

LABORATORIO POSIX #1

1. In quale sezione del manuale troviamo i formati di file ?

Nei sistemi POSIX, i formati di file si trovano nella sezione 5 del manuale tramite il comando **man 5**.

2. A che cosa serve -p in mkdir(1) ?

L'opzione **-p** in **mkdir** consente di creare directory genitore mancanti e non segnala errore se la directory esiste già.

3. Che cosa fa il comando apropos -s 1 directory | grep current ?

Cerca nei manuali (sezione 1) le descrizioni che contengono il termine **directory** e filtra le righe che contengono **current**.

Spiegazione del comando in generale:

apropos: Questo comando serve per cercare nelle descrizioni delle pagine di manuale di Unix/Linux parole chiave specifiche. Restituisce i nomi delle pagine di manuale e una breve descrizione per tutte le voci che corrispondono al termine cercato.

-s 1: L'opzione **-s** indica ad **apropos** di limitare la ricerca solo a una determinata sezione del manuale. Qui, **-s 1** limita la ricerca alla **sezione 1**, che contiene i comandi utente (come **ls**, **cd**, **mkdir**, ecc.). Questo evita che vengano mostrate voci con lo stesso nome provenienti da sezioni diverse (ad esempio, **open(1)** per aprire un file e **open(2)** per le chiamate di sistema relative ai file).

directory: È il termine di ricerca. **apropos** restituirà tutte le voci della sezione 1 del manuale che contengono la parola "directory" nella descrizione o nel titolo.

| grep current: La pipe **|** permette di prendere l'output del comando precedente (**apropos -s 1 directory**) e di passarlo come input al comando **grep current**.

grep current: Filtra l'output ricevuto dalla ricerca di **apropos** per visualizzare solo le righe che contengono la parola "**current**". Quindi, se una delle descrizioni delle voci di manuale trovate da **apropos** contiene anche la parola "current", quella riga verrà mostrata come risultato.

4. Come faccio a “stampare” il nome della directory di lavoro ?

Usando il comando **pwd**.

5. Come posso vedere se un comando, per esempio cd, è interno (built-in) o esterno ?

Utilizzando **type cd** per sapere se **cd** è un comando built-in o esterno.

6. Come faccio, con il comando ls, a elencare anche i file che iniziano con . (il carattere punto) ?

Usando **ls -a** per mostrare anche i file nascosti (quelli che iniziano con “.”)

7. Cosa cambia se uso -h con ls ?

ls -h visualizza le dimensioni dei file in un formato leggibile (KB, MB) anziché in byte.

8. È possibile creare una directory il cui nome contenga degli spazi ?

Sì, è possibile. Dobbiamo mettere il nome della directory tra virgolette o usare il carattere di escape \ per i singoli spazi:

```
mkdir "Nuova directory"
```

oppure

```
mkdir Nuove\ directory
```

9. Dove porta il comando `cd` senza argomenti? Come potete fare in modo che vi porti da un'altra parte ?

cd senza argomenti ci porta nella home directory. Se vogliamo che ci porti da un'altra parte, dobbiamo specificare un percorso:

```
cd /percorso/directory
```

10. Che differenza c'è fra cd e pushd ?

cd cambia directory. **pushd** cambia directory e aggiunge la directory precedente allo stack, permettendo di tornare indietro facilmente con **popd**.

11. A cosa serve which ? Qual è il suo exit-status ?

which mostra il percorso di un comando. L'exit-status è 0 se il comando è trovato, 1 se non lo è.

12. Create un alias (usando alias) per elencare i file con estensione .c

Usiamo **alias cfiles='ls *.c'** per creare un alias che elenca i file .c

13. Stampate le prime 3 linee del file `/etc/group`, usando head(1). Poi, stampatene le ultime 3, usando tail(1).

- Prime 3 righe: `head -n 3 /etc/group`
- Ultime 3 righe: `tail -n 3 /etc/group`

14. Scrivete un comando per contare quanti utenti hanno `/bin/false` come shell.

Si usa:

```
grep '/bin/false' /etc/passwd | wc -l.
```

15. Elencate, senza duplicazioni, le shell specificate in passwd.

Si usa:

```
cut -d: -f7 /etc/passwd | sort | uniq.
```

16. Elencate tutti gli utenti del sistema, in ordine alfabetico inverso.

Si usa:

```
cut -d: -f1 /etc/passwd | sort -r.
```

17. Elencate i PID (e nient'altro) di tutti i processi del sistema.

Si usa:

```
ps -e -o pid=, e per contarli ps -e -o pid= | wc -l.
```

18. Dire cosa fanno i seguenti comandi:

- **top**: Visualizza i processi in esecuzione in tempo reale.
- **pstree**: Mostra la struttura dei processi in forma di albero.
- **xdg-open .** : Apre la directory corrente nel file manager di sistema.
- **find ~ -type d -empty -ls** : Trova e mostra tutte le directory vuote nella home.

19. Scrivete, compilate e testate un programma C che esce con exit-status 42.

Codice:

```
#include <stdlib.h>
int main() {
    return 42;
}
```

Compilazione: gcc -o 19 19.c

Esecuzione: ./19

Controllo dell'exit-status: echo \$? (dovrebbe stampare 42).