

Documentazione in Linux

- Mi serve qualcosa che faccia vedere il contenuto di un file senza modificarlo. Che comando uso?
- A che serve il comando *mkbootdisk* ?
- Quando uso *ls* non mi compaiono i files nascosti, come faccio a vederli?
- Mi hanno detto che esiste un meraviglioso comando che dice a Linux di trovare una parola in tutti i file di una cartella; come faccio a sapere che comando è e come si usa?
- Ho avuto accesso al codice sorgente di un programma di un guru della programmazione che va linkato con *-lXXX*. Ci ho provato ma il compilatore mi dice che non trova *libXXX*. Che devo fare?

apropos e man

Se sappiamo cosa vogliamo fare, ad esempio vedere un file, una possibilità è usare il comando *apropos* seguito dalla parola magica. Si può provare *apropos file* o *apropos view*. Otterremo una lista di comandi con una breve descrizione di ciò che fanno. Possiamo quindi vedere se uno di questi fa al caso nostro. Altrimenti cambiate la parolina dopo *apropos*

Supponendo di aver trovato quello che cerchiamo (ad esempio il comando *less*) c'è ancora bisogno di capire come va usato. A questo proposito si possono avere ulteriori informazioni da *man less*

Le opzioni dei comandi sono di solito precedute da - (meno) e da una o due lettere nella forma breve, oppure da – (meno meno) e una descrizione nella forma lunga (esempi: *ls -a* e *grep –files-with-matches*)

Aiuto! la pagina man dice che non è più aggiornata e che bisogna guardare la pagina info. Cosa sarà mai?

info

Le pagine info sono nate dal tentativo di documentare i programmi in forma di ipertesto quando ancora non esisteva la grafica

L'uso di info è

[info](#) oppure [info programma](#) (esempio: [info gcc](#))

Per navigare nelle pagine info i comandi più utili sono

spazio	schermata successiva
delete	schermata precedente
n	nodo successivo
p	nodo precedente
u	nodo superiore
b	inizio del nodo
e	fine del nodo
/	cerca
q	esci da info

Anche i tasti cursore e quelli che spostano di una schermata possono essere usati. I punti segnati da asterischi sono link che si possono seguire con [invio](#).

Come semplificarsi la vita

info e *man* possono essere consultati più facilmente con il browser web *konqueror*. basta sostituire nella riga dell'URL, ad esempio, *man:/grep* oppure *info:/gsl-ref* per avere una serie di pagine facilmente scorribili e navigabili, senza conoscere in dettaglio tutte le istruzioni precedenti.

Per chi usa GNOME l'applicazione da usare è *yelp* (o *gnome-help*) con la seguente sintassi
yelp man:ls e *yelp info:gsl-ref*

Librerie preconfezionate

Cosa sono:

- pezzi di programma già scritto da altri; di solito ben collaudati;
- comprendono le definizioni delle funzioni e il codice eseguibile;
- per librerie di pubblico dominio è disponibile anche il sorgente;

Cosa richiedono:

- inclusione di un file d'intestazione. Se la libreria ci chiama *gsl* si includerà *gsl.h* all'interno del file sorgente;
- link con opportune librerie extra *-lgsl*. il corrispondente file si chiamerà *libgsl.la* o *libgsl.so*
- *Importante*: lettura molto attenta della documentazione.

Un esempio: le librerie gsl

- GNU Scientific Library (GNU's Not Unix);
- includono *gsl.h* o file con nomi simili;
- link sempre *-lgsl* e spesso *-lgslcblas*.

Contengono anche molto generatori di numeri pseudocasuali per i quali bisogna includere anche *gsl/gsl.h* o altri file d'intestazione.

Ma perché allora ho seguito questo corso?

- per guidare l'auto bisogna saper almeno cambiare una ruota;
- un programma che metta insieme cose sconosciute è facilmente sbagliato;
- Un algoritmo standard non va bene per la maggior parte delle applicazioni; i risultati originali hanno bisogno di programmi originali

Programmi con le librerie gsl

- costanti matematiche

`M_PI = π`

- funzioni elementari e funzioni speciali

`hypot(3, 4)`

`gsl_pow_4(2)`

- valutazione di polinomi

$p(x) = c_0 + c_1x + \dots + c_{n-1}x^{n-1}$

`gsl_poly_eval(c, n, x)`

- ordinamento di un vettore

`gsl_sort(v, k, n)`

`v(0), v(k), v(2k), ...`

per un vettore di n elementi

In pratica

Includo i file header

```
#include < gsl/gsl_XXX.h >  
XXX = math, poly, sort_vector
```

Linking

```
gcc -lgsl -lgslcblas programma.c
```

BLAS = **B**asic **L**inear **A**lgebra **S**ubroutines

Documentazione

Mi documento con info gsl guardando soprattutto gli esempi.