

Il terminale per tutti

un'introduzione all'uso della shell su Linux

Simone Capodicasa

`simone.capo@gmail.com`

Linux Day Torino

26 ottobre 2013



Indice

1. Introduzione
2. Comandi
3. Qualche trucco
4. Scripting
5. Un esempio di vita reale



Obiettivo di oggi

Primo approccio con il terminale

Farvi venire la voglia di approfondire la conoscenza di questo strumento, farvi assaggiare la sua potenza, senza essere esauriente (il tempo non lo permetterebbe comunque)



Introduzione



Convenzioni

Per parlare la stessa lingua

1.

Terminale = Shell = Linea di comando

2.

Directory = cartella

3.

Indipendenza dalla distribuzione

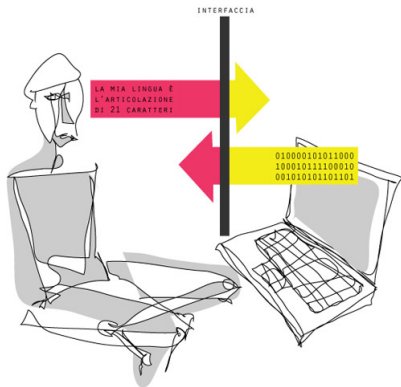
4.

Comandi scritti in carattere a spaziatura fissa



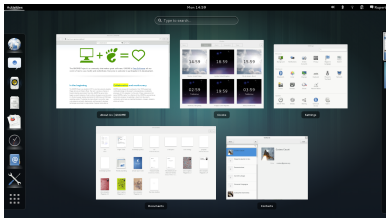
Cosa è il terminale

Un'interfaccia tra programma e utente



GUI vs. CLI

Tutto ciò che si fa tramite
l'interfaccia grafica



```
terminal — bash — 81x24
cheip@macBurgPro ld2013 $ ls
terminal
cheip@macBurgPro ld2013 $ cd terminal/
cheip@macBurgPro terminal $ ls
terminalDefence.tex
cheip@macBurgPro terminal $ mkdir img
cheip@macBurgPro terminal $ ls
img terminalDefence.tex
cheip@macBurgPro terminal $ echo 'Questo sarà uno screenshot per il LD 2013'
Questo sarà uno screenshot per il LD 2013
cheip@macBurgPro terminal $ cd ..
cheip@macBurgPro ld2013 $ ls
terminal
cheip@macBurgPro ld2013 $ cd ..
cheip@macBurgPro Documents $ cd ld2013/
cheip@macBurgPro ld2013 $ ls
terminal
cheip@macBurgPro ld2013 $ cd terminal/
cheip@macBurgPro terminal $ ls
terminalDefence.out terminalDefence.tex
terminalDefence.aux terminalDefence.pdf terminalDefence.toc
terminalDefence.log terminalDefence.sm
terminalDefence.nav terminalDefence.synctex.gz
cheip@macBurgPro terminal $
```

si può replicare usando il
terminale

spesso in modo più efficiente!



Una scelta intelligente

GUI

Nuova versione \Rightarrow funzionalità (come minimo) spostate

CLI

Nuova versione \Rightarrow funzionalità (al massimo) aggiunte



Non si deve imparare di nuovo a guidare



Emulatore di terminale

Come si usa il terminale oggi

Un'interfaccia grafica che simula una linea di comando

- ★ xterm
- ★ Konsole
- ★ Gnome Terminal
- ★ LXTerminal
- ★ infiniti altri...

```

cheip@acurpro ld2013 $ ls
terminal
cheip@acurpro ld2013 $ cd terminal/
cheip@acurpro terminal $ ls
terminalDefence.tex
cheip@acurpro terminal $ mkdir img
cheip@acurpro terminal $ ls
img
cheip@acurpro terminal $ echo 'Questo sarà uno screenshot per il LD 2013'
Questo sarà uno screenshot per il LD 2013
cheip@acurpro terminal $ cd ..
cheip@acurpro ld2013 $ ls
terminal
cheip@acurpro ld2013 $ cd ..
cheip@acurpro Documents $ cd ld2013/
cheip@acurpro ld2013 $ ls
terminal
cheip@acurpro ld2013 $ cd terminal/
cheip@acurpro terminal $ ls
img
cheip@acurpro terminal $ ls
img terminalDefence.out terminalDefence.tex
terminalDefence.aux terminalDefence.pdf terminalDefence.toc
terminalDefence.log terminalDefence.snm
terminalDefence.nav terminalDefence.synctex.gz
cheip@acurpro terminal $

```

Sicuramente si usa ancora la linea di comando pura
(server, ssh, ecc...)



Tanti tipi di shell

nei sistemi operativi di tipo UNIX

- ★ sh
- ★ zsh
- ★ tcsh
- ★ C shell
- ★ Korn shell

noi ci riferiremo sempre alla

BASH

Bourne Again Shell



Il prompt

Chi siamo e dove siamo

Utente

```
username@hostname:directory$
```

Amministratore (root)

```
root@hostname:directory#
```

Esempio (quasi) reale

```
simone@linuxdayto2013:~ $
```



Comandi



La struttura di un comando bash

In generale:

```
comando -opzione argomento
```

Un esempio:

```
rm -f divinaCommedia.pdf
```

Anche più opzioni e più argomenti:

```
rm -rf divinaCommedia.pdf promessiSposi.pdf
```



ls

Stampa la lista del contenuto della directory passata come argomento

```
cheip@macBurpPro ~ $ echo 'Questa è una prova di echo'
Questa è una prova di echo
cheip@macBurpPro ~ $ ls
Applications      Dropbox           Pictures
Calibre Library   Google Drive     Public
Desktop           Library          Sites
Documents         Movies           bin
Downloads         Music            uni
cheip@macBurpPro ~ $
```



cd

Navigare nelle cartelle

Ci spostiamo nella directory passata come argomento

```
bin — bash — 55x14
cheip@macBurpPro ~ $ echo 'Questa è una prova di echo'
Questa è una prova di echo
cheip@macBurpPro ~ $ ls
Applications      Dropbox           Pictures
Calibre Library   Google Drive     Public
Desktop           Library          Sites
Documents         Movies           bin
Downloads         Music            uni
cheip@macBurpPro ~ $ cd bin
cheip@macBurpPro bin $
```



cp

Copiare files e cartelle

Copia il file o la cartella passato come argomento

```
○ ○ ○ cheip — bash — 55x14
Downloads      Music          uni
cheip@macBurpPro ~ $ cd bin
cheip@macBurpPro bin $ ls
dat2LaTeX.sh
cheip@macBurpPro bin $ cp dat2LaTeX.sh ~
cheip@macBurpPro bin $ cd ~
cheip@macBurpPro ~ $ ls
Applications   Google Drive   Sites
Calibre Library Library        bin
Desktop        Movies         dat2LaTeX.sh
Documents      Music          uni
Downloads      Pictures
Dropbox        Public
cheip@macBurpPro ~ $
```



mkdir

Creare directory

Crea la directory specificata

```
Documents Music uni
Downloads Pictures
Dropbox Public
cheip@macBurpPro ~ $ mv dat2LaTeX.sh Desktop
cheip@macBurpPro ~ $ cd Desktop/
cheip@macBurpPro Desktop $ ls
dat2LaTeX.sh
cheip@macBurpPro Desktop $ mv dat2LaTeX.sh nonLoSo.sh
cheip@macBurpPro Desktop $ ls
nonLoSo.sh
cheip@macBurpPro Desktop $ mkdir provaMkdir
cheip@macBurpPro Desktop $ ls
nonLoSo.sh provaMkdir
cheip@macBurpPro Desktop $
```



rm

Rimuove file o directory **senza passare dal cestino!**
Per le directory aggiungere l'opzione **-r**

```
Scrivania — bash — 55x14
cheip@macBurpPro Desktop $ ls
nonLoSo.sh provaMkdir
cheip@macBurpPro Desktop $ rm nonLoSo.sh
cheip@macBurpPro Desktop $ ls
provaMkdir
cheip@macBurpPro Desktop $ rm -r provaMkdir
cheip@macBurpPro Desktop $ ls
cheip@macBurpPro Desktop $
```



su e sudo

Diventare amministratore (utente **root**)



per altre vignette visitare <http://xkcd.com>

Due metodi:

Anteporre **sudo**

al singolo comando

oppure

su

Per loggarsi in una shell di root



Le pagine man

Cercare aiuto

Come si trovano

`man comando`

Cosa c'è dentro

Tutto ciò che c'è da sapere su un programma
(e anche di più)

Nessuna voglia di leggere

`comando -h`



Qualche trucco



Autocompletamento

Il tabulatore



Esiste una lista di comandi e il terminale la conosce

1.

Si scrive un pezzo di comando e si preme TAB

2a.

Se la corrispondenza è univoca,
avremo l'autocompletamento

2b.

Se non è univoca, avremo una
lista delle possibilità (premendo
di nuovo TAB)



Pipe

Il carattere |

Manda l'output del primo comando come input del secondo.

```
cheip — bash — 55x14
cp.png      mkdir.png   wildCard.png
echo.png    mv.png
emulatore.png mv2.png
cheip@macBurpPro img $ cd
cheip@macBurpPro ~ $ ls
Applications  Dropbox      Pictures
Calibre Library Google Drive Public
Desktop       Library      Sites
Documents     Movies       bin
Downloads     Music        uni
cheip@macBurpPro ~ $ ls | grep Do
Documents
Downloads
cheip@macBurpPro ~ $
```



Wild Card

Il carattere *

Si usa l'asterisco per sostituire uno o più caratteri di nomi di file e directory

```

img — bash — 55x14
cd.png          emulatore.png    mv.tiff
cd.tiff          interfaccia.jpg  mv2.png
cowsayDragon.png ls.png           mv2.tiff
cp.png           ls.tiff          rm.png
cp.tiff           mkdir.png      rm.tiff
echo.png          mkdir.tiff       tab.jpg
echo.tiff         mv.png
cheip@macBurpPro img $ rm *.tiff
cheip@macBurpPro img $ ls
cd.png          emulatore.png    mv.png
cowsayDragon.png interfaccia.jpg  mv2.png
cp.png           ls.png           rm.png
echo.png          mkdir.png       tab.jpg
cheip@macBurpPro img $

```



Questa cartella e quella superiore

. e ..

. indica la **working directory**
.. indica la **parent directory**

Non vale aggiungere altri puntini

Non si usa (non funziona) ...
ma ../..

Dove ci si trova?

pwd



Cronologia

Con su (▲) e giù (▼) si scorrono gli ultimi comandi utilizzati

Usando la combinazione di tasti Ctrl+R si effettua una ricerca tra gli ultimi comandi utilizzati



Interrompere l'esecuzione di un comando

`Ctrl+C`

A volte può essere necessario interrompere con la forza l'esecuzione di un programma lanciato da terminale.

Si può fare premendo la combinazione di tasti:
`Ctrl+C`

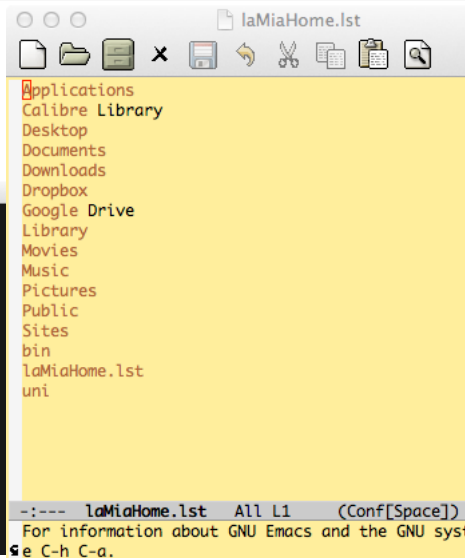


Reindirizzamento dell'output

Il carattere >

Utilizzando il carattere > si manda l'output di un comando in un file di testo

```
cheip@macBurpPro ~ $ ls > laMiaHome.lst
cheip@macBurpPro ~ $ emacs laMiaHome.lst
```

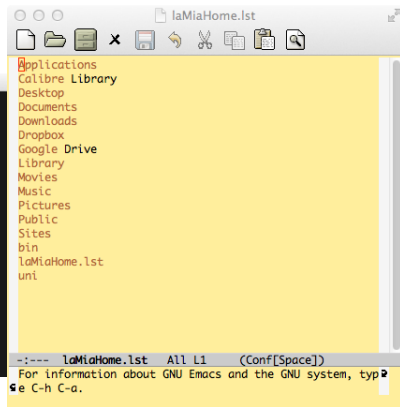


Programmi in background

&

Postponendo & al comando lo si esegue in background, mantenendo le funzionalità del terminale

```
cheip@macBurpPro ~ $ ls > laMiaHome.lst
cheip@macBurpPro ~ $ emacs laMiaHome.lst
cheip@macBurpPro ~ $ emacs laMiaHome.lst &
[1] 9983
cheip@macBurpPro ~ $
```



Scripting



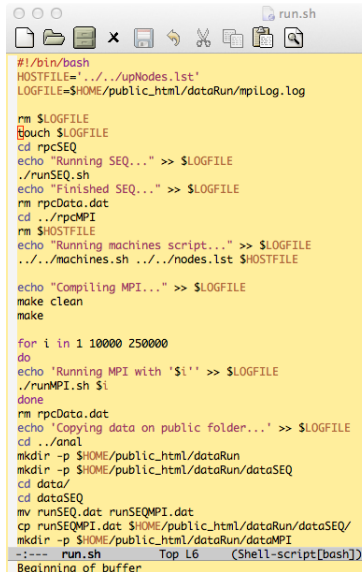
Cosa è uno script?

Un file di testo che contiene una serie di comandi su righe diverse

Esecuzione riga per riga

Lo strumento principe

Un editor di testo
(Emacs, vi, Gedit, ecc...)



```
#!/bin/bash
HOSTFILE='../..upNodes.lst'
LOGFILE=$HOME/public_html/dataRun/mpiLog.log

rm $LOGFILE
touch $LOGFILE
cd rpcSEQ
echo "Running SEQ..." >> $LOGFILE
./runSEQ.sh
echo "Finished SEQ..." >> $LOGFILE
rm rpcData.dat
cd ../rpcMPI
rm $HOSTFILE
echo "Running machines script..." >> $LOGFILE
../machines.sh ../nodes.lst $HOSTFILE

echo "Compiling MPI..." >> $LOGFILE
make clean
make

for i in 1 10000 25000
do
echo 'Running MPI with '$i'' >> $LOGFILE
./runMPI.sh $i
done
rm rpcData.dat
echo 'Copying data on public folder...' >> $LOGFILE
cd ../anal
mkdir -p $HOME/public_html/dataRun
mkdir -p $HOME/public_html/dataRun/dataSEQ
cd data/
cd dataSEQ
mv runSEQ.dat runSEQMPI.dat
cp runSEQMPI.dat $HOME/public_html/dataRun/dataSEQ/
mkdir -p $HOME/public_html/dataRun/dataMPI
-:--- run.sh Top L6 (Shell-script[bash])
Beginning of buffer
```

Solo due regole da rispettare

```
#!/bin/bash
```

Si dichiara quale tipo di shell si vuole usare per lo script

Usiamo l'estensione `.sh`

È solamente una convenzione



Bisogna dare il permesso

e poi eseguire

Dare il permesso di esecuzione

```
chmod u+rx myScript.sh
```

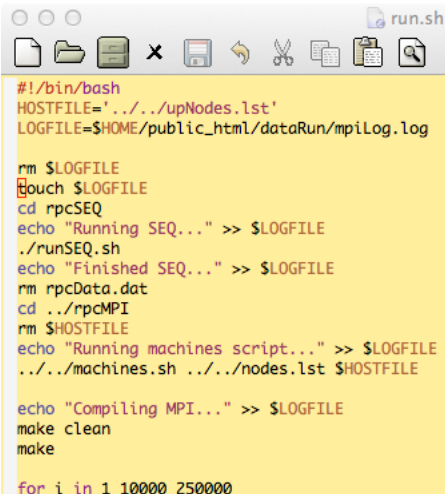
Eseguire

```
./myScript.sh
```



Le variabili

Sono dei nomi brevi (o intuitivi) per richiamare stringhe di testo o numeri



```
#!/bin/bash
HOSTFILE='../..../upNodes.lst'
LOGFILE=$HOME/public_html/dataRun/mpiLog.log

rm $LOGFILE
touch $LOGFILE
cd rpcSEQ
echo "Running SEQ..." >> $LOGFILE
./runSEQ.sh
echo "Finished SEQ..." >> $LOGFILE
rm rpcData.dat
cd ../rpcMPI
rm $HOSTFILE
echo "Running machines script..." >> $LOGFILE
../..../machines.sh ../..../nodes.lst $HOSTFILE

echo "Compiling MPI..." >> $LOGFILE
make clean
make

for i in 1 10000 250000
```



I cicli

Come nei linguaggi di programmazione esiste il modo di ripetere le stesse righe un numero definito (o indefinito) di volte

Esiste anche il costrutto **IF-THEN-ELSE**



Un esempio di vita reale



Mettere i dati in tabella

Situazione

- ★ L'utente ha un file di testo contenente dati
- ★ Vuole mettere questi dati in una tabella \LaTeX

Problema

I dati sono moltissimi!



Problema

Cosa si ha

capTens.dat

1	69.8	70.56	70.41
5	33.7	34.59	34.43
10	24.4	25.15	24.91
15	20.3	20.80	20.67
20	17.7	18.15	17.95
25	15.8	16.49	16.28
30	14.6	15.30	15.07
35	13.7	14.35	14.14
40	12.9	13.60	13.30
45	12.3	12.2	12.40
50	11.8	12.5	12.2
55	11.5	12.07	11.8
60	11.1	11.66	11.41
65	10.7	11.52	11.28
70	10.6	11.44	11.19
75	10.5	11.38	11.12
80	10.5	11.32	11.08
85	10.4	11.28	11.03
90	10.4	11.25	11.00
95	10.35	11.22	10.97
100	10.3	11.19	10.95
105	10.3	11.17	10.92
110	10.27	11.15	10.90
115	10.24	11.13	10.89
120	10.22	11.12	10.87
125	10.2	11.10	10.85
130	10.19	11.09	10.84
135	10.17	11.08	10.82
140	10.16	11.06	10.81
145	10.4	11.05	10.80
150	10.13	11.04	10.79
155	10.11	11.04	10.78
160	10.10	11.02	10.77
165	10.09	11.01	10.76
180	10.04	10.97	10.72
190	10.02	10.96	10.71

~:--- capTens.dat Top L1 (Fundamental)

Cosa si vuole ottenere

diodoCapTend.tex

```

\begin{tabular}{|c|c|c|c|}
\hline
 $\mathcal{V}_{\text{BIAS}}$  (V)$ & \mathcal{S}\mathcal{C}_{\{10\}} (pF)$ & \mathcal{S}\mathcal{C}_{\{100\}} (pF)$ & \mathcal{S}\mathcal{C}_{\{50\}} (pF)$ \\
\hline
1 & 69.80 & 70.56 & 70.41 \\
\hline
5 & 33.70 & 34.59 & 34.43 \\
\hline
10 & 24.40 & 25.15 & 24.91 \\
\hline
15 & 20.30 & 20.80 & 20.67 \\
\hline
20 & 17.70 & 18.15 & 17.95 \\
\hline
25 & 15.80 & 16.49 & 16.28 \\
\hline
30 & 14.60 & 15.30 & 15.07 \\
\hline
35 & 13.70 & 14.35 & 14.14 \\
\hline
40 & 12.90 & 13.60 & 13.30 \\
\hline
45 & 12.30 & 12.20 & 12.40 \\
\hline
50 & 11.80 & 12.50 & 12.20 \\
\hline
55 & 11.50 & 12.07 & 11.80 \\
\hline
60 & 11.10 & 11.66 & 11.41 \\
\hline
65 & 10.70 & 11.52 & 11.28 \\
\hline
70 & 10.60 & 11.44 & 11.19 \\
\hline
75 & 10.50 & 11.38 & 11.12 \\
\hline
\end{tabular}

```

~:--- diodoCapTend.tex Top L34 (LaTeX)

Beginning of buffer

Soluzione

L'utente ha scritto uno script
Con un solo comando la tabella
è pronta!

$V_{BIAS}(V)$	$C_{10}(pF)$	$C_{100}(pF)$	$C_{50}(pF)$
1	69.80	70.56	70.41
5	33.70	34.59	34.43
10	24.40	25.15	24.91
15	20.30	20.80	20.67
20	17.70	18.15	17.95
25	15.80	16.49	16.28
30	14.60	15.30	15.07
35	13.70	14.35	14.14
40	12.90	13.60	13.30
45	12.30	12.20	12.40
50	11.80	12.50	12.20
55	11.50	12.07	11.80

```

dat2LaTeX.sh

#stampo l'help se lo script è senza argomenti
if [ $# -eq 0 ]
then
echo
echo -e $HELP
exit 1
fi

#setto i defaults
COLS=0
HSTYLE=""
VSTYLE=0

#leggo le opzioni e setto le variabili opportune
while getopts ":c:s:v:h" OPT
do
case $OPT in
c ) COLS=$OPTARG;;
s ) HSTYLE=$OPTARG;;
v ) VSTYLE=$OPTARG;;
h ) echo -e $HELP; exit 0;;
esac
done
shift $((OPTIND - 1))
INFILE=$1
OUTFILE=$2

#controlla come sono state passate le opzioni
while [ $COLS -eq 0 ]
do
echo 'You did not set any number of columns. Set a number before continuing.'
read COLS
done
while [ $VSTYLE -eq 0 ]
do
echo 'You did not set any vertical style. Set one style before continuing.'
read VSTYLE
done
1:--- dat2LaTeX.sh 33% L23 (Shell-script[bash])

```

Conclusioni

Il terminale è un modo alternativo di interagire con la macchina

Esiste un efficace linguaggio che permette di comunicare con il computer

Questo linguaggio può essere usato per automatizzare e/o velocizzare certe operazioni ripetitive oppure lunghe e complicate



Un ultimo consiglio

In rete si trovano moltissime guide che usano comandi da terminale per effettuare varie operazioni

È importante capire bene cosa si sta facendo

Evitare il copia-incolla alla cieca!



Riferimenti

Guida completa

<http://www.pluto.it/files/ildp/guide/abs/>

Queste slide

<http://linuxdaytorino.org/2013>

oppure

<http://newton.ph.unito.it/~capodica>



Appendici

6. Altri comandi

7. Altri trucchi

8. Ambiente di lavoro

9. Aggiornare il sistema

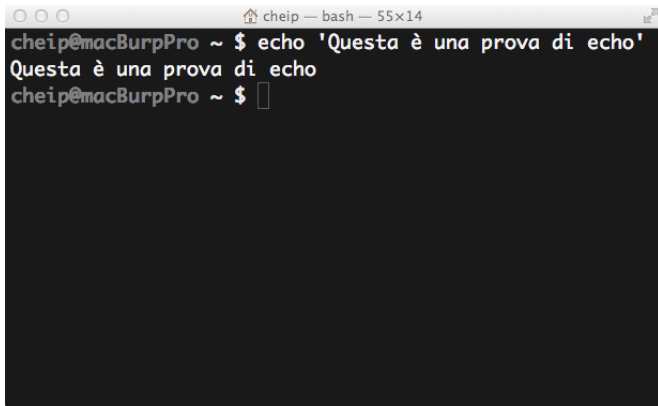


Altri comandi



echo

Stampa a video una stringa di testo

A screenshot of a terminal window with a dark background. The window title bar shows three window control buttons on the left, a home icon followed by 'cheip — bash — 55x14' in the center, and a close button on the right. The terminal content shows the prompt 'cheip@macBurpPro ~ \$' followed by the command 'echo 'Questa è una prova di echo''. The output 'Questa è una prova di echo' is displayed on the next line. The prompt 'cheip@macBurpPro ~ \$' is shown again on the third line, followed by a cursor icon.

```
cheip@macBurpPro ~ $ echo 'Questa è una prova di echo'
Questa è una prova di echo
cheip@macBurpPro ~ $
```



mv

Per spostare...

Sposta il file nella posizione indicata (due argomenti)

```
Scrivania — bash — 55x14
cheip@macBurpPro bin $ cp dat2LaTeX.sh ~
cheip@macBurpPro bin $ cd ~
cheip@macBurpPro ~ $ ls
Applications      Google Drive      Sites
Calibre Library   Library           bin
Desktop           Movies            dat2LaTeX.sh
Documents         Music             uni
Downloads         Pictures
Dropbox          Public
cheip@macBurpPro ~ $ mv dat2LaTeX.sh Desktop
cheip@macBurpPro ~ $ cd Desktop/
cheip@macBurpPro Desktop $ ls
dat2LaTeX.sh
cheip@macBurpPro Desktop $
```



mv

... o per rinominare

Può essere usato anche per rinominare file o directory

```
Scrivania — bash — 55x14
Applications  Google Drive  Sites
Calibre Library Library      bin
Desktop       Movies        dat2LaTeX.sh
Documents     Music         uni
Downloads     Pictures
Dropbox       Public

cheip@macBurpPro ~ $ mv dat2LaTeX.sh Desktop
cheip@macBurpPro ~ $ cd Desktop/
cheip@macBurpPro Desktop $ ls
dat2LaTeX.sh
cheip@macBurpPro Desktop $ mv dat2LaTeX.sh nonLoSo.sh
cheip@macBurpPro Desktop $ ls
nonLoSo.sh
cheip@macBurpPro Desktop $
```



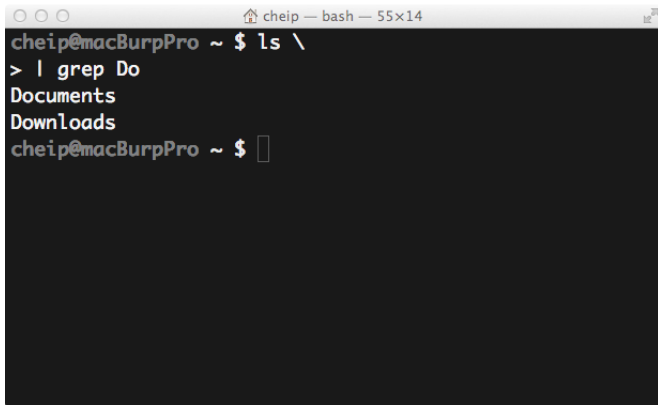
Altri trucchi



Andare a capo

Il carattere \

Comandi lunghi si possono scrivere su più righe, inserendo alla fine della riga un \

A terminal window titled 'cheip — bash — 55x14' showing a multi-line command. The prompt is 'cheip@macBurpPro ~ \$'. The command 'ls \> | grep Do' is entered across two lines. The output shows 'Documents' and 'Downloads'. The prompt returns to 'cheip@macBurpPro ~ \$' with a cursor.

```
cheip@macBurpPro ~ $ ls \> | grep Do
Documents
Downloads
cheip@macBurpPro ~ $
```



Comandi con condizioni

&& e ||

Si inseriscono tra due comandi sulla stessa linea

&&

Esegue il primo comando e, se questo va a buon fine, viene eseguito anche il secondo

||

Prova a eseguire il primo comando e, se questo non va a buon fine, si prova a eseguire il secondo



Ambiente di lavoro



Files importanti

Quali sono?

- ★ `/etc/profile`
- ★ `~/.bash_profile`, `~/.bash_login`, `~/.profile`
- ★ `~/.bashrc`

Vengono letti e caricati ogni volta che si apre un terminale



/etc/profile

Contiene le variabili d'ambiente e i programmi di avvio comuni a tutti gli utenti del sistema

Un esempio:

La variabile PATH

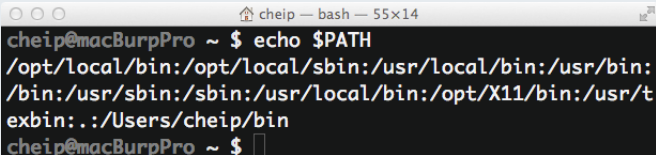
Definisce la lista dei comandi conosciuti al sistema (utilizzata per l'autocompletamento)



La variabile PATH

È una **lista di directory** dove la bash andrà a cercare gli eseguibili quando scriviamo un comando sul terminale

La mia PATH



```
cheip@macBurpPro ~ $ echo $PATH
/opt/local/bin:/opt/local/sbin:/usr/local/bin:/usr/bin:
/bin:/usr/sbin:/sbin:/usr/local/bin:/opt/X11/bin:/usr/t
exbin:./Users/cheip/bin
cheip@macBurpPro ~ $
```



~/ .bash_profile

Variabili d'ambiente e programmi di avvio dell'utente

Un esempio:

La variabile PATH

È possibile ridefinirla



~ /.bashrc

Configurazioni utente della **bash**

Alcuni esempi:

- ★ alias
- ★ personalizzazioni del prompt
- ★ definizioni di funzioni



Gli alias

Sono abbreviazioni di comandi lunghi o ricorrenti

Un alias dal mio `~/.bashrc`

```
alias r='root -l'
```



Aggiornare il sistema



Aggiornare il sistema

Debian (e derivate: Ubuntu, Linux Mint, ecc...)

```
sudo apt-get update && sudo apt-get upgrade
```

Fedora (e simili: CentOS, ecc...)

```
sudo yum update
```

Archlinux

```
sudo pacman -Syu
```

