

Repositories\Laboratorio-Sistemi-Operativi\Comandi.md

man

`man <term>`

Cerca la documentazione (o manuale) del comando specificato

Ex: `man cd`

cd

`cd [location]`

Cambia la directory corrente a quella specificata

Ex: `cd /some/path`

pwd

`pwd`

Stampa il percorso della directory corrente

Ex: `pwd`

ls

`ls [options] [location]`

Stampa una lista di tutti i file e directory nella directory corrente (se omesso il parametro location)

- **-l**: mostra i dettagli dei file, come la data di modifica, la dimensione, i permessi, ecc.
- **-a**: mostra anche i file nascosti
- **-s**: mostra la dimensione dei file
- **-S**: ordina i file per dimensione

Le opzioni possono essere combinate, ad esempio `-la` mostra i dettagli dei file e anche i file nascosti

Ex: `ls -la`

```
drwxr-xr-x  7  user  group  4096  Oct 23 21:07 file
|---1---|  |-2-|  |-3-|  |-4-|  |-5-|  |---6---|  |---7---|
```

1. Primo carattere: indica il tipo di file

d: directory

-: file

l: link simbolico

I 3 gruppi di 3 caratteri successivi indicano i permessi per il proprietario, per il gruppo e per gli altri.
r = readable, w = writeable, x = executable

2. Numero di link hard

3. Proprietario

4. Gruppo proprietario

5. Dimensione in byte

6. Data di ultima modifica

7. Nome del file

id

id [options] [user]

Stampa informazioni sull'utente (utente corrente se omesso il parametro user)

- **-u**: stampa l'ID dell'utente
- **-un**: stampa il nome dell'utente
- **-g**: stampa l'ID del gruppo
- **-gn**: stampa il nome del gruppo
- **-G**: stampa gli ID dei gruppi a cui appartiene l'utente
- **-Gn**: stampa i nomi dei gruppi a cui appartiene l'utente

Ex: id -un

mkdir

mkdir [options] [directory]

Crea una nuova directory

- **-p**: crea anche le directory intermedie nel percorso specificato

Ex: mkdir -p /some/path

du

`du [options] [location]`

Stampa la dimensione dei file e delle directory in formato di pagine occupate dal file

- **-h**: stampa la dimensione in formato leggibile
- **-s**: stampa la dimensione totale

Ex: `du -hs /some/path`

touch

`touch [options] [file]`

Cambia i timestamp di accesso di un file oppure *crea un nuovo file se non esiste*

Ex: `touch file.txt`

cp

`cp [options] [sources] [destination]`

Copia uno o più file o directory da una posizione a una directory.

- **-r**: copia ricorsivamente una directory

Ex: `cp -r ./file1.txt ./file2.c ./destination/test` Ex: `cp -r ./source ./destination` EX: `cp some-file copied-file`

ln

`ln [source] [destination]`

Crea un link simbolico o fisico da un file o directory a un altro file o directory

Ex: `ln -s /some/path/file.txt /some/other/path/file.txt`

mv

`mv [options] [source] [destination]`

Sposta uno o più file o directory da una posizione a una directory, può essere usato anche per rinominare un file

- **-r**: sposta ricorsivamente una directory

Ex: `mv -r ./file1.txt ./file2.c ./destination/test` Ex: `mv -r ./source ./destination` Ex: `mv old-name new-name`

rmmdir

`rmmdir [options] [directory]`

Rimuove una directory **vuota**

- **-p**: rimuove ricorsivamente tutte le directory sottostanti

Ex: `rmmdir -p /some/path`

rm

`rm [options] [file]`

Rimuove uno o più file o directory

- **-r**: rimuove ricorsivamente una directory e tutti i suoi file
- **-f**: non chiede conferma prima di rimuovere un file
- **-i**: rimozione interattiva, chiede conferma prima di rimuovere un file

Ex: `rm -rf file1.txt some-folder`

alias

`alias [name] [command]`

Crea un alias per un comando, ogni volta che viene eseguito l'alias verrà sostituito con il comando originale

Ex: `alias ls="ls -la"`

unalias

`unalias [name]`

Rimuove un alias

Ex: `unalias ls`

cat

`cat [options] [file]`

La funzione di `cat` è quella di concatenare il contenuto di uno o più file e stamparli in output, viene anche usato per mostrare il contenuto di un file, appropriato solo per file piccoli

- **-n**: mostra i numeri di riga

Ex: `cat file.txt`

less

`less [file]`

Permette di visualizzare il contenuto di un file, adatto per file di grandi dimensioni, permette di navigare all'interno del file usando le frecce direzionali.

Ex: `less file.txt`

nano

`nano [file]`

Editor di testo semplice, permette di modificare il contenuto di un file. Ex: `nano file.txt`

xattr

`xattr [file]`

`xattr -p [attribute-name] [file]`

`xattr -w [attribute-name] [value] [file]`

`xattr -d [attribute-name] [file]`

Permette di visualizzare o modificare gli attributi di un file

- **-l**: mostra i valori insieme ai nomi degli attributi
- **-p**: mostra l'attributo specificato
- **-w**: imposta un attributo
- **-d**: rimuove un attributo

Ex: `xattr -l file.txt` Ex: `xattr -p user.name file.txt` Ex: `xattr -w user.name "John Doe" file.txt` Ex: `xattr -d user.name file.txt`

basename

`basename [path]`

Ritorna il nome del file o directory specificato rimuovendo il percorso assoluto

Ex: `basename /some/path/file.txt -> file.txt`

chmod

`chmod [modes] [file]`

Permette di modificare i permessi di uno o più file I permessi possono essere aggiunti precedendoli con un +, e rimossi con un -, = resetta. I permessi sono r: readable, w: writeable, x: executable. Possono anche essere specificata a quale classe di utente si riferiscono i permessi, o a tutti. Si fa precedendo il +,-,= da:

- **u**: utente
- **g**: gruppo
- **o**: altri
- **a**: tutti Può anche essere usata la notazione octal, dove i primi 3 bit indicano i permessi per l'utente, i secondi 3 per il gruppo e gli ultimi 3 per gli altri, in ordine rwx. Es: 777 = rwxrwxrwx, 644 = rw-r--r-- Opzioni:
- **-N**: Rimuove ACL del file Ex: `chmod u+x file.txt` Ex: `chmod a-x file.txt`

head & tail

`head [options] [file]`

`tail [options] [file]`

Head stampa l'inizio di un file, di default le prime 10 righe Tail stampa la fine di un file, di default le ultime 10 righe

- **-n**: numero di righe da stampare
- **-c**: numero di caratteri da stampare

Ex: `head -n 5 file.txt` Ex: `tail -c 100 file.txt`

sort

`sort [options] [file]`

Ordina le righe di un file, di default in ordine alfabetico

- **-n**: ordina in ordine numerico
- **-r**: ordina in ordine inverso
- **-R**: ordina in modo casuale

Ex: `sort file.txt`

nl

`nl [options] [file]`

Stampa il contenuto di un file con numeri di riga iniziali

Ex `nl file.txt`

WC

`wc [options] [file]`

Conta le righe, parole e caratteri di un file

- **-l**: conta le righe
- **-w**: conta le parole
- **-c**: conta i caratteri

Ex: `wc -l file.txt`

cut

`cut [options] [file]`

Legge riga per riga e stampa la prima parte prima del delimitatore specificato, default a tab.

- **-d [DELIMITATORE]**: delimitatore da usare
- **-f [CAMPO]**: Numero del campo/campi da stampare (separati da virgola)

Ex: `cut -d ";" -f 1,2 file.txt`

uniq

`uniq [options] [file]`

Stampa le righe duplicate di un file.

- **-c**: inserisce anche il numero di occorrenze di ogni riga

- **-d**: stampa solo le righe duplicate
- **-u**: stampa solo le righe uniche

Ex: `uniq file.txt`

diff

`diff [options] [file1] [file2]`

Confronta due file linea per linea e stampa le differenze. Le linee precedute da `<` sono quelle del primo file, quelle precedute da `>` sono quelle del secondo file. Le sezioni sono separate da `---`.

- **-q**: stampa solo se i file sono diversi
- **-s**: stampa solo se i file sono uguali
- **-y**: stampa le differenze in modo side-by-side

Ex: `diff file1.txt file2.txt`

egrep

`egrep [options] [pattern] [file]`

Permette di cercare un pattern in un file, è una versione estesa di `grep` che supporta espressioni regolari. Trova e stampa le righe che contengono il pattern. Il pattern può essere una stringa o un'espressione regolare.

- **-n**: stampa il numero di righe che hanno match

Ex: `egrep "pattern" file.txt`

sed

`sed [options] [pattern] [file]`

Permette di sostituire le occorrenze di un input, sostituendo o eliminando parti di testo. Il pattern è un'espressione regolare ed è della forma:

`s/[regex]/[rimpiazzamento]/[flag]`

dove i flag sono:

- **numero**: sostituisce solo la n-esima occorrenza nella riga, es: `s/abc/ABC/2` (sostituisce solo la seconda occorrenza di abc)

- **g**: sostituisce tutte le occorrenze, es: `s/abc/ABC/g`
- **p**: stampa le righe e le righe che hanno sostituzioni (uno dopo l'altro)
- **-e**: concatena più comandi da eseguire

Ex: `sed "s/pattern/replacement/" file.txt`

ps

ps

Mostra una lista di processi in esecuzione

Ex: `ps`

kill

`kill [options] [pids]`

Termina uno o più processi con il pid specificato

- ****-l**: mostra tutti i segnali (TERM) disponibili
- **-s [TERM]**: invia un segnale al processo
- **-9**: termina il processo, non ignorabile

Ex: `kill 1234`

su

`su [options] [user]`

Permette di cambiare utente, se non specificato viene usato l'utente super user (root).

- **-c [command]**: esegue solo il comando specificato come utente specificato Ex: `su root` Ex: `su someuser -c "echo hello"`

sudo

`sudo [options] [command]`

Permette di eseguire un comando come un altro utente, se non specificato viene usato il super user (root).

- **-u [user]**: esegue il comando come l'utente specificato
- **-s**: esegue il comando con lo shell nella variabile \$SHELL

Ex: `sudo ./script.sh`

which

`which [command]`

Mostra il percorso completo di un comando

Ex: `which ls`

echo

`echo [options] [string]`

Stampa una stringa

Ex: `echo "hello"`