

Corsi Linux 2018: Kit di sopravvivenza per studenti

Fondamenti del terminale

lorenzo.ribis@gmail.com 1vercesig@gmail.com



POLITECNICO OPEN
unix LABS

Sezione 1

Introduzione

Basi

Comandi base

Comandi avanzati

I/O

Perchè il terminale ?

- Universale: è presente in ogni distribuzione GNU/Linux
- Si può usare da remoto e/o su una macchina senza schermo
- Consente di automatizzare compiti ripetitivi

Fate demande



Sezione 2

Introduzione

Basi

Comandi base

Comandi avanzati

I/O

Come si scrivono i comandi

comando [opzioni] [argomenti]

- Tutto ciò che è tra parentesi quadre è facoltativo
- le opzioni si passano con uno o due trattini (- o --)

ESEMPIO: $\underbrace{\text{ffmpeg}}_{\text{comando}} \underbrace{-i}_{\text{opzione 1}} \underbrace{\text{"video.avi"}}_{\text{argomento 1}} \underbrace{-c : v \text{ HEVC}}_{\text{opzione 2}} \underbrace{\text{"video.mkv"}}_{\text{argomento 2}}$

Muoversi nel terminale

- **cd** <nomecartella>
 - spostarsi nella cartella <nomecartella>
- **ls**
 - lista i contenuti della cartella corrente
- **:**
 - riferimento rispetto alla cartella padre
- **.**
 - riferimento alla cartella corrente
- **~**
 - riferimento alla cartella home

Shortcut utili

- **TAB**
 - permette di autocompletare i comandi
- **Ctrl + C**
 - invia un segnale di arresto ai comandi
- **SHIFT + CTRL + C / SHIFT + CTRL + V**
 - copia/incolla sul terminale
- **↑ (FRECCIA SU)**
 - permette di richiamare i comandi precedenti

Sezione 3

Introduzione

Basi

Comandi base

Comandi avanzati

I/O

Man

man permette di leggere la documentazione di un **comando** passato come **argomento**

- Per cercare all'interno di man basta digitare */termine_cercato* e premere *Enter*, *n* permette di passare al risultato successivo
- premo *p* si torna all'inizio
- con *q* si esce dal manuale

Creiamo cose

touch nomefile crea un file vuoto con il nome indicato, se esiste già ne aggiorna la data e l'ora di accesso

mkdir nomecartella crea una cartella con il nome indicato

- -p crea tutte le sottodirectory necessarie

RICORDATE il filesystem Linux, a differenza di quello di Windows è case sensitive, per esempio `foo` \neq `Foo` \neq `FOO`

Cancelliamo cose

rm nomefile cancella un file

- -r se il file è una cartella ne cancella ricorsivamente il contenuto

rmdir nomecartella cancella una cartella solo se vuota

Muoviamo cose

cp nomefile1 nomefile2 copia il file1 nella posizione del file 2

mv nomefile 1 nomefile2 muove il file1 nella posizione del file2

**CONGRATULAZIONI! SIETE DIVENTATI
UNA DIVINITÀ INDIANA**

Sezione 4

Introduzione

Basi

Comandi base

Comandi avanzati

I/O

Comandi avanzati

cat nomefile concatena e stampa a schermo il file

less nomefile stampa il file a schermo con una finestra scorrevole

echo stringa stampa a schermo la stringa

find directory -name nomefile cerca dei file all'interno della di una gerarchia di cartelle, eventualmente con dei parametri di ricerca

Sezione 5

Introduzione

Basi

Comandi base

Comandi avanzati

I/O

Standard streams

stdin (0) il canale che riceve l'input, di default riceve ciò che l'utente scrive sull terminale

stdout (1) il canale che stampa l'output del programma di default su schermo

stderr (2) un canale per segnalare errori senza mischiarli con l'output, di default scrive su schermo

Ridirezione dei file

comando < **nomefile** connette lo *stdin* di un processo ad un file

comando > **nomefile** connette lo *stdout* di un processo ad un file, se il file esiste viene cancellato e sovrascritto

- usando invece >> il risultato viene aggiunto al file se esistente

comando 2> **nomefile** connette *stderr* di un processo ad un file

comando %>**nomefile** connette *stderr* e *stdout* di un processo ad un file

comando1 | **comando2** lo *stdout* di **comando1** diventa lo *stdin* di **comando2**

comando1 && **comando2** esegue **comando2** se **comando1** ha successo (codice di ritorno ==0)

comando1 || **comando2** esegue **comando 2** se e solo se **comando1** fallisce (codice di ritorno ≠0)

Filtri

Un filtro è un programma in grado di ricevere dati in ingresso sullo *stdin*, trasformare questi dati e produrre dell'output sullo *stdout*, l'output può essere una versione trasformata dell'input oppure altro

- **GREP** mostra le righe di un testo in base ad una ricerca
- **CUT** tool per la selezione di dati tabulati

Fine

Grazie per l'attenzione!



Queste slides sono licenziate Creative Commons Attribution-ShareAlike 4.0

<http://www.poul.org>