

# LINUX

Il sistema operativo open source  
filesystem e shell



# Il filesystem: i file

---

Un file è essenzialmente un contenitore di dati omogenei e può contenere del testo (una lettera) o dati binari (un file musicale, un'immagine, un video, un programma). Il nome di un file può contenere caratteri alfabetici, numeri, caratteri di sottolineatura o di punteggiatura e può essere lungo al massimo 256 caratteri (come per Windows). Il nome di un file non può iniziare con un numero e non può contenere caratteri particolari come punti interrogativi, spazi o asterischi (perché questi caratteri hanno dei significati particolari per il sistema). Linux è un sistema case sensitive, cioè sensibile ai caratteri minuscoli e maiuscoli. Ciò significa che per Linux 'CIAO', 'ciao', 'CiAo' e 'ciAO' sono 4 file diversi. Questo non è vero per altri sistemi come Windows, dove 'CIAO' o 'ciao' rappresentano sempre lo stesso file.



# Il filesystem: le directory

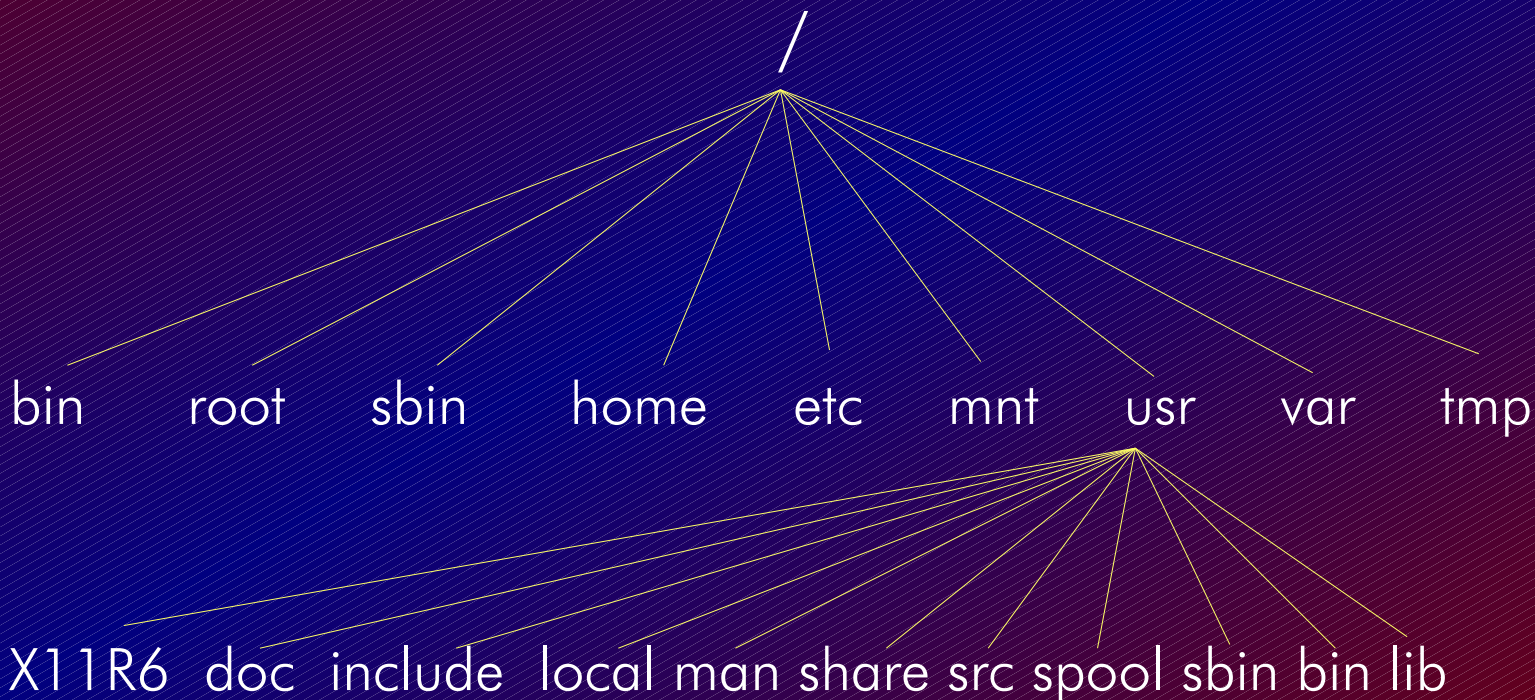
---

In un sistema Linux ogni file viene collocato all'interno di una directory ben precisa. Una directory (o cartella o folder, come viene chiamata nei sistemi Windows) è essenzialmente un contenitore di file. Un filesystem strutturato secondo un sistema di directory può essere assimilato ad una libreria, dove ogni scaffale che identifica una categoria (narrativa, umorismo, cucina, storia) rappresenta la directory stessa, mentre i libri in esso riposti rappresentano i file. Una directory può contenere file ma può contenere anche altre directory (che come abbiamo visto sono un particolare tipo di file). Una directory contenuta all'interno di un'altra directory viene definita subdirectory o sottodirectory (il primo termine sarebbe più corretto ma in Italia abbiamo ormai acquisito l'abitudine di italianizzare i termini inglesi creandone di nuovi come ad esempio shiftare, ciattare, etc.) pertanto la seconda locuzione è quella più usata). Un filesystem composto da directory può anche essere paragonato ad un albero, dove la directory principale o root o radice o /, rappresenta il tronco, mentre le sottodirectory rappresentano i rami dell'albero.



# Le directory di Linux

Esiste una struttura gerarchica di directory 'abbastanza' standard. Abbastanza standard significa che puo' variare leggermente da distribuzione a distribuzione, ma piu' o meno le directory predefinite di qualsiasi distribuzione Linux sono le seguenti:





# Alcune directory

---

## La directory /etc

La directory `/etc` contiene tutti i file di configurazione del sistema; sono file di testo e possono essere modificati sia manualmente sia tramite opportuni programmi di utilità.

## La directory /usr

La directory `/usr` contiene tutti i programmi installati, dunque è spesso contenuta in una partizione a parte, la quale può addirittura risiedere su un'altra macchina. Una sottodirectory comunemente presente è `/usr/local` dove è possibile installare programmi destinati all'uso locale.



# Alcune directory

---

## La directory /home

Contiene i dati e le configurazioni personali di ciascun utente; ad esempio:

l'utente `utente1` avrà a disposizione una directory `/home/utente1`

## La directory /dev

In Linux ad ogni dispositivo (CDROM, floppy, HD, porte seriali, telecamere, schede sonore e video) sono associati uno o più file speciali detti device e contenuti nella directory `/dev`.

I nomi dei file rispecchiano la natura del dispositivo: ad esempio il primo disco sul canale primario IDE si chiamerà

`/dev/hda`, mentre il secondo disco SCSI si chiamerà `/dev/sdb`.



# La Shell

---

La shell è un interprete di comandi. I comandi digitati dall'utente vengono letti dalla shell, interpretati e inviati al kernel per essere eseguiti. La shell è perciò una interfaccia per gli umani, cioè un programma che accetta i comandi digitati dall'utente, li trasforma in istruzioni eseguibili dal computer e li invia al kernel per l'esecuzione. Supponiamo di voler intrattenere una conversazione con una persona straniera, ad esempio un (una) norvegese): conoscendo la lingua, non avremmo problemi, in caso contrario avremmo bisogno di un interprete, cioè di una persona che, conoscendo entrambe le lingue possa tradurre da una lingua all'altra e viceversa. La shell è qualcosa di simile: converte i nostri comandi in istruzioni eseguibili dalla macchina.



# Il Prompt

---

I comandi vengono digitati dall'utente sulla riga comandi della shell. La riga comandi è una riga digitabile che comincia dal prompt. Il prompt è un carattere o un insieme di caratteri che possono essere personalizzati dall'utente. Il prompt dell'utente root (il superuser) inizia con il carattere '#' (cancelletto), mentre il prompt di un utente qualsiasi inizia con il carattere '\$' (dollaro). Ecco un esempio di prompt:

```
mau@localhost mau]$ _
```

il carattere di sottolineatura '\_' (underscore) rappresenta il cursore dal quale è possibile iniziare a scrivere. Ma un prompt come questo appena visto può essere personalizzato modificando la variabile `PS1` presente nel file di configurazione `.bash_profile`. Ad esempio si potrebbe definire un prompt di questo tipo:

```
sto aspettando un tuo comando: _
```



# Il Comandi

---

I comandi sono dei programmi che sono contenuti all'interno di alcune directory speciali come le directory `/bin` e `/sbin`. Quando l'utente digita il nome del comando/programma e preme il tasto invio, la shell cerca tale file in alcune directory predefinite (come `/bin` e `/sbin` appunto) e se lo trova lo esegue. In caso contrario produce un errore del tipo

```
bash: nome-comando : command not found
```



# Sintassi Comandi

---

il nome del comando deve essere separato dalle opzioni da uno spazio, così come le opzioni e gli eventuali argomenti. Le opzioni sono specificate usando un trattino '-' e la lettera o le lettere che specificano le opzioni. L'ordine è:

1) nome-comando, spazio, 2) -opzioni, spazio, 3) argomenti. Tra il trattino ed i caratteri di opzione non devono essere presenti spazi. Ad esempio il comando `ls` consente l'uso di alcune opzioni come `l` (lista estesa) `a` (tutti i file) e via dicendo. Perciò per visualizzare l'elenco dei file di una directory si può usare il comando `'ls -lai'`. Se si vogliono visualizzare le informazioni di un file ben preciso, occorre specificare il nome del file come argomento. Ad esempio con il comando `'ls -l pippo'` vengono visualizzate le informazioni sul file pippo.



# I comandi

---

Ed eccoci arrivati al momento di imparare una lingua che il sistema saprà riconoscere e che ci permetterà di poter interagire con esso!

Prima di iniziare la carrellata di comandi, ricordo che non sarà possibile, per evidenti motivi, spiegare tutte le opzioni per ogni singolo comando: per ogni informazione aggiuntiva che vi interessi avere rivolgetevi alle pagine del manuale utilizzando i comandi proposti nelle due diapositive che seguono.



# Il comando **man**

---

**man** fornisce la sintassi completa di un comando

**man [OPZIONI] COMANDO**

**man** fornisce la sintassi completa di **COMANDO**; le **OPZIONI** sono in genere la *sezione* del manuale legata al **COMANDO**: questo per motivi di omonimia tra i comandi della shell e alcune funzioni dei linguaggi di programmazione.



# Il comando **info**

---

**Info** fornisce ulteriori informazioni sul comando

## **info** COMANDO

info arricchisce le informazioni fornite da man tramite esempi pratici e la spiegazione di concetti fondamentali legati al comando. Fornisce anche un'ottima documentazione sulle principali librerie standard.

A differenza di man le distribuzioni più recenti forniscono sempre maggiori dettagli utilizzando il comando info.



# Il comando **ls**

---

Comando per elencare il contenuto delle directory

**ls directory**

elencare il contenuto della directory specificata

**ls**

elencare il contenuto della directory corrente

**ls -a**

elencare sia i file nascosti che quelli visibili

**ls -l**

elenco dettagliato di file e sottodirectory con i loro attributi come i permessi, le dimensioni, il proprietario, il gruppo di appartenenza e la data di creazione o ultima modifica



# Il comando `cd`

---

Comando per cambiare la propria posizione all'interno del file system

**cd**

con questo comando ci si sposta alla radice della propria home

**cd prossima**

con questa istruzione decidiamo di spostarci all'interno della directory "prossima" che è ubicata sotto la directory corrente

**cd ..**

con questo comando ci spostiamo alla directory superiore

**cd /usr/src**

impartendo queste istruzioni decidiamo di spostarci all'interno della directory /usr/src indipendentemente dal punto in cui ci troviamo all'interno del file system



# Il comando `cp`

---

Comando per copiare un file

```
cp /opt/miofile /root
```

il file verrà copiato così com'è nella directory /root

```
cp /opt/miofile /root/miofilebello
```

il file verrà copiato nella directory /root con il nome "miofilebello"



# Il comando **less**

---

## **less [OPZIONI] FILE**

less permette di visualizzare un qualsiasi file (in genere di testo) e ne permette lo scorrimento in avanti e indietro. Permette anche la ricerca di una stringa all'interno del file stesso.

Per attivare la modalità ricerca digitare **/** e scrivere la parola desiderata.



# Il comando `chmod` 1/3

---

la sintassi del comando `chmod` con l'utilizzo delle chiavi simboliche è:

`chmod [a,u,g,o][+,-][r,w,x] nome file`

dove

"a" sta per all,

"u" sta per user,

"g" sta per gruppo,

"o" sta per other.

E' possibile aggiungere diritti (+) o

togliarli (-), r,w,x stanno per read, write, execute



# Il comando `chmod` 2/3

---

## La codifica ottale

Attraverso la codifica ottale si utilizza una tripletta di numeri per selezionare i permessi.

Con **1** si indica il permesso di esecuzione.

Con **2** si indica il permesso di scrittura.

Con **4** si indica il permesso di lettura.

E' possibile combinare più permessi facendo la somma dei valori, ad esempio per associare i permessi di lettura e scrittura contemporaneamente utilizzeremo  $4+2=6$

Esempi di utilizzo della codifica ottale:

644 Con questa tripletta si concedono i permessi di lettura/scrittura all'utente che possiede il file, permessi di sola lettura ai membri del gruppo, permessi di sola lettura agli altri utenti.

600 Con questa tripletta si concedono i permessi di lettura/scrittura all'utente che possiede il file,

mentre al gruppo ed agli altri utenti non viene concesso alcun tipo di permesso



# Il comando `chmod` 3/3

---

## `chmod [OPZIONI] MODO FILE`

Imposta permessi per un file

`chmod a+rwx miofile` tutti i permessi settati per ogni utente

`chmod g-w,o-w miofile` impedisce a qualunque utente diverso dal proprietario di modificare il file

`chmod 777 miofile` tutti i permessi settati per ogni utente



# Altri comandi

---

**mv** Comando per spostare i file

**rm** Comando per rimuovere files e directory

**rmdir** Comando per eliminare le directory vuote

**pwd** mostra il percorso della directory di lavoro corrente

**mkdir** crea una directory

**rm** permette di cancellare file o directory

**chown** cambia il proprietario di un file o di una directory

**chgrp** cambia il gruppo di un file o di una directory

.....



# Editor vi

---

Per aprire un file di testo con **vi** digitare dalla console  
**vi nomefile**

Premere **i** per andare in inserimento testo

Usare i tasti freccia per spostarsi all'interno del testo

Usare **Backspace** o **Canc** per cancellare il testo

Premere **ESC** per tornare in modalità comandi.

Scrivere **:w** per salvare il file

Scrivere **:q** per uscire

Scrivere **:q!** per uscire senza salvare

Scrivere **:wq** per salvare e uscire nello stesso tempo corrente



# LINUX

Il sistema operativo open source  
filesystem e shell

# FINE