

1 Gestione del sistema Unix

1.1 Gestione degli utenti

1.1.1 Concetti

Permessi Unix

Permessi sui file

In Unix il proprietario di un file, o l'utente root, può accordare permessi sui file a tre tipi di utenti:

- a sè stesso (**owner**)
- agli altri utenti del proprio gruppo (**group**)
- a tutti gli altri utenti (**other**)

a ciascuno di questi tre possono essere accordati i seguenti diritti:

- lettura (r = read)
- scrittura (w = write)
- esecuzione (x = execute)

e le loro combinazioni.

Fra i dati di ogni file, mantenuti dal S.O., ci sono 9 flag che identificano i permessi, ls mostra questi flag ed altre informazioni.

Per esempio:

```
# ls -l
```

può dar luogo a qualcosa di simile:

```
total 7
drwx-----  7 gamon    users      2.0k May 18 17:24 .
drwxr-xr-x   3 root     root       1.0k Feb 22 01:00 ..
-rwx-----  1 gamon    users      57 May 18 12:14 eMMMMMio
drwxr-xr-x   2 gamon    users      1.0k May 18 17:24 GuardateTutti
drwxrwxrwx   2 gamon    users      1.0k May 18 17:22 condivisi
-rw-----   1 gamon    users      21k May 18 17:14 TestoSoloMio.txt
-rw-r--r--   1 root     root       4.7k Apr  7 22:39 FileDelGrandeCapo
```

La prima lettera corrisponde al flag "directory", indicato con d se è un directory, con - se non lo è. Può avere anche il valore l, se si tratta di un link ad altro file.

Le lettere seguenti esprimono i permessi nell'accesso al file. Sono indicati i diritti del proprietario, (tre lettere che possono essere r, w, x o -) quelli del gruppo del proprietario (r, w, x o -) e quelli degli altri utenti.

Segue il numero di collegamenti a questo file che sono attualmente presenti. Il sistema Unix è in grado di fare collegamenti ai file e di usarli come se fossero in tutto e per tutto il "vero" file. E' utile sapere se un file è collegato ad altri perchè se lo cancelleremo i nomi dei collegamenti non punteranno più a nulla di significativo.

Nella riga del comando ls, dopo il numero dei link segue l'username del proprietario ed il groupname del suo gruppo principale; seguono la dimensione del file, in questo caso visualizzata il kByte, ed il mese, giorno, ora della sua ultima modifica.

Per ultimo viene il nome del file.

Cambiare "modo": chmod

Il comando **chmod** ("change mode") permette di cambiare i flag del file e quindi anche i suoi diritti di accesso.

```
chmod <Nome del file o della cartella> -a=rx
```

rende disponibile a tutti (a = all) il file in esecuzione (x = execute) ed in lettura (l = lettura).

chmod con comando ottale

1.1.2 Tilde e home directory

Nel linguaggio di comando della shell il simbolo "~" ("tilde" che in Linux si ottiene con la combinazione di tasti (AltGr ^)) indica l'home directory dell'utente corrente.

1.1.3 Comandi

\$ id

visualizza il nome ed i gruppi dell'utente. Il comando visualizza, nell'ordine:

- **UID** (User Identifier: è un numero univoco che identifica quello specifico utente)
- fra parentesi, l'username.
- tutti i gruppi cui appartiene l'utente, indicando per ciascuno:
 - **GID** (Group Identifier: è un numero che identifica il gruppo)
 - fra parentesi, il nome del gruppo

\$ whoami

(who am I, chi sono io?) visualizza l'username dell'utente attuale. Dato che alcune distribuzioni non scrivono il nome utente al prompt del terminale, questo comando è utile quando si fanno cambi di utente in una stessa sessione (comando "su").

\$ who

visualizza gli username di tutti gli utenti che hanno fatto login con terminali di questa macchina.

\$ w

mostra le stesse informazioni di who ma aggiunge anche da quanto tempo è stato fatto il login e quali processi l'utente sta facendo eseguire.

1.1.4 Comando su

Fa aprire una shell di un altro utente all'interno della shell corrente. Serve per passare da un utente all'altro nella modalità di comando. Se usato senza alcun parametro passa l'utente root. Naturalmente chiede la password dell'utente in cui ci si vuole trasformare. A differenza del processo di login, che porta l'utente che entra sempre nella sua home directory, il comando "su" lascia il nuovo utente nella cartella in cui era l'altroprecedentemente.

Per completare una sessione di lavoro iniziata con il comando "su" e ritornare alla sessione dell'utente precedente, digitare exit.

Sintassi semplificata di "su":

```
$ su [<nome d'utente>]
```

1.1.5

Esempi:

```
$ whoami
ufantozzi
$ su
password: <qui ufantozzi digita la password di root>
# whoami
root
# su pina
password: <qui ufantozzi digita la password di pina>
$ whoami
pina
$ exit
# whoami
# root
# exit
$ whoami
ufantozzi
```

1.2 Utenti

Le informazioni sugli utenti sono nei file: /etc/group, /etc/passwd, /etc/shadow

1.2.1 Formato del file passwd

File dove sono registrati gli utenti

```
<nome utente>:<password crittata>:<UID>:<GID>:<commento>:<home directory>:<interprete dei comandi>
```

<password crittata> è crittata dal sistema, con un hash per aggiungerla la facciamo calcolare dal sistema, con un comando.

<UID> user id unico

<home directory> è la path della directory su cui l'utente può fare tutto. Sono comprese nel diritto di accesso le sottodirectory create dall'utente dentro la sua home.

Esiste una sottodirectory /home sotto la quale vengono definite automaticamente le home directory degli utenti fatti con i normali comandi.

<interprete dei comandi> programma che viene lanciato quando l'utente è autenticato. E' la shell di default dell'utente. La path è assoluta, es. /bin/bash

Piuttosto che aggiungere a mano gli utenti nel file /etc/passwd¹ è meglio usare gli script già presenti nella distribuzione, che rendono più difficile sbagliare.

script per aggiungere un utente:

useradd o adduser (di solito sono presenti entrambi nelle distribuzioni odierne ed uno è un link simbolico all'altro).

```
# useradd [-c <commento>] [-d <home directory>] [-e <expire date>] [-f <inactive time>] [-g <initial group>] [-G <lista di gruppi>] [-s <shell>]
```

<home directory> se non esiste viene creata

<expire date> data di scadenza della password, scritta nel file shadow

<inactive time> scritto nel file shadow

<initial group> gruppo principale, viene scritto nel file passwd

<lista di gruppi> lista, separata da virgole, di gruppi secondari, scritti nel file groups

<shell> interprete dei comandi (path assoluta)

password "ombra", nel file shadow. In questo file ci sono le password. A differenza del file passwd, il file shadow non è leggibile da tutti gli utenti, ma solo da root.

```
# useradd
# -g gruppo di login
# -G lista di altri gruppi
# -m creazione dell'home directory se non esiste
# -k (eventualmente si può far copiare anche una struttura di directory
# "standard da /etc/skeleton)
# -c comment: stringa di commento al nuovo username
# la notazione ${nome:0:2} significa che si prendono le prime due lettere
```

1.2.2 Formato del file group

```
<nome gruppo>:<password crittata>:<GID>:<lista utenti>
```

<password> è criptata, è per chi vuole aggiungersi al gruppo con il comando newgrp: per assumere i diritti del gruppo gli verrà richiesta questa password, questo campo praticamente non è mai usato.

<GID> groupID identificatore numerico del gruppo, univoco

<lista utenti> utenti del gruppo, lista separata da virgole. Ci sono solo gli utenti che hanno questo gruppo come gruppo secondario.

Ogni utente deve appartenere ad un gruppo

il GID del gruppo di root è 0 (zero).

Per aggiungere un gruppo è meglio usare un comando il comando bash:

```
# groupadd <nuovo nome di gruppo>
```

per cancellare un gruppo:

```
# groupdel <gruppo da cancellare>
```

per vedere i gruppi in cui è un utente:

```
$ groups <nome utente>
```

1.2.3 File profile

Contiene gli ID da cui gli script di assegnazione utenti vanno a prendere il numero minimo e massimo di UID e GID; contiene:

```
GID_MIN 500
```

```
GID_MAX 60000
```

```
UID_MIN 500
```

¹ Cosa che si può sempre fare, a proprio rischio e pericolo!

UID_MAX 60000

1.2.4 Aggiunta di un nuovo utente

```
# adduser <nuovo nome di utente>  
script della distribuzione, esiste anche useradd
```

Per definire la prima password di un nuovo utente:

```
# passwd <nuovo nome di utente>  
New UNIX password:  
Retype new UNIX password:
```

per definire un nuovo utente con password vuota si deve usare questo comando:

```
# useradd -p "" <nuovo nome di utente>
```

```
$ chown
```

Cambia il proprietario di un file e, se si vuole, anche il suo gruppo.

```
$ chgrp
```

Cambia il gruppo cui appartiene un file.

File "coredump"

Quando un programma viene interrotto improvvisamente dal S.O., per qualsiasi ragione, tipicamente per un errore irrecuperabile, il S.O. produce un file di nome "core" nel directory in cui quel programma tiene i suoi dati.

Se il sistema od i programmi che vi si provano sono un po' instabili, questi file core possono occupare parti consistenti di hard disk.

Disco di boot

Usare il comando mkbootdisk, che crea un disco di boot in grado di far partire il sistema che sta correntemente girando. E' necessario dare la versione del kernel che si vuole usare.

Esempio:

Se il kernel che si usa è vmlinuz-2.2.14-12 si deve lanciare:

```
root@DottoLinux# mkbootdisk 2.2.14-12
```

in un sistema che ha un solo floppy disk non è necessario specificare il relativo nome di device.

Si noti però che da un po' di tempo il kernel di Linux precompilato nelle normali distribuzioni non sta più in un dischetto floppy.

Per vedere quali sono i servizi di default, lanciati al boot (programma Red Hat):

```
# ntsysv
```