

Editor, Utenti e Permessi

Insegnamento di Sistemi Operativi di Rete
Master Universitario in Tecnologie Internet

Domenico De Guglielmo

E-mail: domenicodegu@virgilio.it Telefono: 050 2217 468

Pervasive Computing & Networking Lab (PerLab) <http://www.perlab.it>

Dipartimento di Ingegneria dell'Informazione, Università di Pisa

Sommario



- ✓ Editor
 - vi
- ✓ Utenti e gruppi
- ✓ Permessi
- ✓ Gestione utenti

Editor di testo

Editor di testo



Esistono vari editor di testo:

vi

emacs

vim

nano

...

Editor vi

- Per creare o aprire un file già esistente:

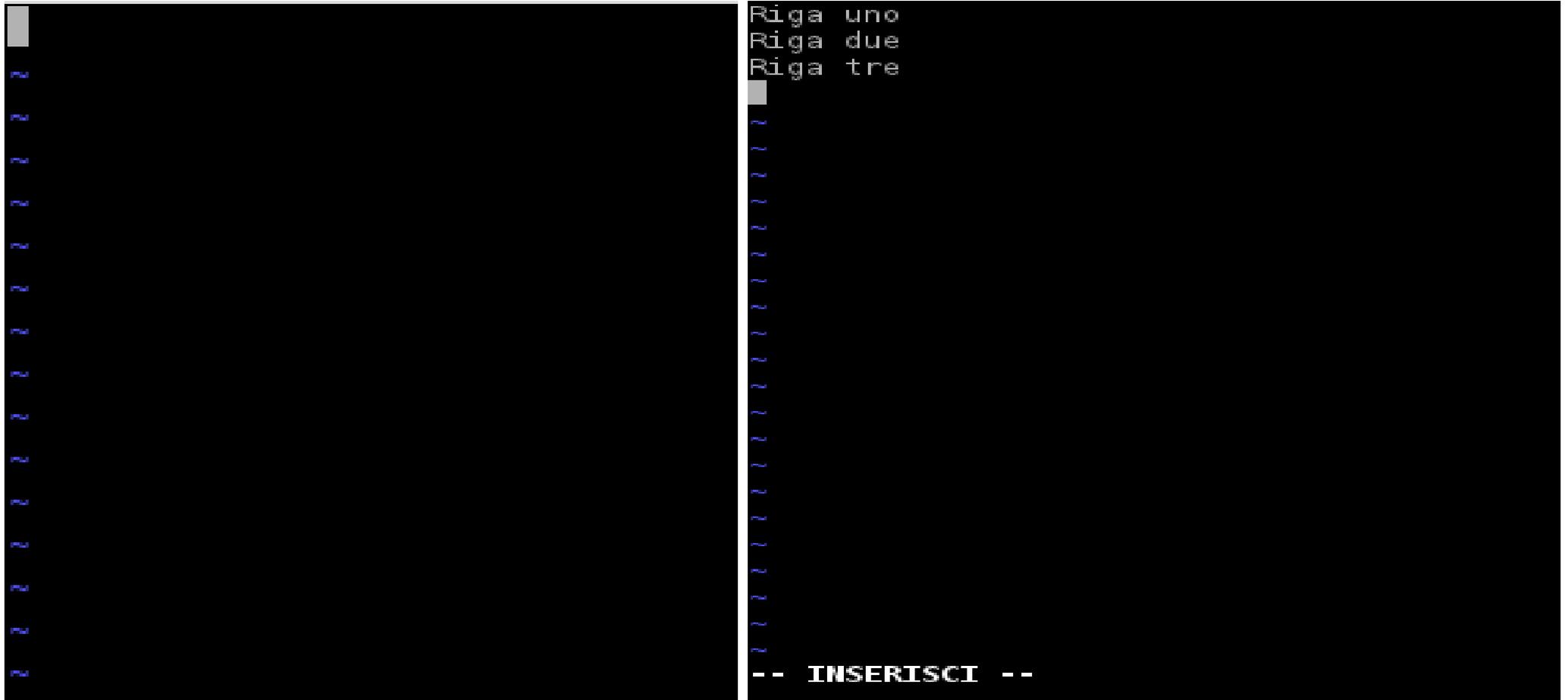
```
$ vi nomeFile
```

- Esistono due modalità di funzionamento per **vi**:

1.modalità comandi: permette di inserire comandi e scegliere quale azione compiere;

2.modalità editing: permette di inserire e cancellare testo (come se fosse un editor di testo quale blocco note).

Editor vi



```
Riga uno
Riga due
Riga tre

```

-- INSERISCI --

Editor vi

Esistono 2 tipologie di comandi:

- **Visuali:** sono i comandi più semplici e si compongono di sequenze di uno o più tasti il cui inserimento non appare in nessuna parte dello schermo. Si concludono senza la pressione del tasto Invio.
- **Due punti:** iniziano tutti con il simbolo : e terminano con Invio. Dato che sono i più complessi appaiono sulla riga inferiore dello schermo

Editor vi

Elenco di alcuni comandi dell'editor vi

<code>esc</code>	Passa in modalità comandi.
<code>i</code>	Passa in modalità inserimento nella posizione corrente.
<code>o</code>	Apri in inserimento una nuova linea sotto la posizione corrente.
<code>x</code>	Cancella un carattere.
<code>u</code>	Elimina gli effetti dell'ultimo comando.
<code>r?</code>	Sostituisce con ? il carattere su cui si trova il cursore.
<code>dd</code>	Taglia la riga corrente.
<code>n dd</code>	Taglia n righe.
<code>yy</code>	Copia una riga.
<code>n yy</code>	Copia n righe.

Editor vi

Elenco di alcuni comandi dell'editor vi

<code>p</code>	Incolla la selezione nella riga sotto il cursore.
<code>/word</code>	Ricerca nel testo la parola word
<code>n</code>	Si posiziona sull'occorrenza successiva (nella ricerca).
<code>N</code>	Si posiziona sull'occorrenza precedente (nella ricerca).
<code>:q</code>	Esce (solo se non si sono fatte modifiche).
<code>:w</code>	Salva.
<code>:wq</code>	Salva ed esce.
<code>:q!</code>	Esce senza salvare.
<code>:help</code>	

Editor con interfaccia grafica



gvim

kvim

emacs

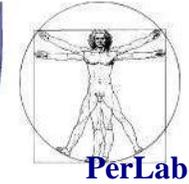
kate

gedit

...

Utenti e gruppi

Utenti



Ogni utente è identificato da:

- uno **username**
- un **UID** (user ID) numerico
- Username e UID sono pubblici.
- Ogni utente può appartenere ad uno o più gruppi.

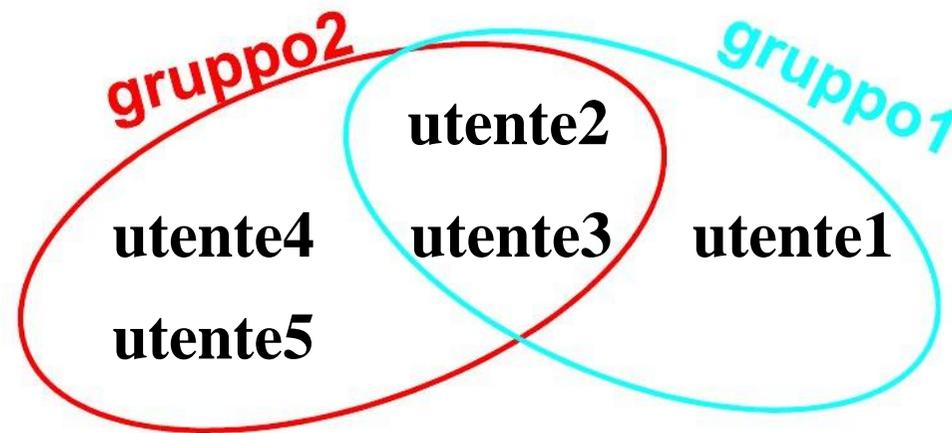
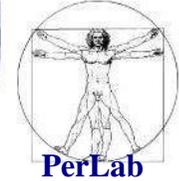
Gruppi (1 di 2)



Ogni gruppo è identificato da:

- un **groupname**,
- un **GUID** (group ID) numerico.
- L'utente **root** appartiene di default al gruppo **wheel**.

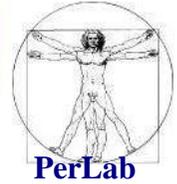
Gruppi (2 di 2)



- **utente1** appartiene al gruppo **gruppo1**.
- **utente2** appartiene ai gruppi **gruppo1** e **gruppo2**.
- **utente3** appartiene ai gruppi **gruppo1** e **gruppo2**.
- **utente4** appartiene al gruppo **gruppo2**.
- **utente5** appartiene al gruppo **gruppo2**.

Permessi

Gestione dei permessi



I permessi possono essere di:

- **lettura,**
- **scrittura,**
- **esecuzione.**

I permessi da applicare vengono differenziati in base alla natura dell'utente che vuole “operare” sul file, directory, ... :

- a) utente proprietario del file: **owner**
- b) gruppo proprietario: **group owner,**
- c) gli altri utenti: **others.**

Gestione dei permessi: file



Attributo	Significato
Lettura	Permette di leggere il contenuto del file
Scrittura	Permette di modificare il contenuto del file
Esecuzione	Permette di eseguire un file (binario o script)

Il permesso di scrittura:

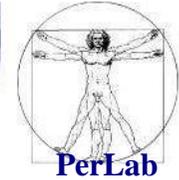
- non incide sulla possibilità di cancellare un file
bisogna considerare i permessi della directory.

Gestione dei permessi: directory



Attributo	Significato
Lettura	Permette di leggere il contenuto (l'elenco dei file)
Scrittura	Permette di modificare il contenuto della cartella
Esecuzione	Permette di attraversare una cartella

- Negare l'accesso in **lettura** impedisce l'esecuzione con successo del comando ls.
- Negare l'accesso in **scrittura** impedisce di creare, cancellare e rinominare file.
- Negare l'accesso in **esecuzione** impedisce di fare un cd sulla directory.



I permessi di un file, directory,... possono essere visualizzati con il comando `ls -l`:

`drwxr-xr-x ... prova`

`t rwx rwx rwx`

others

group

owner

t : tipo di file, **d** : directory, **-** : file, ...

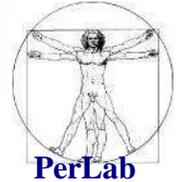
r: permesso di lettura,

w: permesso di scrittura,

x: permesso di esecuzione,

- indica l'assenza del permesso corrispondente.

Rappresentazione ottale

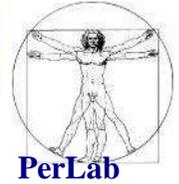


La **rappresentazione ottale** dei permessi consiste in un numero di 4 cifre in base 8.

- **lettura:** 4
- **scrittura:** 2
- **esecuzione:** 1

Il numero che rappresenta i permessi attribuiti ad un tipo di utente (owner, group, others) si ottiene sommando i numeri corrispondenti ai privilegi che si vogliono concedere.

Rappresentazione ottale



Esempi:

644: lettura e scrittura per l'owner, lettura per il gruppo proprietario e lettura per gli altri

750: lettura, scrittura ed esecuzione per l'owner, lettura ed esecuzione per il gruppo, nulla per gli altri

Domande:

600 : quali sono i permessi che ha ogni tipo di utente?

Se volessi dare lettura e scrittura all'owner e gruppo e nulla agli altri?

Permessi speciali



Attributo	Significato
SUID	Durante l'esecuzione il comando acquisisce i privilegi del proprietario del file stesso
SGID	Durante l'esecuzione il comando acquisisce i privilegi del gruppo a cui il file appartiene
Sticky	Durante l'esecuzione di un comando salva l'immagine nella memoria virtuale

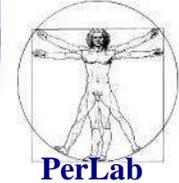
Attributo **sticky** su cartelle

- non consente ad un utente di cancellare o rinominare un file a meno che non ne sia il proprietario (anche se ha il permesso in scrittura sulla directory).

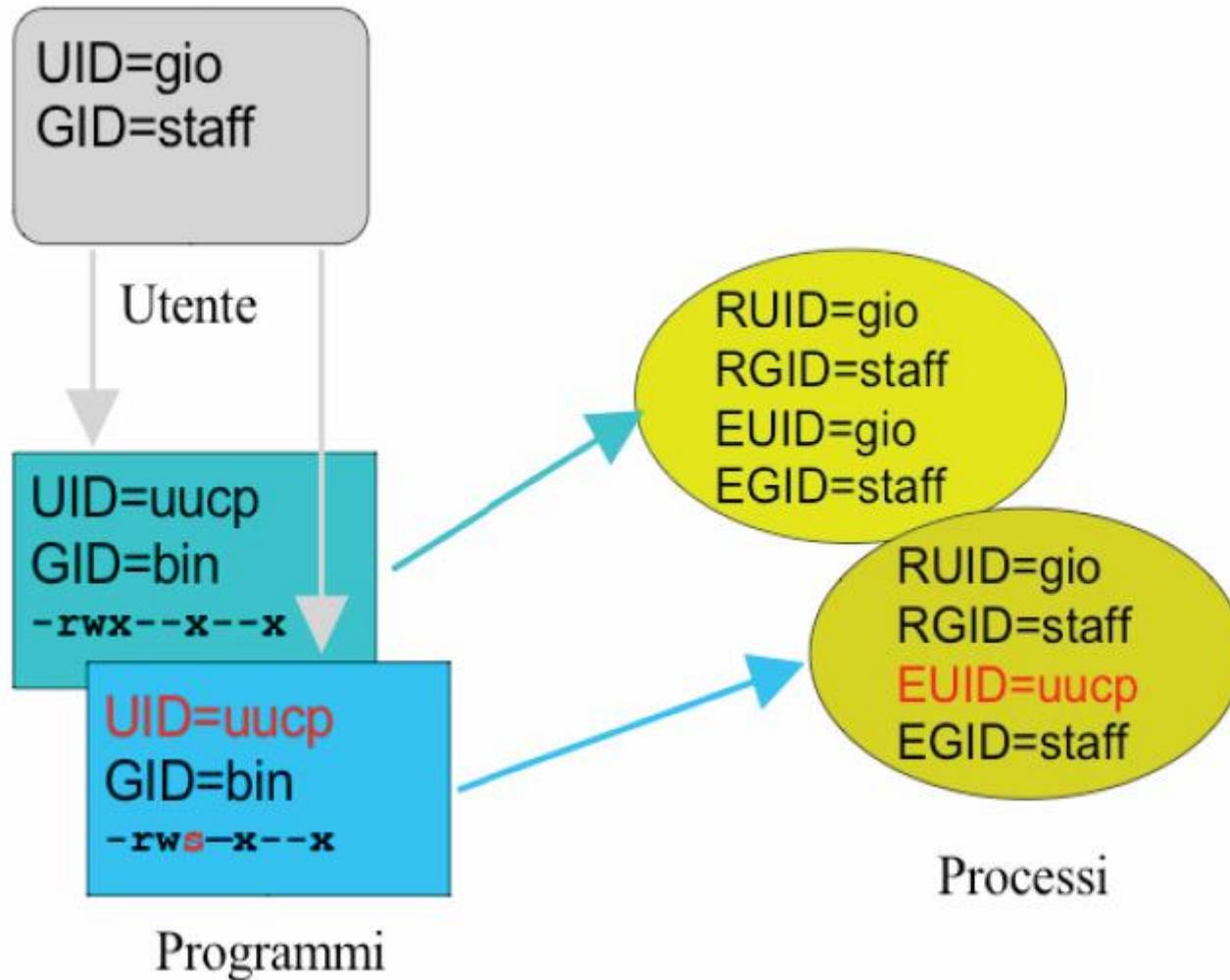


- Il permesso di esecuzione in **owner** può essere anche:
 - s** : SUID ed esecuzione permessi,
 - S** : solo SUID.
- Il permesso di esecuzione in **group owner** può essere anche:
 - s** : SGID ed esecuzione permessi,
 - S** : solo SGID.
- Il permesso di esecuzione in **others** può essere anche:
 - t** : sticky ed esecuzione permessi,
 - T**: solo sticky.

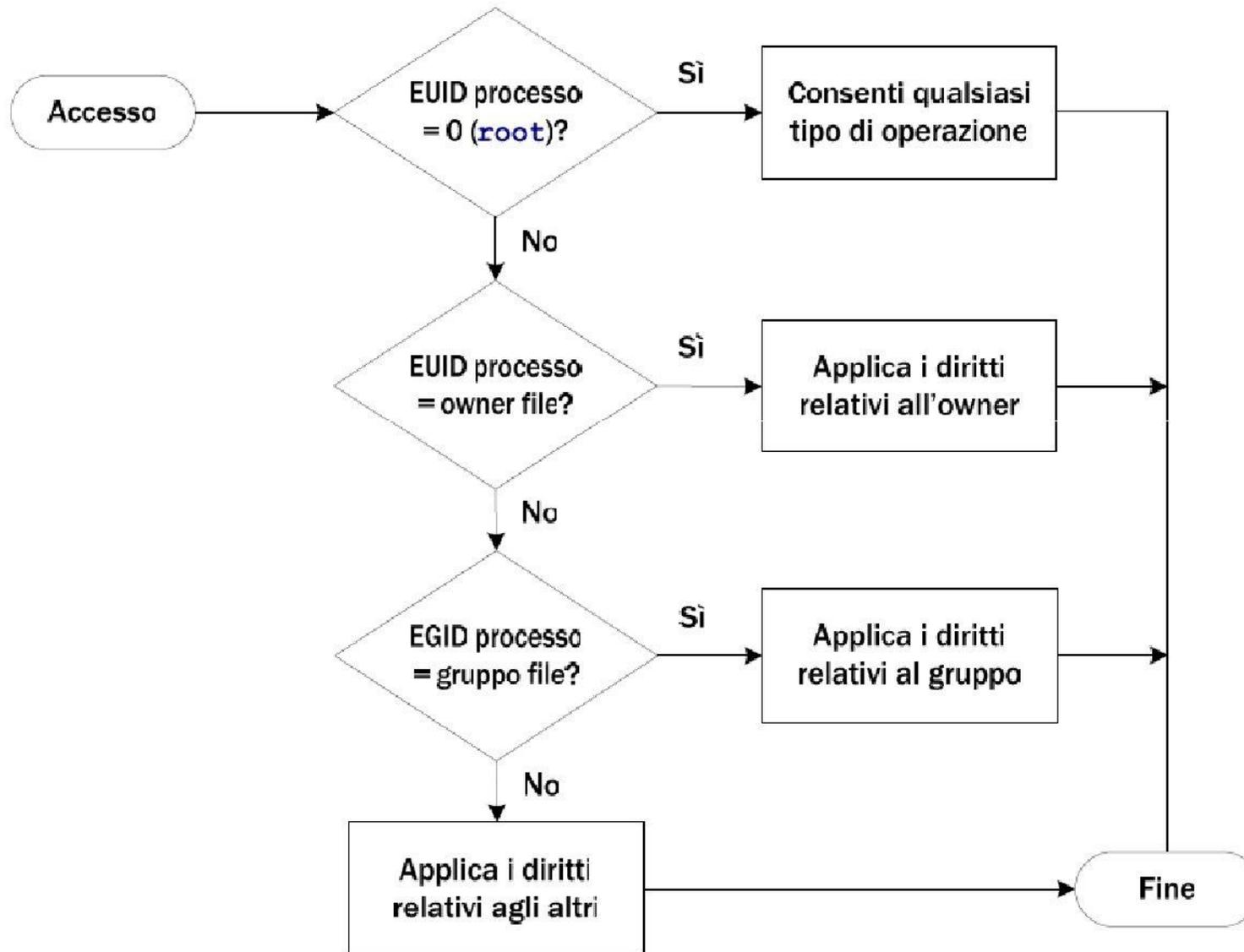
Real e Effective User ID



Suffisso	Gruppo	Descrizione
UID	Real	Utente che ha mandato in esecuzione il programma
GID	Real	Gruppo principale dell'utente che ha mandato in esecuzione il programma
EUID	Effective	Utente usato nel controllo degli accessi (privilegi effettivi utente)
EGID	Effective	Gruppo usato nel controllo degli accessi (privilegi effettivi gruppo)



Applicazione dei permessi



Applicazione dei permessi

Il file `/etc/passwd` contiene le password associate ad ogni utente. Naturalmente agli utenti è concesso cambiare la propria password, esiste infatti il comando “passwd” che serve appositamente a questo. Se si esegue un `ls -l` di `/etc/passwd` si ottiene:

```
-rw-r--r--  root  root  /etc/passwd
```

Questo significa che l'unico utente abilitato a modificare il file delle password è il root. Ma allora come fa un utente normale a cambiare la password senza avere i privilegi di root?

Se si fa un `ls -l` in `/usr/bin/passwd` si ottiene:

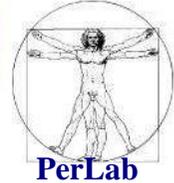
```
-r-sr-xr-x  root  root  /usr/bin/passwd
```

Proprietari dei file



Quando viene creato un file/directory/...

- l'**owner** è rappresentato dall' **EUID**.
- il **group owner** è rappresentato dall' **EGUID**.



chmod permette di modificare i permessi.

La sintassi generale di **chmod** è la seguente:

chmod [who][how][which]

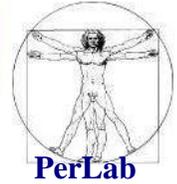
who: a chi modificare i permessi: owner (u),
group (g), others (o), all (a).

how:

'+' aggiunge un permesso,

'-' toglie un permesso,

'=' assegna un permesso.



Rappresentazione simbolica:

```
chmod a+rwx directory1  
chmod a-rw directory1  
chmod u=x directory1
```

Rappresentazione ottale:

```
chmod 0741 file
```

opzione -R : modifica ricorsivamente i permessi delle directory indicate e del loro contenuto.

Comandi chown, chgrp



Il comando `chown` permette di modificare il proprietario e/o il gruppo assegnato ad un file o directory.

`chown federico prova`

L'utente root cambia l'utente proprietario del file prova, facendo in modo che diventi federico.

`chown fede.users prova`

L'utente root cambia l'utente ed il gruppo proprietario del file prova, facendo in modo che diventino rispettivamente fede e users.

Il comando `chgrp` permette di modificare il gruppo assegnato ad un file o ad una directory.

Comando umask



Quando viene creato un file:

- il file appartiene automaticamente all'utente che lo crea ed al gruppo principale dell'utente stesso,
- i permessi gli vengono attribuiti in base alla **maschera dei permessi** (umask):
 - tale maschera rappresenta i permessi che **NON** vengono attribuiti.
 - il suo valore, in generale, è tale da non attribuire il permesso di scrittura né al gruppo proprietario, né agli altri utenti.
 - il valore di questa maschera può essere modificato (temporaneamente) con il comando: umask.

Gestione utenti

Comando su



Il comando su (switch user o substitute user) viene tipicamente usato da:

- **utenti non privilegiati** per avviare una **shell** come **root**
 - per poterlo eseguire l'utente deve appartenere anche al gruppo **wheel**
- **root** per avviare una **shell** come **utente ordinario**.

Richiede autenticazione

- password dell'utente **che si vuole diventare**.

Comando sudo



- **sudo** (superuser *do*) consente di eseguire un comando come se si fosse un altro utente, previa autorizzazione.
- Per eseguire dei comandi con privilegi d'amministrazione è sufficiente digitare **sudo** e successivamente il comando che si desidera eseguire come utente **root**.
- Mentre con il comando **su** si cambia utente fino al termine della sessione del terminale, **sudo** assegna i privilegi dell'utente **target** al solo processo che viene con esso avviato.

Comandi passwd, whoami, id, groups



- passwd : permette di cambiare la password.
- whoami : visualizza l'EUID.
- id : visualizza UID e *GID dell'utente selezionato o di quello corrente.*
- groups : visualizza i gruppi ai quali l'utente appartiene.

Creazione e rimozione utenti

Creazione e rimozione degli utenti



- `adduser` : crea un nuovo utente “nomeUtente”.
- `adduser nomeUtente`

- `userdel` : rimuove l'utente “nomeUtente”.
- `userdel nomeUtente`

File di configurazione utenti



File con le informazioni sugli utenti:

- /etc/passwd

File con le informazioni sui gruppi di utenti:

- /etc/group

File con le informazioni sugli utenti e le password criptate:

- /etc/shadow
- Nasconde le password cifrate ai processi che non hanno privilegi di root.

File /etc/passwd (1 di 2)



password dati personali

osor:*:1001:1001:osor:/home/osor:/bin/sh

utente UID GID home shell

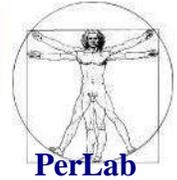
Diagram illustrating the structure of a line in the /etc/passwd file. The line is: osor:*:1001:1001:osor:/home/osor:/bin/sh. Red arrows point from labels to the corresponding fields in the line: 'password' points to the asterisk (*), 'dati personali' points to the osor field, 'utente' points to osor, 'UID' points to 1001, 'GID' points to 1001, 'home' points to /home/osor, and 'shell' points to /bin/sh.

File /etc/passwd (2 di 2)



Campo	Significato
utente	Nome utente
password	Password
UID (user id)	Identificatore univoco dell'utente (numero)
GID (group id)	Identificatore univoco del gruppo (numero)
dati personali	Nominativo completo e altre informazioni
cartella home	Percorso assoluto della cartella personale (home)
shell	Interprete dei comandi da utilizzare

Formato del file /etc/group



Campo	Significato
gruppo	Nome del gruppo
password	Password cifrata del gruppo
GID (group id)	Identificatore univoco del gruppo (numero)
lista utenti	Utenti appartenenti al gruppo (separati da virgole)

password di gruppo

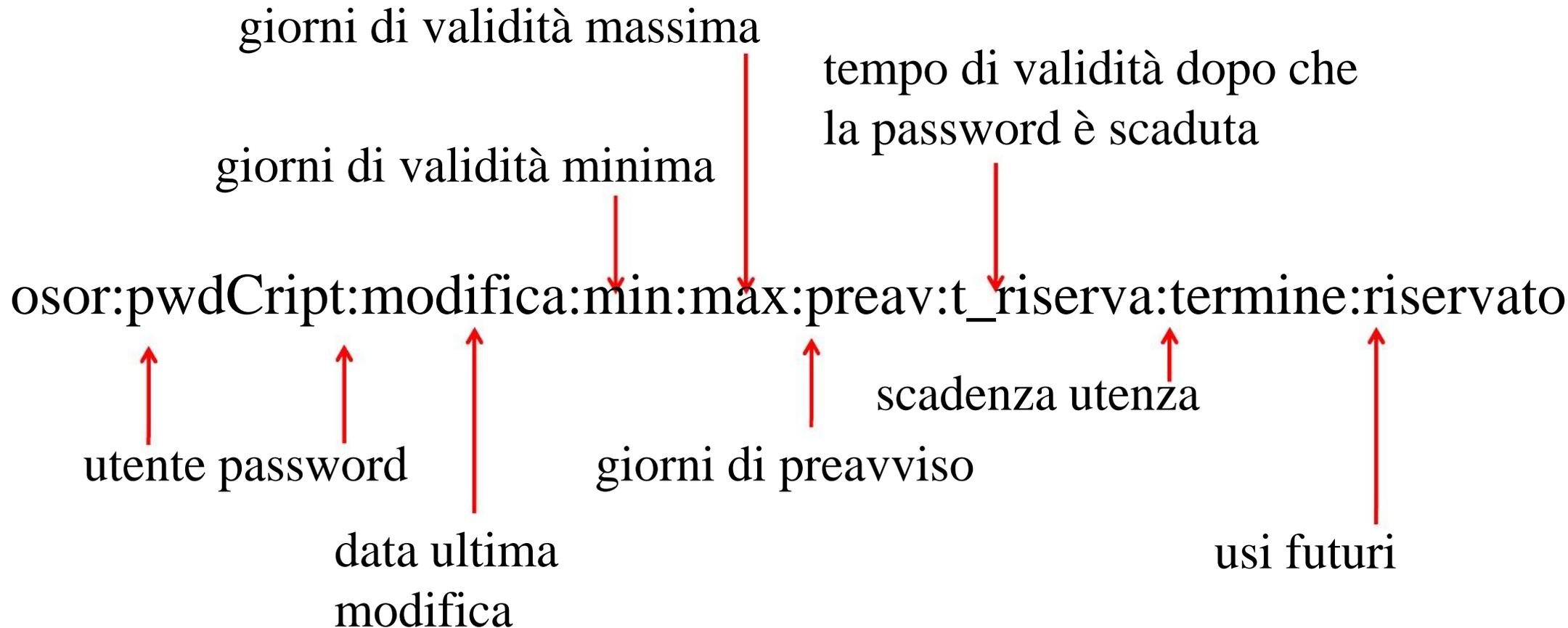
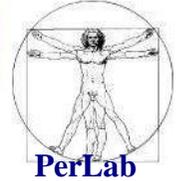
elenco utenti

studenti:x:1010:alessio,giuseppe,giovanni

gruppo

GID

File /etc/shadow



Procedura manuale creazione utente 1/3



1. Editare il file `/etc/passwd` aggiungendo una riga per il nuovo utente:
 1. utilizzare un **UID** libero
 2. conviene lasciare la password vuota (x) e cambiarla in seguito con il comando `passwd`.

`nomeutente:x:UID:GID:grupputente:/home/nomeutente:/bin/shellutente`

2. Editare `/etc/group` aggiungendo un nuovo gruppo per il nuovo utente:
 1. utilizzare un **GID** libero.

`nomeutente:x:GID:`

Procedura manuale creazione utente 2/3



3. Editare `/etc/shadow` aggiungendo una nuova riga per l'utente creato.

```
nomeutente:::::::::
```

4. Creare la home del nuovo utente.

```
mkdir /home/nomeutente
```

5. Creare l'ambiente base nella **home**

1. copiare tutti i file (compresi i file nascosti) in `/etc/skel` nella home dell'utente.

```
cp /etc/skel/* /home/nomeutente/
```

Procedura manuale creazione utente 3/3



6. Cambiare proprietario della home del nuovo utente.

```
chown -R nomeutente:nomegruppo  
/home/nomeutente
```

7. Cambiare i permessi della cartella **home**

1. accesso completo al solo proprietario.

```
chmod 700 /home/nomeutente
```

8. Impostare la password dell'utente

```
passwd nomeutente
```

Risorse e riferimenti



Riferimenti su dispensa:

'Amministrazione di un Sistema UNIX in Rete'

Cap. 3, 7 e 8