



LABORATORIO DI SISTEMI OPERATIVI

Corso di Laurea in Ingegneria Informatica
A.A. 2019/2020

Ing. Guglielmo Cola

 g.cola@iet.unipi.it (specificare "sistemi operativi" nell'oggetto)

 www.iet.unipi.it/g.cola/sistemioperativi

ESERCITAZIONE 2

Utenti e gruppi (prima parte)

Permessi di accesso al filesystem

Editor di testo

Esercizi

Utenti e gruppi

- Ogni utente è identificato da:
 - Username
 - UID (user ID) numerico

- Ogni gruppo è identificato da:
 - Group name
 - GID (group ID) numerico

- Ogni utente deve avere appartenere almeno ad un gruppo (primary group)

Gestione utenti – comandi vari

- `passwd`
 - Permette di cambiare la password (sfrutta il permesso SUID)
- `id [username]`
 - Visualizza UID, gruppo principale e altri gruppi dell'utente corrente o di quello selezionato
- `groups [username]`
 - Visualizza i nomi dei gruppi dell'utente corrente o di quello selezionato

Gestione utenti – creazione e rimozione

- Per aggiungere/rimuovere utenti è necessario avere i privilegi di root
- Creazione di un utente:
 - `adduser username`
- Rimozione di un utente:
 - `deluser username`

Gestione utenti – su e sudo

- `su` (switch user)
Permette di accedere al terminale di un altro utente, o dell'utente root
 - Se l'utente non è specificato si richiede di accedere al terminale di root
 - Viene chiesta la password dell'utente specificato
- `sudo nome_comando`
Permette di eseguire un comando con i privilegi di un altro utente
 - Se non specificato si usa l'utente root
 - Viene chiesta la password dell'utente corrente
 - L'utente deve far parte del gruppo sudoers

Permessi di accesso al filesystem

- Il meccanismo dei permessi gestisce l'accesso al file system da parte dei vari utenti del sistema
- Per ogni file (e directory) sono definiti
 - Un utente proprietario (owner)
 - Un gruppo proprietario (group owner)
- Di conseguenza, per ogni file ci sono tre classi di utenti:
 - Il proprietario del file (owner)
 - Gli utenti appartenenti al gruppo proprietario
 - Gli altri utenti (others)

Permessi di accesso al filesystem

- A ciascuna classe di utenti (proprietario, appartenenti al gruppo proprietario, altri) vengono applicati permessi specifici
- I permessi possono essere di accesso in:
 - `r` – read (lettura)
 - `w` – write (scrittura)
 - `x` – eXecute (esecuzione)

Permessi di accesso al filesystem

- Quando un utente prova ad utilizzare un file, vengono applicati i permessi:
 - Relativi all'owner se l'utente è il proprietario del file
 - Relativi al group owner, se l'utente non è proprietario del file, ma appartiene al gruppo proprietario
 - Validi per tutti gli altri utenti (others), se l'utente non è proprietario e non appartiene al gruppo proprietario

Permessi – file

Attributo	Significato
r	Permette di leggere il contenuto del file
w	Permette di modificare il contenuto del file
x	Permette di eseguire un file (binario o script)

- Il permesso di scrittura non permette di cancellare un file:
 - Per la cancellazione di file valgono i permessi della directory

Permessi – directory

Attributo	Significato
r	Permette di leggere il contenuto (elenco dei file)
w	Permette di modificare il contenuto
x	Permette di attraversare una cartella

- Negare l'accesso in lettura impedisce l'esecuzione del comando ls
- Negare l'accesso in scrittura impedisce di creare, rinominare, cancellare file
- Negare l'accesso in esecuzione impedisce di utilizzare il comando cd sulla directory

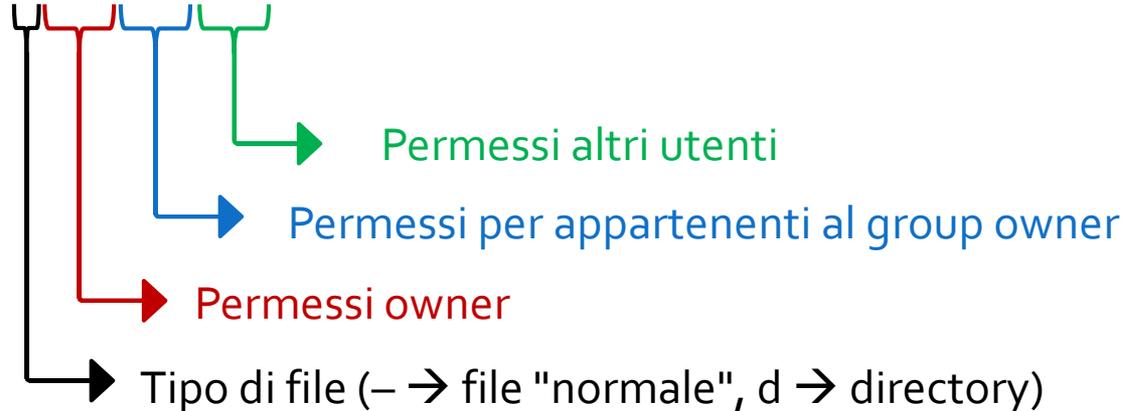
Permessi – rappresentazione simbolica

- I permessi di un file (o directory) possono essere visualizzati con il comando

```
ls -l
```

- Esempio di output:

```
-rw-r--r-- 1 studenti studenti 10 ott 9 13:17 esempio.txt
-rw-r--r-- 1 root      root      0 ott 9 13:28 root_file.txt
drwxr-xr-x 2 studenti studenti 4096 ott 9 13:19 subDir
```



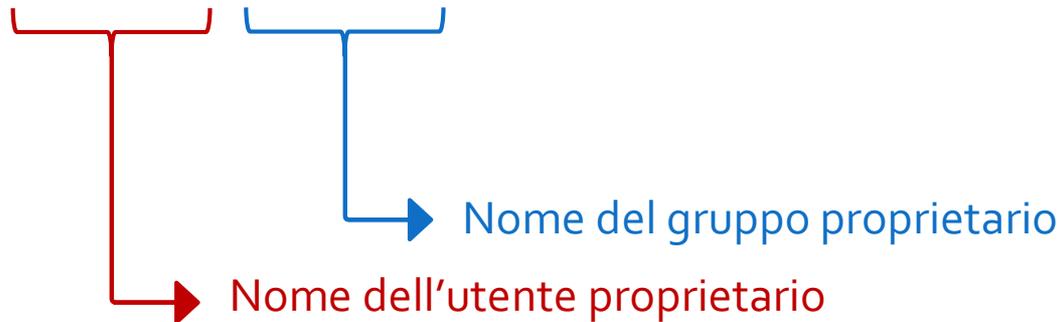
Permessi – rappresentazione simbolica

- I permessi di un file (o directory) possono essere visualizzati con il comando

```
ls -l
```

- Esempio di output:

```
-rw-r--r-- 1 studenti studenti 10 ott 9 13:17 esempio.txt
-rw-r--r-- 1 root      root      0 ott 9 13:28 root_file.txt
drwxr-xr-x 2 studenti studenti 4096 ott 9 13:19 subDir
```



Permessi – rappresentazione ottale

- Tre cifre in base 8
 - Rappresentano: i permessi dell'owner, del group owner, degli altri utenti
 - Ciascuna di queste cifre è ottenuta sommando:
 - 4 se è permessa la lettura
 - 2 se è permessa la scrittura
 - 1 se è permessa l'esecuzione
- Esempi:
 - 777
 - Sono garantiti tutti i permessi (4+2+1) a tutti gli utenti
 - 750
 - Il proprietario ha tutti i permessi (4+2+1), il group owner ha permesso in lettura ed esecuzione (4+1), gli altri utenti non hanno nessun permesso

Permessi – comando chmod

- Il comando `chmod`
 - Permette di modificare i permessi relativi ad uno o più file
 - È possibile usare la rappresentazione simbolica o quella ottale
 - L'opzione `-R` permette di modificare in modo ricorsivo i permessi di una directory e dei file/directory in essa contenuti
- Esempio con la rappresentazione ottale

```
chmod 755 file
```

Permessi – comando chmod

- Sintassi con rappresentazione simbolica

```
chmod [who][how][which] fileName
```
- `who` indica la classe di utenti per cui devono essere modificati i permessi:
 - `u` per l'owner
 - `g` per il group owner
 - `o` per gli altri
- `how` indica in che modo devono essere modificati i permessi
 - `+` per aggiungere permessi
 - `-` per togliere permessi
 - `=` per assegnare permessi
- Esempio

```
chmod go-rwx file
```

(toglie tutti i permessi di accesso a «file» a group owner e altri utenti)

Permessi aggiuntivi – SUID, SGID

Attributo	Significato
SUID	Durante l'esecuzione il processo acquisisce i privilegi del proprietario del file (normalmente un processo acquisisce i privilegi di chi lo esegue)
SGID	Durante l'esecuzione il processo acquisisce i privilegi del gruppo proprietario del file (normalmente un processo ha i privilegi del gruppo di chi lo esegue)

Permessi aggiuntivi – SUID, SGID

- Rappresentazione simbolica di SUID
 - Si utilizza il campo relativo ai permessi in esecuzione del file owner e la lettera s (invece di x) per indicare il permesso in esecuzione con SUID
- Rappresentazione simbolica di SGID
 - Si utilizza il campo relativo ai permessi in esecuzione del group owner, allo stesso modo del SUID
- Ad esempio, eseguendo
`ls -l /usr/bin/passwd`
si ottiene come rappresentazione simbolica:

```
-rwsr-xr-x
```

Permessi aggiuntivi – SUID, SGID

- Per la rappresentazione ottale si utilizza un'ulteriore cifra prima delle 3 cifre relative alle classi di utenti. Questa cifra ottale è ottenuta come somma di
 - 4 se è attivo il permesso SUID
 - 2 se è attivo il permesso SGID
- Ad esempio 6754 corrisponde alla seguente rappresentazione simbolica

`rwsr-sr--`

Comandi chown e chgrp

- `chown username file`
Permette di impostare username come nuovo proprietario di file
 - Può essere eseguito solo dall'utente root
- `chgrp groupname file`
Permette di impostare groupname come gruppo proprietario di file
 - Un utente normale può eseguire il comando solo se appartiene a groupname
 - Altrimenti è necessario essere root

Editor di testo da terminale

- vi
- emacs
- vim
- nano
- ...

vi

- Per modificare un file esistente o creare un nuovo file:

```
vi nome_file
```

- Esistono due modalità di funzionamento:

1. Modalità comandi – permette di inserire comandi e scegliere quale azione compiere
2. Modalità editing – permette di inserire e cancellare testo (come un normale editor di testo).

vi

Elenco di alcuni comandi dell'editor vi

Esc	Passa in modalità comandi
i	Passa in modalità inserimento nella posizione corrente
o	Inserisce una nuova linea dopo quella corrente
x	Cancella il carattere corrente
u	Annulla l'ultimo comando sulla linea corrente
r?	Sostituisce con ? il carattere su cui si trova il cursore
dd	Cancella la riga corrente
ndd	Cancella n righe
yy	Copia una riga
nyy	Copia n righe

vi

Elenco di alcuni comandi dell'editor vi

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

Editor con interfaccia grafica

- gedit
- kate
- gvim
- kvim
- ...

ESERCIZI

Esercizio 1

- Lavorare nella propria cartella home
- Creare una cartella con nome *visibile* e al suo interno una cartella con nome *segreta*
- Scrivere la stringa *vero* nel file *notizia.txt* all'interno di *visibile*
- Copiare *notizia.txt* all'interno di *segreta* assegnandole il nome *cronaca.txt*
- Lavorare sui permessi di *visibile*
 - Togliere il permesso di esecuzione (proprietario) a *visibile* usando la rappresentazione simbolica
 - Ripristinare il diritto di esecuzione (proprietario) a *visibile* usando la rappresentazione simbolica
 - Togliere di nuovo il diritto di esecuzione (proprietario) usando la rappresentazione ottale e lasciando invariati gli altri permessi

Esercizio 1

- A questo punto:
 - Si riesce a vedere il contenuto di visibile?
 - Si riesce a vedere il file notizia.txt dentro visibile?
 - Si riesce a vedere il contenuto di segreta?
 - Si riesce a vedere il file cronaca.txt dentro segreta?
- Ripristinare il permesso di esecuzione a visibile e togliere il permesso in lettura a segreta (per l'utente proprietario)
 - Riesco a vedere il contenuto di segreta?
 - Riesco a leggere il contenuto di cronaca.txt dentro segreta?

Esercizio 2

- Creare un utente utente2
 - Si riesce a vedere il contenuto della home di utente2 con le proprie credenziali utente?
 - Eventualmente cambiare i diritti in modo che gli altri utenti non riescano a vedere il contenuto della home di utente2
- Controllare a quali gruppi appartiene l'utente root

- Creare un utente utente3
- Creare la cartella temp nella home di utente3
- Quali sono l'utente proprietario e il gruppo proprietario di temp?
- Cambiare utente proprietario e gruppo proprietario di temp con utente3 e verificare che sia avvenuto l'aggiornamento di tali campi
- Rimuovere utente2 ed utente3