

CORSO LINUX DA ZERO AD AVANZATO - BLOCCO BASE

INDICE

1. Chi siamo veramente?
2. Navigare le directory e visualizzare il loro contenuto
3. Creazione, modifica ed eliminazione di file e directory
4. User e superuser
5. Gestione utenti e gruppi
6. Gestione permessi
7. Ricerca nel filesystem e tipologie di file
8. Aggiornamento e management dei pacchetti
9. Comandi e strumenti utili

1. CHI SIAMO VERAMENTE?

Non c'era modo migliore per iniziare con una bella crisi di identita'. Ma presto guariremo grazie ai comandi di Linux.

- **WHOAMI**

Questo comando ci consente di sapere l'effective user su cui stiamo lavorando.

- **ID**

Simile al precednete, ci consente di sapere l'user, il suo gruppo e poi tutti gli altri gruppi. Insieme a queste info vengono dati anche I numeri corrispondenti a ciascuna voce.

Se $n \geq 1000$ gli user/groups sono pensati per essere usati da noi umani

Se $n < 1000$ stiamo indicando system users utili durante l'esecuzione dell'os

ESEMPIO	COMMENTO
<code>uid=1000(spidy)</code>	User id
<code>gid=1000(spidy)</code>	Group id
<code>groups=1000(spidy),4(adm),20(dialout),24(cdrom)...</code>	I gruppi presenti su questo account

2. NAVIGARE LE DIRECTORY ED VISUALIZZARE IL LORO CONTENUTO

Analizziamo ora i comandi che ci permettono di spostarci nelle varie directory.

- **PWD:**
Prima di tutto per sapere dove ci troviamo possiamo usare questo comando che mostrerà la working directory. Semplice ma efficace.
- **CD:**
Stand for change directory. Viene usato per spostarsi nella directory che gli indichiamo.
Vediamo alcuni esempi ed alcune particolarità:

ESEMPIO	COMMENTO
cd snap/ubuntu-desktop-installer	Dalla mia directory mi sposto in una delle sottodirectory
cd ..	Mi sposto alla directory precedente
cd ../../	Mi sposto di 2 o più directory in su
cd ../Desktop	Mi sposto alla directory precedente ed entro in Desktop
cd /	Mi sposto alla root directory
cd ~	Mi sposto alla home directory
cd <percorso_assoluto>	Mi sposto nella directory che gli specifico

- **LS:**
Stand for list. Questo comando ci lista i file presenti all'interno di una directory. Ci sono vari parametri che ci permettono di visualizzare i contenuti in modo diverso e con varie caratteristiche. In particolare vediamo:

	COMMENTO
-l	Lista i file ma con un formato esteso andando a specificare l'user, il gruppo, i permessi...
-a	Lista anche i file nascosti
-s	Lista indicando la dimensione di ogni elemento in blocchi
-h	Lista solo file human readable
-R	Lista in modo ricorsivo (quindi anche tutte le sottodirectory della directory corrente)

3. CREAZIONE, MODIFICA ED ELIMINAZIONE DI FILE E DIRECTORY

Vediamo ora come si perfomano varie azioni su di un file o una directory.

- **TOUCH:**

Il comando serve, molto semplicemente, per creare un file

- **MV:**

Stand for move. Questo comando puo' avere due utilizzi: spostare o rinominare un file/directory

1. move : mv <nome_file> <nuovo_nome_file>
2. rename : mv <nome_file> <percorso_dove_spostarlo>

- **CP:**

Stand for copy. Copia un file/cartella in una destinazione precisa. Per poter copiare le cartelle bisogna utilizzare il parametro -r

- **RM:**

Stand for remove. Rimuove un file/cartella in una destinazione precisa. Per poter rimuovere le cartelle bisogna utilizzare il parametro -r. Attenzione a come si usa perche' e' molto potente

- **LN:**

Stand for link. Crea link tra file. Ci sono due principali tipi di link: gli hard links e i symbolic links.

- Hard links: sono puntatori aggiuntivi a un file che puntano allo stesso inode (struttura dati che rappresenta un file) del file originale. Non possono puntare a directory o file su file system diversi. Se il file originale viene eliminato, il link fisico accederà comunque ai contenuti del file poiché puntano agli stessi dati sul disco. I link fisici non possono collegarsi a file o directory inesistenti e non sono influenzati dallo spostamento del file originale all'interno dello stesso file system.
- Soft links: sono semplici puntatori al file di destinazione. Simili ai shortcut in Windows, possono puntare a file su diversi file system. Se il file di destinazione viene eliminato o spostato, il link simbolico sarà interrotto. I link simbolici possono puntare sia a file che a directory e anche collegarsi a file o directory inesistenti.

I flag da utilizzare sono rispettivamente -P per gli hard (physical) e -s (soft/symbolic) per quelli simbolici.

N.B.

Gli SPACE vengono interpretati come parametri di un comando. Ad esempio:

```
cat file di prova
```

Creera' tre file distinti. Per poter creare un file con SPACE al suo interno bisogna includere il nome nelle virgolette:

```
cat "file di prova"
```

Lo stesso vale per i parametri di mv, touch, cp e rm.

Il TAB spesso e volentieri risolve molti di questi problemi poiche' completa automaticamente il nome del file.

Ci sono altri due comandi, che vedremo piu' approfonditamente nel corso intermedio ma che vale la pena accennare:

- **NANO**

Nano e' un editor su terminale. Molto utile e veloce se bisogna apportare qualche modifica (non eccessivamente impegnativa) ad un file.

Allo stesso tempo e' abbastanza primitivo e limitato, quindi non ottimale se si deve programmare qualcosa di complesso.

Con nano si puo' anche direttamente creare un nuovo file e scriverci dentro. Basta inserire il nome di un file che su quella directory non e' gia' presente.

- **ECHO**

Echo di per se stampa a video (stdout) un messaggio che gli diamo come parametro.

La vera utilita' sta nel reindirizzare i messaggi che gli diamo ad un determinato canale di output. Ad esempio un file.

Per ora vediamo queste due opzioni:

echo "messaggio" > path/to/file	Sovrascrive il contenuto con la stringa di input
echo "aggiungi" >> path/to/file	Aggiunge una nuova linea al file

4. USER E SUPERUSER

Quando usiamo il computer per la maggior parte del tempo le azioni che performiamo sono svolte in modalita' user.

Talvolta i comandi hanno bisogno dei permessi di amministratore per poter essere eseguiti. Per poter fare cio' si puo' utilizzare il comando sudo per eseguire quello specifico comando in superuser

- **SUDO:**

```
spidy@DESKTOP-5AAC034:~$ apt update
Reading package lists... Done
E: Could not open lock file /var/lib/apt/lists/lock - open (13: Permission
denied)
E: Unable to lock directory /var/lib/apt/lists/

spidy@DESKTOP-5AAC034:~$ sudo apt update
Hit:1 http://archive.ubuntu.com/ubuntu jammy InRelease
Get:2 http://archive.ubuntu.com/ubuntu jammy-updates InRelease [128 kB]
Get:3 http://archive.ubuntu.com/ubuntu jammy-backports InRelease [127 kB]
Get:4 http://security.ubuntu.com/ubuntu jammy-security InRelease [129 kB]
...
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
13 packages can be upgraded. Run 'apt list --upgradable' to see them.
```

Oppure possiamo cambiare utente e diventare superuser

- **SUDO SU:**

```
spidy@DESKTOP-5AAC034:~$ sudo su
[sudo] password for spidy:
root@DESKTOP-5AAC034:/home/spidy# apt update
...
2 packages can be upgraded. Run 'apt list --upgradable' to see them.
root@DESKTOP-5AAC034:/home/spidy#
```

5. GESTIONE UTENTI E GRUPPI

- **ADDUSER/ADDGROUP:**

Il comando come suggerisce il nome consente di aggiungere un utente/gruppo. Per fare cio' sara' sufficiente digitare:

```
adduser <nome_nuovo_user>
addgroup <nome_nuovo_gruppo>
```

e seguire le procedure indicate sul terminale. Potrebbe essere necessario ricorrere al parametro -B per consentire i "bad names"

- **DELUSER/DELGROUP:**

Per rimuovere un utente/gruppo invece ci serviranno del comando deluser che elimina l'utente/gruppo e volendo aggiungiamo la flag per rimuovere tutti i file relativi a quell'user/gruppo:

```
delgroup <nome_gruppo>
deluser -remove-all-files <nome_user>
```

- **USERMOD:**

Questo comando permette di modificare le proprieta' di un account. I parametri sono molteplici e di seguito ho elencato i piu' diffusi e utilizzati:

PARAMETRO	COMMENTO
-c, --comment COMMENT	Aggiunge un commento all'account dell'user
-d, --home HOME_DIR	Specifichiamo la nuova home directory dell'user
-g, --gid GROUP	Cambia il gruppo dell'user
-G, --groups GROUPS	Aggiunge all'user nuovi gruppi di appartenenza
-s, --shell SHELL	Cambia la shell di login dell'user
-l, --login NEW_LOGIN	Cambia l'username dell'user
-a, --append	Aggiunge gruppi all'user senza rimuovere i gruppi a cui appartiene
-e, --expiredate EXPIRE_DATE	Imposta una data di scadenza dell'account dell'user
-p, --password PASSWORD	Imposta una password criptata all'account
-U, --unlock	Sblocca l'account

6. GESTIONE PERMESSI

Breve descrizione dei permessi e delle caratteristiche di un file:

```
shum@so1:~$ ls -l
total 20
drwx----- 2 shum staff 4096 Jan 16 22:04 Mail
drwx----- 3 shum staff 4096 Jan 16 14:15 csc128
drwxr-xr-x 2 shum staff 4096 Jan 13 16:42 public
drwxr-xr-x 2 shum staff 4096 Jan 16 14:07 public_html
-rw-r--r-- 1 shum staff 628 Jan 15 20:04 verse
```

- **CHMOD:**

Stand for change mode. Questo comando consente di modificare i permessi di un file.

I permessi possono essere modificati principalmente in due modi.

→ Attraverso le lettere

u → user	+ → add	r → read
g → group	- → remove	w → write
o → others	= → set bits to	x → execute

ESEMPIO	COMMENTO
chmod g+x, o+x script.sh	Aggiunto permesso di exec al gruppo e agli altri
chmod u-w, g-w documento.txt	Rimosso permesso di scrittura all'user e al gruppo
chmod u=rwx, g=r,o= data.csv	Permesso di lettura, scrittura ed exec ad user, di lettura al gruppo e nessuno agli altri

→ Attraverso i bit secondo la seguente tabella

drwxrwxrwx

d = Directory
r = Read
w = Write
x = Execute

chmod 777

 ↙ ↓ ↘
rwx | rwx | rwx
Owner | Group | Others

7	rwx	111
6	rw-	110
5	r-x	101
4	r--	100
3	-wx	011
2	-w-	010
1	--x	001
0	---	000

ESEMPIO	COMMENTO
chmod 700 script.sh	Tutti i permessi ad user, nessuno agli altri
chmod 664 documento.txt	Lettura e scrittura ad user e group, solo lettura a others
chmod 711 data.csv	Tutti i permessi ad user, solo di exec agli altri

- **CHOWN:**
Stands for change owner. Permette di modificare gli owner del file nonche' il proprietario e il gruppo di appartenenza.

ESEMPIO	COMMENTO
chown user1 file.txt	Cambio l'owner del file
chown -R user1:group1 file.txt	Cambio l'owner e il gruppo del file in modo ricorsivo
chown :group1 file.txt	Cambio il gruppo del file

7. RICERCA NEL FILESYSTEM E TIPOLOGIE DI FILE

- **FIND:**

Ci consente di ricercare ricorsivamente un file attraverso svariati parametri per poter filtrare i risultati della nostra ricerca e renderli piu' o meno precisi a seconda dell'esigenza che abbiamo.

PARAMETRO	ARGOMENTO	COMMENTO
-type	f : files d : directory	Ricerca sul tipo di file
-size	c : size in bytes k : size in kylobytes M : size in megabyes G : size in gigabytes +n : piu' di -n : meno di n : uguale	Ricerca sulle dimensioni di un file. Viene specificato un numero seguito dalla sua unita' di misura. Aggiungendo + e - possiamo specificare se deve essere piu' o meno grande di
-mtime	giorni +n : piu' di -n : meno di n : uguale	Ricerca sull'ultima modifica del file in giorni espresso in numero.
-user	user name	Ricerca sull'user a cui appartiene
-group	group name	Ricerca sul gruppo a cui appartiene
-perm	permessi /nnn : almeno uno dei perm +nnn : almeno questi perm -nnn : questi perm esatti	Ricerca sui permessi che deve avere
-iname	nome del file	Ricerca sul nome
-maxdepth	numero di livelli	Specifica quanto in profondita' andare
-not or !	un altro parametro dopo	Nega il parametro successivo

E' possibile utilizzare un altro comando molto potente, l'exec.
Questo comando ci consente di eseguire un comando sui file trovati da find.

```
find /path/to/search -exec command {} \;
```

Vediamo ora degli esempi che aiutano molto a capire:

COMANDO	COMMENTO
find ... -exec rm {} \;	Cerca i file e li elimina
find ... -exec cp {} /destination_path \;	Cerca i file e li copia nella destinazione

- **FILE:**

Se invece vogliamo sapere il tipo del file possiamo utilizzare l'omonimo comando. I file che il comando categorizza sono principalmente questi:

TIPO DI FILE	COMMENTO
ASCII text	Testo codificato in ASCII
UTF-8 Unicode text	Testo codificato in UTF-8
ELF executable	File binario per Linux
JPEG image data	Immagine JPEG
PDF document	File di tipo portable document file
ZIP archive data	Archivio compresso ZIP
PNG image data	Immagine PNG
HTML document	File Hypertext Markup Language
Shell script	Script scritto in linguaggio shell scripting
GZIP compressed data	File compresso con GZIP

COMANDI E STRUMENTI UTILI

- **HISTORY**

Ci permette di vedere i comandi utilizzati fino ad ora in ordine cronologico e di riutilizzarli inserendo il loro codice.

Ad esempio:

```
...
580 rm car*
581 rm -r car*
582 ls -l
583 exit
584 man echo
585 history
spidy@DESKTOP-5AAC034:~$ !582
ls -l
total 4
drwx----- 3 spidy spidy 4096 Jun  5 18:25 snap
spidy@DESKTOP-5AAC034:~$
```

- **DU E DF**

Sono due comandi che ci permettono rispettivamente di conoscere lo spazio occupato da un file e quello libero sul disco

```
spidy@DESKTOP-5AAC034:~$ df -h
Filesystem      Size  Used Avail Use% Mounted on
/dev/sda1       20G   8.5G   9.8G   47% /
/dev/sdb1       50G   20G   28G   42% /home

spidy@DESKTOP-5AAC034:~$ du -sh /var/log
2.5G    /var/log
```

Alcuni flag utili per du sono:

	COMMENTO
-h	La grandezza dei file viene indicata in Bytes
-s	Viene fatto un riassunto cioe' non vengono elencate le grandezze di ogni sottodirectory
-a	Vengono elencate le grandezze di tutti i file

- **I CARATTERI * E ?**

Noti come caratteri jolly, sono utilissimi quando non sappiamo esattamente cosa cercare o conosciamo solo una parte o il numero di caratteri di qualsiasi cosa.

COMANDO	COMMENTO
ls *.txt	Mostra solo i file con estensione .txt
rm log?.log	Elimina solo i file del tipo "log?.log" dove ? puo' essere qualsiasi carattere
grep "pattern" /etc/*/file*.txt	Prende solo i file che contengono la parola "pattern" solo dei file che iniziano con "file" e che abbiano estensione .txt all'interno di qualsiasi sottodirectory di etc/
cp image?.jpg /path/to/destination	Copia tutti i file "image?.jpg", dove il ? puo' essere qualsiasi carattere, in una specifica directory
ls /path/to/directory/file?.txt	Mostra solo i file con questo formato "file?.txt" all'interno di una directory specifica