

Come gestire i permessi con GNU/Linux

Autore: Stefano Pardini

<http://www.viareggiolinux.org>

Linux User Group: ACROS <http://www.lug-acros.org>

Premessa

In fase di installazione di un sistema GNU/Linux normalmente si crea un utente comune che è quello che solitamente viene utilizzato per il login. L'utente comune a differenza di root non dispone di qualsiasi permesso sui files e le directory presenti nell'intero file system, ma in termini di sicurezza è meglio accedere come utente comune, piuttosto che fare il login come root. In caso di necessità potrete sempre diventare root aprendo una shell e digitando "su" seguito dalla password di root. In questo modo avrete la possibilità di modificare anche i files di sistema e i files di configurazione che normalmente non sono accessibili all'utente comune (ricordate sempre di fare un backup del file prima di modificarlo! - potete fare così: tar cf nomefile.tar nomefile.txt).

Gestione dei Permessi

In GNU/Linux la gestione e l'accesso a files e directory è regolata dai permessi. Ogni utente dovrebbe cercare di apprendere come avviene la gestione dei permessi, questa cosa è molto importante perché prima o poi vi troverete a dover fare i conti con i diritti di lettura, scrittura ed esecuzione dei files. Oltre all'utente comune e root ci sono i gruppi ai quali possono appartenere più utenti. Per ogni file avremo quindi un proprietario, un gruppo di appartenenza, i permessi di lettura, scrittura ed esecuzione. Il permesso di lettura viene indicato dalla lettera "r", quello di scrittura dalla lettera "w" e quello di esecuzione dalla lettera "x". Il comando per assegnare i permessi di un file dalla console è: chmod. Esistono poi due modi di assegnare i permessi tramite chmod, attraverso l'utilizzo di "chiavi simboliche" oppure attraverso l'utilizzo della "codifica ottale". Per chiarire meglio il concetto e spiegare la differenza fra questi due diversi modi di attribuzione dei permessi vediamoli in dettaglio con degli esempi pratici.

Utilizzo delle chiavi simboliche:

L'utilizzo delle chiavi simboliche con chmod è il seguente: chmod [a,u,g,o] [+,-] [r,w,x] nome file, dove "a" sta per tutti (all), "u" sta per utente (user), "g" sta per gruppo (group), "o" sta per altri (other). E' possibile aggiungere diritti con il segno "+" o toglierli con "-", allo stesso modo r,w,x stanno rispettivamente per lettura (read), scrittura (write) ed esecuzione (execute). Prima di procedere con un esempio di assegnazione dei permessi vediamo l'output del comando ls con l'opzione -l su alcuni files all'interno di una directory della mia home:

stefano@linux

```
stefano@linux:~/dati/info>ls -l
```

```
-rwxr-xr-x 1 stefano users 420 2004-09-26 00:53 soci.txt
drwxr-xr-x 2 stefano users 216 2006-07-05 19:42 disegni
-rw-r--r-- 1 stefano users 0 2006-10-17 16:45 test.txt
drwxr-xr-x 2 stefano users 88 2006-05-29 17:39 lettere
```

Come avrete notato ogni riga contiene all'inizio una tripletta (in tutto nove caratteri) preceduta nella prima e terza riga dal segno "-", nella seconda e quarta riga dalla lettera "d". Questo è dovuto al fatto che il "-" prima della tripletta identifica un file mentre la lettera "d" prima della tripletta identifica una directory. Subito dopo il "-" o la "d" c'è la prima serie di tre caratteri, poi la seconda serie e infine la terza. Nel caso della prima riga avremo quindi un file (-) dove:

- rwx indica i permessi che ha il proprietario del file;
- r-x indica i permessi che hanno gli utenti del gruppo;
- r-x indica i permessi che hanno gli altri utenti.

ricordate u,g,o,? Se avete letto attentamente il documento credo di sì...Dopo la tripletta la riga presenta un numero:1 per i file, mentre quando si tratta di una directory il numero dipende da quante sottocartelle contiene. Dopo la tripletta e il numero ci sono il nome dell'utente e il gruppo, la dimensione in bit, la data e l'ora di creazione ed infine il nome del file o cartella. Allo stesso modo vi potete allenare a leggere l'output degli altri files e directory...

Vediamo adesso come possiamo modificare i permessi del file test.txt con un esempio:

```
-rw-r--r-- 1 stefano users 0 2006-10-17 16:45 test.txt
```

l'output di `ls -l` sul file test.txt ci dice che:

il file è leggibile e scrivibile dall'utente, solo leggibile dal gruppo, solo leggibile dagli altri; supponendo di voler dare i permessi di scrittura anche al gruppo e agli altri digitiamo il comando:

`chmod go+w test.txt` in questo modo avremo aggiunto i permessi di scrittura con "+" al gruppo e agli altri (go), se adesso li volessimo togliere di nuovo dovremo usare il segno "-"; con `chmod a+w test.txt` si concede a tutti gli utenti (a) di scrivere il file (w). Questi sono semplici esempi, fate pratica e vedrete che queste operazioni diventeranno presto normale routine.

Utilizzo della codifica ottale:

l'utilizzo della codifica ottale avviene attraverso una tripletta di numeri invece delle lettere:

- con 0 nessun tipo di permesso;
- con 1 si indica file eseguibile ma non leggibile ne modificabile;
- con 2 si indica file modificabile ma non leggibile ne eseguibile;
- con 3 si indica file modificabile ed eseguibile ma non leggibile;
- con 4 si indica file leggibile ma non modificabile ne eseguibile;
- con 5 si indica file leggibile ed eseguibile ma non modificabile;
- con 6 si indica file leggibile e modificabile ma non eseguibile;
- con 7 si indica file leggibile, modificabile ed eseguibile.

E' possibile combinare più permessi sommando dei valori, ad esempio per associare i permessi di lettura e scrittura contemporaneamente si può utilizzare $4+2=6$

Ecco un esempio 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, praticamente:

[u g o]

[6 4 4]

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, praticamente:

[u g o]

[6 0 0]

Confronto fra i due diversi modi di attribuzione dei permessi

Infine paragoniamo il primo metodo al secondo e diciamo che:

`chmod u=rw-,g=rw-,o=rw- test.txt` è uguale a: `chmod 666 test.txt`

Conclusione

Concludendo, possiamo dire che non è tanto importante quale metodo di attribuzione utilizzate ma è importante capire il concetto di gestione dei permessi, la pratica e il tempo faranno il resto.

Enjoy with GNU/Linux...



Questo/a opera è pubblicata sotto una [Licenza Creative Commons](https://creativecommons.org/licenses/by-nc-sa/4.0/).