

Dispense Corso Intermedio

GNU/Linux



lucalug
Lucca linux user group

30 maggio 2008

<http://www.lucalug.it>

La shell

In un sistema operativo, la shell è il programma che permette agli utenti di comunicare con il sistema e di avviare i programmi. È una delle componenti principali di un sistema operativo, insieme al kernel.

La shell è l'"ambiente di lavoro" attraverso il quale è possibile impartire al computer comandi, richiedendo l'esecuzione di programmi.

Esistono molti tipi di shell, che si dividono principalmente in testuali e grafiche. Quando si parla semplicemente di "shell", si intende di solito una shell testuale.

Shell testuali

Una shell testuale è un programma con interfaccia a linea di comando, che viene eseguito all'interno di un terminale testuale. L'utente digita un comando, ovvero richiede l'esecuzione di un programma, e il programma eseguito può interagire con l'utente e/o stampare dati sul terminale.

Una delle più note shell testuali è il tradizionale 'prompt dei comandi', ben noto a quanti hanno familiarità con i sistemi operativi DOS (MS-DOS, DR-DOS, FreeDOS). Per gli utenti del vecchio MS-DOS la shell era il programma 'command.com'. Anche le recenti versioni di Windows hanno una shell, il programma 'cmd.exe'.

In ambiente Unix, esistono diverse shell, una tra le più famose è sicuramente la 'Bash' (Bourne-Again Shell), ma ne esistono altre come la Korn shell e la C shell, con un insieme di funzionalità e caratteristiche di base in comune.

La shell testuale è un ambiente di lavoro che funziona 'a riga di comando', con istruzioni testuali che vengono impartite attraverso la tastiera per avviare programmi, eseguire comandi, visualizzare il file system, interagire in ogni modo con il computer. Un programma viene invocato digitando il suo nome, eventualmente seguito da altri dati, che vengono passati come argomenti al programma. Completata una riga di comando, si preme invio per richiederne l'esecuzione.

Sono inoltre presenti potenti strumenti per collegare tra loro diversi programmi per svolgere compiti complessi, come le pipe e la redirectione. I programmi unix più propensi ad essere collegati in questo modo sono detti filtri.

Le moderne shell per Unix posseggono funzionalità ergonomiche molto ricche, come la "command history", ovvero la possibilità di ripetere gli ultimi comandi digitati, la "completion", ovvero la possibilità di completare automaticamente nomi di programmi e di file, il "job control", che permette di gestire diversi programmi contemporaneamente. Queste funzionalità, una volta apprese, contribuiscono in modo determinante a rendere le shell uno strumento di lavoro altamente produttivo.

Le shell testuali per Unix sono molto apprezzate dagli sviluppatori perché hanno integrato anche un linguaggio di scripting molto potente con il quale è possibile scrivere veri e propri programmi (script) o delle procedure che agevolano moltissimo l'amministrazione di sistema. La sintassi di tale linguaggio è un'estensione di quella usata interattivamente, per cui chi è familiare con l'uso interattivo della shell trova facile e naturale creare degli script.

La shell su ubuntu

Per prima cosa in Ubuntu potete andare su *Applicazioni > Accessori > Terminale*; invece in ambiente Kde potete aprire *Konsole*. In realtà, qualunque programma abbiate aperto, che vi presenti qualcosa del genere:

```
utente@computer:~$
```

è un terminale. Anzi, più precisamente si tratta di un programma (a volte detto emulatore di terminale) che vi permette di avere in una finestra il computer senza interfaccia grafica, ovvero di interagire direttamente con esso attraverso i **comandi**.

Analizziamo la scrittura che vi appare all'inizio. *Utente* rappresenta il nome dell'utente che state utilizzando in questo momento, *computer* rappresenta il nome unico con cui il computer viene visto in rete. La tilde (~) vi dice che siete nella vostra home directory (cioè */home/utente/*) e il simbolo del dollaro vi dice che siete un utente normale (e non con i privilegi di amministratore).

Perché usarlo?

In realtà, le moderne distribuzioni Linux (soprattutto le più semplici come Ubuntu) forniscono una buona interfaccia grafica che permette di evitare praticamente quasi tutte le manovre da terminale. Infatti si può fare quasi tutto con un interfaccia grafica.

Tuttavia, in rete si trovano spesso dei comandi da impartire da terminale. Se li si sa interpretare si possono anche sostituire con azioni effettuate dall'interfaccia grafica, ma è molto più comodo copiarli ed eseguirli, inoltre è più rapido.

Non è detto che esista un interfaccia grafica per ogni operazione possibile sul proprio pc. Mentre TUTTI i programmi disponibili possono essere avviati dalla shell.

I Comandi Unix

Per comando si deve intendere generalmente un programma sia del kernel sia esterno ad esso, che la shell manda in esecuzione in risposta ad istruzioni digitate da tastiera. Per sintassi di un comando si deve intendere il modo corretto di digitare il comando stesso con la tastiera, affinché sia compreso dalla shell. La sintassi tipica dei comandi UNIX è la seguente :

nome del comando [opzioni] [argomenti]

Gli argomenti sono gli oggetti su cui il comando esegue le proprie azioni (per es., file e directory); le opzioni sono dei parametri che consentono variazioni all'azione compiuta dal comando, e vanno precedute dal segno - (per es. -p, -c, -d ...). Non tutti i comandi hanno una tale struttura, potendo mancare delle opzioni, o degli argomenti, o di entrambi .

Per conoscere le opzioni d'uso e ottenere aiuto su un comando si possono utilizzare le seguenti strutture:

man [comando] – mostra il manuale del comando (per uscire premere Q)

[comando] --help – mostra le opzioni disponibili per il comando

Terminal Control Keys

Esistono diverse combinazioni di tasti che implementano speciali funzioni sul terminale. Questi tasti di controllo sono ottenuti tenendo premuto il tasto CTRL mentre si digita il tasto successivo.

Per esempio, CTRL-c significa tenere premuto il tasto CTRL mentre si digita la lettera c. I tasti di controllo più usati sono i seguenti :

CTRL-u cancella tutto quello che si è digitato sulla linea di comando

CTRL-c interrompe bruscamente l' esecuzione di un comando

CTRL-h torna indietro di un carattere come il tasto backspace

CTRL-z sospende un comando

CTRL-s interrompe lo scorrimento (scrolling) dello schermo

CTRL-q continua lo scrolling

Reindirizzamento

E' possibile reindirizzare l'output di un comando in un file.

Per fare cio' basta scrivere in fondo alla linea del comando il simbolo di maggiore (>) seguito dal nome del file che verra' creato (o sovrascritto). Invece per aggiungere l'output di un comando in fondo ad un file gia' esistente basta utilizzare due simboli di maggiore (>>) al posto di uno singolo.

Esempi:

comando > nomefile

comando >> nomefile

Comandi principali:

Sotto è riportata una lista con i comandi più comuni:

File e Directory

tar - crea archivi, e aggiunge o estrae file

cat - mostra su video e concatena

cd - cambia la directory corrente

chgrp - cambia l'appartenenza a un gruppo di un file

chmod - cambia i permessi di un file

chown - cambia il proprietario di un file

cmp - confronta byte a byte due file

comm - seleziona o annulla linee comuni a due file ordinati

cp - copia uno o più file

cpio - copia archivi di file in e out

cut - trascrive byte, caratteri, o campi selezionati da ciascuna linea di un file

df - riporta lo spazio libero sul disco del file system

diff - mostra le differenze linea per linea fra coppie di file di testo

du - mostra il numero di blocchi di disco usati per un file o una directory

file - determina il tipo di file esaminandone il contenuto

find - trova file per nome o per altre caratteristiche

grep - ricerca un file attraverso una stringa in esso contenuta

head - mostra alcune linee iniziali del file specificato

ln - crea un symlink

ls - mostra il contenuto di una directory

mkdir - crea una directory

more - scorre il contenuto di un file su video

mv - sposta o rinomina un file

page - scorre le pagine di un file di testo

paste - unisce linee da uno o più file

pr - visualizza uno o più file

pwd - mostra il pathname della directory corrente

rccp - remote file copy

rev - inverte l'ordine dei caratteri in ciascuna riga

rm - cancella un file

rmdir - rimuove una directory

size - mostra la dimensione di un file oggetto

sort - ordina linee di un file

split - divide un file in sottofile

strings - trova le stringhe stampabili in un file binario o oggetto

tail - mostra l'ultima parte di un file

tar - crea archivi tape, e aggiunge o estrae file

tee - replica lo standard output

tftp - trivial file transfer program

touch - aggiorna la data di accesso e di modifica di un file

unexpand - riconverte caratteri TAB da caratteri SPACE (vedi expand)

wc - mostra il numero di linee, parole e caratteri

Informazioni

apropos - trova comandi mediante ricerca di una chiave

hostname - stampa il nome dello host corrente

rup - mostra lo stato di host della macchina locale (versione RPC)

ruptime - mostra lo stato di host della macchine locali

uptime - mostra da quanto tempo è attivo il sistema

whatis - descrive la funzione di un comando

whereis - localizza sorgenti, file binari, o manuali per un programma

which - localizza un programma, includendo aliases e paths

Filesystem

df - mostra lo spazio libero sul disco fisso

free - mostra lo stato della memoria

mount - monta un filesystem [è previsto un articolo dedicato, ndr]

umount - disattiva un filesystem

ps - visualizza un elenco dei processi correnti

kill - invia un messaggio [TERM di default] ad un processo

Spegnimento

halt - chiude il sistema [per lo spegnimento è suggerito il comando shutdown -h now, ndr]

reboot - riavvia il sistema

Fonti:

Da completare.