Amministrare GNU/Linux

Seconda serata 22.04.2014
rev. 2

Shell: interprete dei comandi

Ciò che permette a un utente di interagire con un sistema operativo è la shell, che si occupa di interpretare ed eseguire i comandi dati dall'utente.
Dal punto di vista pratico, il funzionamento di un sistema Unix dipende molto dalla shell utilizzata, di conseguenza, la scelta della shell è molto importante. La shell standard del sistema GNU/Linux è Bash (il programma **`bash`**).
Una shell Unix normale svolge i compiti seguenti:

- mostra l'invito, o *prompt*, all'inserimento dei comandi;
- interpreta la riga di comando data dall'utente;
- esegue delle sostituzioni, in base ai caratteri jolly e alle variabili di ambiente
- mette a disposizione alcuni comandi interni;
- mette in esecuzione i programmi;
- gestisce la ridirezione dell'input e dell'output;
- è in grado di interpretare ed eseguire dei file script di shell.

Ogni comando inviato genera una system call (chiamata di sistema) che viene eseguita dal kernel, l'unico programma che ha i privilegi per controllare l'hardware.

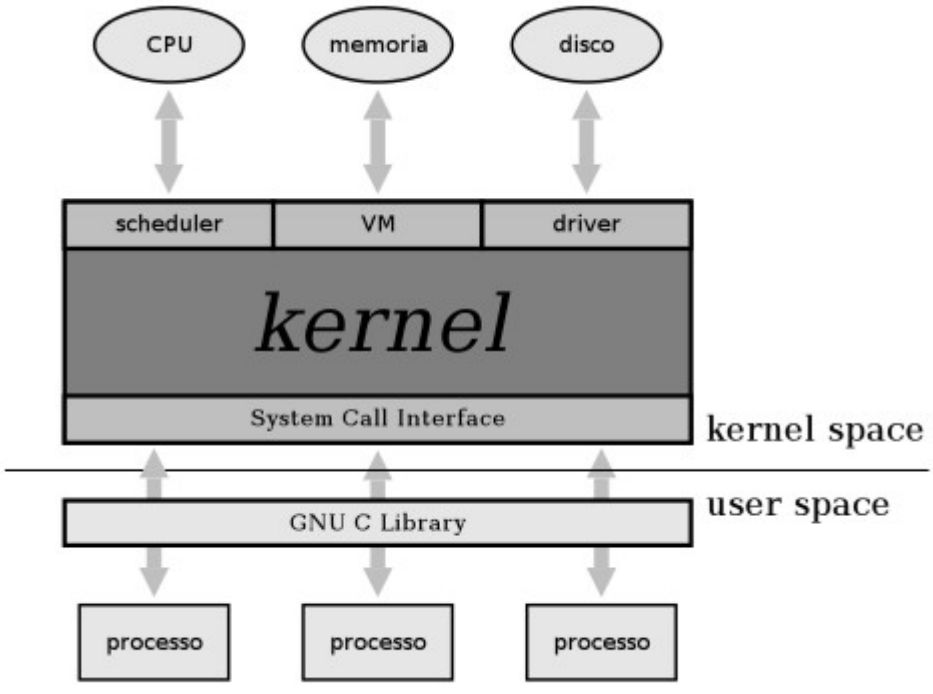

Figura 1.1: Schema della struttura del sistema operativo GNU/Linux.

Prompt
La riga comandi e' una riga digitabile che comincia dal prompt. Il prompt è uno o più caratteri che

possono essere personalizzati dall'utente modificando il file .bashrc (Debian). Il prompt dell'utente root (il superuser) inizia con il carattere '#' (cancelletto), mentre il prompt di un utente qualsiasi inizia con il carattere '$' (dollaro). Ecco un esempio di prompt:

stefano@stefano:~$

I comandi sono dei programmi che sono contenuti all'interno di alcune directory speciali come le directory /bin, /usr/bin, /sbin, /usr/sbin ecc. Quando l'utente digita il nome del comando/programma e preme il tasto invio, la shell cerca tale file in alcune directory predefinite (come /bin e /sbin appunto) e se lo trova lo esegue. In caso contrario produce un errore del tipo:

bash: nome-comando : command not found

cioe' comando non trovato. Un comando e' composto dal nome del comando stesso, da una o piu' opzioni facoltative e da uno o piu' argomenti facoltativi. Un argomento e' solitamente il nome di un file sul quale il comando deve agire. Alcuni comandi prevedono degli argomenti obbligatori, altri no:

nome-comando -opzione argomento

il nome del comando deve essere separato dalle opzioni da uno spazio, cosi' come le opzioni e gli eventuali argomenti. Le opzioni sono specificate usando un trattino '-' e la lettera o le lettere che specificano le opzioni. L'ordine e':
1) **nome-comando**, spazio, 2) **-opzioni**, spazio, 3) **argomenti**.
Tra il trattino ed i caratteri di opzione non devono essere presenti spazi. Ad esempio il comando ls consente l'uso di alcune opzioni come -l (lista estesa) a (tutti i file) i (visualizza inode) e via dicendo. Percio' per visualizzare l'elenco dei file di una directory si puo' usare il comando 'ls -lai'. Se si vogliono visualizzare le informazioni di un file ben preciso, occorre specificare il nome del file come argomento. Ad esempio con il comando 'ls -l pippo' vengono visualizzate le informazioni sul file pippo.

$ ls - l
$ ls -ail
$ ls -lia
$ ls -l -a -i
$ ls --all
$ ls --all -li

Alcuni comandi accettano opzioni senza il trattino ed esempio:

$ ps aux

Un esempio di comando che puo' accettare piu' argomenti e' il comando cp (CoPy, cioe' copia) che consente di effettuare la copia di un file in un altro file con un nome diverso.

$ cp documento /home/stefano/Salvataggi/documento.bak

Ma anche ls:

$ ls /home/ /home/stefano/Documenti/

Il metacarattere '\' può essere usato per scrivere testo su più righe facendo in modo che la shell ignori la pressione del tasto 'invio' come segnale di 'esegui questo comando'.
Es:
prova a digitare:

Ciao, come stai \ *(invio)*
io sto bene \ *(invio)*
e tu \ *(invio)*

Tipi di variabile locali

intere	var=1234
	var=miavariabile
stringa	var="Ciao mondo"

Visualizzare variabile

	echo $var
Esportare variabile	
	var=miavariabile
	export $var
oppure	
	export var=miavariabile

---

posizionale

1-creare uno script ed incollare all'interno:

```
#!/bin/bash
#dati anagrafici

echo "il nome dello script è $0"

nome=$1
cognome=$2
eta=$3

echo "il tuo nome è $1"
echo "il tuo cognome è $2"
echo "la tua età è $3"

echo "Hai passato $# parametri/o"

exit 0
```

2-salvare con nome **dati_anagrafici.sh** cambiare i permessi con **chmod +x dati_anagrafici.sh**
3-lanciare con:
./dati_anagrafici.sh Paolo Rossi 28

---

aritmetiche

```
let var=10+20
echo $var
let var1=22/2
let var=var-var1
```

---

ambiente

```
env             #visualizza variabili ambiente
printenv        #visualizza variabili ambiente
.bashrc         # rendere predefinite per la sessione
/etc/profile    # rendere predefinite per l'utente
/etc/environment        # rendere predefinite per l'utente
```

---

array

```
rapaci=(aquila falco allocco gheppio)
echo $rapaci
echo ${rapaci[0]}
echo ${rapaci[1]}
echo ${rapaci[2]}
echo ${rapaci[3]}
echo ${rapaci[*]}
```

Inserire spazi nell'argomento

```
var=nome\ file
```

Visualizzare numero caratteri variabile

```
var=tre
echo ${#var}
```

Alias

alias è un comando, che abilita l'uso di una parola al posto di un'altra stringa. Esso viene spesso utilizzato per abbreviare un comando, o per aggiungere opzioni o argomenti ad un comando utilizzato regolarmente.
Gli alias personalizzati sono preferibilmente contenuti nel file `~/.bashrc`, mentre gli alias di sistema (che verranno utilizzati da tutti gli utenti) devono essere inseriti in `/etc/bash.bashrc`.
Un esempio da un estratto di un file `~/.bashrc` contenente diversi utili alias (dal sito di ArchLinux):

`~/.bashrc`

```
# comandi modificati
alias diff='colordiff'              # richiede il pacchetto colordiff
alias grep='grep --color=auto'
```

```
alias more='less'
alias df='df -h'
alias du='du -c -h'
alias mkdir='mkdir -p -v'
alias nano='nano -w'
alias ping='ping -c 5'
alias ..='cd ..'

# nuovi comandi
alias da='date "+%A, %B %d, %Y [%T]"'
alias du1='du --max-depth=1'
alias hist='history | grep $1'      # richiede un argomento
alias openports='ss --all --numeric --processes --ipv4 --ipv6'
alias pg='ps -Af | grep $1'         # richiede un argomento (nota: /usr/bin/pg
viene installato dal pacchetto util-linux package; forse sarebbe meglio usare
un altro alias)

# elevazione dei privilegi
if [ $UID -ne 0 ]; then
    alias sudo='sudo '
    alias scat='sudo cat'
    alias svim='sudo vim'
    alias root='sudo su'
    alias reboot='sudo reboot'
    alias halt='sudo halt'
    alias update='sudo pacman -Su'
    alias netcfg='sudo netcfg2'
fi

# ls
alias ls='ls -hF --color=auto'
alias lr='ls -R'                    # ls ricorsivo
alias ll='ls -l'
alias la='ll -A'
alias lx='ll -BX'                   # ordinati per estensione
alias lz='ll -rS'                   # ordinati per dimensione
alias lt='ll -rt'                   # ordinati per data
alias lm='la | more'

# misure di sicurezza
alias cp='cp -i'
alias mv='mv -i'
alias rm='rm -I'                    # 'rm -i' chiede conferma per
l'eliminazione di ogni file
alias ln='ln -i'
alias chown='chown --preserve-root'
alias chmod='chmod --preserve-root'
alias chgrp='chgrp --preserve-root'
```

STRINGHE DI PROMPT (da man bash)

Quando eseguita in modalità interattiva, bash mostra il prompt primario quando è pronta per leggere un comando e il prompt secondario PS2 quando è necessario ulteriore input per completare il comando.
Bash permette di personalizzare queste stringhe di prompt inserendo vari caratteri di escape speciali che vengono interpretati come segue:

- \a il carattere ASCII beep (07)
- \d la data nel formato "Giorno-della-settimana Mese Data" (e.g., "Tue May 26")
- \e un carattere di escape ASCII (033)
- \h l'hostname fino al primo `.'
- \H l'hostname
- \n il carattere "newline"
- \r il carattere "carriage return"
- \s il nome della shell, il nome base di $0 (la parte che segue lo slash finale)
- \t l'ora corrente nel formato 24-ore HH:MM:SS
- \T l'ora corrente nel formato 12-ore HH:MM:SS
- \@ l'ora corrente nel formato 12-ore am/pm
- \u lo username dell'utente corrente
- \v la versione di bash
- \V la release di bash, versione + patchlevel
- \w la directory di lavoro corrente
- \W il nome di base della directory di lavoro corrente
- \! il numero cronologico (history number) di questo comando
- \# il numero di questo comando
- \$ se l'UID effettivo è 0, un #, altrimenti un $
- \nnn il carattere corrispondente al numero ottale nnn
- \\ un backslash
- \[comuncia una sequenza di caratteri non stampabili, che potrebbero essere usati per inserire una sequenza di controllo del terminale nel prompt
- \] termina la sequenza di caratteri non stampabili

Possiamo cambiare il prompt sostituendo la variabile di sistema $PS1 facendone prima un backup con:
PS1BAK=$PS1
poi impostando a piacere:
$PS1='\t \d @\h:$'

Decommentando la riga:
force_color_prompt=yes
possiamo sfruttare le impostazioni del prompt colorato.

Per rendere predefinita la modifica del prompt dobbiamo modificare file .bashrc

Per utilizzare i colori nel prompt:

‘\e[x;ym $PS1 \e[m’

- `\e[` : Inizio schema colore
- `x;y` : Coppia Colore (x;y)
- `$PS1` :la variabile del prompt
- `\e[m` : fine schema colore

con i seguenti schemi colore:

nero (0;30)	Rosso (0;31)	Verde (0;32)	Marrone (0;33)
Blu (0:34)	Viola (0;35)	Ciano (0:36)	

Usando l’1 al posto dello zero si hanno delle versioni cromatiche più leggere.
Ecco un esempio:

PS1="\e[0;35m{`uname -r`}\$ \e[m "

Per personalizzare una TTY possiamo avvalerci dei programmi 'setterm', 'setfont /usr/share/consolefonts/.....' , aggiungere il mouse installando il pacchetto 'gpm',

---

Registrare una sessione di bash

Da terminale avviare:

$ script
$ cd /usr/bin
$ ls -l

Per rivedere i comandi impartiti apriamo il file typescript nella nostra home:

$ cat typescript

ora per vedere i tempi di input:

$ cat typescript | less

Per poter rivedere tutta la sessione nella shell avviare script con l'opzione -t e la ridirezione degli error all'interno del file timefile:

$ script miofile 2> timefile

$ cd /usr/bin
$ ls -l

Ora per rivedere:

scriptreplay timefile miofile

setleds

Il programma **`setleds`** interviene esclusivamente su una console virtuale attiva.
Modalità

```
+num | -num
```

Attiva o disattiva [*BlocNum*].

```
+caps | -caps
```

Attiva o disattiva [*Fissamaiuscole*].

```
+scroll | -scroll
```

Attiva o disattiva [*BlocScorr*].

Per rendere predefinita l'impostazione aggiungere a /etc/profile come root

---

TTY

`stty` permette di modificare le caratteristiche della connessione del terminale al sistema.

```
$ stty -a
```

esempio di interruzione e ripresa con ^S e ^Q

```
$ find / -print
```

^S e ^Q ^C

```
# cat /dev/vcs1
```

I file di dispositivo /dev/vcs*, definiti *virtual console capture device*, possono essere usati per visualizzare lo schermo di una console particolare.

Usare Screen

screen	# invio
C^ a + ?	# lista comandi
C^ a + c	# nuova screen
C^ a + C^ a	# swithc tra scren (nella stessa screen)
C^ a + K	# chiude screen
C^ a + N	# restituisce numero screen
C^ a + S	# split verticale
C^ a + |	# split orizzontale
C^ a + 'TAB'	# cambia il fuoco tra le screen
C^ a + x	# lockscreen
screen -ls	# lista sessioni attive
screen -d	# stacca screen
sceen -r	# attacca screen