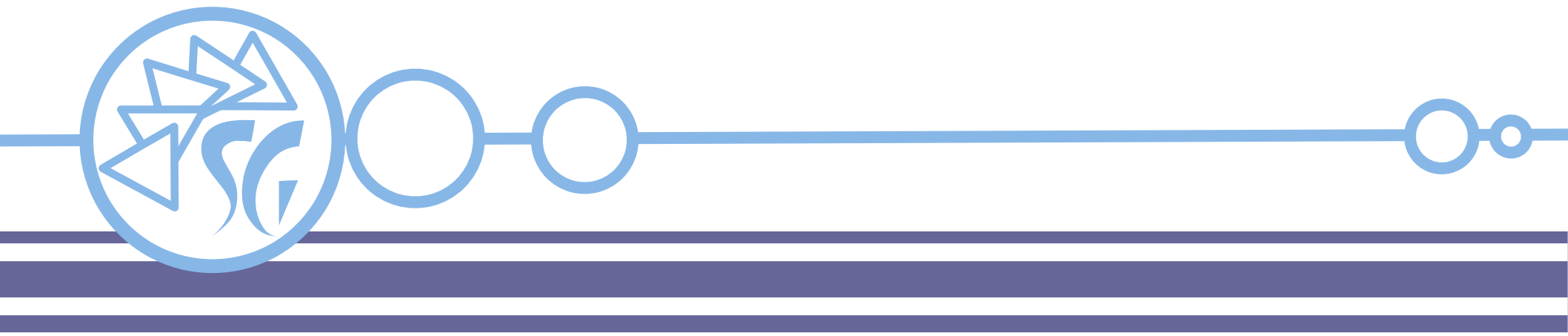

LINUX
Gestione dei Permessi

Ing. Simone Giustetti

studiosg@giustetti.net

I Permessi

Linux gestisce direttamente i permessi e demanda invece l'autenticazione a servizi esterni.

La gestione dei permessi è integrata nel sottosistema dei file system.

Non tutti i file system compatibili con Linux supportano il sottosistema dei permessi. I file system della famiglia **FAT**, ad esempio, non consentono la gestione dei permessi.

Le Risorse

Le risorse di sistema generalmente includono:

- File di testo o di dati.
- Programmi.
- Dispositivi.

I programmi sono file aventi uno speciale formato binario.

Tutti i dispositivi collegati ad un computer sono gestiti attraverso uno specifico **file descrittore**.

In conclusione **tutte le risorse sono file**.

Organizzazione di File e Directory

Ogni file system è organizzato come un albero avente una radice comune, che si biforca in numerose diramazioni chiamate directory o cartelle.

Attraverso i permessi è possibile impostare:

- Chi abbia il diritto di accesso ad una determinata cartella.
- Quali file possa leggere o scrivere.
- Quali programmi possa eseguire.

Permessi dei File

Il sistema di controllo degli accessi di Linux è molto semplice e consiste di una combinazione di modalità di accesso e tipologie di "persone".

Esistono 3 modalità d'accesso:

- **Lettura** (**r** = read): Possibilità di leggere il contenuto di un file.
- **Scrittura** (**w** = write): Possibilità di scrivere un file, cioè di crearne uno o di modificarne il contenuto.
- **Esecuzione** (**x** = execute): Possibilità di eseguire un file / programma.

Permessi dei File

I file possono essere acceduti da 3 differenti tipologie di "persone":

- Il **proprietario** del file ossia l'utenza (**u** = user).
- Un **membro del gruppo** cui il file è assegnato (**g** = group).
- **Chiunque altro** (**o** = others).

Permessi delle Directory

Le directory sono gestite in maniera del tutto analoga ai file:

- Ogni directory è assegnata ad un proprietario (**u** = user).
- Ogni directory è assegnata ad un gruppo (**g** = group).
- Tutte le utenze o gruppi che non sono il proprietario o non appartengono al gruppo assegnato sono inclusi nel grande insieme degli altri (**o** = others).

Permessi delle Directory

I permessi hanno un significato specifico leggermente diverso da quello descritto per i file:

- **Lettura** (**r** = read): Possibilità di elencare il contenuto di una directory.
- **Scrittura** (**w** = write): Possibilità di aggiungere, rinominare o cancellare file o sottodirectory.
- **Esecuzione** (**x** = execute): Possibilità di entrare in una directory ed aprire file.

Per navigare un file system servono i permessi **rx** sulle directory.

Permessi e root

root, il super-utente, è sempre in grado di entrare in qualsiasi directory e creare o modificare i file in essa contenuti.

Il super-utente deve essere in grado di operare sempre, anche in casi di emergenza quando un file system è danneggiato.

Per consentire a **root** di adempiere alle proprie funzioni non gli sono imposte le regole applicate a tutte le altre utenze o gruppi.

Gestione dei Permessi

I permessi possono essere impostati da una utenza standard, quando è proprietaria di un file, oppure da **root**.

Quando viene creato un nuovo file oppure una nuova directory gli **vengono automaticamente assegnati permessi predefiniti**, che possono essere cambiati in seguito.

I permessi dell'utente **sono ereditati dai programmi** che vengono lanciati.

Comandi di Amministrazione

Il comando **ls** consente di visualizzare le informazioni estese di un file tra cui i permessi.

L'opzione *-l* rende l'elenco di file e directory con informazioni estese.

```
ls -l
total 0
-rw-r--r-- 1 jim-big users 0 Aug 16 18:19 file_test_01
```

I caratteri nella colonna immediatamente a destra sono i permessi del file o della directory.

Notazione Testuale

I permessi sono rappresentati da una terna di caratteri: **rwx**. Il carattere “-” (Meno) funge da segnaposto per i permessi non assegnati.

La lettera **r** rappresenta il permesso di **lettura** di un file.

La lettera **w** rappresenta il permesso di **scrittura** di un file.

La lettera **x** il permesso di **esecuzione** di un programma.

Comandi di Amministrazione

Il comando **chown**, abbreviazione di "change owner" o "cambia proprietario", consente di assegnare un file / directory ad una utenza.

Può essere eseguito solo da root.

La sintassi del comando è:

chown <opzioni> <utenza> <directory / file>

chown jim-big file_test_01

chown *-R* jim-big dir_01 Opera su file/directory in maniera ricorsiva.

Comandi di Amministrazione

Il comando **chgrp**, abbreviazione di "change group" o "cambia gruppo", consente di assegnare un file / directory ad un gruppo.

Può essere eseguito dal proprietario del file.

La sintassi del comando è:

chgrp <opzioni> <gruppo> <directory / file>

chgrp dba file_test_01

chgrp *-R* dba dir_01 Opera su file/directory in maniera ricorsiva.

Comandi di Amministrazione

Un modo alternativo per cambiare il gruppo consiste nell'usare il comando **chown** impostando sia l'utenza che il gruppo di appartenenza. La sintassi è:

chown <opzioni> <utenza>**:**<gruppo>
<directory o file>

chown jil-big:developer file_test_11

La soluzione è consentita solo a root.

Comandi di Amministrazione

Per impostare i permessi si usa il comando **chmod** contrazione di "change mode" o "cambia modalità".

La sintassi del comando è:

chmod <opzioni> <permessi> <directory o file>

Il formato dei permessi è composto da 3 parti:

- Un identificatore delle terne di permessi che verranno modificate.
- La tipologia di operazione eseguita.
- Il permesso oggetto della modifica.

Identificatore della Terna di Permessi

La prima parte consiste in una qualsiasi combinazione delle lettere "a", "g", "o" ed "u" ove:

- "**a**" = all, cioè tutte le utenze e corrisponde all'unione di g, o ed u.
- "**g**" = group. Il gruppo assegnatario del file o della directory.
- "**o**" = others. Tutte le utenze diverse dal proprietario e da tutti i membri del gruppo cui il file è assegnato.
- "**u**" = user, il proprietario del file o della directory.

Tipologia di Operazione

La tipologia di operazione è rappresentata dai caratteri "+", "-" o "=" ove:

- "**+**" **aggiunge** i permessi impostati nella riga di comando a quelli esistenti.
- "**-**" **rimuove** i permessi impostati nella riga di comando da quelli esistenti.
- "**=**" **assegna** i permessi impostati **rimuovendo nel contempo tutti gli altri**.

Permesso Modificato

I permessi che si vogliono impostare o rimuovere sono rappresentati attraverso le lettere "r", "w", "x"o "X" ove:

- "**r**" è il permesso di lettura (read).
- "**w**" è il permesso di scrittura (write).
- "**x**" è il permesso di esecuzione o di ricerca nelle directory (execute).
- "**X**" coincide con "x", ma la modifica avviene se e solo se un'altra utenza ha assegnato il permesso "x".

Chmod Esempi

chmod go-rwx dir_test_02/ Rimuove tutti i permessi al gruppo ed alle altre utenze.

chmod a+r file_test_111 Assegna a tutte le utenze il permesso di leggere il file.

chmod ug=rw file_test_112 Assegna i Permessi di leggere e scrivere il file a proprietario e gruppo, rimuove nel contempo il permesso di esecuzione ed ogni permesso agli altri.

Chmod Esempi

chmod a= dir_test_04/ Revoca i permessi di accesso a chiunque. La directory potrà essere acceduta solo da root.

Un utente può rimuovere il proprio permesso di scrittura da un suo file. È un buon modo per assicurarsi che lo stesso non sia **cancellato o sovrascritto per errore**.

Notazione Numerica

I permessi possono essere rappresentati in maniera più compatta con la notazione numerica.

I permessi delle risorse sono salvati direttamente nel file system sotto forma di 4 valori numerici compresi tra 0 e 7.

Ogni cifra rappresenta un insieme di permessi:

- **Permessi speciali**.
- **Permessi del proprietario**.
- **Permessi dei membri del gruppo**.
- **Permessi di tutti gli altri utenti**.

Notazione Numerica

Le assegnazioni in notazione numerica sono sempre assolute. Funziona come l'operatore "=".

Notazione testuale	Notazione numerica
Nessun carattere (Nessun permesso)	0
r (read)	4
w (write)	2
x (execute)	1

Chmod Esempi

chmod 764 file_test_150 Assegna il permesso di lettura, scrittura ed esecuzione al proprietario; lettura e scrittura al gruppo; sola lettura a tutti gli altri.

chmod 640 file_test_118 Assegna il permesso di lettura e scrittura al proprietario; lettura al gruppo nessun permesso a tutti gli altri.

chmod 000 file_test_112 Revoca tutti i permessi a tutte le utenze.

Chmod Esempi

chmod 664 file_test_115 Assegna il permesso di lettura e scrittura a proprietario e gruppo ed il permesso di sola lettura a tutti gli altri.

chmod 555 file_test_115 Assegna a tutte le utenze il permesso di lanciare un programma (Lettura / Accesso + esecuzione).

Il permesso di esecuzione accompagna sempre quello di lettura perché **è necessario accedere ad un programma per poterlo lanciare**.

Gestione dei Collegamenti (Link)

Linux consente di creare dei collegamenti ai file.

Esistono due tipologie di collegamenti:

- Soft: Un puntatore ad altro file.
- Hard: Duplica tutta la struttura a basso livello del file (Gli I-node).

I collegamento soft hanno sempre permessi **777** perché comunque fanno fede quelli del file puntato.

Gestione dei Collegamenti (Link)

I collegamento hard sono “copie” indipendenti del medesimo file perciò hanno gli stessi permessi del file.

Il file system si occupa di aggiornare i permessi di tutti i file collegati hard perciò **i permessi sono sempre allineati** tra le “copie”.

Permessi Predefiniti

I permessi predefiniti assegnati ai file di nuova creazione sono gestiti mediante il comando **umask**.

Il comando usa la notazione numerica.

Notazione numerica	Directory	File
0	Nessun permesso	Nessun permesso
4	Elenco del contenuto	Lettura
2	Creazione nuove risorse	Scrivi / Modifica / Cancella
1	Accesso alla directory e lettura dei file contenuti	Esecuzione di un programma o uno script

Permessi Predefiniti

umask Rende i permessi predefiniti

Non riporta i permessi assegnati, ma quelli sottratti a 777.

Un umask pari a 022 significa che ogni nuovo file generato avrà i permessi 644 o 755 per una directory.

Il permesso di esecuzione non è considerato per i programmi perché **deve sempre essere assegnato esplicitamente**.

Permessi Predefiniti

Il comando umask può essere utilizzato per modificare i permessi predefiniti.

umask 033 Coincide con 022 per i file → 644
Cambia invece per le directory → 744

Umask 027 → 640 / 750

Umask 077 → 600 / 700

Permessi Predefiniti

I permessi impostati con umask dureranno fino a che l'utente non si scolleghi.

La maggior parte delle distribuzioni usa permessi predefiniti pari a **022** oppure **027**.

077 è solitamente consigliato per aumentare il livello di sicurezza.

Permessi Speciali

Esistono 3 permessi speciali che cambiano il comportamento standard dei file o delle directory a cui sono assegnati.

I permessi speciali sono:

- **SUID bit**.
- **GUID / SGID bit**.
- **Sticky bit**.

SUID Bit

Quando un utente lancia un programma questo **eredita i privilegi dell'utente.**

Quando il permesso SUID (Set User ID) è impostato, il processo assume identità e privilegi del proprietario del programma anziché dell'utente che lo ha lanciato.

Il proprietario dei comandi di sistema è **root** per cui l'impostazione di SUID bit consente ad un utente standard di **impersonare root** avendo così a disposizione risorse illimitate.

SGID Bit

l GUID / SGID bit o Set Group ID bit svolge funzioni analoghe al SUID bit, ma limitate al gruppo.

Quando viene eseguito un programma avente il bit abilitato lo stesso **gira con i permessi del gruppo cui è assegnato il programma** anziché del gruppo associato all'utente che lo ha lanciato.

Sticky Bit

Lo sticky bit viene utilizzato per modificare la gestione dei permessi delle directory e dei file in esse contenuti.

Lo sticky bit assegnato ad una directory imposta che le risorse ivi contenute possano essere **cancellate solo dal proprietario** della risorsa.

Lo sticky bit ha la priorità sui permessi standard.

Gestione dei Permessi Speciali

I permessi speciali vengono amministrati con il comando **chmod**.

Permesso Speciale	Notazione Numerica	Notazione Testuale
Nessun permesso	0	-
SUID bit	4	s o S (Al posto del permesso di esecuzione del proprietario)
GUID / SGID bit	2	s o S (Al posto del permesso di esecuzione del gruppo)
Sticky bit	1	t o T (Al posto del permesso di esecuzione degli altri utenti)

Gestione dei Permessi Speciali

chmod u+s <file> Imposta il SUID bit
chmod 4750 <file> Imposta il SUID bit con notazione numerica

chmod u-s <file> Rimuove il SUID bit
chmod 0750 <file> Rimuove il SUID bit con notazione numerica

chmod o+t <directory> Imposta lo Sticky bit
chmod 1777 <directory> Imposta lo Sticky bit con notazione numerica

Informazioni & Licenze