dev/fd0) o il nome della directory in cui è montato (ad esempio /floppy).

# Links

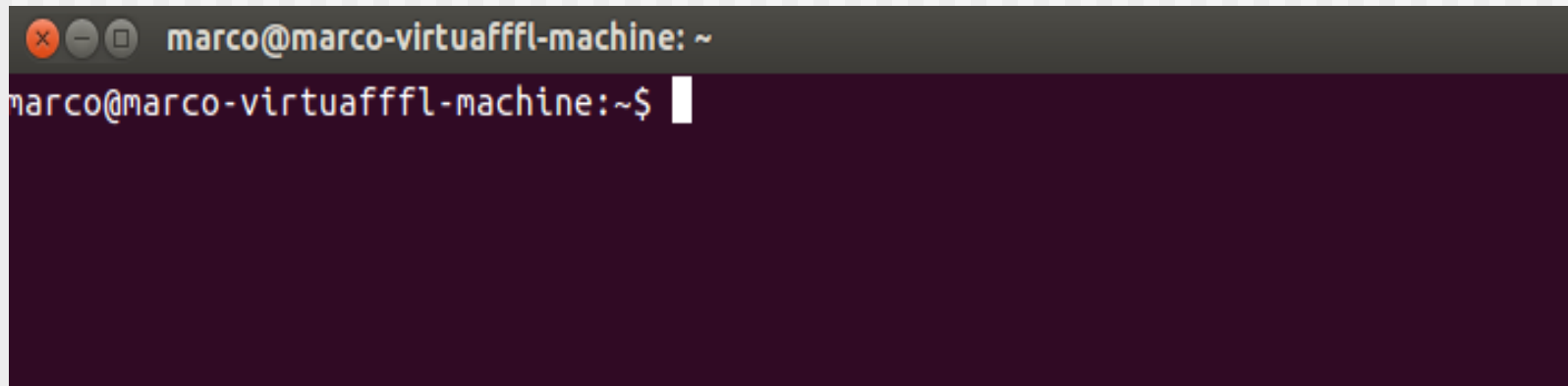
---

- Ubuntu: [www.ubuntulinux.org](http://www.ubuntulinux.org)
  - Distribuzione semplice e veloce da installare disponibile anche in versione live.

# La Shell di Linux

---

- In Linux, oltre all'interfaccia grafica, è possibile utilizzare la riga di comando
- Il terminale visualizza il nome del login (in questo esempio, marco), il nome host del computer (qui, marco-virtuafffl-machine), e il percorso corrente.



```
marco@marco-virtuafffl-machine: ~
```

The image shows a terminal window with a dark background. The title bar at the top contains three window control icons (close, minimize, maximize) and the text 'marco@marco-virtuafffl-machine: ~'. The main area of the terminal displays the prompt 'marco@marco-virtuafffl-machine:~\$' followed by a white cursor block.



# Comandi Linux

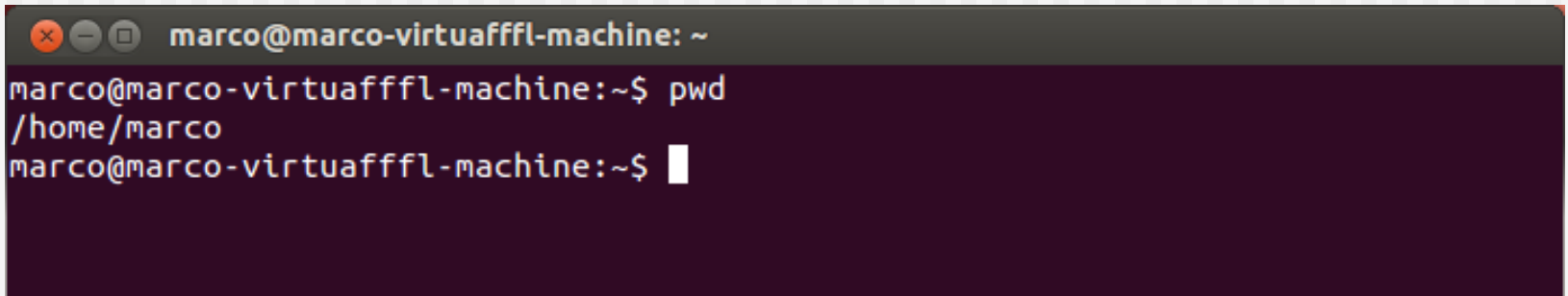
---

- Un comando è composto da vari elementi.
- Il primo elemento è sempre il comando effettivo, seguito da parametri od opzioni.
- È possibile digitare un comando e modificarlo, utilizzando ←, →, ←—, Canc e Space.
- È possibile anche aggiungere opzioni o correggere errori di battitura.
- Il comando viene eseguito premendo Invio (Enter).

# Il comando `pwd`

---

- Il comando `pwd` (Present Working Directory) fornisce il percorso assoluto della directory in cui ci si trova in quel momento.

A terminal window with a dark purple background. The title bar shows 'marco@marco-virtuafffl-machine: ~'. The terminal text shows the command 'pwd' being executed, resulting in the output '/home/marco'.

```
marco@marco-virtuafffl-machine: ~  
marco@marco-virtuafffl-machine:~$ pwd  
/home/marco  
marco@marco-virtuafffl-machine:~$
```

# Percorso assoluto e relativo

---

- Per individuare un file o una cartella determinati, occorre specificare il percorso che vi conduce. Vi sono due modi per specificare un percorso:
  - Il percorso completo (assoluto) dalla directory root al file rispettivo. Comincia sempre con uno /
  - Un percorso che parte dalla directory corrente (percorso relativo)
- Il Linux fa distinzione tra caratteri maiuscoli e minuscoli. Ad esempio, i file `test.txt` e `Test.txt` in Linux sono due file diversi

# Comando ls

- Il comando `ls` (list segments, elenca segmenti), utilizzato per elencare il contenuto di una directory. Il comando può essere utilizzato con o senza opzioni. Immettendo semplicemente `ls` il comando mostra i contenuti della directory corrente.

```
marco@marco-virtuafffl-machine: ~  
marco@marco-virtuafffl-machine:~$ ls  
Documenti      Modelli      OpenCV-2.4.2.tar.bz2  Scaricati  
examples.desktop  Musica      prova                Scrivania  
Immagini       OpenCV-2.4.2  Pubblici             Video  
marco@marco-virtuafffl-machine:~$
```

# Comando ls

---

- In Linux i file possono, ma non necessariamente devono, avere un'estensione.
- In questo modo è difficile differenziare tra file e cartelle.
- Di default, i colori possono venire in aiuto dell'utente: le directory sono solitamente visualizzate in blu, mentre i file in nero se lo sfondo è bianco, o in bianco se lo sfondo è nero.

# Comando ls con opzioni

---

- Un modo migliore di ottenere più dettagli sul contenuto di una directory è tramite il comando ls con una stringa di opzioni.
- Le opzioni modificano il modo di funzionamento di un comando in maniera tale da poter compiere operazioni specifiche.
- Le opzioni sono separate dal comando da uno spazio e sono prefissate da un trattino.

# Comando ls -l

- Es. il comando `ls -l` mostra i contenuti della stessa directory in modo dettagliato (formato listato lungo).

```
marco@marco-virtuafffl-machine: ~
marco@marco-virtuafffl-machine:~$ ls -l
totale 51384
drwxr-xr-x  3 marco marco    4096 apr 11 10:23 Documenti
-rw-r--r--  1 marco marco    8445 ott 15 17:38 examples.desktop
drwxr-xr-x  3 marco marco    4096 apr 11 10:22 Immagini
drwxr-xr-x  2 marco marco    4096 ott 15 17:45 Modelli
drwxr-xr-x  2 marco marco    4096 ott 15 17:45 Musica
drwx----- 13 marco marco    4096 ott 24 14:29 OpenCV-2.4.2
-rw-rw-r--  1 marco marco 52562911 ott 24 14:25 OpenCV-2.4.2.tar.bz2
drwxrwxr-x  2 marco marco    4096 apr 11 10:04 prova
drwxr-xr-x  2 marco marco    4096 ott 15 17:45 Pubblici
drwxr-xr-x  2 marco marco    4096 ott 24 14:26 Scaricati
drwxr-xr-x  2 marco marco    4096 ott 24 14:23 Scrivania
drwxr-xr-x  2 marco marco    4096 ott 15 17:45 Video
marco@marco-virtuafffl-machine:~$
```

# Comando ls -l

---

- A sinistra di ciascun nome di oggetto, vengono visualizzate su diverse colonne informazioni relative all'oggetto stesso. I più importanti sono i seguenti:
  - La prima colonna mostra il tipo di file dell'oggetto (in questo esempio, d per directory oppure - per i file normali).
  - Le successive nove colonne mostrano le autorizzazioni dell'utente relative all'oggetto.
  - Le colonne 11 e 12 mostrano il nome del proprietario del file e il gruppo, mentre la colonna successiva mostra la dimensione del file in byte.
  - Infine, vengono visualizzate data e ora dell'ultima modifica.



# Permessi

---

- Su ogni file in Linux possono essere definiti tre tipi di permessi:
  - Lettura
  - Scrittura
  - Esecuzione
- I permessi vanno definiti per:
  - Utente
  - Gruppo
  - Altri

# Il comando cd

- Per passare da una directory ad un'altra si usa il comando *cd* (change directory) seguito dal nome della directory o dal percorso assoluto
- Per ritornare alla directory superiore si usa *cd ..*
- *cd* senza opzioni mi riporta alla home

```
marco@marco-virtuafffl-machine: ~  
marco@marco-virtuafffl-machine:~$ ls  
Documenti      Modelli      OpenCV-2.4.2.tar.bz2  Scrivania  
examples.desktop  Musica      Pubblici          Video  
Immagini      OpenCV-2.4.2  Scaricati  
marco@marco-virtuafffl-machine:~$ cd Immagini/  
marco@marco-virtuafffl-machine:~/Immagini$ ls  
1.png 2.png 3.png 4.png Foto  
marco@marco-virtuafffl-machine:~/Immagini$ cd ..  
marco@marco-virtuafffl-machine:~$ ls  
Documenti      Modelli      OpenCV-2.4.2.tar.bz2  Scrivania  
examples.desktop  Musica      Pubblici          Video  
Immagini      OpenCV-2.4.2  Scaricati  
marco@marco-virtuafffl-machine:~$ █
```

# Cronologia e Completamento

- Di default, la shell "ricorda" i comandi digitati. Questa caratteristica è denominata cronologia. Per ripetere un comando precedentemente inserito, premere ↑ (freccia verso l'alto) fino a quando non appare il comando.
- Il completamento di un nome di file o di directory dopo la digitazione delle prime lettere è un'altra utile funzione della shell di Linux. Per eseguire questa operazione, digitare le prime lettere, quindi premere →| (TAB)

# Creare una directory: mkdir

- Per creare una directory si usa il comando *mkdir* seguito dal nome della directory (se la si vuole creare all'interno della directory corrente) o dall'intero percorso se la si vuole creare in un'altra zona del disco.

```
marco@marco-virtuafffl-machine: ~/primaDirectory
marco@marco-virtuafffl-machine:~$ ls
Documenti      Modelli      OpenCV-2.4.2.tar.bz2  Scrivania
examples.desktop  Musica      Pubblici           Video
Immagini      OpenCV-2.4.2  Scaricati
marco@marco-virtuafffl-machine:~$ mkdir primaDirectory
marco@marco-virtuafffl-machine:~$ ls
Documenti      Modelli      OpenCV-2.4.2.tar.bz2  Scaricati
examples.desktop  Musica      primaDirectory        Scrivania
Immagini      OpenCV-2.4.2  Pubblici           Video
marco@marco-virtuafffl-machine:~$ mkdir primaDirectory/secondaDirectory
marco@marco-virtuafffl-machine:~$ cd primaDirectory/
marco@marco-virtuafffl-machine:~/primaDirectory$ ls
secondaDirectory
marco@marco-virtuafffl-machine:~/primaDirectory$
```

# Creare un file: touch

- Per creare un file vuoto si usa il comando *touch* seguito dal nome del file o dal percorso

```
marco@marco-virtuafffl-machine: ~/primaDirectory/secondaDirectory
marco@marco-virtuafffl-machine:~$ ls
Documenti      Modelli      OpenCV-2.4.2.tar.bz2  Scaricati
examples.desktop  Musica      primaDirectory        Scrivania
Immagini       OpenCV-2.4.2  Pubblici              Video
marco@marco-virtuafffl-machine:~$ cd primaDirectory/
marco@marco-virtuafffl-machine:~/primaDirectory$ ls
secondaDirectory
marco@marco-virtuafffl-machine:~/primaDirectory$ touch secondaDirectory/file.txt

marco@marco-virtuafffl-machine:~/primaDirectory$ cd secondaDirectory/
marco@marco-virtuafffl-machine:~/primaDirectory/secondaDirectory$ ls
file.txt
marco@marco-virtuafffl-machine:~/primaDirectory/secondaDirectory$
```

# Eliminare una directory o un file

---

- Se la directory è vuota si può usare il comando *rmdir* seguito dal nome della directory o dall'intero percorso
- Se non è vuota si può usare il comando *rm -r* nomeDirectory (la *r* indica la ricorsione, elimina la cartella e tutto ciò che c'è all'interno)
- Per rimuovere un file usare il comando *rm* seguito dal nome del file

# Eliminare directory e file: esempio

```
marco@marco-virtuafffl-machine: ~
marco@marco-virtuafffl-machine:~/primaDirectory/secondaDirectory$ ls
file.txt
marco@marco-virtuafffl-machine:~/primaDirectory/secondaDirectory$ rm file.txt
marco@marco-virtuafffl-machine:~/primaDirectory/secondaDirectory$ ls
marco@marco-virtuafffl-machine:~/primaDirectory/secondaDirectory$ cd ..
marco@marco-virtuafffl-machine:~/primaDirectory$ rmdir secondaDirectory/
marco@marco-virtuafffl-machine:~/primaDirectory$ ls
marco@marco-virtuafffl-machine:~/primaDirectory$ touch file.txt
marco@marco-virtuafffl-machine:~/primaDirectory$ ls
file.txt
marco@marco-virtuafffl-machine:~/primaDirectory$ cd ..
marco@marco-virtuafffl-machine:~$ ls
Documenti      Modelli      OpenCV-2.4.2.tar.bz2  Scaricati
examples.desktop  Musica      primaDirectory        Scrivania
Immagini      OpenCV-2.4.2  Pubblici              Video
marco@marco-virtuafffl-machine:~$ rmdir primaDirectory/
rmdir: rimozione di "primaDirectory/" non riuscita: Directory non vuota
marco@marco-virtuafffl-machine:~$ rm -r primaDirectory/
marco@marco-virtuafffl-machine:~$ ls
Documenti      Modelli      OpenCV-2.4.2.tar.bz2  Scrivania
examples.desktop  Musica      Pubblici              Video
Immagini      OpenCV-2.4.2  Scaricati
marco@marco-virtuafffl-machine:~$
```

# Copiare e/o spostare un file

---

- Per copiare un file, si usa il comando  
*cp File directoryDestinazione*
- Per copiare una directory si usa il comando:  
*cp -r nomeDirectory1 directoryDestinazione*
- Per spostare un file si usa il comando  
*mv nomeFile1 percorso*
- Per spostare una directory si usa il comando  
*mv nomeDirectory percorso*

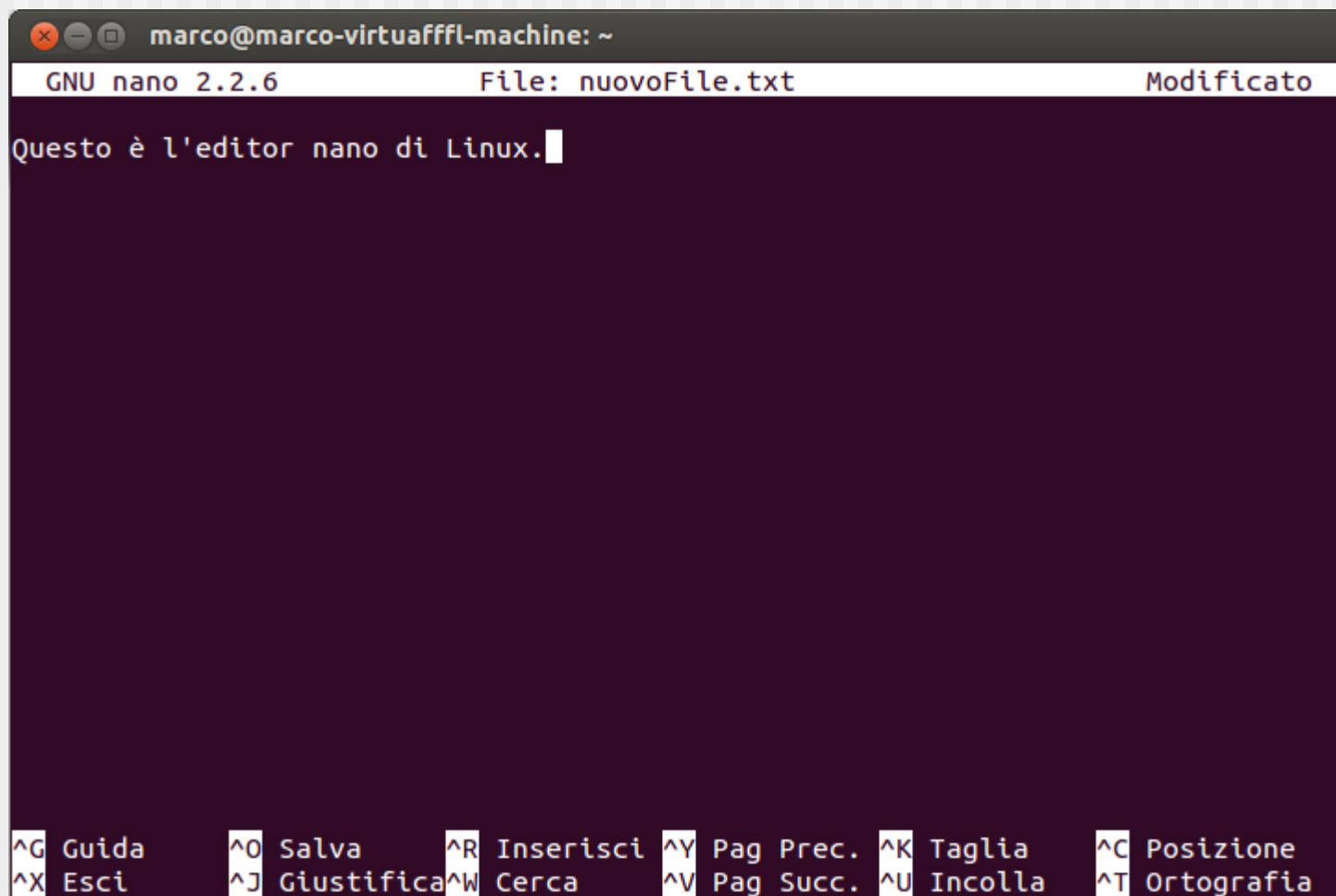


# Editare un file di testo: nano

- Un editor testuale di Linux molto semplice da utilizzare è **nano**
  - Permette di creare, visualizzare e modificare file di testo
  - Per avviare l'editor, digitare *nano* seguito dal nome del file

```
marco@marco-virtuafffl-machine: ~  
marco@marco-virtuafffl-machine:~$ ls  
Documenti      Modelli      OpenCV-2.4.2.tar.bz2  Scrivania  
examples.desktop  Musica      Pubblici      Video  
Immagini      OpenCV-2.4.2  Scaricati  
marco@marco-virtuafffl-machine:~$ nano nuovoFile.txt
```

# L'editor nano



```
marco@marco-virtuaafffl-machine: ~
GNU nano 2.2.6 File: nuovoFile.txt Modificato
Questo è l'editor nano di Linux.
^G Guida      ^O Salva      ^R Inserisci  ^Y Pag Prec. ^K Taglia     ^C Posizione
^X Esci       ^J Giustifica ^W Cerca     ^V Pag Succ. ^U Incolla    ^T Ortografia
```

# L'editor nano: comandi principali

---

- Salvare: `ctrl+o`
- Uscire: `ctrl+x`
- Per copiare una riga: `alt + 6`
- Per tagliare una riga: `ctrl+k`
- Per incollarla: `ctrl+u`
- Per selezionare solo una parte del testo, posizionarsi all'inizio della parte da selezionare, utilizzare i tasti `alt+a` e spostare il cursore, poi copiare (`alt+6`) o tagliare (`ctrl+k`)

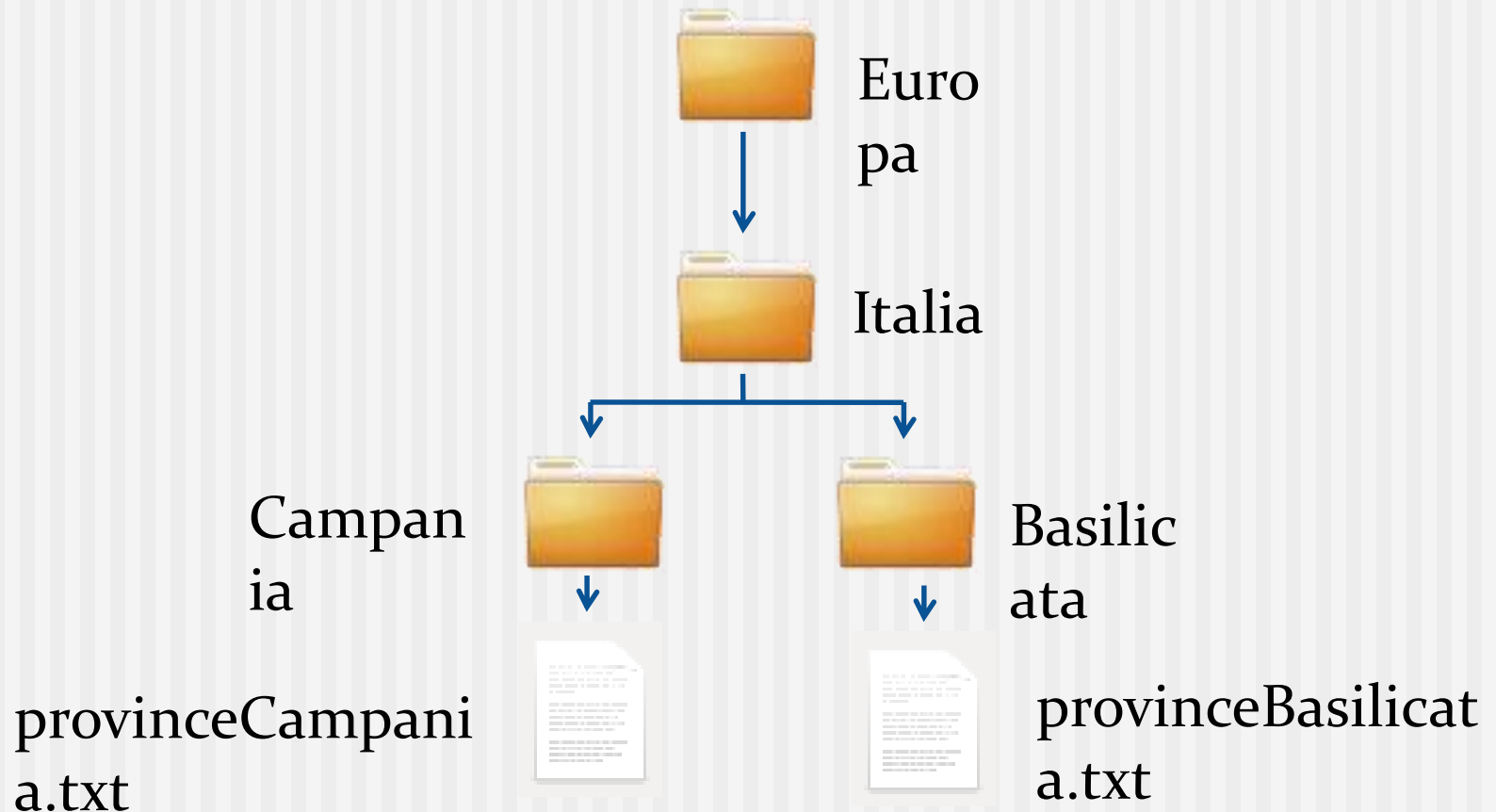
# Compilare ed eseguire un file

- Per compilare un file si può utilizzare il compilatore gcc di linux. La sintassi è: *gcc nomeFileDaCompilare -o NomeFileDiOutput*
- Per eseguire il file: *./nomeFileDiOutput*

```
marco@marco-virtuaafffl-machine: ~  
marco@marco-virtuaafffl-machine:~$ ls  
Documenti      Immagini  OpenCV-2.4.2      Scaricati  Video  
examples.desktop  Modelli   OpenCV-2.4.2.tar.bz2  Scrivania  
helloWorld.c    Musica    Pubblici           usb  
marco@marco-virtuaafffl-machine:~$ gcc helloWorld.c -o helloWorld.o  
marco@marco-virtuaafffl-machine:~$ ls  
Documenti      helloWorld.o  Musica            Pubblici  usb  
examples.desktop  Immagini      OpenCV-2.4.2      Scaricati  Video  
helloWorld.c     Modelli       OpenCV-2.4.2.tar.bz2  Scrivania  
marco@marco-virtuaafffl-machine:~$ ./helloWorld.o  
Ciao mondo  
marco@marco-virtuaafffl-machine:~$
```

# Esercizio 1

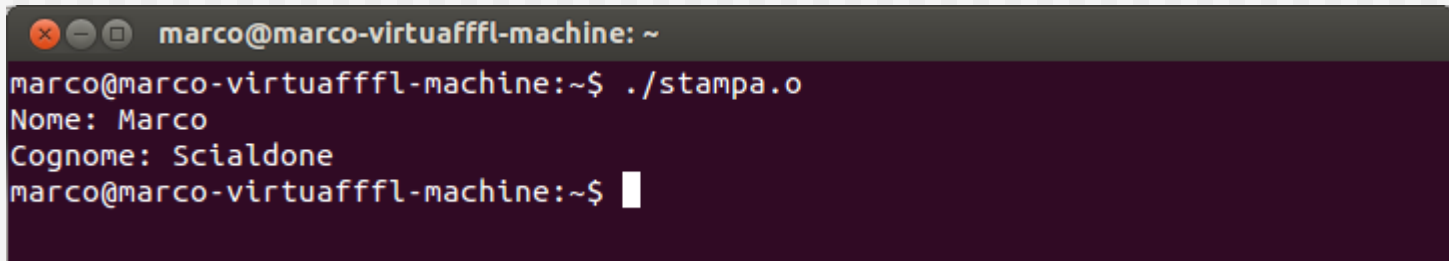
---



# Esercizio 2

---

- Creare un programma in c che stampi a video il vostro nome e cognome, come mostrato in figura:



```
marco@marco-virtuafffl-machine: ~  
marco@marco-virtuafffl-machine:~$ ./stampa.o  
Nome: Marco  
Cognome: Scialdone  
marco@marco-virtuafffl-machine:~$
```