

Guida rapida ai comandi Linux

Andrea Sivieri (tigre@dei.unipd.it)

Versione 1.1.2, 3 Maggio 1999

(documento in formato testo, Andrea Sivieri, tigre@dei.unipd.it - <http://www.magicmill.net/linux/grap/grap.html>)

Versione 1.1.2, 13 Aprile 2002

(documento in formato pdf, Daniele D'Andreti, - <http://www.glugitalia.org>)

Questo documento si propone di aiutare i nuovi utenti Linux a prendere rapidamente confidenza con i comandi del sistema.

1. Introduzione

2. Panoramica dei comandi

- 2.1 Documentazione
- 2.2 Gestione archivi
- 2.3 Gestione file e directory
- 2.4 Gestione processi
- 2.5 Gestione stampa
- 2.6 Gestione testo
- 2.7 Operazioni su file di dati
- 2.8 Informazioni spazio su disco
- 2.9 Informazioni su di noi e sul sistema
- 2.10 Modifica attributi dei file
- 2.11 Operazioni di ricerca
- 2.12 Gestione dischi DOS
- 2.13 Varie

3. Descrizioni lunghe

- 3.1 nome del comando

4. Documentazione

- 4.1 man

5. Gestione archivi

- 5.1 bunzip2
- 5.2 bzip2
- 5.3 gunzip
- 5.4 gzip
- 5.5 tar
- 5.6 unzip
- 5.7 zip

6. Gestione file e directory

- 6.1 cd
- 6.2 cp
- 6.3 ln
- 6.4 ls
- 6.5 mkdir
- 6.6 mv
- 6.7 rm
- 6.8 rmdir

7. Gestione processi

- 7.1 kill
- 7.2 ps

8. Gestione stampa

- 8.1 lpr
- 8.2 lprm
- 8.3 lpq

9. Gestione testo

- 9.1 cat
- 9.2 dos2unix
- 9.3 head
- 9.4 less
- 9.5 more
- 9.6 tail

10. Operazioni su file di dati

- 10.1 cat
- 10.2 head
- 10.3 split
- 10.4 tail

11. Informazioni spazio su disco

- 11.1 df
- 11.2 du

12. Informazioni su di noi e sul sistema

- 12.1 date
- 12.2 finger
- 12.3 ps
- 12.4 pwd
- 12.5 uname
- 12.6 who

13. Modifica attributi dei file

- 13.1 chgrp
- 13.2 chmod
- 13.3 chown

14. Operazioni di ricerca

- 14.1 find
- 14.2 grep

15. Gestione dischi DOS

- 15.1 mcd
- 15.2 mcopy
- 15.3 mdel
- 15.4 mdir
- 15.5 mmd
- 15.6 mrd

16. Varie

- 16.1 exit

17. Indice alfabetico

18. Note legali

- 18.1 Copyright
- 18.2 Distribuzione
- 18.3 Lavori derivati
- 18.4 Attenzione

1. Introduzione

La "Guida rapida ai comandi Linux" vuole essere un valido aiuto per chi da poco si è avvicinato a Linux e per tutti coloro che non sono ancora familiari con i suoi comandi.

Per scrivere questa guida mi sono basato sulle pagine di manuale di sistema, ma ho scelto un'impostazione diversa rispetto ad esse.

Ecco secondo me le principali differenze:

Caratteristiche delle man pages:

- danno una descrizione lunga e analiticamente esaustiva del comando;
- a volte bisogna scorrere schermate e schermate prima di trovare quello che si cerca;
- spesso sono prive di esempi;
- all'inizio l'utente non sa che pagine guardare, non conosce i nomi dei comandi, non sa cosa può fare;
- descrivono decine di opzioni che si usano raramente.

Caratteristiche della "Guida rapida ai comandi Linux":

- spiegazioni brevi,
- non viene trattato l'uso avanzato dei comandi;
- vengono descritte solo le opzioni di uso più frequente;
- vengono forniti esempi, ed altri esempi verranno aggiunti;
- un capitolo della guida presenta una panoramica veloce di tutti i comandi;
- data la brevità si può pensare di leggere tutta la guida o alcuni suoi capitoli mentre non ha senso leggere tutte le man pages.

Chiaramente questa guida non vuole sostituire le man pages, ma affiancarsi ad esse. Appena ci sia bisogno di usare in modo avanzato un certo comando rimane necessario consultare la documentazione completa (man pages, info, ...).

La riga di comando è uno dei punti di forza di Linux, perché in molti casi permette di fare in modo semplice e veloce cose che con un'interfaccia grafica richiederebbero centinaia di clic. D'altronde rappresenta anche uno dei principali scogli che allontanano molti potenziali utenti da Linux.

Spero che questo mio lavoro contribuisca a rendere meno ripida la curva di apprendimento di questo meraviglioso sistema operativo e ne favorisca in questo modo la diffusione.

Quello che ho scritto non pretende in nessun modo di essere privo di errori o definitivo. Anzi invito tutti a mandarmi contributi e correzioni, suggerimenti e critiche.

Un vivo saluto a tutti e buona lettura,

Andrea Sivieri
tigre@dei.unipd.it

2. Panoramica dei comandi

In questo capitolo vengono appena sfiorati tutti i comandi, giusto per farsi un'idea d'insieme e prendere confidenza con i nomi.

Per ogni comando vengono dati il nome, una nota in corsivo per capire l'origine del nome e una breve descrizione della funzione del comando.

2.1 Documentazione

man

manual pages

visualizza pagine di documentazione

2.2 Gestione archivi

bunzip2

contrario di bzip2

scomprime un file del tipo "bz2"

bzip2

block-sorting file compressor, successor of bzip

comprime un file in formato "bz2"

gunzip

GNU unzip

scomprime un file del tipo "gz"

gzip

GNU zip

comprime un file in formato "gz"

split

vedi split in **Operazioni su file di dati**

tar

il nome deriva da "tape archive"

programma per la gestione di archivi di tipo "tar"

unzip

contrario di zip

espande archivi compressi di tipo "zip"

zip

zip

crea archivi compressi in formato "zip"

2.3 Gestione file e directory

cd

change working directory

entra nella directory specificata (cambia directory)

cp

copy files

copia file

ln

link

crea collegamenti a files o directory

ls

list contents of directories

visualizza il contenuto delle directory specificate

mkdir

make directory

crea una directory nuova

mv

move

sposta un file o una directory nel posto specificato

pwd

print working directory

stampa a schermo il nome della directory corrente con l'intero percorso

rm

remove files

cancella i file specificati

rmdir

remove directories

cancella le directory specificate

2.4 Gestione processi

kill

kill process

elimina processi in esecuzione

ps

report process status

stampa a schermo un resoconto dei processi attivi in quel momento

2.5 Gestione stampa

lpr

off line print

Manda file nella coda di stampa

lprm

remove jobs from the line printer spooling queue

Elimina file dalla coda di stampa

lpq

line printer queue

Esamina la coda di stampa

2.6 Gestione testo

cat

vedi cat in **Operazioni su file di dati**.

dos2unix

convert dos files to unix files

converte file di testo DOS in formato UNIX

head

vedi head in **Operazioni su file di dati**.

less

has more functions than "more"

visualizza il file di testo paginandolo

more

stops scrolling and asks for more

visualizza un testo paginandolo (ma non torna indietro)

tail

vedi tail in **Operazioni su file di dati**.

2.7 Operazioni su file di dati

cat

concatenate files and print on the standard output

dà in uscita il contenuto dei file specificati concatenandoli

head

head of the file

fornisce la parte iniziale di un file

split

split a file into pieces

divide un file in parti più piccole

tail

tail of the file

fornisce la parte finale di un file

2.8 Informazioni spazio su disco

df

summarize free disk space

visualizza le statistiche sull'utilizzo degli hard disk (spazio libero...)

du

summarize disk usage

visualizza la quantità di spazio occupato sul disco da file e directory

2.9 Informazioni su di noi e sul sistema

date

show date

fornisce data e ora corrente

finger

finger users

fornisce informazioni su di altri utenti

ps

vedi ps in **Gestione processi**.

uname

print system informations

fornisce informazioni sul sistema operativo e sulla macchina

who

who are you?

fornisce informazioni sugli utenti collegati al sistema

2.10 Modifica attributi dei file

chgrp

change group

cambia il gruppo del file

chmod

change the access permissions of files

cambia gli attributi di un file (i modi di accesso permessi)

chown

change owner

cambia proprietario e/o gruppo del file

2.11 Operazioni di ricerca

find

search for files in a directory hierarchy

cerca file nelle directory specificate

grep

grep pattern = print lines matching a pattern

cerca la stringa specificata nei file indicati

2.12 Gestione dischi DOS

mcd

msdos cd

cambia directory corrente sul disco dos

mcopy

msdos copy

copia file da/verso disco dos

mdel

msdos del

cancella file sul disco dos

mdir

msdos dir

visualizza contenuto disco dos

mmd

msdos md

crea directory sul disco dos

mrdd

msdos rd

cancella directory sul disco dos

2.13 Varie

exit

exit

esce dalla sessione in corso

3. Descrizioni lunghe

Nella seguente sezione viene esaminato il modello che si è deciso di seguire per presentare i diversi comandi nei successivi capitoli. Verrà infatti dedicato un capitolo per approfondire la conoscenza dei comandi di ogni categoria fino ad ora considerata.

3.1 nome del comando

sintassi

In questa posizione della tabella viene indicato qual è lo schema da seguire quando si vuole chiamare un certo comando.

I diversi stili di carattere e le parentesi sono stati utilizzati secondo le seguenti convenzioni:

-> caratteri minuscoli e spazi (esempio: `cat`)

le parti scritte con tali caratteri vanno riportate fedelmente quando si utilizza il comando;

-> caratteri maiuscoli e underscore (esempio: `FILE_ORIGINE`) in questo modo si sono volute indicare le parti della riga di comando che cambiano di volta in volta: le parole scritte in maiuscolo vanno considerate come dei parametri. Nell'esempio considerato al posto di `FILE_ORIGINE` si deve inserire il nome effettivo del file che interessa;

-> segno OR (esempio: `FILE|DIRECTORY`)

indica che l'argomento da inserire può essere indifferentemente del tipo indicato da uno dei parametri separati dal segno OR. Nell'esempio l'argomento può essere di tipo `FILE` oppure `DIRECTORY`;

-> parentesi tonde (esempio: `(FILE|DIRECTORY)`)

queste parentesi sono state inserite in certi casi per unire diverse parti in un unico blocco al quale va applicato poi un certo operatore e in altri con il solo scopo di aumentare la leggibilità. Non vanno mai inserite quando si vuole chiamare un comando;

-> parentesi quadre (esempio: `[FILE]`)

queste parentesi sono state invece usate per indicare quali parti possono essere eventualmente omesse: anche queste parentesi non compariranno nella riga di comando;

-> puntini (esempio: `FILE...`)

indicano che possono essere inseriti un numero arbitrario di elementi del tipo indicato dall'espressione che precede i puntini.

descrizione

Nella descrizione si è cercato di illustrare in che modo viene usato più di frequente il comando considerato.

Nel caso si sia interessati ad un uso più avanzato conviene utilizzare il comando `man` per consultare la documentazione completa disponibile per quel comando.

opzioni

Anche per le opzioni si è scelto di riportare solo quelle di uso più frequente. Questa scelta è stata fatta perché spesso consultando le man pages si rimane disorientati dal gran numero di opzioni e si perde tempo anche per riuscire a fare le cose più stupide. Le opzioni usate tutti i giorni non sono mai molte.

esempi

Gli esempi sono utili per dissipare (si spera) eventuali incertezze che sono rimaste dopo la lettura delle parti precedenti.

4. Documentazione

4.1 man

sintassi

man [SEZIONE] [OPZIONE...] NOME...

descrizione

man formatta e visualizza le eventuali pagine del manuale che riguardano (NOME...). NOME può essere il nome di un comando, il nome di una funzione, il nome di un file di configurazione (solo pochi sono documentati) o altro.

Se SEZIONE viene specificata, man cerca la documentazione soltanto in quella sezione (utile se un certo NOME è presente in più sezioni, nel qual caso la prima sezione consultata da man maschera le altre).

esempi

man cat

visualizza la pagina di manuale di cat

man 3 printf

cerca la documentazione per printf nella sezione 3

5. Gestione archivi

5.1 bunzip2

sintassi

bunzip2 [OPZIONE...] [FILE_BZIP2...]

descrizione

bunzip2 sostituisce ognuno dei file compressi specificati con la versione espansa a cui è stato tolto il suffisso ".bz2".

esempi

bunzip2 molletta.txt.bz2

bunzip2 archivio.tar.bz2

5.2 bzip2

sintassi

bzip2 [OPZIONE...] [FILE...]

descrizione

bzip2 comprime o decomprime file in formato bzip2.

Il comando usato senza opzioni sostituisce ognuno dei file specificati con la versione compressa a cui è stato aggiunto il suffisso ".bz2".

opzioni

-d

Espande i file specificati: è equivalente a bunzip2.

esempi

bzip2 molletta.txt

bzip2 archivio.tar

5.3 gunzip

sintassi

gunzip [OPZIONE...] [FILE_GZIP...]

descrizione

gunzip sostituisce ognuno dei file compressi specificati con la versione espansa a cui è stato tolto il suffisso ".gz".

Nel caso il suffisso originale sia invece ".tgz" il nuovo suffisso usato al posto di quello originale sarà ".tar".

esempi

gunzip molletta.txt.gz

gunzip archivio.tgz

5.4 gzip

sintassi

gzip [OPZIONE...] [FILE...]

descrizione

gzip comprime o decomprime file in formato gzip.

Il comando usato senza opzioni sostituisce ognuno dei file specificati con la versione compressa a cui è stato aggiunto il suffisso ".gz".

opzioni

-d

Espande i file specificati: è equivalente a gunzip.

esempi

gzip molletta.txt

gzip archivio.tar

5.5 tar

sintassi

uso classico:

tar [-] c|r|t|x [v] f NOME_ARCHIVIO [FILE...] [DIRECTORY...]

descrizione

tar gestisce gli archivi di tipo ".tar". Un archivio di questo tipo non è altro che un insieme di file e directory conservati in modo ordinato in un unico file non compresso. In tale file sono presenti tutte le informazioni per ricostruire correttamente la gerarchia di directory originale con tutto il suo contenuto. Generalmente si crea un archivio di questo tipo per ottenere un unico file da comprimere poi con gzip.

opzioni

f NOME_ARCHIVIO

Con questa opzione si specifica il nome dell'archivio da creare (o processare).

v

Opera in modalità verbosa (dice quello che sta facendo).

c

Crea un archivio.

r

Aggiunge file ad un archivio esistente.

t

Mostra il contenuto di un archivio.

x

Estrae i file contenuti in un archivio e ricrea la struttura di directory originale (a partire dalla directory corrente).

z

Permette di comprimere o decomprimere gli archivi senza chiamare esplicitamente gzip. **esempi**

tar cvf prog.tar programmi (*Mette la directory programmi e tutto il suo contenuto in prog.tar.*)

tar zcvf archivio.tgz * (*Mette tutto il contenuto della directory corrente nel file compresso archivio.tgz.*)

tar zxvf archivio.tar.gz (*Estrae il contenuto di archivio.tar.gz nella directory corrente.*)

tar ztvf prog.tgz (*Visualizza il contenuto di prog.tgz.*)

5.6 unzip

sintassi

unzip [OPZIONE...] FILE_ZIP [FILE...]

descrizione

unzip viene principalmente usato per espandere archivi compressi creati con il comando zip oppure con il programma pkzip; dà comunque anche la possibilità di testare tali file o di esaminarne il contenuto.

Se non si specifica altrimenti l'azione effettuata è la decompressione:

il FILE_ZIP (che deve essere un archivio ZIP o PKZIP) viene espanso nella directory corrente.

Se l'argomento [FILE...] viene specificato allora l'azione prescelta viene effettuata soltanto sui file così specificati.

opzioni

-d DIRECTORY

l'archivio viene espanso in DIRECTORY invece che nella directory corrente -l visualizza il nome dei file contenuti nell'archivio

-t

controlla che l'archivio non sia corrotto

-x (XFILE)...

i file indicati con questa opzione non vengono processati

esempi

unzip zippone.zip

espande il file indicato nella directory corrente

5.7 zip

sintassi

zip [OPZIONE...] [FILE_ZIP [FILE...]]

descrizione

zip crea o aggiorna archivi compressi con il formato zip. Gli archivi creati possono essere espansi con unzip oppure con il pkzip.

Il nome dell'archivio che viene processato è indicato con FILE_ZIP, se esso non è già esistente ne viene creato uno vuoto con tale nome. Fatto ciò vengono presi in considerazione i file indicati con l'argomento [FILE...]; essi vengono compressi e aggiunti all'archivio. Nel caso essi siano già presenti nell'archivio, avviene una sostituzione e la versione vecchia viene persa.

opzioni

-r

Prendi in esame anche le sottodirectory in modo ricorsivo. Se viene usata questa opzione [FILE...] può contenere anche il nome di directory.

esempi

zip nuovo_zip.zip *

Crea un archivio contenente i file della directory corrente (non viene preso in considerazione il contenuto delle sottodirectory)

zip -r divina.zip .

Crea un archivio completo del contenuto della directory corrente e di tutte le sottodirectory

6. Gestione file e directory

6.1 cd

sintassi

cd [DIRECTORY]

descrizione

cd entra nella directory specificata.

Se non è fornita alcuna directory allora entra nella home directory dell'utente.

esempi

cd /usr/include

entra nella directory /usr/include

6.2 cp

sintassi

modo1:

cp [OPZIONE...] FILE_ORIGINE... DIRECTORY

modo2:

cp [OPZIONE...] FILE_ORIGINE FILE_DESTINAZIONE

descrizione

modo1:

(si riconosce perchè l'ultimo argomento è una directory esistente)

Copia i file specificati con [FILE_ORIGINE...] nella directory DIRECTORY.

modo2:

Copia il file FILE_ORIGINE nel file FILE_DESTINAZIONE.

opzioni

-b, --backup

crea copie di backup dei file che stanno per essere sovrascritti

-f, --force

non chiede conferma quando l'operazione indicata richiede la sovrascrittura (con conseguente perdita) di file esistenti

-i, --interactive

prima di procedere ad eventuali sovrascritture di file esistenti chiede conferma

-R, --recursive

copia ricorsivamente directory e sottodirectory

-v, --verbose

mentre copia i file visualizza in uscita i loro nomi

esempi

cp bianca.txt neve.txt /home/skywalker

cp claudia.old claudia.new

6.3 ln

sintassi

modo1:

ln [OPZIONE...] FILE_ORIGINE... DIRECTORY

modo2:

ln [OPZIONE...] FILE_ORIGINE [LINK_DA_CREARE]

descrizione

modo1: (l'ultimo argomento è una directory esistente)

Per ogni file specificato crea nella directory **DIRECTORY** un hard link collegato a quel file, avente lo stesso nome del file.

modo2:

Crea un hard link di nome **LINK_DA_CREARE** collegato a **FILE_ORIGINE**. Se non è specificato **LINK_DA_CREARE**, viene creato un link nella directory corrente avente lo stesso nome del file di origine.

opzioni

-b, --backup

effettua una copia di sicurezza di eventuali file che stanno per essere rimossi

-f, --force

effettua eventuali sovrascritture di file esistenti senza fare domande

-i, --interactive

chiedi conferma, prima di procedere con eventuali sovrascritture

-s, --symbolic

viene creato un link simbolico invece che hard

-v, --verbose

elenca i collegamenti che sono stati effettuati

6.4 ls

sintassi

ls [OPZIONE...] [NOME_DIRECTORY...] [NOME_FILE...]

descrizione

ls lista il contenuto di ogni directory indicata in **[NOME_DIRECTORY...]** e i file individuati da **[NOME_FILE...]**.

opzioni

-a, --all

elenca tutti i file compresi quelli che iniziano con **.** (i file nascosti)

-l, --format=long

per ogni voce della lista fornisce anche le seguenti informazioni:

tipo di file, permessi, numero di hard link, nome del possessore, nome del gruppo, dimensione, momento di ultima modifica.

-R, --recursive

discende ricorsivamente anche tutte le sottodirectory

-t, --sort=time

gli elementi della lista vengono elencati in ordine temporale, per primi quelli modificati più di recente

-X, --sort=extension

ordina gli elementi della lista per estensione (estensione=caratteri che seguono l'ultimo punto presente)

6.5 mkdir

sintassi

mkdir [OPZIONE...] NOME_DIRECTORY...

descrizione

mkdir crea le directory specificate con (NOME_DIRECTORY...).

esempi

mkdir pippo

6.6 mv

sintassi

modo1:

mv [OPZIONE...] FILE_ORIGINE... DIRECTORY

modo2:

mv [OPZIONE...] FILE_ORIGINE FILE_DESTINAZIONE

descrizione

modo1: (si riconosce perchè l'ultimo argomento è una directory esistente)

Sposta i file specificati con [FILE_ORIGINE...] nella directory DIRECTORY.

modo2:

Rinomina il file FILE_ORIGINE: il nuovo nome sarà FILE_DESTINAZIONE.

opzioni

-b, --backup

crea copie di backup dei file che stanno per essere mossi -f, --force

non chiede conferma quando l'operazione indicata richiede la sovrascrittura (con conseguente perdita) di file esistenti

-i, --interactive

prima di procedere ad eventuali sovrascritture di file esistenti chiede conferma

-v, --verbose

mentre sposta i file visualizza in uscita i loro nomi

esempi

mv jane.png giungla.png /home/tarzan

mv libro.txt libro.old

6.7 rm

sintassi

rm [OPZIONE...] NOME_FILE...

descrizione

rm rimuove i file specificati, non però le directory.

opzioni

-f, --force

cancella senza fare domande

-i, --interactive

chiede all'utente la conferma di ogni rimozione

-r, -R, --recursive

se viene specificata questa opzione vengono rimosse ricorsivamente anche eventuali directory e il loro contenuto

6.8 rmdir

sintassi

rmdir [OPZIONE...] NOME_DIRECTORY...

descrizione

rmdir rimuove le directory specificate, a patto che siano vuote.

esempi rmdir /home/gianni/varie

7. Gestione processi

7.1 kill

sintassi

kill [-S] PID...

descrizione

kill manda il segnale S ai processi specificati con (PID...) (PID = numero di identificazione del processo (fornito ad esempio da ps)).

Se l'utente non dà indicazioni sul segnale, kill spedisce il segnale TERM, che chiude tutti i processi che non lo intercettano.

S=9 corrisponde al segnale KILL, che non può essere intercettato; questo segnale quindi chiude anche i processi che bloccano il segnale TERM.

esempi

kill -9 277

chiude il processo il cui PID è 277

7.2 ps

sintassi

ps [OPZIONE...]

descrizione

PS fornisce le seguenti informazioni per ogni processo in esecuzione:

PID = numero di identificazione del processo,

TTY = codice di identificazione del terminale,

STAT = stato del processo,

TIME,

COMMAND = riga di comando con cui e' stato fatto partire il processo.

opzioni

a

mostra anche i processi degli altri utenti

X

mostra i processi non controllati da un terminale

8. Gestione stampa

8.1 lpr

sintassi

lpr [OPZIONE...] [NOME_FILE]

descrizione

Il file di ingresso viene mandato nella coda di stampa.

8.2 lprm

sintassi

lprm [OPZIONE...] [-|ID]

descrizione

Se come argomento viene passato -, allora vengono tolti dalla coda di stampa tutti i file che erano stati inseriti dall'utente che ha lanciato lprm.

Si può invece eliminare un file specifico passando semplicemente l'identificativo ID associato a quel file. Gli identificativi in questione si possono ottenere semplicemente utilizzando il comando lpq.

8.3 lpq

sintassi

lpq [OPZIONE...]

descrizione

Mostra la lista dei file sospesi in coda che attendono di essere stampati. Per ognuno dei file vengono riportati anche il nome dell'utente che ha richiesto la stampa e un codice numerico identificativo.

9. Gestione testo

9.1 cat

vedi cat nel capitolo **Operazioni su file di dati**.

9.2 dos2unix

sintassi

```
dos2unix FILE_ORIGINE FILE_DESTINAZIONE
```

descrizione

dos2unix converte un file di testo che segue le convenzioni MSDOS in uno che segue le convenzioni UNIX. **esempi**

```
dos2unix dos.txt unix.txt
```

9.3 head

vedi head nel capitolo **Operazioni su file di dati**.

9.4 less

sintassi

```
less [OPZIONE...] [NOME_FILE...]
```

descrizione

less riporta in uscita gli ingressi specificati una schermata alla volta. E' simile al comando **more**, ma ha più possibilità; ad esempio oltre a poter avanzare, si può anche tornare indietro. Viene generalmente utilizzato quando si vuole velocemente prendere visione di un lungo file di testo oppure in abbinamento con altri comandi, quando essi forniscono in uscita informazioni che non hanno spazio su di una singola schermata.

Sono possibili le seguenti operazioni attivate dalla pressione del tasto indicato all'inizio della riga:

ENTER -> avanza di una riga

BARRA SPAZIATRICE -> avanza di una schermata

B -> torna indietro di una schermata

Q -> esci

/ -> chiede una parola o una espressione regolare su cui fare una ricerca

esempi

```
ls |less
```

questo abbinamento mi permette di esaminare comodamente il contenuto di directory che contengono centinaia di file.

```
less ulisse.txt
```

```
less <ulisse.txt
```

```
cat ulisse.txt |less
```

queste ultime tre forme hanno lo stesso effetto

9.5 more

sintassi

more [OPZIONE...] [NOME_FILE...]

descrizione

more riporta in uscita gli ingressi specificati una schermata alla volta.

La pressione del tasto ENTER permette di avanzare una riga alla volta, la pressione della barra spaziatrice invece fa avanzare di una schermata alla volta. Per uscire prima della fine degli ingressi è sufficiente la pressione del tasto Q o di CTRL-C.

esempi

```
cat divina_commedia.txt |more
```

```
more divina_commedia.txt
```

```
more <divina_commedia.txt
```

Le diverse forme presentate hanno tutte lo stesso effetto.

9.6 tail

vedi tail nel capitolo **Operazioni su file di dati**.

10. Operazioni su file di dati

10.1 cat

sintassi

cat [OPZIONE...] [FILE...]

descrizione

cat concatena i suoi ingressi e li scrive sullo standard output. Se vengono specificati dei file, cat considera questi come ingressi; altrimenti prende lo standard input.

opzioni

-n --number

le righe vengono numerate

-v --show-nonprinting

vengono visualizzati i caratteri di controllo presenti negli ingressi

esempi

cat pippo.txt (*Visualizza il contenuto di pippo.txt*)

cat -n pippo.txt (*come prima, ma le righe vengono numerate*)

cat pippo.txt pluto.txt (*Concatena i file nell'ordine in cui vengono proposti e li visualizza*)

Come posso ricomporre file spezzati in pezzi da split?

Ecco due modi per unire diversi file in uno unico.

modo 1:

```
cat solemare.tgz.1 solemare.tgz.2 > solemare.tgz
```

modo 2:

```
cat solemare.tgz.1 > solemare.tgz
```

```
cat solemare.tgz.2 >> solemare.tgz
```

10.2 head

sintassi

head [OPZIONE...] [FILE...]

descrizione

head fornisce la parte iniziale dei file in ingresso. Se non viene specificato altrimenti l'ingresso viene considerato semplice testo e ne vengono date le prime 10 righe.

opzioni

-c N, --bytes N

Vengono forniti i primi N byte; eventualmente si può cambiare l'unità di misura aggiungendo una delle seguenti lettere:

b = blocchi da 512 byte,

k = blocchi da 1 kilobyte,

m = blocchi da 1 megabyte.

-n N, --lines N

Vengono fornite le prime N righe del file.

esempi

```
head -n 10 Inbox
```

Richiedo le prime 10 righe del file di testo Inbox.

```
head -c 44 noise.wav
```

Richiedo i primi 44 byte del file binario noise.wav.

10.3 split

sintassi

split [OPZIONE...] [FILE [PREFISSO_FILE]]

descrizione

split divide il file FILE in più file di dimensioni inferiori. Eventualmente con PREFISSO_FILE si può specificare il prefisso che verrà utilizzato come parte iniziale del nome dei file creati.

opzioni

-RIGHE, -l RIGHE, -lines=RIGHE

Questa opzione si usa quando si processa un file di testo. Il file viene diviso in parti che contengono un numero di linee di testo pari a RIGHE. L'ultima parte generalmente è più piccola per ovvi motivi.

-b BYTES[b|k|m], -bytes=BYTES[b|k|m]

Questa opzione viene utilizzata quando si vuole dividere un file binario, essa specifica la lunghezza delle parti da creare. La misura della lunghezza è espressa in byte a meno che al numero BYTES non venga aggiunto come suffisso una delle seguenti lettere:

b -> unità=byte

k -> unità=kilobyte

m -> unità=megabyte

esempi

split -b 1400k mega.tgz

Divide il file in parti che possono trovare posto in alcuni dischetti.

10.4 tail

sintassi

tail [OPZIONE...] [FILE...]

descrizione

tail fornisce la parte finale dei file in ingresso. Se non viene specificato altrimenti l'ingresso viene considerato semplice testo e ne vengono date le ultime 10 righe.

opzioni

-c N, --bytes N

Vengono forniti gli ultimi N byte; eventualmente si può cambiare l'unità di misura aggiungendo una delle seguenti lettere:

b = blocchi da 512 byte,

k = blocchi da 1 kilobyte,

m = blocchi da 1 megabyte.

-f

Prova continuamente a leggere nuovi dati dalla fine del file, assumendo che il file si stia ingrandendo.

-n N, --lines N

Vengono fornite le ultime N righe del file.

esempi

tail -n 10 Inbox (*Richiedo le ultime 10 righe del file di testo Inbox.*)

tail -f /var/log/messages

Richiedo le ultime righe di /var/log/messages e di rimanere in attesa dell'eventuale arrivo di nuove righe. Questa opzione è utile in generale per monitorare un file di log in crescita. Come al solito Control-C interrompe il comando.

tail -c 22k noise.wav (*Richiedo gli ultimi 22 kilobyte del file binario noise.wav.*)

11. Informazioni spazio su disco

11.1 df

sintassi

df [OPZIONE...] [NOME_FILE...]

descrizione

Se [NOME_FILE...] non viene dato df visualizza le seguenti informazioni per ogni filesystem che è stato attivato con mount: spazio complessivo, spazio occupato, spazio libero.

Se invece è presente [NOME_FILE...] le stesse informazioni vengono fornite per i filesystem contenenti i file specificati.

opzioni

-k, --kilobytes

Lo spazio viene misurato in blocchi da 1024 byte. Questo è il comportamento di default se la variabile d'ambiente POSIXLY_CORRECT non è stata attivata.

esempi

df -k

fornisce una panoramica dello spazio disponibile misurato in Kbytes.

11.2 du

sintassi

du [OPZIONE...] [(NOME_FILE|NOME_DIRECTORY)...]

descrizione

du visualizza la quantità di spazio occupato da ogni elemento specificato in [(NOME_FILE|NOME_DIRECTORY)...] e dalle sottodirectory di ogni directory specificata. In assenza di indicazioni viene assunto NOME_DIRECTORY="." (cioè mostra quanto spazio occupa la directory corrente e ogni sua sottodirectory).

opzioni

-a, --all

Visualizza anche quanto spazio occupano i file contenuti nelle directory specificate e nelle loro sottodirectory.

-k, --kilobytes

La misura viene espressa in blocchi da 1024 bytes. Questo è il comportamento di default.

-s, --summarize

visualizza lo spazio occupato dalle directory specificate, ma non quanto ne viene occupato dalle loro sottodirectory (in definitiva fornisce un numero solo per ogni argomento specificato).

-c, --total

alla fine fornisce la somma totale dello spazio occupato dagli argomenti specificati

12. Informazioni su di noi e sul sistema

12.1 date

descrizione

date fornisce ora e data corrente.

12.2 finger

sintassi

```
finger [OPZIONE...] [(NOME_DI_LOGIN)[@MACCHINA])...]
```

descrizione

Se l'argomento [@MACCHINA] viene omesso allora viene assunto MACCHINA=localhost.

Se NOME_DI_LOGIN viene specificato allora finger fornisce informazioni su tale particolare utente della MACCHINA, altrimenti fornisce informazioni sugli utenti connessi in questo momento a MACCHINA.

esempi

```
finger mario
```

```
finger mario@luna
```

```
finger @luna
```

12.3 ps

vedi ps nel capitolo **Gestione processi**.

12.4 pwd

descrizione

pwd visualizza il nome della directory corrente con il percorso completo.

12.5 uname

sintassi

```
uname [OPZIONE...]
```

descrizione

uname fornisce informazioni sul sistema operativo e sulla macchina a cui si è collegati. Le opzioni permettono di specificare quali informazioni si desiderano.

L'assenza di opzioni equivale all'opzione -S.

opzioni

-m, --machine *(tipo della macchina)*

-n, --nodename *(nome del nodo della rete associato alla macchina)*

-r, --release *(codice di rilascio del sistema operativo)*

-s, --sysname *(nome del sistema operativo)*

-v *(informazioni sulla versione del sistema operativo)*

-a, --all *(tutte le informazioni)*

esempi

```
uname -a (fornisce informazioni sulla macchina che si sta usando e sul sistema operativo.)
```

12.6 who

sintassi

who [am i] [OPZIONE...]

descrizione

who invocato senza argomenti restituisce in uscita le seguenti informazioni per ogni utente collegato al sistema:

nome di login,

terminale usato dall'utente,

ora e giorno in cui è avvenuto il login,

nome dell'host remoto o codice dell' X display.

La forma **who am i** invece restituisce soltanto le informazioni riguardanti l'utente che ha lanciato il comando.

13. Modifica attributi dei file

13.1 chgrp

sintassi

chgrp [OPZIONE...] GROUP FILE...

descrizione

chgrp permette di modificare il gruppo di appartenenza dei file specificati, mantenendo inalterato il proprietario. GROUP specifica il nuovo gruppo.

esempi

chgrp astronauti nasa.doc

il nuovo gruppo del file nasa.doc sarà astronauti

13.2 chmod

sintassi

sintassi 1: chmod [OPZIONE...] XYZ FILE...

sintassi 2: chmod [OPZIONE...] UGA FILE...

descrizione

chmod cambia i permessi di un file. Specifica cioè le modalità di accesso al file consentite ai diversi utenti.

sintassi 1:

X indica gli utenti nei confronti dei quali voglio modificare gli accessi.

a = tutti (dall'inglese (a)ll)

g = le persone dello stesso gruppo (deriva da (g)roup)

u = l'utente stesso (deriva da (u)ser)

Y indica che azione voglio compiere sugli accessi che intendo modificare.

+ = attivare

- = togliere

Z indica che tipo di accessi voglio modificare.

r = lettura (da (r)ead)

w = scrittura (da (w)rite)

x = esecuzione (da e(x)ecute)

Infine vanno indicati i file di cui si vogliono modificare gli attributi.

sintassi 2:

Gli attributi riguardanti l'utente, il suo gruppo e gli altri utenti possono essere pensati come 3 numeri binari di 3 cifre ciascuno. Le tre cifre rappresentano nell'ordine gli accessi in lettura, scrittura ed esecuzione. Porre a 1 le cifre corrispondenti agli accessi che si vogliono attivare, a 0 le altre. Convertire ora ognuno dei 3 numeri in base 8, si otterranno 3 numeri di una cifra ottale ciascuno.

U -> numero riguardante l'utente

G -> numero riguardante il gruppo dell'utente

A -> numero riguardante tutti

In definitiva specificando le tre cifre ottali UGA in modo opportuno è possibile impostare tutti gli attributi in un colpo solo. Bisogna ovviamente specificare file a cui si vogliono cambiare gli attributi.

esempi

chmod a+r topolino.txt (*do a tutti la possibilità di aprire in lettura il file topolino.txt*)

chmod 644 pippo.txt (*attivo la lettura per tutti, la scrittura solo per me*)

13.3 chown

sintassi

`chown [OPZIONE...] OWNER [:GROUP] FILE...`

descrizione

`chown` permette di modificare il proprietario dei file specificati. Il nuovo proprietario è specificato in **OWNER**. L'argomento opzionale **GROUP** permette eventualmente di indicare anche un nuovo gruppo di appartenenza per il file. Root può usare questo comando senza limiti, gli altri utenti solo sui propri file.

esempi

`chown luca libro.txt`

cedo la proprietà di libro.txt all'utente luca

`chown marco:simpatici scherzi.html` in questo caso il nuovo proprietario sarà marco e il gruppo simpatici (marco appartiene anche a quel gruppo)

14. Operazioni di ricerca

14.1 find

sintassi

Uso elementare:

```
find [(FILE|DIRECTORY)...] [VINCOLO...]
```

Uso avanzato:

```
find [(FILE|DIRECTORY)...] ESPRESSIONE
```

descrizione

Uso elementare:

Vengono dati in uscita i nomi dei file che soddisfano i vincoli specificati.

In assenza di argomenti si assume `DIRECTORY=DIRECTORY_CORRENTE`.

Vengono presi in esame tutti i file e le directory specificati, inoltre in modo ricorsivo si analizza il contenuto delle directory.

Uno dei vincoli usati più di frequente quando si cerca un file è il seguente:

```
-name 'PATTERN'
```

dove `PATTERN` è una stringa composta dai caratteri comunemente usati per riferirsi al nome di un file, compresi i caratteri jolly. Tale vincolo è considerato soddisfatto per ogni file il cui nome è compatibile con `PATTERN`.

Uso avanzato:

Il comando `find` è veramente molto potente e utile, vale quindi la pena di leggere la sua man page per un uso più evoluto.

esempi

```
find /home/caos -name '*.cc'
```

cerca nella directory `/home/caos` tutti i file con estensione `.CC`

14.2 grep

sintassi

```
grep [OPZIONE...] PATTERN [FILE...]
```

descrizione

Uso elementare:

`grep` cerca la stringa definita in `PATTERN` all'interno dei file specificati. Ogni volta che trova un'occorrenza di tale stringa stampa la riga corrispondente per intero.

`PATTERN` può avere le seguenti forme illustrate con esempi:

stringa_semplice

```
'stringa con eventuali spazi'
```

```
-e '-stringa con-eventuali trattini-'
```

Se non viene usata l'ultima forma di `PATTERN` alcuni segni contenuti in esso potrebbero essere interpretati in modo diverso da come vorremmo.

esempi

```
grep cioccolata dispensa.txt
```

```
grep -e 'Il gatto e la volpe' *
```

15. Gestione dischi DOS

15.1 mcd

sintassi

mcd [a:][DIRECTORY_MSDOS]

descrizione

mcd entra nella directory del dischetto MSDOS specificata. Cambia cioè la directory attiva.

esempi

mcd a:pippo

entra nella directory pippo

15.2 mcopy

sintassi

modo1:

mcopy [OPZIONE...] FILE_ORIGINE FILE_DESTINAZIONE

modo2:

mcopy [OPZIONE...] FILE_ORIGINE... DIRECTORY

descrizione

mcopy permette di trasferire dati da/verso dischi MSDOS.

Viene seguita la seguente convenzione: i nomi di file o directory che cominciano con **a:** si riferiscono a dati sul disco MSDOS, tutti gli altri vengono cercati nel filesystem UNIX. Se non vengono date indicazioni diverse la directory attiva sul disco MSDOS è la directory radice.

modo1:

FILE_ORIGINE viene copiato su FILE_DESTINAZIONE.

modo2:

(FILE_ORIGINE...) vengono copiati in DIRECTORY.

Conviene notare i seguenti fatti:

- 1) I nomi dei file vengono memorizzati anche nel formato esteso per i nomi lunghi riconosciuto da Windows, quindi generalmente non si perdono i nomi originali dei file. Ci sono però alcuni caratteri (di uso meno frequente) che non sono consentiti nei nomi di file Windows. In questo caso in nomi originali vengono alterati in modo da essere compatibili.
- 2) Nei nomi lunghi Windows l'informazione su quali lettere siano maiuscole e quali minuscole viene conservata. Però nomi di file diversi soltanto per quanto riguarda maiuscole e minuscole vengono considerati equivalenti. Eventuali conflitti possono essere risolti rinominando qualche file su richiesta di mcopy.
- 3) Per i caratteri jolly vengono seguite le convenzioni UNIX anche quando ci si riferisce a nomi di file MSDOS.

esempi

mcopy * a:

copiati tutti i file della directory corrente nella directory attiva del dischetto MSDOS

mcopy ../mag/clone.bak a:

copiati clone.bak nel dischetto MSDOS

mcopy a:* .

copiati tutti i file contenuti nella directory attiva del dischetto MSDOS nella directory in cui ci troviamo

15.3 mdel

sintassi

mdel [a:][FILE_MSDDOS]

descrizione

mdel cancella i file specificati presenti su di un disco MSDOS.

esempi

mdel a:*.*

cancella tutti i file contenuti nella directory attiva del dischetto

15.4 mdir

sintassi

modo1:

mdir [a:][DIRECTORY_MSDDOS]

modo2:

mdir [a:]FILE_MSDDOS...

descrizione

mdir visualizza i nomi dei file richiesti o il contenuto della directory indicata fornendo anche altre informazioni nello stile del comando MSDOS dir. In mancanza di indicazioni la directory esaminata è la directory precedentemente attivata con il comando mcd, oppure la directory radice del disco se tale comando non è stato usato.

Per ogni file elencato vengono visualizzati nell'ordine:

nome corto tipo MSDOS, lunghezza in byte, data e ora di ultima modifica, nome lungo tipo Windows. Inoltre viene riportato il numero di file contenuti nella directory in esame, lo spazio complessivo da essi utilizzato e lo spazio libero rimanente sul disco.

esempi

mdir

visualizza il contenuto della directory corrente sul disco MSDOS

15.5 mmd

sintassi

mmd [a:]DIRECTORY_MSDDOS...

descrizione

mmd crea le directory specificate nel dischetto MSDOS.

esempi

mmd a:pippo

crea la directory pippo

15.6 mrd

sintassi

mrd [a:]DIRECTORY_MSDDOS...

descrizione

mrd rimuove le directory specificate dal dischetto MSDOS, ma solo se sono vuote.

esempi

mrd a:pippo

cancella la directory pippo

16. Varie

16.1 exit

descrizione

exit chiude la sessione di lavoro.

17. Indice alfabetico

A =	F =	M =	rmdir
B =	find	man	S =
bunzip2	finger	mcd	split
bzip2	G =	mcopy	T =
C =	grep	mdel	tail
cat	gunzip	mdir	tar
cd	gzip	mkdir	U =
chgrp	H =	mmd	uname
chmod	head	more	unzip
chown	J =	mrdd	V =
cp	K =	mv	W =
D =	kill	N =	who
date	L =	O =	X =
df	less	P =	Y =
du	ln	pwd	Z =
dos2unix	lpr	ps	zip
E =	lprm	Q =	
exit	lpq	R =	
	ls	rm	

18. Note legali

18.1 Copyright

Copyright (c) 1998, 1999 Andrea Sivieri. All rights reserved.

L'autore originario della "Guida rapida ai comandi Linux" è Andrea Sivieri.

18.2 Distribuzione

Questo documento può essere riprodotto e distribuito in tutto o in parte, con ogni mezzo fisico o elettronico, finché queste note legali sono mantenute su tutte le copie.

18.3 Lavori derivati

Versioni modificate e/o tradotte di questo documento o lavori che contengono parti modificate e/o tradotte di questo documento devono essere distribuiti sotto gli stessi termini che coprono la distribuzione di questo documento. Inoltre tali lavori devono segnalare di essere lavori derivati dalla "Guida rapida ai comandi Linux" scritta da Andrea Sivieri e portare indicato in maniera ben visibile dove poter trovare l'ultima versione ufficiale della "Guida rapida ai comandi Linux" rilasciata dall'autore originario.

Eccezioni a queste regole possono essere garantite sotto certe condizioni. Contattare l'autore per eventuali domande o per suggerire eventuali modifiche a queste note legali.

18.4 Attenzione

Le informazioni contenute in questa guida sono state raccolte nella speranza che possano essere utili in un qualche modo, ma potrebbero essere errate o incomplete. Quindi nel caso si stia operando con dati importanti consiglio di fare prima delle prove con dati non importanti e comunque di tenere copie di sicurezza.

In nessun caso può essermi attribuita la responsabilità per eventuali danni fisici, morali o di altro tipo conseguenti all'utilizzo della "Guida rapida ai comandi Linux". Declino anche qualsiasi altra responsabilità legata in un qualche modo a questo documento. In definitiva l'uso di questo documento è a proprio rischio e pericolo.

Tutti i marchi citati sono dei rispettivi proprietari.

Ringrazio in anticipo per ogni correzione e suggerimento che gli utilizzatori di questa guida mi vorranno fornire.

Andrea Sivieri
tigre@dei.unipd.it