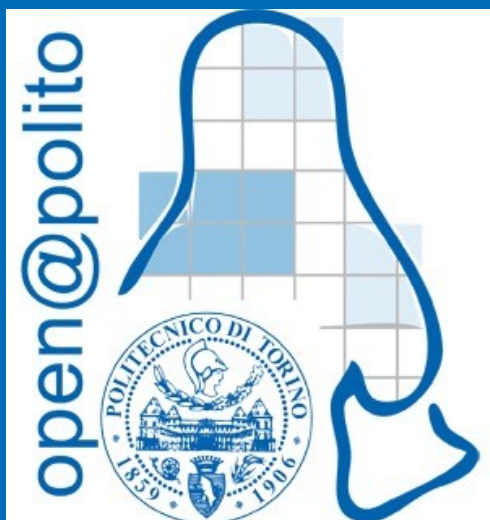




Con il supporto di



FOLLOW-UP SOPRAVVIVENZA NELLA SHELL

Andrea Mannarella

25/11/2021

ANATOMIA DI UN COMANDO

<comando> <qualificatori> <parametro1> <parametro2> ...

```
ls -lah /home/utonto/Documenti
```

Generalmente i qualificatori si possono raggruppare e possono essere scritti sia in forma estesa con -- che abbreviata con -

GREP

Cerca una stringa all'interno di un file

- i case **i**nsensitive
- n precede ogni linea visualizzata con il numero della riga
- c numero totale delle linee che contengono il testo ricercato

General **R**egular **E**xpression **P**rint

HEAD

Stampa l'inizio di un file

-n scegliamo quante linee visualizzare (10 di default)

TESTA

TAIL

Stampa la fine di un file

- n scegliamo quante linee visualizzare (10 di default)
- f non esce quando finisce di stampare

CODA

LESS

Visualizza il contenuto di un file come cat, ma permette di scorrere e di effettuare ricerche al suo interno

/testo	per cercare del testo		
n	occorrenza successiva	N	precedente
q	esce		

Il significato del suo nome ha ragioni storiche

ECHO

Stampa cosa riceve da parametro

- n non va a capo
- e interpreta i caratteri di escape per modificare caratteristiche del testo
"\e[36;1;5mTesto ciano grassetto lampeggiante"

ECO

LN

Crea un collegamento "hard link"

-s crea un "soft link"
ls -lh riconosciamo i collegamenti

LiNk

FIND

Cerca dei file in una directory

-name	definiamo il nome del file che deve cercare
-type d	cerchiamo una cartella

TROVA

HISTORY

Stampa la lista dei comandi eseguiti da quell'utente

↑ ↓	scorrono nella cronologia
!!	ultimo comando
!NUM	comando NUM
ctrl + r	cerca nella cronologia

CRONOLOGIA

ALIAS

Crea un comando personalizzato.

```
alias l="ls -lah"
```

Eseguito da solo elenca li alias gi impostati.

temporaneo. Per renderlo permanente bisogna editare ~/.bashrc

NANO

Editor di testo da console user-friendly

Tutti i comandi sono scritti nella parte bassa, per muoversi si usano le freccette

ma3e.github.io/brains

:wq

\$ killall vim

Press Alt+F4

Press X button
on the terminal
window

\$ sudo shutdown -r now



VIM

Editor di testo avanzato, difficile da padronegiare

Ha un'infinitezza di comandi tra cui

:wq

salva ed esci

i

modalità inserimento

esc

esci dalla modalità attuale

↑ ↓ → ←

per spostarsi



TRICK DA TASTIERA

`ctrl + l`

pulisce il terminale

`TAB`

autocompleta il testo

`ctrl + c`

termina l'esecuzione di un comando

`ctrl + d`

chiude la shell

`ctrl + a`

va all'inizio della riga che stiamo scrivendo

`ctrl + e`

va alla fine della riga che stiamo scrivendo

`ctrl + u`

cancella una riga

`ctrl + w`

cancella una parola

TRICK COPIA INCOLLA

ctrl + shift + c

copia

ctrl + shift + v

incolla

tasto rotella del mouse

copia e incolla

tre dita con il touchpad

come sopra

REDIREZIONE OUTPUT

comando > file

cattura l'output del comando e lo scrive nel file

comando >> file

come sopra ma appende invece che riscriverlo

comando < file

ridirige lo standard input da un file

comando 2> file

cattura gli errori del comando e lo scrive nel file

comando1 | comando2

incanala l'output del comando1 nell'input del comando2

COME IMPARO DA SOLO?

- **MAN**uale nel terminale
- `--help` come parametro
- Documentazione online, wiki
- Cheatsheet
- [Cerca sul Web](#)



GoSquared		VI Help Sheet
Modes & Controls		
Command Mode ESC (commands preceded by :) Insertion Mode Entered on insertion or change		
Starting VI (command line)		
<code>vi <filename></code>	Edit filename	
<code>vi -r <filename></code>	Edit last version of filename after crash	
<code>vi + n <filename></code>	Edit filename at line n	
<code>vi <filename></code>	Edit filename at end of file	
<code>vi +/str <filename></code>	Edit filename at first occurrence of str	
In insertion mode the following should be preceded by ESC:		
<code>:w</code>	Save	
<code>:x</code>	Save & Exit	
<code>:q</code>	Exit if no changes made	
<code>:q!</code>	Exit & discard any changes	
Cursor Navigation		
<code>h</code> or <code>←</code>	Cursor left	
<code>j</code> or <code>↓</code>	Cursor down	
<code>k</code> or <code>↑</code>	Cursor up	
<code>l</code> or <code>→</code>	Cursor right	
<code>w</code>	Next word	
<code>W</code>	Next blank delimited word	
<code>b</code>	Start of word	
<code>B</code>	Start of blank delimited word	
<code>e</code>	End of word	
<code>E</code>	End of blank delimited word	
<code>(</code>	Back a sentence	
<code>)</code>	Forward a sentence	
<code>{</code>	Back a paragraph	
<code>}</code>	Forward a paragraph	
<code>0</code>	Beginning of line	
<code>\$</code>	End of the line	
<code>1G</code>	Start of file	
<code>G</code>	End of file	
<code>:n</code>	nth line of file	
<code>f<char></code>	Forward to char	
<code>F<char></code>	Back to char	
<code>H</code>	Top of screen	
<code>M</code>	Middle of screen	
<code>L</code>	Bottom of screen	
<code>%</code>	Matching bracket	
<code>gg</code>	Start of document	
Inserting Text		
<code>i</code>	Insert before cursor	
<code>a</code>	Append after cursor	
<code>I</code>	Insert before line	
<code>A</code>	Append after line	
<code>o</code>	Add new line after current line	
<code>O</code>	Add new line before current line	
<code>r</code>	Overwrite one character	
<code>R</code>	Overwrite many characters	
<code>τ <file></code>	Reads file and inserts it after this line	
<code>p</code>	Put after the position or line	
<code>P</code>	Put before the position or line	
<code>C</code>	Rewrite the whole line	
Deleting Text		
<code>x</code>	Delete character to right of cursor	
<code>X</code>	Delete character to left of cursor	
<code>D</code>	Delete the rest of line	
<code>dd</code> or <code>xd</code>	Delete current line	
<code>ndw</code>	Deletes the next n words	
<code>ndb</code>	Deletes the previous n words	
<code>ndd</code>	Deletes n lines starting with current	
<code>xyd</code>	Delete lines x through y	
<code>d(nav_cmd)</code>	Overwrite many characters	
Searching		
<code>/string</code>	Search forward for string	
<code>?string</code>	Search backwards for string	
<code>n</code>	Go to next match	
<code>N</code>	Go to previous match	
<code>:set ic</code>	Ignore case while searching	
<code>:set noic</code>	Case-sensitive searching	
<code>:set nu</code>	Turn on line numbers	
<code>:x,yg/str</code>	Search for str from line x to line y	
<code>g/str/cmd</code>	Run cmd on lines containing str	
<code>*</code>	Search for next instance of current word	
<code>#</code>	Search for last instance of current word	
Replacing		
<code>:s/pt/str/flag</code>	Replace pattern with string	
Flags		
<code>g</code>	Replace all occurrences of pattern	
<code>c</code>	Confirm replaces	
<code>&</code>	Repeat last :s command	
Other		
<code>u</code>	Undo last change	
<code>J</code>	Join lines	
<code>nJ</code>	Join next n lines	
<code>.</code>	Repeat last command	
<code>U</code>	Undo all changes to line	
<code>:N</code>	Open split screen	
<code>v</code>	Visual mode	
<code>ctrl + c</code>	Escape insert mode	

Download this Help Sheet now at gosquared.com/liquidity
Put it on your wall Referenced from <https://www.lagmonster.org/docs/vi.html>

© 2010 Go Squared Ltd.

BONUS

&

wc

sl

&&

diff

awk

tee

sort

cowsay

cmatrix

watch

lolcat

!\$

date

dd

unzip

lsblk

tlldr