

Esercitazione GNU/Linux n. 5

5.1 Sicurezza nel filesystem (i diritti d'accesso)

Dalla directory **corsolinux**, digita

```
ls -l (l sta per lunga lista)
```

Vedrai che ora otterrai maggiori informazioni sul contenuto della directory, come nell'esempio sotto

```
-rwxrw-r-- 1 stefano users 2450 Sept 29 11:52 file1
```

A ciascun file (e directory) sono associati i diritti d'accesso, che si possono sapere digitando appunto `ls -l`.

Nella colonna a sinistra c'è un gruppo di 10 lettere che consistono nei simboli `d`, `r`, `w`, `x`, `-`, `e`, occasionalmente, `s` o `S`. Se la lettera `d` è presente all'inizio di questa sequenza allora significa che si tratta di una directory, altrimenti `-` è il simbolo con cui inizia la stringa.

Le rimanenti 9 lettere indicano i permessi, o i diritti d'accesso, e sono in gruppi di 3.

- il gruppo a sinistra mostra i permessi dell'utente cui appartiene il file (o la directory) (stefano nell'esempio precedente);
- il gruppo centrale mostra i permessi per il gruppo di utenti a cui il file (o la directory) appartiene (users nell'esempio precedente);
- il gruppo più a destra mostra i permessi per tutti gli altri.

Le lettere `r`, `w`, ecc., hanno significati leggermente diversi a seconda se si riferiscono a un semplice file o a una directory.

Diritti d'accesso ai file

- `r` (or `-`), indicates read permission (or otherwise), that is, the presence or absence of permission to read and copy the file
- `w` (or `-`), indicates write permission (or otherwise), that is, the permission (or otherwise) to change a file
- `x` (or `-`), indicates execution permission (or otherwise), that is, the permission to execute a file, where appropriate

Diritti d'accesso alle directory

- `r` allows users to list files in the directory;
- `w` means that users may delete files from the directory or move files into it;

- `x` means the right to access files in the directory. This implies that you may read files in the directory provided you have read permission on the individual files.

So, in order to read a file, you must have execute permission on the directory containing that file, and hence on any directory containing that directory as a subdirectory, and so on, up the tree.

Alcuni esempi

<code>-rwxrwxrwx</code>	a file that everyone can read, write and execute (and delete).
<code>-rw-----</code>	a file that only the owner can read and write - no-one else can read or write and no-one has execution rights (e.g. your mailbox file).

5.2 Cambiare i diritti d'accesso

chmod (changing a file mode)

Only the owner of a file can use `chmod` to change the permissions of a file. The options of `chmod` are as follows

Carattere	Significato
<code>u</code>	utente
<code>g</code>	gruppo
<code>o</code>	gli altri
<code>a</code>	tutti
<code>r</code>	lettura
<code>w</code>	scrittura (e cancellazione)
<code>x</code>	esecuzione (e accesso alle directory)
<code>+</code>	aggiungi un permesso
<code>-</code>	rimuovi un permesso

For example, to remove read write and execute permissions on the file **listagrande** for the group and others, type

```
chmod go-rwx listagrande
```

This will leave the other permissions unaffected.
To give read and write permissions on the file **listagrande** to all,

```
chmod a+rw listagrande
```

Esercizio 5a

Try changing access permissions on the file **science.txt** and on the directory **backups**. Use `ls -l` to check that the permissions have changed.

5.3 Processi e job

A process is an executing program identified by a unique PID (process identifier). To see information about your processes, with their associated PID and status, type

```
ps
```

A process may be in the foreground, in the background, or be suspended. In general the shell does not return the UNIX prompt until the current process has finished executing. Some processes take a long time to run and hold up the terminal. Backgrounding a long process has the effect that the UNIX prompt is returned immediately, and other tasks can be carried out while the original process continues executing.

Eseguire i processi in background

To background a process, type an **&** at the end of the command line. For example, the command `sleep` waits a given number of seconds before continuing. Type

```
sleep 10
```

This will wait 10 seconds before returning the command prompt `%`. Until the command prompt is returned, you can do nothing except wait. To run `sleep` in the background, type

```
sleep 10 &  
[1] 6259
```

The **&** runs the job in the background and returns the prompt straight away, allowing you to run other programs while waiting for that one to finish.

The first line in the above example is typed in by the user; the next line, indicating job number and PID, is returned by the machine. The user is notified of a job number (numbered from 1) enclosed in square brackets, together with a PID and is notified when a background process is finished. Backgrounding is useful for jobs which will take a long time to complete.

Mettere in background un processo attivo

At the prompt, type

```
sleep 1000
```

You can suspend the process running in the foreground by typing **^Z**, i.e. hold down the **[Ctrl]** key and type **[z]**. Then to put it in the background, type

```
bg
```

Note: do not background programs that require user interaction e.g. vi

5.4 Elenca i processi sospesi e in background

When a process is running, backgrounded or suspended, it will be entered onto a list along with a job number. To examine this list, type

```
jobs
```

An example of a job list could be

1. Suspended sleep 1000
2. Running netscape
3. Running matlab

To restart (foreground) a suspended processes, type

```
fg %numero_del_job
```

For example, to restart sleep 1000, type

```
fg %1
```

Typing fg with no job number foregrounds the last suspended process.

5.5 Uccidere i processi

kill (uccide un processo)

It is sometimes necessary to kill a process (for example, when an executing program is in an infinite loop)

To kill a job running in the foreground, type **^C** (control c). For example, run

```
sleep 100  
^C
```

To kill a suspended or background process, type

```
kill %numero_del_job
```

For example, run

```
sleep 100 &  
jobs
```

If it is job number 4, type

```
kill %4
```

To check whether this has worked, examine the job list again to see if the process has been removed.

ps (process status)

Alternatively, processes can be killed by finding their process numbers (PIDs) and using kill PID_number

```
sleep 1000 &  
ps  
PID TT S TIME COMMAND  
20077 pts/5 S 0:05 sleep 1000  
21563 pts/5 T 0:00 firefox
```

To kill off the process **sleep 1000**, type

```
kill 20077
```

and then type ps again to see if it has been removed from the list.
If a process refuses to be killed, uses the **-9** option, i.e. type

```
kill -9 20077
```

Note: It is not possible to kill off other users' processes !!!

Sommario

Comando	Significato
ls -lag	list access rights for all files
chmod [options] file	change access rights for named file
comando &	run command in background

<code>^C</code>	kill the job running in the foreground
<code>^Z</code>	suspend the job running in the foreground
<code>bg</code>	background the suspended job
<code>jobs</code>	list current jobs
<code>fg %1</code>	foreground job number 1
<code>kill %1</code>	kill job number 1
<code>ps</code>	list current processes
<code>kill 26152</code>	kill process number 26152

whitone@netstudent.polito.it, © 15 dicembre 2009

m.stonebank@surrey.ac.uk, © 9 ottobre 2000

Questa esercitazione è rilasciata sotto [Creative Commons License](https://creativecommons.org/licenses/by/4.0/).