I PERMESSI SUI FILE E SULLE DIRECTORY

E' importante saper riconoscere quali file poter **leggere**, **scrivere** ed **eseguire** e quali no.
Linux è un **sistema multiutente**, ciò significa che in unico sistema possiamo trovare svariati utenti. Ma allora come proteggere i nostri documenti o impedire che utenti normali abbiamo accesso a risorse o file di configurazione del sistema? Semplice, impostando dei diritti d'accesso.

Individuazione delle autorizzazioni sui file o cartelle

Abbiamo **otto combinazioni** assegnabili ad un file o ad una directory, in modo tale da decidere quali tipi di autorizzazioni fanno al caso nostro. Possiamo decidere se utilizzare il **sistema numerico ottale** o una **terzina di lettere**:

0 --- Nessuna autorizzazione
1 --x Solo esecuzione
2 -w- Solo scrittura
3 -wx Possibilità di scrittura ed esecuzione
4 r-- Solo lettura
5 r-x Possibilità di lettura ed esecuzione
6 rw- Possibilità di lettura e scrittura
7 rwx Possibilità di lettura, scrittura ed esecuzione

Per visualizzare i diritti relativi a dei file o directory, digitare nella shell quanto segue:

ls –l

```
[scoleri@linux ~]$ ls -l
totale 16
drwxr-xr-x  2 scoleri scoleri 4096 29 ott 11:10 Desktop
drwxrwxr-x  2 scoleri scoleri 4096  1 nov 22:24 prove
```

Spostiamoci nella directory prove e digitiamo **ls –l prova1.txt.**

```
[scoleri@linux prove]$ ls -l prova1.txt
```

-rw-rw-r-- 1 scoleri scoleri 0 1 nov 22:24 prova1.txt

Vengono visualizzati i diritti relativi al file prova1.txt. Notate che la prima lettera che compare indica con che cosa abbiamo a che fare (-rwxrwxr--):
- File normale
d Directory
p Named pipe
c Character special file
l Link simbolico ad un file

Nel nostro caso c'è un - quindi si tratta di un file. Successivamente le sequenze che seguono vogliono raggruppate in 3 parti:

rw- Autorizzazioni del proprietario
rw- Autorizzazioni del gruppo del proprietario
r-- Autorizzazioni degli altri utenti

Assegnazione delle autorizzazioni

Per cambiare le autorizzazioni sui file o sulle directory bisogna utilizzare dalla shell il comando **chmod**. Mettiamo il caso che del file *prova1.txt* voglia cambiare le autorizzazioni in lettura, esecuzione e scrittura per il proprietario (naturalmente possono cambiare le autorizzazioni solo l'utente root o il proprietario del file) e sola lettura per gli altri utenti del gruppo e utenti normali; allora scriverò:

chmod 744 prova1.txt (7 per il **proprietario**, 4 per gli **utenti del gruppo del proprietario**, 4 per gli **altri utenti**)

Vediamo cosa è successo

ls -l prova1.txt

-rwxr--r-- 1 scoleri scoleri 0 1 nov 22:24 prova1.txt

I diritti si sono modificati da **-rw-rw-r-- (rw-=6 rw-=6 r--=4 quindi 664)** a -rwxr--r-- (rwx= 7 r--=4 r--=4 quindi 744)

Riepilogando:
r (lettura) **w** (scrittura) **x** (esecuzione)
Se c'è un meno (-) in una posizione, vuol dire che quel permesso non è concesso.

N.B. Se associamo ad un permesso concesso il bit 1 (uno) e ad un permesso negato il bit 0 (zero) otteniamo per le tre terne di permessi (ugo: **u** sta per **user**, il proprietario del file, **g** sta per **group**, il gruppo del proprietario, **o** sta per **other**, tutti gli altri utenti del sistema) un numero che tradotto in base 8 va da 0 a 7. Esempio:

rw-r--r-- si codifica in binario come 110100100, in ottale (base 8) abbiamo 110=6, 100=4, 100=4 che dà luogo alla stringa ottale 644 (ricorda: 110 100 100, inteso in base 2, corrisponde a

1	1	0	1	0	0	1	0	0
$1*2^2$	$1*2^1$	$0*2^0$	$1*2^2$	$0*2^1$	$0*2^0$	$1*2^2$	$0*2^1$	$0*2^0$
4	2	0	4	0	0	4	0	0
6			4			4		

Un utile comando per ottenere informazioni sui permessi di un file: **GETFACL**

Portatevi nella directory **prove** e digitate **getfacl prova1.txt**
[scoleri@linux prove]$ getfacl prova1.txt
file: prova1.txt
owner: scoleri
group: scoleri
user::rwx
group::r--
other::r--

Altro uso di **chmod** (**ch**ange **mod**e).

u: utente che possiede il file (user)
g: membri del gruppo che possiede il file (group)
o: tutti gli altri utenti del sistema (other)
a: tutti gli utenti del sistema (equivale a **ugo**)

+ : aggiunge permessi ai permessi esistenti all'utente
- : rimuove permessi ai permessi esistenti all'utente
= : revoca tutti permessi agli utenti indicati

Esempi: **u+w** aggiunge il permesso di scrittura all'utente che possiede il file; **a+rw** aggiunge i permessi di lettura e scrittura ai permessi esistenti di tutti (a) gli utenti (è equivalente ad **ugo+rw**).

Proteggere un file da scrittura: **chmod go-w (g** sta per gruppo, **o** per other, **-w** nega la scrittura)
Provare questo comando con il file prova1.txt

Rendere un file privato: **chmod go=** (questo comando revoca tutti permessi di accesso a group e other, quindi il file diventa accessibile solo al proprietario, a colui che lo ha creato). Dopo aver lanciato il comando

chmod go= prova1.txt
con ls –l prova1.txt
dovrà ottenersi una cosa del genere (notare i permessi: rwx al proprietario, nessuno a tutti gli altri)
-rwx------ 1 sc0550 users 0 2004-11-23 20:17 prova1.txt

Rendere un file pubblico: **chmod a+rw prova1.txt** rende il file **prova1.txt** leggibile e modificabile da tutti (viene definito *world readable*). Rendere il file prova1.txt world readable; dopo aver letto i permessi rendere il file privato.