

## **Console, terminale e privacy** (emulare un thin client per UNIX e GNU/Linux)

Augusto Scatolini ([webmaster@comunecampagnano.it](mailto:webmaster@comunecampagnano.it)) ([a.scatolini@linux4campagnano.net](mailto:a.scatolini@linux4campagnano.net))  
Miniguia n. 150

Ver. 1.0 Ottobre 2011



Il VT100 (mostrato in figura) è un video terminale costruito dalla Digital Equipment Corporation (DEC) che divenne uno standard de facto per gli emulatori di terminale.

Il VT100 fu il primo terminale della Digital basato su un microprocessore standard (in questo caso l'Intel 8080)

Un terminale che dipende dal computer host per la sua capacità computazionale è chiamato “thin client”.

La configurazione informatica delle aziende negli anni 80' era la seguente:

Un server UNIX, di solito collocato in uno sgabuzzino o in sottoscala, collegato a uno scatolotto dal quale partivano 5, 10, 20 o più cavi seriali che collegavano in parte dei video terminali (come il VT100) e in parte delle stampanti collocati nelle varie stanze dello stesso piano o su altri piani del fabbricato.

Quando il sistemista doveva fare operazioni delicate come l'accensione o lo spegnimento del server accedeva ad una console sul server e controllava i processi.

Gli impiegati dell'azienda "cosiddetti terminalisti" eseguivano il login sul proprio terminale e dopo essere stati autenticati potevano lavorare con un paio, o poco più, di programmi.

Ovviamente si al console che i terminali lavoravano in modalità rigorosamente TESTUALE. Caratteri bianchi su sfondo nero, al massimo caratteri neri su sfondo verde (o viceversa?)

Oggi, nei sistemi Unix-like come GNU/Linux, a livello server, si opera ancora con la console. Questo perché su un server, di solito, non opera nessuno e quando un sistemista deve fare manutenzione, sostanzialmente, non ha bisogno di alcuna applicazione grafica.

Il lavoro dei sistemisti consiste nel creare o editare dei file di testo (solitamente di configurazione) e gli è sufficiente un editor testuale come VI o al massimo come Nano (per le nuove generazioni).

## **RUNLEVEL**

Il termine runlevel si riferisce alle diverse modalità di inizializzazione dei sistemi UNIX-like.

Convenzionalmente, (può variare da distribuzione a distribuzione) il numero (7) e le specifiche dei runlevel è il seguente:

<b>ID</b>	<b>Nome</b>	<b>Descrizione</b>
0	Halt	Spegne il sistema
1	Singolo utente	Modalità per compiti amministrativi
2	Multi utente senza rete	Non configura la scheda di rete
3	Multi utente con rete	Avvio normale senza grafica
4	Non usato	Per scopi speciali
5	Multi utente con rete e grafica	Come il runlevel 3 + interfaccia grafica
6	Reboot	Riavvia il sistema

Quindi, un server si avvia con runlevel 3, senza grafica e quindi con una console

Un sistema Desktop si avvia con un runlevel 5, con interfaccia grafica.

Tuttavia, anche nei sistemi desktop si aprire una o più console virtuali o uno o più terminali.

## **Elementi di base della console**

Prima di tutto bisogna aprire una console. Ci sono almeno due modalità che ci permettono di avere a disposizione lo stesso tipo di console, ma con un aspetto grafico diverso:

Modalità testo; (CONSOLE)

Modalità grafica, emulazione della modalità testuale (TERMINALE)

## MODALITA' CONSOLE (TESTO)

La console, da non confondersi con il terminale, è una sessione puramente testuale  
Si possono aprire, generalmente, 6 diverse console con la combinazione di tasti CTRL, Alt e Fx

Ctrl + Alt + F1	prima console
Ctrl + Alt + F2	seconda
Ctrl + Alt + F3	terza
Ctrl + Alt + F4	quarta
Ctrl + Alt + F5	quinta
Ctrl + Alt + F6	sesta

per chiudere una console premere la combinazione di tasti Ctrl + Alt + F7

Le console simulano i famosi terminali vt100

Ogni comando che digitiamo, oltre ad essere memorizzato in una cronologia, comporta la relativa visualizzazione sullo schermo e potrebbe essere letto da chiunque abbia accesso al nostro computer.

il file nascosto **.bash\_logout** (nella propria Home) contiene il comando **clear console** per aumentare la riservatezza (privacy) e cancellare appunto la cronologia quando si effettua il logout o si usa il comando exit

```
# ~/.bash_logout: executed by bash(1) when login shell exits.  
# when leaving the console clear the screen to increase privacy  
if [ "$SHLVL" = 1 ]; then  
    [ -x /usr/bin/clear_console ] && /usr/bin/clear_console -q  
fi
```

## MODALITA' TERMINALE (GRAFICO)

Il terminale permette di utilizzare la **riga di comando** detta anche **prompt dei comandi** detta anche **shell** tipicamente la **Bash**

all'avvio del terminale compare **utente@computer:~\$**

utente è il nome con il quale avete eseguito il login

computer è il nome del computer

la tilde (~) sta ad indicare che vi trovate nella vostra cartella home,

~ si può scrivere con Alt + 126 oppure AltGr + ì

mentre il simbolo del dollaro (\$) significa che state operando come un utente normale

I comandi li possiamo digitare direttamente da tastiera o anche (...e meno male...) utilizzando il "copia ed incolla"

Per acquisire, momentaneamente, i privilegi di root, possiamo scrivere, prima del comando, la parolina magica: **sudo**

la password rimane valida per circa cinque minuti dall'ultimo utilizzo del comando

utente@computer:~\$ sudo -s                   input per diventare root (primo metodo)  
root@ubuntu:~#                               risposta

utente@ubuntu:~\$ su                         input per diventare root (secondo metodo)  
Password: inserite la password di root     risposta  
root@ubuntu:~#

quando si diventa root il simbolo dollaro cambia con sharp o semplicemente cancelletto

procedura per settare la password di root e quindi attivarlo:

```
utente@ubuntu:~$ sudo passwd root  
Password: inserite la vostra password utente  
Enter new UNIX password: inserite la nuova password di root  
Retype new UNIX password: ripetete la nuova password di root
```

## La cronologia del terminale

Se lanciate il comando **history** nel terminale, verrà visualizzata una lunga serie di comandi: si tratta, in pratica, di tutti quelli che avete impartito nel tempo. I vecchi comandi possono essere riesumati con il tasto freccia su. (e freccia giù)

La lista dei comandi impartiti è registrata nel file nascosto **.bash\_history** (nella propria home) cancellando questo file si elimina la cronologia storica dei comandi impartiti

Se la nostra cronologia dei comandi è troppo affollata, bash ci permette di cercare un particolare comando attraverso una ricerca, con la combinazione di tasti [CTRL] + [r]

Un'altra funzione per ottimizzare la digitazione dei comandi è il completamento: ogni comando o percorso del filesystem può essere completato in automatico con la pressione del tasto **TAB**.

## Esecuzione di un processo in background

La console (terminale) supporta le funzionalità multitasking di GNU/Linux, cioè ci permette di eseguire più comandi senza rendere inusabile il prompt dei comandi. Infatti, se lanciamo un processo che impiega molto tempo per essere eseguito, rischiamo di restare minuti ad attendere che il prompt si liberi.

Lanciando un comando, posponendo il carattere **& (ampersand)** si indica a bash che l'esecuzione deve essere in background, non in foreground come avviene normalmente.

Il caso più frequente in cui conviene eseguire in processo in background è quando si lancia un programma con interfaccia grafica. Per esempio, lanciando firefox da terminale così:

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

 il prompt resterà bloccato finché la finestra di firefox resterà aperta.

Lanciando il comando così, invece:

```
utente@computer:~$ firefox &
```

l'applicazione viene lanciata ugualmente, ma ci viene subito restituito il prompt pronto per ricevere un altro comando.

## CONSOLE REMOTA

Una volta installato **OpenSSH-server** sulla macchina GNU/Linux 1 dalla macchina GNU/Linux 2 (o anche da una macchina Windows) si può aprire una console tramite il programma **Putty** (emulatore terminale) usando il protocollo **SSH**

## GRAFICAMENTE

La macchina 2 (Linux) può aprire la home di un utente della macchina 1 (Linux) tramite il programma “**connessione al server**” usando il protocollo **SSH e SFTP**

oppure la macchina windows può aprire un desktop remoto (grafico) sulla macchina 1 (Linux) tramite **Putty+Xming**

Un desktop remoto grafico da linux a linux può essere fatto con il programma **NoMachine nx** (proprietario) o con **FreeNX** (Open Source)

C'è anche la soluzione trasversale **VNC** che funziona con tutte le combinazioni  
Windows Windows  
Windows GNU/Linux  
GNU/Linux GNU/Linux  
GNU/Linux Windows

FINE

Questo documento è rilasciato con licenza Copyleft  
(tutti i rovesci sono riservati)

altre miniguide su

<http://www.comunecampagnano.it/gnu/miniguide.htm>  
oppure direttamente su <http://miniguide.tk>

sito consigliato: <http://www.linux4campagnano.net>