

Guida base Linux

Cos'è Linux

Linux è un sistema operativo creato da Linus Torvalds. Ha un discreto seguito come sistema operativo per pc domestici ed ha ampia diffusione come sistema operativo per i Server. Differentemente da Windows è open source, vuol dire che il codice che lo implementa è visibile e ciò lo rende modificabile da chi ne abbia le competenze tecniche. Grazie a ciò sono nate diverse distribuzioni di Linux, ciascuna derivata dal core base di Linux, tra le più diffuse: Ubuntu, Centos, Red Hat ecc.

L'accesso al sistema operativo, da parte di un utente, avviene con l'operazione di log in che prevede l'inserimento di una username e di una password. Il sistema operativo Linux ha un utente predefinito, root, che è l'utente principale, detto anche superutente: ha i permessi più ampi tali da consentirgli di amministrare tutto. E' possibile aggiungere e eliminare utenti.

Il kernel

Il kernel (nucleo) è il centro del sistema operativo, ovvero la parte che contiene l'implementazione delle funzionalità basilari che consentono la gestione delle risorse hardware.

Il filesystem

Il filesystem è la parte del sistema operativo che gestisce il salvataggio del file e il recupero del file quando viene richiesto. Più genericamente, quando ci si riferisce al filesystem di un computer, si intende la struttura logica ad albero, organizzata in cartelle, sottocartelle e file che rappresenta la posizione di file e cartelle all'interno dello spazio di archiviazione del dispositivo.

I comandi

L'utente può inviare comandi attraverso l'interfaccia utente, cioè l'interfaccia grafica che può essere presente nei sistemi operativi Linux, simile all'interfaccia grafica dei sistemi Windows, tuttavia essa non è sempre presente soprattutto nelle installazioni Linux in ambito Server.

Assume quindi una particolare valenza la capacità di saper eseguire comandi inviandoli attraverso la cosiddetta riga di comando. L'uso della riga di comando, seppure più complesso dell'uso dell'interfaccia grafica, conferirà all'utente funzionalità e potenzialità non disponibili attraverso l'interfaccia utente.

Di seguito tratteremo i principali comandi che è possibile eseguire tramite riga di comando.

Nei sistemi operativi non dotati di interfaccia grafica, dopo l'avvio, il sistema operativo mostra all'utente il prompt dei comandi, ovvero l'interfaccia che richiede l'inserimento di dati da tastiera.

Nei sistemi operativi Linux dotati di interfaccia grafica, per poter visualizzare il prompt dei

comandi occorre avviare l'applicazione “Terminale” .

Il prompt dei comandi richiederà l'inserimento di username e, successivamente, password.

Una volta effettuato il processo di login, l'utente sarà condotto all'interno della cartella a lui assegnata come home directory.

A questo punto l'utente potrà digitare i comandi voluti.

La sintassi generale di un comando Linux è la seguente:

parola chiave -opzioni argomenti nomefile

La parola chiave è lo specifico comando che si vuole utilizzare tra quelli previsti da Linux. Le opzioni sono istruzioni che modificano il comportamento del comando. Ogni comando supporta un ben preciso set di opzioni.

Le opzioni possono essere combinate tra di loro accodando le lettere delle opzioni che si vogliono utilizzare per uno specifico comando. Ad esempio `ls -la` utilizza l'opzione `l` e l'opzione `a` del comando `ls`.

L'opzione `-help` consente di accedere alla guida del comando, di conseguenza:

`ls -help` mostrerà la guida del comando `ls`

Gli argomenti sono parametri, dati di ingresso di cui alcuni comandi hanno bisogno (ad esempio per creare una cartella va specificato il nome della cartella da creare: il nome della cartella da creare è quindi l'argomento del comando che crea la cartella).

Nomefile è il nome del file su cui opererà il comando.

Cd

Per cambiare cartella si utilizza il comando `cd`

`Cd utente` consente di accedere alla cartella utente (se è una sottocartella della cartella corrente) altrimenti si ottiene un errore

`cd ..` consente di accedere alla cartella padre della cartella corrente, ovvero salire al livello superiore del file system.

Mkdir

Consente di creare una cartella. Il comando `mkdir` deve essere seguito dal nome della cartella che si vuole creare, esempio:

`mkdir giovanni` crea la cartella `giovanni` all'interno della cartella corrente

ls

`ls -la`

Consente di visualizzare files e cartelle contenute dentro la cartella corrente. Utilizzando l'opzione `-l` verranno visualizzate delle informazioni aggiuntive sui file e sulle cartelle.

Utilizzando l'opzione `-a` la lista comprenderà anche file e cartelle nascoste (file e cartelle nascoste hanno il nome che inizia con il carattere `."`).

cp

Il comando `cp` copia un file creandone una copia con il nome specificato

```
cp nomefile nomecopia
```

Il secondo parametro può anche essere il path (percorso) di una cartella, in questo caso la copia verrà creata all'interno di questa cartella e il nome del file sarà uguale a quello originario. Un esempio della seconda possibilità:

```
cp prova.txt /usr/lavoro
```

creerà una copia del file `prova.txt` dentro la cartella `lavoro`

Mv

Il comando `mv` (move) consente di rinominare un file o spostarlo in un'altra cartella. Il suo funzionamento è simile a `cp`, con la differenza che il file originario non esisterà più.

```
mv prova.txt documento.txt
```

 rinomina il file `prova.txt` in `documento.txt`

```
mv prova.txt /usr/lavoro
```

 sposta il file `prova.txt` dentro la cartella `lavoro`

Rm

Il comando `rm` elimina un file.

```
rm riunione.pdf
```

 elimina il file `riunione.pdf`

Utilizzando l'opzione `-r` è possibile rimuovere tutti i file all'interno di una cartella e la cartella stessa:

```
rm -r /usr/lavoro
```

 rimuove tutti i file e le sottocartelle dentro la cartella `lavoro` e la cartella `lavoro` stessa.

Chmod

Il comando `chmod` consente di assegnare i permessi a files o cartelle.

```
chmod 777 relazione.pdf
```

 assegna i permessi al file `relazione.pdf`. L'argomento verrà trattato in maniera più dettagliata più avanti

Chown

Il comando `chown` cambia il proprietario di un file. Ogni file ha un proprietario, che di norma è l'utente che ha creato il file. In qualsiasi momento è possibile cambiare il proprietario del file usando il comando `chown`.

```
chown giovanni:docenti verifica.xls
```

 assegna il proprietario `giovanni` e il gruppo `docenti` al file `verifica.xls`

I permessi su file e cartelle

Linux è basato su permessi che possono essere concessi a files e cartelle. Questi permessi riguardano la possibilità di leggere, scrivere ed eseguire file e cartelle. I permessi sono divisi su tre livelli: permessi del proprietario, permessi del gruppo e permessi per tutti gli utenti.

A titolo di esempio, potremmo voler assegnare al proprietario del file `newsletter.php` i permessi di lettura scrittura e esecuzione, al gruppo i permessi di lettura e esecuzione e a tutti gli altri utenti solo il permesso di lettura.

Con il comando `ls -l` si ottiene la lista dei file e delle cartelle presenti nella cartella corrente e, per ciascun file e cartella, i permessi che sono stati settati.

I permessi appariranno formattati nel seguente modo:

```
-rwxrw-r--
```

Si tratta di 10 caratteri, il primo dei quali specifica se la risorsa è una cartella (d) o un file (-). I successivi 9 caratteri vanno considerati a gruppi di tre. La prima terna si riferisce ai permessi concessi al proprietario, la seconda al gruppo, la terza a tutti gli altri utenti. Se un permesso è concesso, allora è presente la lettera corrispondente (r: lettura, w: scrittura, x:esecuzione), mentre se il permesso è negato, è presente il carattere -.

In fase di assegnazione dei permessi tramite il comando `chmod`, si fa riferimento a una notazione numerica per l'attribuzione dei permessi. Ad esempio:

```
chmod 761 controller.php
```

assegna i permessi "761" al file `controller.php`, bisogna quindi capire cosa si intende con 761 alla luce delle notazioni sui permessi che sono state enunciate precedentemente.

Il numero 761 va considerato una cifra per volta, la prima cifra rappresenta la prima terna dei permessi, quella che si riferisce al proprietario, la seconda rappresenta la terna per il gruppo, e la terza i permessi per tutti gli utenti. E' necessario ora definire come ogni cifra possa rappresentare la configurazione dei permessi per ogni livello. Ciò avviene assegnando un'etichetta numerica ad ogni permesso, ed in particolare:

permesso	valore
r	4
w	2
x	1

Il valore della cifra è dato dalla somma dei valori dei permessi che si intende attribuire a quel livello, pertanto: 7 corrisponde ad assegnare i permessi di lettura, scrittura e esecuzione, 6 può essere ottenuto solo come sommo di 4 e 2, pertanto corrisponde ad assegnare i permessi di lettura e scrittura, 1 corrisponde a concedere il solo permesso di esecuzione.

Come si può intuire, non si possono mai generare condizioni di ambiguità interpretativa circa i permessi assegnati.

Gli utenti

Ogni utente è individuato, nel sistema operativo, da un identificativo, l'uid, ovvero l'user id, e da uno gid, ovvero l'identificativo del gruppo. Ogni utente appartiene perciò a un gruppo. Il sistema operativo Linux ha un utente predefinito, root, che è l'utente principale, detto anche superutente, quello che ha i permessi più ampi tali da consentirgli di amministrare tutto il resto.

Sudo

Sudo consente di eseguire comandi con qualsiasi utente spacciandosi per l'utente root. Ciò è particolarmente importante perchè alcuni comandi possono essere eseguiti soltanto dall'utente root e perchè alcune distribuzioni Linux non consentono di effettuare il login come utente root.

Sudo va anteposto al comando e richiederà l'inserimento della password di root.

Aggiungere un utente

`sudo adduser nomeutente nomegruppo` aggiunge un utente con nome nomeutente e gruppo di appartenenza nomegruppo

Eliminare un utente

`userdel nomeutente` elimina l'utente con nome nomeutente

Aggiungere un gruppo

`addgroup nomegruppo` crea il gruppo con nome nomegruppo

Standard input, standard output, standard error. Ridirezione di input e output

Linux, normalmente, visualizza i risultati dei comandi sul monitor, il monitor svolge quindi il ruolo di standard output, mentre i dati vengono normalmente acquisiti dalla tastiera, la tastiera svolge quindi il ruolo di standard input. I possibili errori che possono verificarsi all'esecuzione delle istruzioni vengono inviati al monitor: il monitor è lo standard error.

L'operazione che consente di modificare questi standard è definita ridirezione. Con la ridirezione dell'input si intende la possibilità di inviare dati a un comando non digitandoli dalla tastiera (che è lo standard input) ma fornendoli da un'altra fonte, come ad esempio, un file.

Ad esempio `ls > elenco` invierà il risultato del comando ls non al monitor ma al file con nome elenco (il cui contenuto verrebbe sostituito con il risultato del comando). Se il file non esistesse verrebbe creato.

`ls 2> elenco` invia al file elenco gli eventuali errori generatesi nell'esecuzione del comando ls: si tratta di una ridirezione dello standard error.

`mail < testo` invia al comando mail (comando che consente l'invio di una email) il contenuto del file testo, si tratta quindi di un esempio di ridirezione dell'input

I comandi sui file

Linux dispone di diversi comandi che consentono di cercare un file o effettuare delle operazioni sul contenuto dei file.

find

Consente di cercare un file all'interno del file system di Linux

`find docente -name esercizio.xls` cerca il file con nome esercizio.xls all'interno della cartella docente (se docente è una sottocartella della cartella corrente)

Il nome del file può contenere i caratteri jolly * e ?: * ha il significato di qualsiasi sequenza di qualsiasi carattere, mentre ? Qualsiasi carattere una sola volta.

Ad esempio:

`find docente -name *esercizio.xls` cercherà all'interno della cartella docente i file il cui nome termina con esercizio.xls

grep

Il comando grep cerca all'interno di un file restituendo le righe che contengono una certa parola (o un'espressione regolare)

`grep giovanni nomi.txt` mostrerà sul monitor le righe del file nomi.txt che contengono la parola giovanni

tail

Il comando tail mostra le ultime righe di un file

`tail -15 elenco.txt` mostra le ultime 15 righe del file elenco.txt

Le pipeline

Le pipeline sono uno strumento che consente di utilizzare il risultato di un comando come dato di ingresso per un successivo comando. Ogni comando viene separato dal successivo utilizzando il carattere |.

`ls -la | grep elenco` Il risultato del comando ls -la viene inviato al comando grep elenco, costituendo quindi il dato su cui quest'ultimo opererà.

I processi

Un processo rappresenta l'istanza, ovvero la concretizzazione di un programma o di un comando che è stato lanciato e che sarà eseguito dal processore fino a che non sarà completato.

In Linux alcuni processi sono lanciati direttamente dal sistema operativo, altri vengono lanciati dagli utenti.

Per conoscere la lista dei processi attivi si utilizza il comando `ps` con le opzioni `aux`, tale comando visualizzerà anche il pid (process id) ovvero il numero identificativo del processo.

`ps aux` visualizza i processi in corso di tutti gli utenti con indicazione della data di inizio del processo e del tempo di esecuzione.

E' possibile interrompere anticipatamente l'esecuzione di un processo usando il comando `kill`

`kill 6530` termina il processo con pid 6530

Il comando `kill` prevede l'uso dell'opzione `-SIGKILL` che consente di arrestare anche i processi che ignorano il comando `kill` se sprovvisto dell'opzione `-SIGKILL`