

Linux

LINUX E' TUTTO UN FILE!!

Linux in sé è un nucleo (kernel) di un sistema operativo, sviluppato originariamente da Linus Torvalds nel 1991. Il kernel è la parte centrale di un sistema operativo che gestisce le risorse hardware e consente la comunicazione tra il software e l'hardware.

Il Sistema Operativo Completo

Quando parliamo di "Linux" come sistema operativo, ci riferiamo a un sistema operativo completo che include il kernel Linux insieme a una varietà di software aggiuntivo che ne permette l'utilizzo. Questo software aggiuntivo include programmi, librerie, interfacce utente, e strumenti di gestione.

Spesso, un sistema operativo Linux è anche chiamato distribuzione (o distro), che è una combinazione del kernel Linux e vari pacchetti software. Alcuni esempi di distribuzioni Linux includono:

Ubuntu

Debian

Fedora

Arch Linux

Quindi, mentre Linux è tecnicamente il kernel, quando si parla di "Linux" nel linguaggio comune, si intende generalmente una distribuzione Linux, ovvero un sistema operativo completo.

Sintesi:

Linux = il kernel (parte centrale del sistema operativo).

Sistema operativo Linux = il kernel Linux + software aggiuntivo (un intero sistema operativo).

In breve, Linux è un sistema operativo, ma più precisamente è un nucleo di un sistema operativo, e quando si parla di "Linux" come sistema operativo, ci si riferisce alla combinazione del kernel e del software associato.

BASH: Una **Bash** (Bourne Again SHell) è un interprete di comandi e un linguaggio di scripting utilizzato nei sistemi operativi basati su Unix, come Linux e macOS. È una delle shell più diffuse e serve per interagire con il sistema operativo attraverso il terminale.

Con Bash puoi:

- **Eseguire comandi** come `ls`, `cd`, `mkdir`, `rm`, ecc.
- **Automatizzare operazioni** creando script (`.sh`) che eseguono una serie di comandi.
- **Usare variabili, condizioni, cicli e funzioni** per scrivere script complessi.

Kernel: oggetto che si occupa di parlare con il sistema operativo (in cuore del computer insomma)

È il software che gestisce direttamente le risorse hardware del computer e fornisce un'interfaccia tra l'hardware e i programmi in esecuzione.

Ubuntu: **Ubuntu** è una distribuzione Linux basata su **Debian**, sviluppata da **Canonical Ltd.** È una delle distribuzioni più popolari grazie alla sua facilità d'uso, stabilità e ampia comunità di supporto.

Distribuzione mantenuta da CANONICA, rimane gratuita perché l'azienda si fa pagare quando si rompe qualcosa e col supporto, oppure con le versioni che includono funzionalità più complesse magari per le aziende.

Noi usiamo una ubuntu 24.10 cioè del 2024 di ottobre (è garantita 5 anni)

```
tommaferre@ubu01:~$
```

```
username@hostname:_tilde_tipodishell
```

username: chi ha fatto il login

hostname: in che macchina si sta lavorando

tilde: indica la home

il dollaro: indica con quale potenza si sta lavorando (\$) utente normale

su linux esiste anche l'utente root cioè un super utente che può fare qualsiasi cosa compresi i danni (#)

Struttura della gran parte dei comandi su linux, ma anche su altre shell, è composta da:
comando [opzioni] [argomenti]

- il comando Il nome dell'istruzione (es. ``ls``, ``cd``, ``mkdir``).
- un parametro/opzione/switch/flag in genere `-qualcosa` modificano il comportamento del comando (es. ``-l``, ``-a``)
- l'argomento del comando, se non lo passiamo si riferisce al posto dove siamo quindi `ls -l` ad esempio farà la lista della directory nella quale ci troviamo

I comandi possono essere suddivisi in:

1. **Comandi interni (built-in):** Sono integrati nella shell (es. `cd`, `echo`, `pwd`).
2. **Comandi esterni:** Sono eseguibili salvati in `/bin`, `/usr/bin` (es. `ls`, `grep`, `nano`).

Per sapere se un comando è interno o esterno, si usa:

```
type [comando]
```

Tipi di Shell

Linux supporta diverse shell, tra cui:

- **Bash** (Bourne Again Shell) → la più diffusa.
- **Zsh** → avanzata, con autocompletamento migliorato.
- **Ksh** (Korn Shell) → usata nei sistemi UNIX.
- **Fish** (Friendly Interactive Shell) → intuitiva e moderna.

Navigazione nel Filesystem

Alcuni comandi essenziali per navigare nel filesystem:

- `pwd` → mostra la directory corrente.
- `ls` → elenca i file e le cartelle.
- `cd [directory]` → cambia directory.
- `cd ..` → torna alla cartella superiore.
- `cd ~` → torna alla home dell'utente.

Uso delle Wildcard

Le **wildcard** permettono di selezionare file con pattern specifici:

- `*` → sostituisce qualsiasi carattere.
- `?` → sostituisce un solo carattere.
- `[abc]` → seleziona una lettera tra quelle specificate.

Quoting e Escape Characters

Per gestire stringhe speciali:

- **Virgolette doppie** (`" "`) → Espandono variabili.
- **Virgolette singole** (`' '`) → Mantengono il contenuto letterale.
- **Backslash** (`\`) → Escape per caratteri speciali.

Auto completamento e Cronologia

- **Tab** → Auto completa comandi e percorsi.
- **Freccia Su/Giù** → Scorre la cronologia dei comandi.
- **history** → Mostra i comandi precedenti.

Comandi usati a lezione

Clear

ls -l /home == lista long form di quello che c'è nella home

-l = --format=long ---> uso -l perchè è molto più comodo

ls -al = per elencare i file e le directory presenti in una cartella, mostrando anche i file nascosti e dettagli aggiuntivi.

- -a (**all**) → Mostra **tutti** i file, inclusi quelli nascosti (che iniziano con `.`).
- -l (**long format**) → Mostra i dettagli di ogni file in un formato esteso.

ls --help = apre una guida

ctrl+c per uscire/arrestare un caricamento

type per vedere da dove prende un comando (ad esempio type cd è "builtin" cioè già presente nella shell, mentre type python3 da la cartella dove è contenuto python)