

COMANDI PRINCIPALI DI LINUX

1) **cambiare la password: passwd**

2) **documentazione : 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

3) Gestione file e directory

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`

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
```

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)

mkdir

sintassi

```
mkdir [OPZIONE...] NOME_DIRECTORY...
```

descrizione

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

esempi

```
mkdir pippo
```

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
```

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`

chide 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

rmdir**sintassi**

```
rmdir [OPZIONE...] NOME_DIRECTORY...
```

descrizione

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

esempi

```
rmdir /home/gianni/varie
```

4) Gestione testo**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.

5) Informazioni spazio su disco**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.

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

6) Informazioni su di noi e sul sistema

date

descrizione

date fornisce ora e data corrente.

finger

sintassi

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

descrizione

Se l'argomento [@MACCHINA] viene omissso 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
```

pwd

descrizione

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

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.

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.

7) Operazioni di ricerca

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`

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' *
```

8) Gestione dischi DOS

mcd

sintassi

```
mcd [a:][DIRECTORY_MSDDOS]
```

descrizione

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

esempi

`mc` `a:pippo`

entra nella directory `pippo`

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 i 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:`

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

`mcopy ../mag/clone.bak a:`

copie `clone.bak` nel dischetto MSDOS

`mcopy a:* .`

copia tutti i file contenuti nella directory attiva del dischetto MSDOS nella directory in qui ci troviamo

mdel

sintassi

```
mdel [a:][FILE_MSDOS]
```

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

mdir

sintassi

modo1:

```
mdir [a:][DIRECTORY_MSDOS]
```

modo2:

```
mdir [a:]FILE_MSDOS...
```

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 `cd`, 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

mmd

sintassi

```
mmd [a:]DIRECTORY_MSDOS...
```

descrizione

`mmd` crea le directory specificate nel dischetto MSDOS.

esempi

```
mmd a:pippo
```

crea la directory `pippo`

mrd

sintassi

```
mrd [a:]DIRECTORY_MSDOS...
```

descrizione

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

esempi

```
mrd a:pippo
```

cancella la directory `pippo`

9) Varie

exit

descrizione

`exit` chiude la sessione di lavoro.

10) 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
```

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.

11) Gestione archivi**bunzip2****sintassi**

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

descrizione

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

esempi

```
bunzip2 molletta.txt.bz2
```

```
bunzip2 archivio.tar.bz2
```

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 qui è stato aggiunto il suffisso ".bz2".

opzioni

```
-d
```

Espande i file specificati: è equivalente a bunzip2.

esempi

```
bzip2 molletta.txt
```

```
bzip2 archivio.tar
```

gunzip**sintassi**

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

descrizione

gunzip sostituisce ognuno dei file compressi specificati con la versione espansa a qui 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
```

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 qui è stato aggiunto il suffisso ".gz".

opzioni

-d

Espande i file specificati: è equivalente a gunzip.

esempi

```
gzip molletta.txt
```

```
gzip archivio.tar
```

tar**sintassi**

uso classico:

```
tar [-] c|x|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.

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

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