

Laboratorio di Sistemi di Elaborazione e Trasmissione dell'Informazione (SETI)

POSIX Intro #1

16 ottobre 2021

Lo scopo di questo laboratorio è familiarizzare con l'uso della riga di comando e abituarsi a consultare la documentazione in un sistema POSIX. L'idea è “costringervi” a leggersi un po' di pagine di manuale e fare qualche prova.

Non abbiate paura di sperimentare, anzi! È molto meglio rispondere a poche domande, ma avendo capito esattamente il perché (e, sperabilmente, avendo imparato qualche comando/switch nuovo lungo la strada), rispetto all'arrivare subito in fondo avendo un'idea vaga del tipo “ma sì, più o meno...”.

Vi ricordiamo che potete consultare la documentazione di un comando built-in della *bash* con `help`, per esempio `help type`, e dei comandi esterni con `man`; per esempio, `man ls` o `man 1 ls`. Specificare il numero di sezione è utile per le situazioni in cui lo stesso nome è definito in più sezioni diverse, per esempio `open(1)` e `open(2)`.

Questo è un laboratorio “di riscaldamento” e il suo svolgimento NON deve essere consegnato.

Vi consigliamo di svolgere le domande/esercizi di questo laboratorio nell'ordine proposto, poiché dovrebbero essere in ordine di difficoltà crescente e, a volte, nel rispondere a una domanda potreste scoprire (se già non le sapete) informazioni importanti per affrontare quelle successive.

Domande/esercizi

bash

1. In quale sezione del manuale troviamo i formati di file?
2. A che cosa serve `-p` in `mkdir(1)` ?
3. Che cosa fa il comando `apropos -s 1 directory | grep current` ?
Per i pigri: <http://explainshell.com/explain?cmd=apropos+-s+1+directory+%7C+grep+current>
Nota: su alcuni sistemi (Mac OS X?) potrebbe non essere presente l'opzione `-s`
4. Come faccio a “stampare” il nome della directory di lavoro?
5. Come posso vedere se un comando, per esempio `cd`, è interno (*built-in*) o esterno? (Suggerimento: leggete bene il testo, il comando per farlo è già stato nominato ☺)
6. Come faccio, con il comando `ls`, a elencare anche i file che iniziano con `.` (il carattere punto)?
7. Cosa cambia se uso `-h` con `ls`?
8. È possibile creare una directory il cui nome contenga degli spazi?
9. Dove porta il comando `cd` senza argomenti? Come potete fare in modo che vi porti da un'altra parte? Potrebbe servirvi `pwd` per vedere la differenza
10. Che differenza c'è fra `cd` e `pushd`?
11. A cosa serve `which`? Qual è il suo *exit-status*? (l'*exit-status* dell'ultimo comando viene memorizzato nella variabile `$_`, quindi potete vederlo con `echo $_`)
12. Create un alias (usando `alias`) per elencare i file con estensione `.c`; ricordatevi che la shell separa il suo input dove trova degli spazi ed espande le wildcard...
13. Stampate le prime 3 linee del file `/etc/group`, usando `head(1)`. Poi, stampatene le ultime 3, usando `tail(1)`

14. Scrivete un comando per contare quanti utenti hanno `/bin/false` come shell.
- per sapere dove guardare, `passwd(5)`
 - per contare le linee che contengono un certo *pattern*, `grep(1)`
 - per estrarre, da ogni linea, solo il campo “optional user command interpreter” (cioè la shell dell’utente), `cut(1)`
15. Elencate, senza duplicazioni, le shell specificate in `passwd`; vi servirà `uniq(1)` ma non solo: leggete bene cosa fa `uniq` e che altro vi suggerisce di guardare la sua man-page
16. Elencate tutti gli utenti del sistema, in ordine alfabetico inverso
17. Elencate i PID (e nient’altro) di tutti i processi del sistema (sono *tanti*, se ve ne escono meno di cento c’è probabilmente qualcosa che non quadra); vedete `ps(1)`
- Assicuratevi che `ps` scriva solo i PID, senza nessuna intestazione
 - Scrivete un comando per contare quanti sono; vedete `wc(1)`
18. Dire cosa fanno i seguenti comandi:
- `top`; se installato (non lo è di default su molte distribuzioni), potete anche provare `htop`
 - `pstree`
 - `xdg-open .`
 - `find ~ -type d -empty -ls`

C

Scrivete, compilate e testate un programma C che...

19. esce con exit-status 42; potete usare, indifferentemente, `return` dal `main` o la funzione `exit(3)` (dove il numero 3 indica la sezione del manuale, ovviamente). Verificate che il programma funzioni visualizzando, dopo averlo lanciato, il valore della variabile `$?` dalla shell