

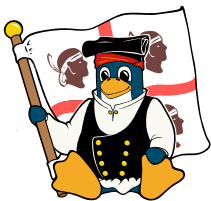
Il Magico Mondo della Linea di Comando

– Linuxday 2016 –

Francesco Versaci

(CRS4)

22 ottobre 2016



- 1 Filesystem
- 2 La shell Bash
- 3 Comandi
 - Comandi Principali
 - Comandi Avanzati



Organizzazione del Filesystem

- **Everything is a file** (dischi, terminali, scheda audio, ...)
- Un unico albero su cui si montano i dispositivi

Caratteri e directory speciali

/ Separatore directory

.nome Un punto a inizio nome indica un file "nascosto"

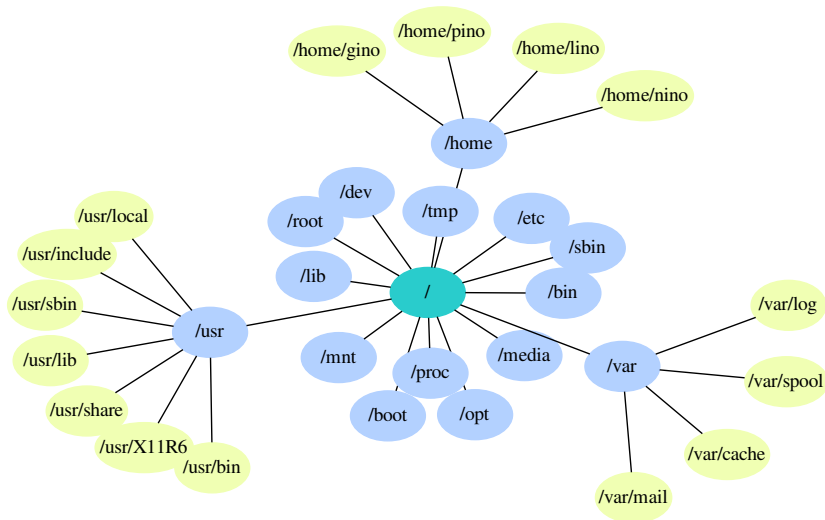
~ La home directory dell'utente

. La directory corrente

.. La directory superiore



Filesystem Hierarchy Standard (FHS) – 1/3



Filesystem Hierarchy Standard (FHS) – 2/3

- `/` La directory radice
- `/bin` Esecuibili importanti per l'utente (es. `cat`, `ls`, `cp`)
- `/sbin` Esecuibili per root (amministratore) (es. `mount`, `ifconfig`)
- `/dev` Dispositivi (es., `/dev/sda`, `/dev/random`, `/dev/audio`)
- `/boot` Kernel
- `/etc` File di configurazione
- `/root` Home dell'amministratore
- `/home` Home degli utenti
- `/lib` Librerie essenziali (per eseguibili in `/bin` e `/sbin`)
- `/media` Montaggio dispositivi rimovibili (DVD, chiavette, usb)
- `/mnt` Montaggio altri filesystem
- `/proc` Informazioni su kernel e processi
- `/tmp` File temporanei
- `/usr` Esecuibili e librerie non essenziali
- `/var` Dati vari (DB, web), log, posta



`/usr/bin` Eseguibili per gli utenti

`/usr/sbin` Eseguibili per
l'amministratore

`/usr/lib` Librerie

`/usr/share` Font, manuali, ...

`/usr/X11R6` L'X Window System

`/usr/local` Programmi installati
manualmente

`/var/mail` La posta degli utenti

`/var/log` I log di sistema

`/var/spool` Dati da processare (per
es. stampe)

`/var/cache` Dati temporanei



Utenti e Permessi

Ad ogni file viene associato un **owner** Il proprietario del file (es. gino)

group Un gruppo di appartenenza (es. utenti)

È possibile specificare permessi di lettura (**r**), scrittura (**w**) ed esecuzione (**x**) per

- Il proprietario del file
- Per utenti del gruppo di appartenenza
- Per tutti gli altri

Esempio

```
$ ls -l /dev/sd*
brw-rw---- 1 root disk 8, 0 24 feb 15:29 /dev/sda
brw-rw---- 1 root disk 8, 1 24 feb 15:29 /dev/sda1
brw-rw---- 1 root disk 8, 5 24 feb 15:29 /dev/sda5
brw-rw---- 1 root disk 8, 6 24 feb 15:29 /dev/sda6
```



1 Filesystem

2 La shell Bash

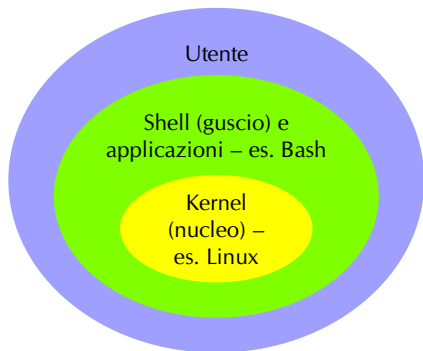
3 Comandi

- Comandi Principali
- Comandi Avanzati



A Cosa Serve una Shell?

- Interagire col kernel
- Gestire file e processi in modo potente e veloce
- Lanciare applicazioni
- Automatizzare operazioni frequenti e/o complesse



Schema a cipolla di un sistema operativo



BASH

- Segue lo standard POSIX
- Disponibile per molti sistemi operativi
- Programmabile, personalizzabile

Login utente

```
spongebob login: cesco
Password:
cesco@spongebob:~$ pwd
/home/cesco
cesco@spongebob:~$ logout
spongebob login:
```

Login root

```
spongebob login: root
Password:
spongebob:~# pwd
/root
spongebob:~# logout
spongebob login:
```



```
$ nome-comando <opzioni> <argomenti>
```

opzioni modificano l'esecuzione del programma

- --nome-opzione
- -o

argomenti obiettivi del comando, di solito nomi di file

Autocompletamento

Premere TAB semplifica molto la vita...

Esempi

```
$ uname
```

```
Linux
```

```
$ uname --machine
```

```
x86_64
```

```
$ uname -m
```

```
x86_64
```

```
$ wc -l /etc/fstab
```

```
16 /etc/fstab
```

Espansioni in linea di comando

Sostituzione comando

```
$ dpkg -S 'which nmtui'  
network-manager: /usr/bin/nmtui
```

* qualsiasi stringa (nulla inclusa)

? carattere qualsiasi

[...] carattere in intervallo

[^...] carattere non in intervallo

Espansione delle graffe

```
$ touch {g,p,tr}ino  
$ ls  
gino  pino  trino
```

Espansione di percorso

```
$ ls ?ino  
gino  pino  
$ ls *ino  
gino  pino  trino  
$ ls [a-m]ino  
gino  
$ ls t*i*  
trino
```

- backslash** per caratteri singoli
- virgolette** permettono di indicare nomi con spazi, ma non bloccano le sostituzioni di comando
- apici** bloccano anche le sostituzioni comando

Esempi

```
$ touch "sale e pepe"
$ touch olio\ e\ aceto
$ ls
olio e aceto  sale e pepe
$ echo "$ (ls)"
olio e aceto
sale e pepe
$ echo '$ (ls)'
$(ls)
```



- Per velocizzare la digitazione di comandi frequenti
- Possono aggiungere opzioni di default ai comandi
- Possono essere aggirati col quoting

Esempi

```
$ alias lc='ls --color=auto'  
$ alias ll='ls -l'  
$ alias mutt='mutt -y'  
$ mutt #==> mutt -y  
$ 'mutt' #==> mutt  
$ unalias ll
```



- Per impostare opzioni per la shell
- Per impostare opzioni predefinite per alcuni programmi

Esempio

```
$ echo $PATH
/usr/local/bin:/usr/bin:/bin:/usr/bin/X11:/usr/
games
$ export PATH=~ /mybin:$PATH
$ echo $PATH
/home/cesco/mybin:/usr/local/bin:/usr/bin:/bin:
/usr/bin/X11:/usr/games
```



~/.bashrc

- Eseguito da shell non di login
- Quello globale è
/etc/bash.bashrc

Esempio

```
alias ls='ls --color=auto'  
alias ll='ls -ctrl'  
alias rm='rm -i'  
. /etc/bash_completion
```

~/.bash_profile

- Eseguito da shell di login
(no terminali)
- Quello globale è
/etc/profile

Esempio

```
. ~/.bashrc  
PATH=~:/bin:"${PATH}"  
umask 077
```



Nel file `~/.bash_history` vengono memorizzati gli ultimi `$HISTSIZE` comandi.

frecche su/giú scorrono l'history

CTRL-R consente la ricerca all'indietro

ALT - . Ripete argomento del comando precedente

Esempio

```
$ less /etc/fstab
$ ...
$ ...
$ <CTRL-R>
(reverse-i-search)'fs': less /etc/fstab
$ ls -l <ALT-.> /etc/fstab
-rw-r--r-- 1 root root 750  8 apr  2009 /etc/fstab
```



Esecuzioni in background

foreground esecuzione in primo piano, blocca la shell

background esecuzione in sfondo, lascia libera la shell; si invoca aggiungendo il carattere & alla fine del comando

Esempio

```
$ firefox &  
[1] 7984  
$ ...  
$ <chiudiamo firefox>  
$ ...  
[1]+  Done      firefox
```



Cambiamento della modalità di esecuzione

Mentre un processo è in esecuzione in foreground si può alterarne l'esecuzione con

CTRL- si chiede al processo di uscire

CTRL-C viene ucciso il processo

CTRL-Z il processo viene sospeso e restituita la shell

Per riattivare un processo:

bg viene mandato in background

fg viene mandato in foreground

Esempio

```
$ mc <CTRL-Z>
[1]+  Stopped      mc
$ vim <CTRL-Z>
[2]+  Stopped      vim
$ fg 1 # si riavvia mc
```



Canali di flusso dei processi

`stdin` canale di input

`stdout` canale di output

`stderr` canale per messaggi
d'errore

Redirezione dei canali

`< nome-file` reindirizza input

`> nome-file` reindirizza output

`2> nome-file` reindirizza stderr

`>> nome-file` aggiunge al file se già
esistente

Esempio

```
$ echo "6*7" > f.txt
$ echo "10+3" >> f.txt
$ cat f.txt
6*7
10+3
$ bc < f.txt
42
13
$ cat /etc/shadow > p 2>&1
$ # = cat /etc/shadow &> p
$ cat p
cat: /etc/shadow: Permissi
on denied
```



Si può anche collegare lo *stdout* di un processo con lo *stdin* di un altro, utilizzando il carattere | e formando una pipeline.

Esempio

```
$ ps -e | grep bash
7191 pts/3 00:00:00 bash
8582 pts/5 00:00:00 bash
8732 pts/5 00:00:00 bash
```

Per combinare comandi:

`com1 ; com2` `com2` è eseguito dopo `com1`

`com1 && com2` idem, ma solo se `com1` non ha dato errori

Esempi

```
$ sleep 20m; sudo halt
$ ./configure && make
```



- Programmi che girano in background (es. sshd, postfix, syslogd, cron)

Gestione dei Servizi

```
# /etc/init.d/ssh status
sshd is running.
# /etc/init.d/ssh stop
Stopping OpenBSD Secure Shell server: sshd.
# vim /etc/ssh/sshd_config
# /etc/init.d/ssh start
Starting OpenBSD Secure Shell server: sshd.
```

Nuova sintassi

```
# service ssh status
# service ssh stop
# service ssh start
```



1 Filesystem

2 La shell Bash

3 Comandi

- Comandi Principali
- Comandi Avanzati



1 Filesystem

2 La shell Bash

- 3 Comandi
- Comandi Principali
 - Comandi Avanzati



echo

- Stampa una stringa nello *stdout*
- Utile negli script e nelle sequenze di comandi

Esempio

```
$ make && echo "Fatto..."
```

touch

- Crea nuovi file vuoti
- Cambia data di accesso o modifica di file esistenti

Esempi

```
$ touch main.cc ; make

# mount -o remount,ro /
# touch me
touch: cannot touch 'me':
Read-only file system
# fsck /
```

cat

- Visualizza uno o piú file sullo *stdout*
- Utile per unire piú file (vedi *split*)
- Visualizza su schermo file piccoli
- Può essere usato anche per scrivere piccoli file

Esempi

```
$ cat /etc/fstab  
$ cat >> TODO.txt
```

less

- Visualizzatore di file
- Usato dal *man*

/ Ricerca nel file

`less -i` Abilita ricerca case-insensitive

Esempio

```
$ dmesg | less
```



head

Visualizza le prime righe di un file

`head -n` Mostra le prime `n` righe

Esempio

```
$ find . | sort | head -3
.
./..abook
./..abook/addressbook
```

tail

- Visualizza le ultime righe di un file
- Utile per seguire file di log

`tail -n` Mostra le ultime `n` righe

`tail -f` Aspetta nuovi caratteri dal file

Esempio

```
# tail -f /var/log/messages
$ tail -f wget-log
```



help, man e info

help

Aiuto sui comandi interni bash

man

Aiuto per programmi e comandi

`man -L en` Richiede il manuale in lingua inglese

Esempi

```
$ help for
$ man -L en vim
$ gunzip -c bash.1.gz | groff
-man > ~/bash-man.ps
```

info

- Manuali in ipertesto
- Spesso piú completi
- Non molto usato

<INVIO> Segue il link

| (L minusc.) Torna indietro

/ Ricerca

Esempi

```
$ info wget
$ info find
```



pwd

Stampa la directory corrente

`pwd -P` stampa directory fisica, ignorando i symlinks

Esempio

```
$ pwd
/home/cesco/mp3
$ pwd -P
/store/audio/mp3
```

cd

Cambia directory

`cd` torna alla home

`cd -` torna alla directory precedente

Esempio

```
$ pwd
/home/cesco/mp3
$ cd /etc
$ pwd
/etc
$ cd -
/home/cesco/mp3
```

Visualizza contenuto directory e informazioni file

`ls -l` visualizza anche informazioni

`ls -R` mostra ricorsivamente sottodirectory

`ls -a` mostra anche file nascosti (.nomefile)

`ls -t` ordina secondo orario

`ls -r` inverti ordine

`ls -d` mostra directory senza entrarci

`ls -h` human-readable, dimensioni leggibili (es. 9,5M)

`ls --color` usa colori

Esempi

```
$ touch {p,g,l}ino
$ mkdir dir{1,2,3}
$ ls -d */
dir1/  dir2/  dir3/
$ ls -ctrl ~/downloads
```



cp

Serve per copiare file o directory

`cp -R` copia directory
ricorsivamente

`cp -p` preserva attributi

`cp -v` elenca file durante la copia

`cp -u` copia solo file piú recenti

mv

Sposta e/o rinomina file

`mv -v` elenca file durante lo
spostamento

`mv -i` chiede conferma prima
di una sovrascrittura

Esempi

```
$ cp -v /media/usbdisk/*.jpg ~/foto
```

```
$ mv vecchio-nome.jpg nuovo-nome.jpg
```



rm

Cancella file e directory

`rm -R` cancella ricorsivamente
le sottodirectory

`rm -i` chiede conferma a ogni
file

`rm -f` forza la rimozione

ln

Crea link fra file, alias per
raggiungerli da diversi percorsi

`ln -s` crea link simbolico (il piú
usato)

`ln -f` cancella un eventuale link
già esistente

Esempio

```
$ rm -R .[^.]*
```

Esempio

```
$ ln -s /store/mp3 ~/mp3
```



chown e chmod

chown

Cambia proprietario e gruppo di un file

chown -R Ricorsivo (si applica ai contenuti delle directory)

chmod

Cambia i permessi

Ottale Permessi espressi in base 8

Mnemonico Per cambiare alcuni permessi

Esempio

```
# touch esempio; ls -l esempio
-rw-r--r-- 1 root root 0 29 mar 15:47 esempio
# chown cesco.cesco esempio; chmod 640 esempio
# ls -l esempio
-rw-r----- 1 cesco cesco 0 29 mar 15:47 esempio
# chmod o+r esempio; ls -l esempio
-rw-r--r-- 1 cesco cesco 0 29 mar 15:52 esempio
```



df

Mostra spazio libero dei dispositivi

df -h human-readable

df -T mostra tipo dei filesystem

du

Mostra spazio usato da file

du -h human-readable

du -s non mostra file ricorsivamente

Esempi

```
$ df -Th
```

Filesystem	Type	Dimens.	Usati	Disp.	Usa%	Montato su
/dev/hda5	ext3	5,5G	2,1G	3,2G	40%	/
/dev/hda6	ext3	3,7G	328M	3,4G	9%	/home
/dev/hda8	ext3	27G	2,1G	24G	9%	/store
tmpfs	tmpfs	118M	0	118M	0%	/dev/shm

```
$ du -sh ~
```

```
264M /home/cesco
```

`ps`

Visualizza i processi attivi

`ps -u user` mostra processi di user

`ps -e` mostra tutti i processi

`ps ax` mostra tutti i processi con linea di comando

Esempio

```
$ ps -u cesco
```

`top`

Programma grafico per gestire i processi attivi

? aiuto

s scegli tempi di aggiornamento

F ordina

u filtra per utente

k manda segnale

r cambia priorità

Esempio

```
$ top
```

```
$ #htop è meglio
```

su

Apri una shell come altro utente (*root* di default)

`su` - Apri una shell di login

Esempi

```
$ su -  
Password:  
# su - postgres  
$ psql
```

nice

Avvia un comando con una certa priorità
priorità numero da -20 (max. priorità) a 20 (minimo)

permessi solo *root* può avviare a priorità negative

`nice -n` Avvia con priorità *n*

`renice n PID` Cambia priorità

Esempio

```
$ nice -15 make  
# nice --10 xcdroast
```

kill e killall

kill

Manda un segnale a un processo

```
kill -s PID Invia segnale s
```

killall

Manda un segnale a tutti i processi con lo stesso nome

```
killall -s nome Invia segnale s
```

Segnali

15 Termina (default)

3 Quit

9 Uccidi

19 Ferma

18 Riattiva

Esempio

```
$ ps -e | grep firefox
4888 ? 00:06:29 firefox
$ kill -15 4888
$ killall -15 firefox
```

1 Filesystem

2 La shell Bash

3 Comandi

- Comandi Principali
- Comandi Avanzati



Crea e gestisce archivi

`tar -c` Crea un archivio

`tar -x` Estrai un archivio

`-f file` Leggi/scrivi su file anziché su stdin/out

`-z` Archivio compresso con gzip

`-v` Elenca file (de)compressi

Esempi

```
$ mkdir roba; touch roba/{p,g}ino
$ tar -czf roba.tar.gz roba
$ rm -R roba; tar -xvzf roba.tar.gz
roba/
roba/pino
roba/gino
```



- t Elenca i contenuti
- r Aggiungi a un archivio
- u Aggiorna un archivio
- j Archivio compresso con bzip2
- p Preserva permessi
- h Segui i link simbolici

Esempi

```
$ touch roba/{c,m}iao
$ gunzip roba.tar.gz
$ tar -rf roba.tar roba/?iao
$ gzip roba.tar

$ tar -xjf pacchetto.tar.bz2
```



split

Divide un file in pezzi piú piccoli
`split file pref` Divide file in
prefaa, prefab, ...

`-b N` Fa pezzi di N byte

Unione

Per riunire i pezzi si usa
semplicemente il `cat`:

```
$ cat pref* > file
```

Esempi

```
$ du -b filone
10485760          filone
$ split -b 3m filone p-
$ du -b p-*
3145728 p-aa
3145728 p-ab
3145728 p-ac
1048576 p-ad
$ cat p-* > unito
$ du -b unito
10485760          unito
$ cmp filone unito
```

Cerca dei file e ci fa qualcosa (default: *stampa*)

`find dir <opzioni>` Cerca file in dir

`-name 'espr'` Cerca file per nome

`-iname 'espr'` Cerca file per nome ignorando (maius/minus)cole

`-type f/d` Cerca solo file/directory

Esempi

```
$ find /store -iname '*venez*.jpg'
```

```
/store/foto/varie/veneziana-01.jpg
```

```
$ find ~ -type d -name 'mutt*'
```

```
/home/cesco/dump/mutt-1.5.9
```



- `-maxdepth n` Scendi ricorsivamente fino al livello n
- `-mindepth n` Scendi ricorsivamente dal livello n
- `-size ±dim` Dimensione (maggi/min)ore di dim
- `-ctime ±num` File modificati da piú/meno di num giorni
- `-perm` Gestisci permessi dei file

Esempi

```
$ find . -maxdepth 1 -size -3k
./file-piccolo.est
$ find . -ctime +30
./oldfile.est
./dir/vecchio.est
```



- `-print0` Stampa separati da carattere *null* (0x00)
- `-delete` Cancella i file
- `-exec` Esegui qualcosa per ogni file

xargs

Prende nomi di file in stdin e costruisce una riga di comando

`xargs -0` Filename separati da *null*

`xargs -r` Se stdin vuoto non eseguire comando

Esempi

```
$ find ~ -name '*~' -delete
$ find . -name '*.eps' -exec epstopdf {} \;
$ find . -type f -perm 755 -print0 | xargs -0r
chmod 644
$ find . -type f -perm 755 -exec chmod 644 {} +
```



- (`espr`) Raggruppa espressioni
- ! `espr` Inverti espressione (NOT)
- `espr1 espr2` AND delle espressioni
- `espr1 -o espr2` OR delle espressioni

Esempi

```
$ find . ! -iname '*.jpg' -ctime -7
$ find . \( -name '*.h' -o -name '*.cc' \) -exec
c wc -l {} + | sort -n
53 ./prog/sim-nonmax/punto.h
69 ./prog/sim-nonmax/test2.cc
92 ./prog/sim-sing/test.cc
214 totale
```



Cerca del testo dentro i file

`grep testo file` Cerca testo in file

`-i` Ignora maiuscole e minuscole

`-l` Stampa solo i nomi dei file

`-Z` Nomi file separati da *null*

Esempi

```
$ find . -name '*.txt' -exec grep -i mucca {} +  
./farm.txt:Mucca, toro, vitello  
$ grep -lZ brutto * | xargs -0r rm
```



Espressioni regolari

(esp) Racchiude
espressione

`^/$` Inizio/fine riga

`.` Carattere qualunque

`e*` da 0 a infinite e

`e{n,m}` e ripetuta da m a n
volte

`e?` e opzionale

Esempi

```
$ D=/usr/share/dict/italian
```

```
$ grep -E '^Lin' $D
```

```
Linus Linux
```

```
$ grep -E 'ucca$' $D
```

```
Lucca mucca parrucca pilucca  
stucca trucca zucca
```

```
$ grep -E '^(vio)?\.ino$' $D
```

```
Gino fino lino pino sino tin  
o vino violino
```

```
$ grep -E '^a.*b.*c.*d.*' $D
```

```
abboccando abbracciando abbr  
acciandola abbracciandolo ab  
dicando
```

Copia di tutto

`if=file` File di input (stdin di default)

`of=file` File di output (stdout di default)

`bs=dim` Dimensione blocchi

`count=c` Numero blocchi

`skip=n` Salta `n` blocchi dall'input

`seek=n` Salta `n` blocchi nell'output

Esempi

```
$ dd if=/dev/sda of=usb.img bs=8M
```

```
$ dd if=/dev/zero of=zero-file bs=1M count=10
```



Scarica file da internet

`wget URL` Scarica l'URL

`-b` Va in background

`-c` Continua un download interrotto

`-i file` Legge le URL da file

`-limit-rate=10k` Limita banda a 10KB/s

`-U Mozilla` Presentati come Mozilla

Esempi

```
$ wget -bc www.sito.it/dir/file.est
```

```
$ tail -f wget-log
```



- r Download ricorsivo
- H Segui link cambiando host
- l15 Livelli di ricorsione (0 = infinito)
- np Non salire di directory
- nc Non ricaricare file già presenti
- k Cambia link per navigazione locale
- p Scarica immagini di ogni pagina html
- A/R es1, ... Accetta/rifiuta file con estensioni es1, ...

Esempi

```
$ wget -r -l0 -np -nc -k -p www.sito.it/par/d/  
$ wget -r -l11 -H -A pdf www.sito.it/par/d/
```



identify

Stampa caratteristiche delle immagini

`-verbose` Stampa *tante* informazioni per ogni file

Esempi

```
$ identify *
openlogo-100.jpg JPEG 100x123 DirectClass 8kb
openlogo-100.png[1] PNG 100x123 100x123+0+0 PseudoClass 33c 2kb
openlogo-75.jpg[2] JPEG 75x92 DirectClass 6kb
openlogo-75.png[3] PNG 75x92 75x92+0+0 PseudoClass 33c 1kb
openlogo.eps[4] PS 110x145 110x145+0+0 DirectClass 47kb
openlogo.pdf[5] PDF 612x792 612x792+0+0 DirectClass 1.4mb
openlogo.xcf[6] XCF 1528x2014 DirectClass 288kb 9.010u 0:11
$ identify -verbose openlogo-100.jpg | wc -l
270
$ identify -verbose openlogo-100.png | wc -l
112
```



mogrify

Modifica una o piú immagini

- resize 50% Dimezza dimensione
- resize 400x400 Ridimensiona entro 400x400 pixel
- resize 400x400! Ridimensiona a 400x400 pixel
- crop 50x20% Scegli una fetta dell'immagine
- gravity center Centro come punto di riferimento
- format type Salva immagine cambiando formato

Esempi

```
$ mogrify -resize 1024x768 sfondo.jpg  
$ mogrify -crop 50% -gravity center img.jpg  
$ mogrify -border 5 -bordercolor red file.jpg
```



convert

Converte/crea immagini

`convert in.png out.jpg` converte formati

`-antialias` Anti-aliasing

`-thumbnail 100x100` Crea thumbnail

`label:stringa` Convertete una stringa in immagine

Esempi

```
$ find . -maxdepth 1 -iname '*.jpg' -exec convert
```

```
-thumbnail 100x100 {} thumbs/{} \;
```

```
$ find . -iname '*.jpg' -exec mogrify
```

```
-format png {} png/ \;
```

```
$ convert -pointsize 50 label:mucca mucca.png
```



import

Salva una finestra come immagine

`-window root` Salva tutto lo schermo

Altri comandi

`display` Visualizza immagini

`composite` Combina immagini

`montage` Affianca immagini

`conjure` Interprete script

Esempi

```
$ import shot.png
```

```
$ display immagine.jpg
```



Rinominare file – Midnight Commander (mc)

È un file-manager in stile Norton Commander.

F5 Copia

F6 Muovi o rinomina

F7 Crea dir

F8 Cancella

ALT+INVIO Incolla
nome-file

(CTRL+)ALT+TAB****
Completion

Maschere

`*.tgz -> *.tar.gz` Gli asterischi si corrispondono

`*.mp3 -> elio-*.mp3` Cambia
ditta.mp3 in elio-ditta.mp3

`*-* -> roma-\2` Cambia img-01.jpg in
roma-01.jpg, ...

`* -> \u*` Cambia roma-01.jpg in
Roma-01.jpg, ...

`*.* -> \L\u*\E.*` Cambia
file-Name.eSt in File-name.eSt



 Brian Fox e Chet Ramey

Manuale Bash

`man bash`

 Autori Vari

Tutte le altre pagine man :-)

```
find /usr/share/man -name '*.[1-9].gz' -exec man -l  
{ } \;
```

