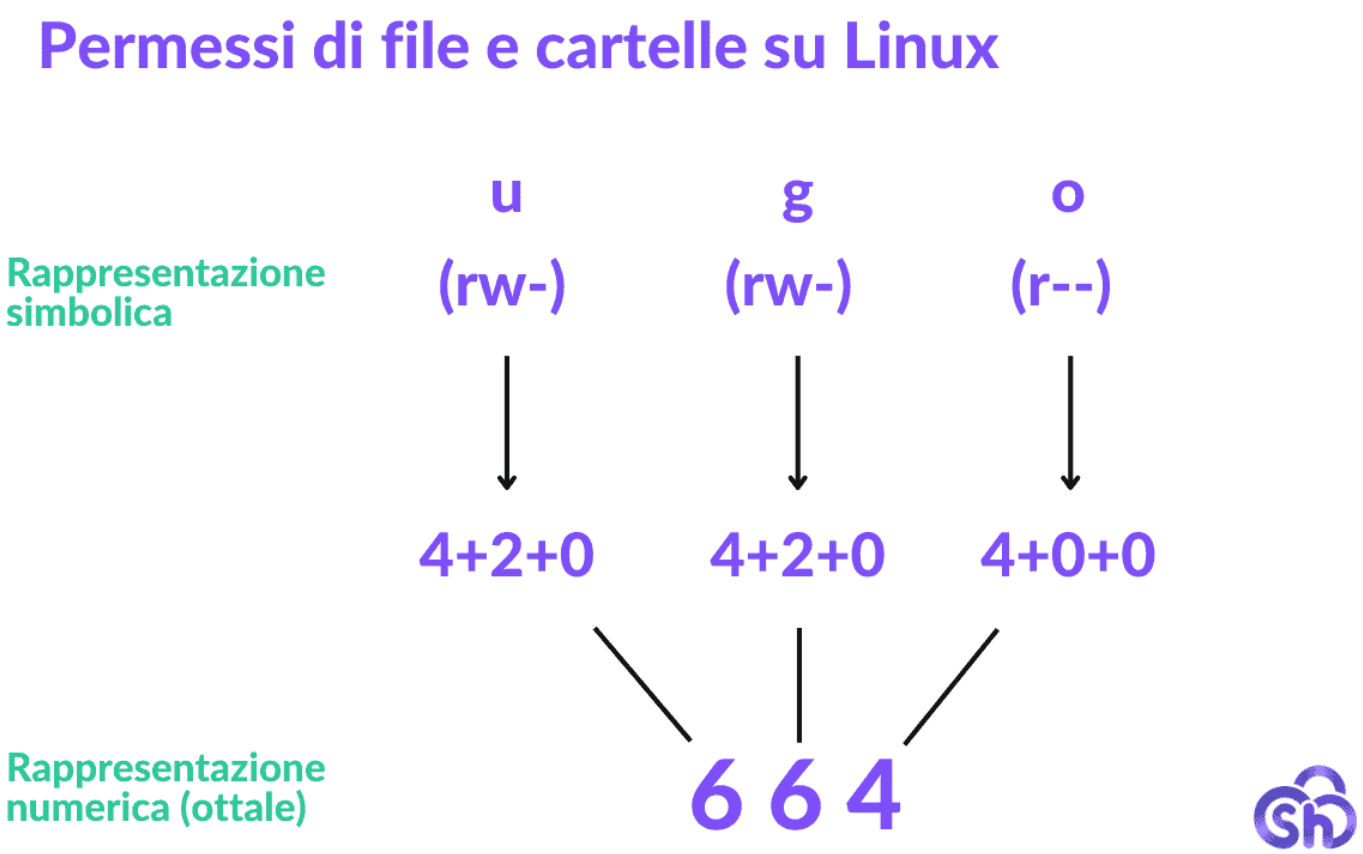
CHMOD e permessi in Linux




I permessi non sono altro che le **autorizzazioni necessarie affinché un utente possa accedere ai file ed alle cartelle del sistema**. In altre parole: ciascun utente può accedere o meno ad un determinato file o cartella a seconda che l'amministratore di sistema gli abbia conferito o meno la necessaria autorizzazione.

I permessi possono consentire ad un utente (o gruppo di utenti) di accedere ad un file o una cartella in lettura (r), scrittura (w) o esecuzione(x). Vediamo nel dettaglio cosa significa:

- lettura (**r** come read) – l'utente (o il gruppo cui appartiene) puoi leggere il contenuto; nel caso delle directory l'utente potrà vedere i file e le sotto-cartelle in quest'ultima contenute;
- scrittura (**w** come write) – l'utente (o il gruppo cui appartiene) può modificare il contenuto di un file; nel caso delle directory permette di modificarne il contenuto aggiungendo o rimuovendo elementi al suo interno;
- esecuzione (**x** come execute) – l'utente (o il gruppo cui appartiene) può eseguire un file eseguibile; nel caso delle directory consente di accedere al loro contenuto.

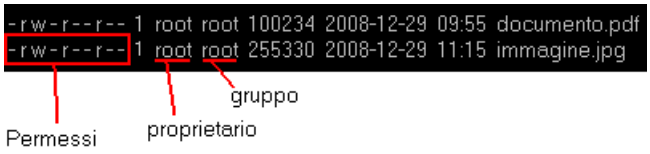
## •Cartelle, file e permessi

In diverse lezioni della nostra guida ci siamo soffermati sulle modalità di gestione di file e cartelle. In questa lezione vedremo un argomento strettamente correlato: la **gestione dei permessi** (argomento molto importante essendo Linux un sistema multi-utente come detto nella lezione precedente e ricordato poco sopra).

Abbiamo già visto il comando **ls**, in particolar modo abbiamo già detto che l'utilizzo della sintassi:

```
ls -al
```

ha la funzione di stampare a video l'elenco dei file (anche quelli nascosti) e delle cartelle presenti nella posizione corrente. L'output generato da questo comando è ricco di informazioni:




Nell'immagine qui sopra ho evidenziato le informazioni utili ai fini di questa lezione:

- colonna dei permessi
- utente proprietario / gruppo

La **colonna dei permessi** contiene 10 lettere (o trattini):

- il primo spazio indica la tipologia dell'elemento e può avere i seguenti valori: **d** (directory), **l** (link simbolico), – (file);
- i nove caratteri successivi indicano, appunto, i permessi. Più precisamente si tratta di tre distinti gruppi di 3 permessi (**r** = lettura; **w** = scrittura; **x** = esecuzione, – = non permesso). Il primo gruppo da tre riguarda il proprietario, il secondo riguarda il gruppo ed il terzo riguarda gli altri utenti.

Nel nostro esempio si tratta di due file, per entrambi il proprietario può leggere e scrivere (rw-), mentre il gruppo e gli altri utenti possono solo leggere (r–).

Le due colonne **proprietario** e **gruppo** indicano, rispettivamente l'utente proprietario del file ed il gruppo di appartenenza.

Dopo questa lunga, ma doverosa, premessa veniamo al nocciolo della questione, ovvero come gestire e modificare queste informazioni. A tal scopo linux dispone di comandi ad hoc. Vediamoli in rassegna:

## chmod

E' il comando che modifica i permessi (lettura, scrittura, esecuzione). Il comando in oggetto ha una duplice sintassi, vediamole entrambe:

Consente di assegnare diversi permessi al proprietario, al gruppo ed agli altri utenti. La sintassi è la seguente:

```
chmod a=rwx nomefile
```

nel nostro esempio abbiamo assegnato a tutti (**a** = all) tutti i permessi (**rwx**). A sinistra del simbolo uguale (=) abbiamo l'assegnatario del permessi, a destra i permessi assegnati. L'assegnatario viene identificato attraverso una lettera:

- **a** (tutti)
- **u** (utente proprietario)
- **g** (gruppo)
- **o** (altri utenti)

I permessi, invece, sono identificati dalle tre lettere **r**, **w** e **x** che abbiamo già visto in precedenza.

chmod con sintassi ottale

Con questa sintassi i permessi vengono assegnati a tutti i livelli simultaneamente. Al posto delle lettere **rwx** si utilizzano 3 numeri. Facciamo un esempio:

```
chmod 777 nomefile
```

Nel nostro esempio abbiamo assegnato a tutti i massimi permessi (cioè "rwx" come nell'esempio precedente). In questa sintassi i tre numeri definiscono i permessi dei tre livelli: il primo numero riguarda l'utente proprietario, il secondo il gruppo, il terzo gli altri utenti. Di seguito una tabella dei valori numerici e del loro significato:

- **7** corrisponde a **rwx (1 1 1 in binario)**
- **6** corrisponde a **rw- (1 1 0 in binario)**
- **5** corrisponde a **r-x (1 0 1 in binario)**
- **4** corrisponde a **r-- (1 0 0 in binario)**
- **3** corrisponde a -**wx (0 1 1 in binario)**
- **2** corrisponde a -**w- (0 1 0 in binario)**
- **1** corrisponde a --**x (0 0 1 in binario)**
- **0** negato ogni accesso --- ( **0 0 0 in binario**)

Se ad esempio avessimo voluto assegnare al proprietario tutti i permessi, al gruppo solo lettura ed esecuzione (ma non scrittura) ed agli altri utenti nulla, avremmo scritto:

```
chmod 750 nomefile
```

N.B. il permesso di scrittura è quello più "delicato": chi possiede questo permesso potrà non solo modificare i file ma anche cancellarli! Si faccia attenzione quindi ad usarlo con la massima attenzione e parsimonia.

## chmod in modalità ricorsiva

La modalità ricorsiva è molto comoda se si deve agire su una cartella e su tutto quanto è in essa contenuto. Per attivare la modalità ricorsiva:

```
chmod -R 777 nomecartella
```