

RIGA DEI COMANDI

(Burn Shell again)

Augusto Scatolini (webmaster@comunecampagnano.it) (a.scatolini@linux4campagnano.net)

Miniguia n. 174

Ver. 1.0 Settembre 2012

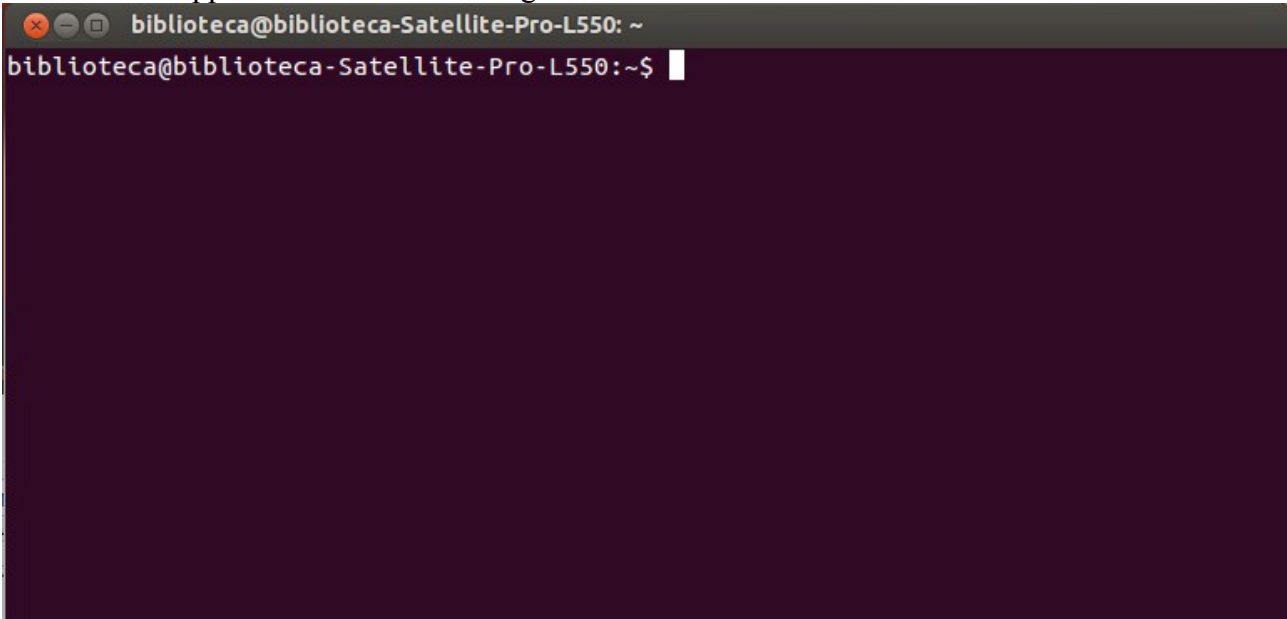


La riga dei comandi

La riga dei comandi su GNU/Linux (in questo caso Ubuntu) è quello strumento (molto potente) al quale si accede in tre diverse circostanze:

- quando si è nel **runlevel 2** ovvero prima di caricare il server grafico che si ottiene con il runlevel 5 (situazione tipica dei server)
- quando si apre una console dall'ambiente grafico con la combinazione **Ctrl+Alt+1** o 2 fino a 6, con **Ctrl+Alt+7** si chiude la console
- quando si apre un **terminale** (emulatore) sempre dall'ambiente grafico

Un terminale appare come mostrato in figura:



```
biblioteca@biblioteca-Satellite-Pro-L550: ~
biblioteca@biblioteca-Satellite-Pro-L550:~$
```

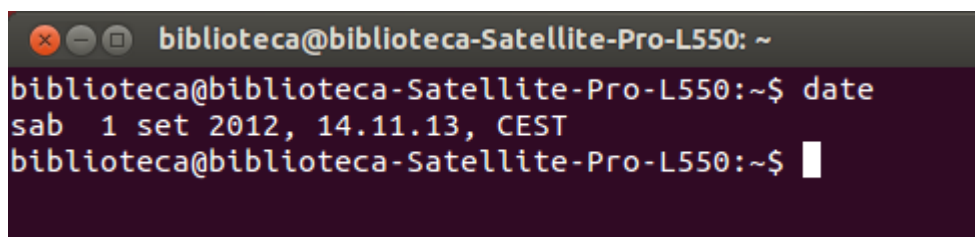
Serve per impartire dei comandi anche molto sofisticati e per lanciare dei programmi, testuali o grafici (nel caso del terminale).

Molti si domanderanno se la riga dei comandi sia presente solo sui sistemi Unix like come GNU/Linux, la risposta è no! Oltre al vecchio DOS è presente sui sistemi Windows (il famoso e poco utilizzato CMD), è presente su iPhone, iPad, Android phone, sui modem e sulle TV più moderne.

Nel mondo GNU/Linux si usa dire continuamente che tutto quello che si può fare graficamente si può fare dalla riga di comando, ma non viceversa.

Il primo comando da imparare è **clear**, serve a ripulire lo schermo dopo che abbiamo pasticciato per diversi minuti.

Continuiamo con **date**, che riporta:



```
biblioteca@biblioteca-Satellite-Pro-L550: ~
biblioteca@biblioteca-Satellite-Pro-L550:~$ date
sab 1 set 2012, 14.11.13, CEST
biblioteca@biblioteca-Satellite-Pro-L550:~$
```

Continuiamo con **whoami** che ci ricorda chi siamo (con quale utente abbiamo fatto il login

```
biblioteca@biblioteca-Satellite-Pro-L550: ~  
biblioteca@biblioteca-Satellite-Pro-L550:~$ whoami  
biblioteca  
biblioteca@biblioteca-Satellite-Pro-L550:~$
```

Sarebbe stato sufficiente osservare bene il prompt per capire che l'utente biblioteca è l'utente attualmente loggato sulla macchina biblioteca-Satellite-Pro-L550

Un altro comando utile può essere **pwd** che ci dice in che posizione del file system ci troviamo

```
biblioteca@biblioteca-Satellite-Pro-L550: ~  
biblioteca@biblioteca-Satellite-Pro-L550:~$ pwd  
/home/biblioteca  
biblioteca@biblioteca-Satellite-Pro-L550:~$
```

Continuiamo con **startx** che avvia il server grafico X (se non è già avviato)

```
biblioteca@biblioteca-Satellite-Pro-L550: ~  
biblioteca@biblioteca-Satellite-Pro-L550:~$ startx  
X: user not authorized to run the X server, aborting.  
XIO: fatal IO error 11 (Resource temporarily unavailable) on X server ":0"  
after 7 requests (7 known processed) with 0 events remaining.  
biblioteca@biblioteca-Satellite-Pro-L550:~$
```

Molto utile è la funzione delle **frecche su e giù** che ripresentano tutti i comandi digitati in precedenza

Un'altra comoda funzione è quella del **tabulatore** dopo che si è iniziato a digitare un comando, il tabulatore lo completerà.

Continuiamo con il comando free che ci mostrerà quanta memoria RAM è ancora disponibile

```
biblioteca@biblioteca-Satellite-Pro-L550: ~  
biblioteca@biblioteca-Satellite-Pro-L550:~$ free  
total used free shared buffers cached  
Mem: 4086216 1731544 2354672 0 162548 959096  
-/+ buffers/cache: 609900 3476316  
Swap: 4937724 0 4937724  
biblioteca@biblioteca-Satellite-Pro-L550:~$
```

Il comando **df /** ci mostrerà su quale partizione è montato il file system, lo spazio su disco occupato, lo spazio libero e lo spazio totale

```
biblioteca@biblioteca-Satellite-Pro-L550: ~  
biblioteca@biblioteca-Satellite-Pro-L550:~$ df /  
File system      1K-blocchi   Usati Disponib.  Uso%  Montato su  
/dev/sda5        115393028  7738868 101875680    8% /  
biblioteca@biblioteca-Satellite-Pro-L550:~$
```

Il comando **hostname -I** ci mostrerà l'indirizzo IP della nostra macchina

```
biblioteca@biblioteca-Satellite-Pro-L550: ~  
biblioteca@biblioteca-Satellite-Pro-L550:~$ hostname -I  
192.168.0.97  
biblioteca@biblioteca-Satellite-Pro-L550:~$
```

Che cosa è il comando **sudo**?

Equivale più o meno a **“Esegui come amministratore”** di windows. Permette di modificare le cose che riguardano tutta la macchina, non solo quelle di un utente.

Proviamo a lanciare il programma **synaptic** “gestore dei pacchetti” senza premettere sudo



Proviamo il comando `find` che serve a trovare uno o più file digitando `find / -name *.py` stiamo cercando dalla radice del file system tutti i file che hanno estensione py ovvero scritti nel linguaggio phyton.

```
biblioteca@biblioteca-Satellite-Pro-L550: ~  
/usr/share/onboard/scripts/sokSettings.py  
/usr/share/onboard/layoutstrings.py  
/usr/share/apt-xapian-index/plugins/app-install.py  
/usr/share/apt-xapian-index/plugins/debtags.py  
/usr/share/apt-xapian-index/plugins/sizes.py  
/usr/share/apt-xapian-index/plugins/software-center.py  
/usr/share/apt-xapian-index/plugins/display_name.py  
/usr/share/apt-xapian-index/plugins/template.py  
/usr/share/apt-xapian-index/plugins/apptags.py  
/usr/share/apt-xapian-index/plugins/sections.py  
/usr/share/apt-xapian-index/plugins/translated-desc.py  
/usr/share/apt-xapian-index/plugins/relations.py  
/usr/share/apt-xapian-index/plugins/aliases.py  
/usr/share/apt-xapian-index/plugins/origin.py  
/usr/share/apt-xapian-index/plugins/cataloged_time.py  
/usr/share/apt-xapian-index/plugins/descriptions.py  
find: "/root": Permessato negato  
find: "/lost+found": Permessato negato  
biblioteca@biblioteca-Satellite-Pro-L550:~$
```

Otterremo una lunga serie di file e di errori dovuti al mancato permesso di cercare nelle directory protette di sistema.

Proviamo ora ad anteporre il comando `sudo`

`sudo find / -name *.py`

```
biblioteca@biblioteca-Satellite-Pro-L550: ~  
/usr/share/ibus-table/engine/tabsqlitedb.py  
/usr/share/ibus-table/engine/factory.py  
/usr/share/onboard/scripts/sokSettings.py  
/usr/share/onboard/layoutstrings.py  
/usr/share/apt-xapian-index/plugins/app-install.py  
/usr/share/apt-xapian-index/plugins/debtags.py  
/usr/share/apt-xapian-index/plugins/sizes.py  
/usr/share/apt-xapian-index/plugins/software-center.py  
/usr/share/apt-xapian-index/plugins/display_name.py  
/usr/share/apt-xapian-index/plugins/template.py  
/usr/share/apt-xapian-index/plugins/apptags.py  
/usr/share/apt-xapian-index/plugins/sections.py  
/usr/share/apt-xapian-index/plugins/translated-desc.py  
/usr/share/apt-xapian-index/plugins/relations.py  
/usr/share/apt-xapian-index/plugins/aliases.py  
/usr/share/apt-xapian-index/plugins/origin.py  
/usr/share/apt-xapian-index/plugins/cataloged_time.py  
/usr/share/apt-xapian-index/plugins/descriptions.py  
biblioteca@biblioteca-Satellite-Pro-L550:~$
```

tutti gli errori sono scomparsi perché il super utente (sudo) può cercare in tutte le directory

Ora proviamo a dirigere l'output di questo comando su un file di nome programmi_python

sudo find / -name *.py > programmi_python

```
biblioteca@biblioteca-Satellite-Pro-L550: ~  
biblioteca@biblioteca-Satellite-Pro-L550:~$ sudo find / -name *.py > programmi_python  
biblioteca@biblioteca-Satellite-Pro-L550:~$
```

Sullo schermo non vedremo niente perché il risultato del comando è andato a creare il file programmi_python

Per vedere il contenuto del file programmi_python possiamo usare il comando cat

cat programmi_python

```
biblioteca@biblioteca-Satellite-Pro-L550: ~  
/usr/share/apt-xapian-index/plugins/app-install.py  
/usr/share/apt-xapian-index/plugins/debtags.py  
/usr/share/apt-xapian-index/plugins/sizes.py  
/usr/share/apt-xapian-index/plugins/software-center.py  
/usr/share/apt-xapian-index/plugins/display_name.py  
/usr/share/apt-xapian-index/plugins/template.py  
/usr/share/apt-xapian-index/plugins/apptags.py  
/usr/share/apt-xapian-index/plugins/sections.py  
/usr/share/apt-xapian-index/plugins/translated-desc.py  
/usr/share/apt-xapian-index/plugins/relations.py  
/usr/share/apt-xapian-index/plugins/aliases.py  
/usr/share/apt-xapian-index/plugins/origin.py  
/usr/share/apt-xapian-index/plugins/cataloged_time.py  
/usr/share/apt-xapian-index/plugins/descriptions.py  
biblioteca@biblioteca-Satellite-Pro-L550:~$
```

Il file (risultante) può essere anche molto lungo ma, possiamo ancora fare qualcosa tipo ordinare alfabeticamente il contenuto del file con il comando sort

sort programmi_python

```
biblioteca@biblioteca-Satellite-Pro-L550: ~  
/usr/share/system-config-printer/troubleshoot/QueueRejectingJobs.py  
/usr/share/system-config-printer/troubleshoot/RemoteAddress.py  
/usr/share/system-config-printer/troubleshoot/SchedulerNotRunning.py  
/usr/share/system-config-printer/troubleshoot/ServerFirewalled.py  
/usr/share/system-config-printer/troubleshoot/Shrug.py  
/usr/share/system-config-printer/troubleshoot/VerifyPackages.py  
/usr/share/system-config-printer/troubleshoot/Welcome.py  
/usr/share/system-config-printer/userdefault.py  
/usr/share/system-config-printer/xml/validate.py  
/usr/share/xul-ext/ubufox/scripts/check-component-enabled.py  
/usr/src/linux-headers-3.2.0-24/scripts/rt-tester/rt-tester.py  
/usr/src/linux-headers-3.2.0-24/scripts/tracing/draw_functrace.py  
/usr/src/linux-headers-3.2.0-29/scripts/rt-tester/rt-tester.py  
/usr/src/linux-headers-3.2.0-29/scripts/tracing/draw_functrace.py  
biblioteca@biblioteca-Satellite-Pro-L550:~$
```

Oppure possiamo cercare la parola tester nella lista

grep "tester" programmi_python

```
biblioteca@biblioteca-Satellite-Pro-L550: ~  
biblioteca@biblioteca-Satellite-Pro-L550:~$ grep "tester" programmi_python  
/usr/lib/python2.7/dist-packages/twisted/test/process_tester.py  
/usr/src/linux-headers-3.2.0-29/scripts/rt-tester/rt-tester.py  
/usr/src/linux-headers-3.2.0-24/scripts/rt-tester/rt-tester.py  
/usr/share/pyshared/twisted/test/process_tester.py  
biblioteca@biblioteca-Satellite-Pro-L550:~$
```

Oppure possiamo contare quante righe ci sono in questo file

wc -l programmi_python

```
biblioteca@biblioteca-Satellite-Pro-L550: ~  
biblioteca@biblioteca-Satellite-Pro-L550:~$ wc -l programmi_python  
7043 programmi_python  
biblioteca@biblioteca-Satellite-Pro-L550:~$
```

Se si volessero concatenare due comandi dirigendo il risultato del primo sul secondo si può usare l'opzione **chiamata pipe**

sudo find / -name *.py | sort

```
biblioteca@biblioteca-Satellite-Pro-L550: ~  
/usr/share/system-config-printer/troubleshoot/QueueRejectingJobs.py  
/usr/share/system-config-printer/troubleshoot/RemoteAddress.py  
/usr/share/system-config-printer/troubleshoot/SchedulerNotRunning.py  
/usr/share/system-config-printer/troubleshoot/ServerFirewalled.py  
/usr/share/system-config-printer/troubleshoot/Shrug.py  
/usr/share/system-config-printer/troubleshoot/VerifyPackages.py  
/usr/share/system-config-printer/troubleshoot/Welcome.py  
/usr/share/system-config-printer/userdefault.py  
/usr/share/system-config-printer/xml/validate.py  
/usr/share/xul-ext/ubufox/scripts/check-component-enabled.py  
/usr/src/linux-headers-3.2.0-24/scripts/rt-tester/rt-tester.py  
/usr/src/linux-headers-3.2.0-24/scripts/tracing/draw_functrace.py  
/usr/src/linux-headers-3.2.0-29/scripts/rt-tester/rt-tester.py  
/usr/src/linux-headers-3.2.0-29/scripts/tracing/draw_functrace.py  
biblioteca@biblioteca-Satellite-Pro-L550:~$
```

Oppure contarli con il comando

sudo find / -name *.py | wc -l

```
biblioteca@biblioteca-Satellite-Pro-L550: ~  
biblioteca@biblioteca-Satellite-Pro-L550:~$ sudo find / -name *.py | wc -l  
7043  
biblioteca@biblioteca-Satellite-Pro-L550:~$
```

oppure quanti programmi contengono la parola “tester”

sudo find / -name *.py | grep “tester” | wc -l

```
biblioteca@biblioteca-Satellite-Pro-L550: ~  
biblioteca@biblioteca-Satellite-Pro-L550:~$ sudo find / -name *.py | grep "tester"  
" | wc -l  
4  
biblioteca@biblioteca-Satellite-Pro-L550:~$
```

un comodo comando è echo

echo “questa è la lista dei programmi scritti in python”

```
biblioteca@biblioteca-Satellite-Pro-L550: ~  
biblioteca@biblioteca-Satellite-Pro-L550:~$ echo “questa è la lista dei programmi  
scritti in python”  
“questa è la lista dei programmi scritti in python”  
biblioteca@biblioteca-Satellite-Pro-L550:~$
```

ora possiamo concatenare il comando echo con il comando find e ancora con il comando echo tramite ;

echo “questa è la lista dei programmi scritti in python” ; sudo find / -name *.py ; echo “fine della lista, arrivederci”

```
biblioteca@biblioteca-Satellite-Pro-L550: ~  
/usr/share/apt-xapian-index/plugins/debtags.py  
/usr/share/apt-xapian-index/plugins/sizes.py  
/usr/share/apt-xapian-index/plugins/software-center.py  
/usr/share/apt-xapian-index/plugins/display_name.py  
/usr/share/apt-xapian-index/plugins/template.py  
/usr/share/apt-xapian-index/plugins/apptags.py  
/usr/share/apt-xapian-index/plugins/sections.py  
/usr/share/apt-xapian-index/plugins/translated-desc.py  
/usr/share/apt-xapian-index/plugins/relations.py  
/usr/share/apt-xapian-index/plugins/aliases.py  
/usr/share/apt-xapian-index/plugins/origin.py  
/usr/share/apt-xapian-index/plugins/cataloged_time.py  
/usr/share/apt-xapian-index/plugins/descriptions.py  
“fine della lista, arrivederci”  
biblioteca@biblioteca-Satellite-Pro-L550:~$
```


se si chiude tutto il comando precedente tra parentesi tonde, si può ri-direzionare l'output sul famoso file programmi_python

(echo "questa è la lista dei programmi scritti in python" ; sudo find / -name *.py ; echo "fine della lista, arrivederci") > programmi_python

```
biblioteca@biblioteca-Satellite-Pro-L550: ~
biblioteca@biblioteca-Satellite-Pro-L550:~$ (echo "questa è la lista dei programmi scritti in python" ; sudo find / -name *.py ; echo "fine della lista, arrivederci") > programmi_python
biblioteca@biblioteca-Satellite-Pro-L550:~$
```

per sapere quanti file ci sono nella nostra directory si può usare il comando

```
biblioteca@biblioteca-Satellite-Pro-L550: ~
biblioteca@biblioteca-Satellite-Pro-L550:~$ ls | wc -l
11
biblioteca@biblioteca-Satellite-Pro-L550:~$
```

ls | wc -l

quando il risultato di un comando è troppo lungo e occupa più schermi si può usare l'opzione more

sudo find / -name *.py | more

```
biblioteca@biblioteca-Satellite-Pro-L550: ~
/etc/python2.7/sitecustomize.py
/etc/python2.6/sitecustomize.py
/usr/bin/pilprint.py
/usr/bin/pilconvert.py
/usr/bin/pilfont.py
/usr/bin/pildriver.py
/usr/bin/miniterm.py
/usr/bin/pilfile.py
/usr/lib/update-notifier/apt_check.py
/usr/lib/update-notifier/backend_helper.py
/usr/lib/rhythmbox/plugins/im-status/im-status.py
/usr/lib/rhythmbox/plugins/artsearch/lastfm.py
/usr/lib/rhythmbox/plugins/artsearch/oldcache.py
/usr/lib/rhythmbox/plugins/artsearch/local.py
--Ancora--
```

premere lo spazio per la prossima pagina, premere Invio per la prossima riga e premere q per terminare

Più potente del comando more è il comando less

sudo find / -name *.py | less

```
biblioteca@biblioteca-Satellite-Pro-L550: ~  
/etc/python2.7/sitecustomize.py  
/etc/python2.6/sitecustomize.py  
/usr/bin/pilprint.py  
/usr/bin/pilconvert.py  
/usr/bin/pilfont.py  
/usr/bin/pildriver.py  
/usr/bin/miniterm.py  
/usr/bin/pilfile.py  
/usr/lib/update-notifier/apt_check.py  
/usr/lib/update-notifier/backend_helper.py  
/usr/lib/rhythmbox/plugins/im-status/im-status.py  
/usr/lib/rhythmbox/plugins/artsearch/lastfm.py  
/usr/lib/rhythmbox/plugins/artsearch/oldcache.py  
/usr/lib/rhythmbox/plugins/artsearch/local.py  
:
```

La gestione degli errori

Abbiamo visto all'inizio che con il comando **find / -name *.py** senza usare sudo l'elenco risultante conteneva sia errori che nomi di file

con le opzioni 1> e 2> possiamo dividere gli errori dai file isolando gli errori nel file “errori”

```
find / -name *.py 1> programmi_python 2> errori
```

```
biblioteca@biblioteca-Satellite-Pro-L550: ~  
biblioteca@biblioteca-Satellite-Pro-L550:~$ find / -name *.py 1> programmi_python  
2> errori  
biblioteca@biblioteca-Satellite-Pro-L550:~$
```

poi potremo vedere il contenuto dei file con **cat programmi_python** e **cat errori**

il file “errori” poi deve essere cancellato con il comando **rm errori**

per evitare di cancellare ogni volta il file “errori” possiamo usare **2> /dev/null**

find / -name *.py 2> /dev/null | wc -l

```
biblioteca@biblioteca-Satellite-Pro-L550: ~  
biblioteca@biblioteca-Satellite-Pro-L550:~$ find / -name *.py 2> /dev/null | wc -  
l  
7043  
biblioteca@biblioteca-Satellite-Pro-L550:~$
```

Come si carica il modulo del suono? Se non è caricato all'avvio?

La configurazione dei moduli del kernel è nel file `/etc/modules`. Quindi per prima cosa facciamo una copia di backup del file con il comando

`sudo cp /etc/modules /etc/modules.old` in modo tale che se dovessimo combinare qualche pasticcio potremmo sempre ripristinarlo con il comando `sudo cp /etc/modules.old /etc/modules`

poi apriamo il file con l'editor testuale nano

`sudo nano /etc/modules`

e aggiungiamo alla fine il comando `snd_bcm2835`

e infine l'ultimo comando per riavviare il computer `sudo reboot`

se invece si volesse solo spegnerlo `sudo shutdown -h now`

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>
blog consigliato: <http://campagnano-rap.blogspot.com>