

UNIVERSITÀ | Dipartimento DEGLI STUDI | di Ingegneria Gestio DI BERGAMO | dell'Informazione e di

Panoramica su Linux

Introduzione alla programmazione della shell Bash

Patrizia Scandurra Marco Abbadini Andrea Bombarda



Outline

Parte I

- Breve storia di Linux, GNU, GNU/Linux
- Perché Linux? Perché imparare ad utilizzare la Shell?
- Distribuzioni Linux (quali sono, differenze principali, ...)
- Set up e configurazione della macchina virtuale Linux con VirtualBox
- Introduzione ai comandi della shell

Parte II

• Programmazione Bash (shell scripting)



Breve Storia di Linux, GNU, GNU/Linux



Un po' di storia

Nella storia di Linux ci sono due persone da ricordare



Richard Stallman

Linus Torvalds





Richard Stallman

<u>Richard Stallman</u> è uno degli Hacker più talentuosi del MIT che aveva, come aspirazione, quella di combattere la progressiva chiusura del codice sorgente a cui si stava assistendo.

In linea con questo suo pensiero, Richard Stallman:

- Annuncia nel 1983 l'intenzione di creare il SO GNU
- Rilascia nel 1985 il manifesto del progetto GNU
- Crea la Free Software Foundation nel 1985
- Rilascia nel 1989 la prima versione della licenza GPL



Licenza GPL

La GPL è una licenza open source che garantisce:

- Libertà di eseguire il programma, come si preferisce
- Libertà di studiarne il codice sorgente e modificarlo
- Libertà di fare copie e distribuirle agli altri
- Libertà di pubblicare altre versioni modificate



di Ingegneria Gestionale, dell'Informazione e della Produzione

Linus Torvalds

Linus Torvalds è uno studente universitario finlandese che voleva accedere al suo account universitario da casa.

Il problema che voleva risolvere era che per accedere al suo account universitario doveva usare il suo PC ma Minix, il SO disponibile all'epoca (<u>https://it.wikipedia.org/wiki/Minix</u>) non funzionava bene sul suo PC 386.

Per risolvere questo problema decise di *scrivere un terminale indipendente da Minix*.



Linus Torvalds

Perciò nel 1991, dopo diversi lavori, inviò questa mail che viene considerata come l'atto di nascita di Linux:

From: torvalds@klaava.Helsinki.FI (Linus Benedict
Torvalds)

[...]

Hello everybody out there using minix -I'm doing a (free) operating system (just a hobby, won't be big and professional like gnu) for 386(486) AT clones. [...]



Caratteristiche iniziali di Linux

All'inizio della storia di Linux:

- Il codice sorgente era completamente disponibile
- Era possibile modificarne il codice
- La sua licenza non imponeva alcuna limitazione nell'utilizzo

Al di fuori del terminale, però, mancavano tutti gli elementi che ne fanno da contorno e consentono ad un sistema operativo di essere utilizzato da tutti. La soluzione a questo problema arriva con l'integrazione con il <u>Progetto</u>



GNU.



Progetto GNU (acronimo ricorsivo di "GNU's Not Unix")

GNU era di fatto un sistema già esistente dal 1983, realizzato da Richard Stallman, al fine di avere un sistema Unix-like con licenza <u>aperta</u>.

Quando Linux venne inventato, GNU aveva

- Compilatori
- Debugger
- Editor
- Core utils

Ma mancava di un kernel!!











Perchè imparare Linux?



di Ingegneria Gestionale, DI BERGAMO dell'Informazione e della Produzione

Perchè imparare Linux?

E' importante imparare Linux perché:

- E' un SO utilizzato da moltissimi dispositivi (ad esempio, Android, condivide molto della struttura di Linux)
- E' il SO più utilizzato in ambito server
- Può succedere spesso che lo sviluppatore utilizzi un differente sistema operativo, ma poi (dato che spesso un SW va in produzione su un server) l'ambiente di deployment è Linux



Perchè è importante imparare ad utilizzare la Console?



Perchè è importante imparare ad utilizzare la Console?

Lavorando con Linux, è fondamentale saper utilizzare la Shell perché:

- Consente di automatizzare compiti ripetitivi
- Permette di creare «programmi» complessi senza scrivere del vero e proprio codice
- I server, spesso, non hanno una interfaccia grafica
- E' diventato lo standard de-facto per interfacciarsi con servizi cloud





Distribuzioni Linux



di Ingegneria Gestionale, DI BERGAMO dell'Informazione e della Produzione

GNU/Linux e distribuzioni

L'insieme del Kernel (Linux) ed una serie di programmi viene distribuito (solitamente scaricabile gratuitamente) in diverse distribuzioni (anche dette «distro»).





Esempi di distribuzioni Linux

Alcune principali distribuzioni di Linux sono:

- Ubuntu, Kubuntu, <u>Xubuntu</u>, Lubuntu, EdUbuntu, ...
- Debian
- Gentoo
- Centos, Fedora, Red Hat Enterprise Linux
- OpenSUSE, SISE
- Slackware
- Arch Linux

... Ma ne esistono centinaia di altre versioni...



Differenze tra le varie distro

Le varie distribuzioni si differenziano l'un l'altra per:

- Tool di configurazione
- Gestione del software

formazione e della Produzione

- Modalità di rilascio
- Posizionamento dei file di configurazione
- Desktop environment (GNU, KDE, <u>Xfce</u>, ...)



Tool di configurazione

Tutto in Linux/UNIX è configurabile tramite appositi file di configurazione:

- Periferiche (mouse, scanner, stampanti, ...)
- Programmi (web server, db server, application server, ...)
- Avvio automatico di servizi e programmi

Tra le varie distribuzioni, ma anche all'interno della stessa distribuzione, questi file possono essere editati per cambiare la configurazione:

- In modo manuale
- Utilizzando dei tool guidati



Posizionamento dei file

Può succedere che alcuni file/directory si trovino in posizioni diverse tra diverse distribuzioni.

Nonostante questo è molto difficile che il contenuto e l'organizzazione di questi file cambi.



Set Up della macchina virtuale



Dove trovare macchine virtuali

Potete creare in autonomia macchine virtuali utilizzando una ISO (o CD/DVD) di installazione già a vostra disposizione.

In alternativa, su <u>https://www.osboxes.org/virtualbox-images/</u> trovate una lista di file di macchine virtuali già pronte che potete scaricare ed installare.

Utilizzeremo VirtualBox come software per l'esecuzione della macchina virtuale (<u>https://www.virtualbox.org</u>).



Cosa è una macchina virtuale

Con *Macchina Virtuale* intendiamo un software che, attraverso un processo di virtualizzazione, crea un ambiente virtuale che emula tipicamente comportamento di una macchina fisica (PC, client o server) grazie all'assegnazione di risorse hardware (porzioni di disco rigido, RAM e CPU) ed in cui alcune applicazioni (o un vero e proprio sistema operativo) possono essere eseguite come se interagissero con tale macchina





Set Up della macchina virtuale (1)

Al link seguente troverete una macchina virtuale con Xubuntu già installato e configurato:

(<u>https://drive.google.com/file/d/1vNhVeeSwduC9vcguuXQg8-5srUVkVvgR/view?usp=sharing</u>)

Quello che dovete fare è importarla all'interno di Virtual Box ed impostare correttamente la cartella condivisa tra sistema ospitato ed ospitante (potete utilizzarla ad esempio per passare i file sulla macchina virtuale). Di seguito le istruzioni per fare il tutto.



Set Up della macchina virtuale (2)

Per prima cosa dobbiamo importare la macchina virtuale che avete scaricato, all'interno di Virtual Box. Cliccate quindi su «Importa»





Set Up della macchina virtuale (3)

Applicazione virtuale da importare

Selezionate la macchina virtuale che avete scaricato e cliccate su «Continua»

Drigine: File system loca	le		
egli un file da cui importar	e l'applicazione virtuale. VirtualBox supporta	attualmente l'importazione di applicazioni salvate in Open Virtu	alization
File: /Users/andrea/De	sktop/prova.ova		
,			
	N N		
		N N	
		Modalità esperta Indietro Continu	a Ann



Set Up della macchina virtuale (4)

Nella schermata successiva vi verranno mostrate delle impostazioni che riguardano le configurazioni che vogliamo utilizzare per la macchina virtuale (quantità di RAM da assegnare, spazio di Hard Disk virtuale da utilizzare, CPU riservata a macchina virtuale, etc.)

Nel nostro caso possiamo tenere quelle di default che ci propone. Confermiamo il tutto e diamo avvio al processo di importazione della macchina virtuale.



Set Up della macchina virtuale (5)

Fatto questo, avremo la macchina virtuale installata!

Dobbiamo solamente impostare (non obbligatorio ma ci viene comodo per gli esercizi che vedremo) una cartella condivisa tra il sistema ospite ed il sistema ospitante.

Vediamo come fare nelle prossime slide.





Set Up della macchina virtuale (6)

Selezionate la macchina virtuale di interesse e cliccate sul pulsante «Impostazioni».

Strumenti	Nuova Impostazioni Scarta Avvia			
Windows XP ③ Spenta	Generale Nome: Xubuntu Sistema operativo: Ubuntu (64-bit) Posizione del file delle impostazioni: /Users/andrea/VirtualBox VMs/ Sistema 2			
Xubuntu Image: Spenta				
	Memoria di base: 1024 MB Ordine di avvio: Floppy, Ottico, Disco fisso Accelerazione: VT-x/AMD-V, Paginazione nidificata, Paravirtualizza			
1	📃 Schermo			
	Memoria video:16 MBFattore di scala:2.00Scheda grafica:VMSVGAServer di desktop remoto:DisabilitatoRegistrazione:Disabilitata			
	2 Archiviazione			
	Controller: IDE IDE master secondario: [Lettore ottico] VBoxGuestAdditions.iso (75 Controller: SATA			



Set Up della macchina virtuale (7)

Spostatevi nella sezione «Cartelle Condivise» e aggiungete una nuova cartella condivisa

🥹 Xubuntu - Impostazion	ni	?	×
Generale	Cartelle condivise		
Sistema	Cartelle condivise		
Schermo	Nome Percorso Accesso Montaggio automatico	Su	
Archiviazione			2
🕩 Audio			
📑 Rete			
🏠 Porte seriali			
S USB			
Cartelle condivise	1		
Interfaccia utente			
	ОК	Annul	а



Set Up della macchina virtuale (8)

Impostate il percorso corretto (ad esempio in una cartella sul desktop), dando il nome che preferite e checkate «Montaggio automatico»

Percorso della cartella:	Ustop/SharedLinux				
Nome della cartella:	SharedLinux				
Sola lettura					
· · · · · · · · · · · · · · · · · · ·	Montaggio automatico				
Punto di mount:					
	Annulla OK				



Set Up della macchina virtuale (9)

A questo punto troverete all'interno del sistema virtualizzato la possibilità di accedere alla cartella condivisa.





Set Up della macchina virtuale (10)

Se dovesse servirvi username o password della macchina virtuale sono le seguenti:

- Username: Linux
- Password: linuxRoot



Installazione di Software



Gestione del software

I programmi in Linux possono essere installati:

- Compilando i programmi direttamente partendo dal codice sorgente
- Utilizzando le versioni pacchettizzate dei programmi

Questa seconda possibilità, se presente, è da preferire in quanto permette di risolvere diversi dei problemi che si possono presentare:

- Vengono risolte in automatico le dipendenze
- Permette l'aggiornamento automatico alle nuove versioni


Cosa contiene un pacchetto

Le versioni pacchettizzate dei programmi contengono:

- Script pre e post installazione
- File utili al programma:
 - Eseguibili
 - Librerie
 - Manuali
 - File di configurazione
 - ...



Package manager

		Package Manager	
Distribuzione	Formato	Alto livello	Basso livello
Centos, Fedora, RHEL		yum, dnf	
Mandriva	rpm	urpmi	rpm
openSUSE, SUSE		zypper	
Debian	dah		dializer
Ubuntu	aeb	apı-gei	иркд



UNIVERSITÀ Dipartimento di Ingegneria Gestionale, dell'Informazione e della Produzione

Package manager

In alcune distribuzioni è offerto anche un package manager con GUI che semplifica l'installazione di software.

Questi package manager gestiscono anche gli aggiornamenti dei software.

¢ 9 Esplora Installato - + X Aggiornamenti www.inkaryme.org Selezionate da noi 0 Selfless Heroes Plex Media Se. Shotcut Xonotic Chromium ark ***** ***** ***** **** ***** ***** Altro... Applicazioni grafiche consigliate X **())** Eye of GNOME VidCutter SmartScreen directpdf matroska-tools mpv **** ***** ***** ** +++ ***** Altro... Applicazioni di produttività consigliate FREEMIND じ 3 \sim Ul X Tusk FreeMind Wekan Ora Firefox Mailspring ***** ***** **** ***** ***** ***** Categorie Produttività 🞵 Audio e video Comunicazione e notizie



Perchè utilizzare pacchetti?

Come accennato in precedenza, quando possibile, è meglio utilizzare software tramite pacchetti, in quanto:

- L'installazione è molto semplice
- Le dipendenze vengono risolte automaticamente
- Il sistema viene <u>tutto</u> aggiornato in automatico



Dipartimento di Ingegneria Gestionale, dell'Informazione e della Produzione

Comandi apt-get

Dato che nel tutorato di oggi vedremo Xubuntu, una distribuzione derivata da Debian (così come il più famoso Ubuntu), vediamo di seguito una lista di comandi utili offerti dal gestore dei pacchetti «apt»

Operazione	Comando
Aggiornamento lista programmi	apt-get update
Aggiornamento Sistema	apt-get upgrade
Ricerca Software	apt-cache search XYZ
Installazione software	apt-get install XYZ



Dipartimento di Ingegneria Gestionale, dell'Informazione e della Produzione

"Contro" dell'installazione dei SW pacchettizzati?

Utilizzare SW pacchettizzati può anche avere qualche «contro»:

- Mischiare repository non ufficiali può portare ad instabilità del sistema
- Generalmente risulta impossibile avere due versioni dello stesso programma installate
- Alcuni software potrebbero non esistere in versione pacchettizzata (specialmente se SW proprietari)
- Il software deve essere installato con privilegi di amministratore, quindi potrebbero esserci dei problemi di sicurezza



Nuove alternative (1)

Per risolvere i problemi delle versioni pacchettizzate sono nate delle alternative:

- Soluzioni basate su Linux container
 - Flatpak
 - <u>https://flatpak.org</u> sito ufficiale del progetto
 - <u>https://flathub.org/home</u> da cui è possibile scaricare applicazioni realizzare con Flatpak
 - Snap
 - <u>https://snapcraft.io/</u> sito ufficiale e «store» per il download delle applicazioni



di Ingegneria Gestionale, dell'Informazione e della Produzione

Nuove alternative (2)

Per risolvere i problemi delle versioni pacchettizzate sono nate delle alternative:

- Soluzioni non basate su Linux container
 - AppImage
 - <u>https://appimage.org</u>
 - <u>https://appimage.github.io/apps/</u>

NB: Nonostante questi metodi risolvano i problemi dei SW pacchettizzati, anche queste soluzioni utilizzano i pacchetti software, quindi essi rimangono ancora rilevanti!



Introduzione alla Console



di Ingegneria Gestionale, DI BERGAMO dell'Informazione e della Produzione

Interazione con Linux

Per interagire con Linux è possibile utilizzare <u>due strade</u>:

- GUI in Linux risulta un componente aggiuntivo ed opzionale del sistema
- Terminale



A | Dipartimento
 I | di Ingegneria Gestionale,
 D | dell'Informazione e della Produzione

GUIs per Linux (1)

Per Linux è possibile parlare di GUI in due termini:

- <u>Window manager</u>: semplice gestione delle finestre
 - Blackbox, fluxbox
 - Icewm
 - Twm
 - Afterstep
 - Enlightenment
 - ...



GUIs per Linux (2)

Per Linux è possibile parlare di GUI in due termini:

- <u>Desktop environment</u>: comprende window manager e programmi integrati
 - GNOME
 - KDE
 - XFCE4
 - Mate
 - ...



Introduzione alla console

Linux integra diversi tipi di console:

- Bash è la più diffusa
- Zsh
- Ksh
- Csh
- ...





A | Dipartimento
 DI | di Ingegneria Gestionale,
 O | dell'Informazione e della Produzione

Operazione	Comando
Creazione directory	mkdir
Spostamento/ridenominazione file	mv
Navigare nelle directory	cd
Rimuovere file/directory	rm
Elencare file	ls

- E' possibile anche combinare più programmi utilizzando la pipe |
- E' possibile redirigere l'output dei programmi



Esercizio: navigare nella console (1)

- 1. Aprire la console nella *home* dell'utente;
- 2. Spostarsi nella cartella Desktop;
- 3. Creare una cartella di nome "cartella1" ed entrarvi;
- 4. Elencare il contenuto della cartella;
- 5. Creare altre due cartelle: "cartella2" e "cartella3";
- 6. Elencare il contenuto della cartella;
- 7. Spostare "cartella3" in "cartella2";
- 8. Entrare in "cartella2" e listare il contenuto;
- 9. Rinominare "cartella3" in "cartella4";
- 10. Eliminare tutte le cartelle create;
- 11. Tornare nella propria home.



Esercizio: navigare nella console (2)

- 1. Cosa succede se si esegue il comando ls .?
- 2. Cosa succede se si esegue il comando ls ..?
- 3. Cosa succede se si esegue il comando ls ~?
- 4. Che differenza c'è tra il comando ls e il comando ls .?
- 5. Cosa succede se si esegue il comando ${\tt cd}$.?
- 6. Cosa succede se si esegue il comando ${\tt cd}$..?
- 7. Cosa succede se si esegue il comando ${\tt cd}$ ~?
- 8. Che differenza c'è tra il comando ${\tt cd}$ e il comando ${\tt cd}$.?
- 9. Che differenza c'è tra il comando ${\tt cd}$ e il comando ${\tt cd}$ ~?



Esercizio: navigare nella console (3)

- 1. Create una cartella "comandi" nella home dell'utente;
- 2. Create una cartella per ognuno dei comandi base appena visti;
- 3. Navigate tra le cartelle prima tramite console e poi tramite interfaccia grafica;
- 4. Rimuovete la cartella "rm";
- 5. Tornate nella home;
- 6. Tornate in "comandi" e copiate la cartella "cp" in "mv";
- 7. Entrate nella cartella "mkdir" e create una cartella "informatica";
- 8. Eliminate tutte le cartelle create;
- 9. Tornate nella home.



Combinare più programmi insieme tramite pipe

Cercare tra i processi di tutti gli utenti (a), mostrando l'user proprietario (u), compresi i processi che non sono dipendenti da un terminale (x), il processo di nome «firefox».

ps aux | grep firefox

Redirigere l'output di un comando

Cercare tutti i gruppi che contengono la parola «users» e salvare il risultato nel file members_of_users_group.

grep users /etc/group > members_of_users_group



Redirigere stderr (2) sullo stdout (1)

rm /tmp/* 2>&1

Redirigere stderr e stdout su file

rm /tmp/* >& everything



Dipartimento
 di Ingegneria Gestionale,
 dell'Informazione e della Produzione

Gestione dei permessi

Aggiunta del permesso di esecuzione al file script.sh.

```
chmod +x script.sh
```

Lista dei processi in esecuzione

Processi di tutti gli utenti (a), mostrando l'user proprietario (u), compresi i processi che non sono dipendenti da un terminale (x)

ps aux



A | Dipartimento
 I di Ingegneria Gestionale,
 D dell'Informazione e della Produzione

Gestione dei processi

Terminazione del processo con PID 900

kill 900

Terminazione di tutti i processi del programma firefox

killall firefox



A | Dipartimento
 I di Ingegneria Gestionale,
 O dell'Informazione e della Produzione

Ricerca all'interno di file di testo

grep <pattern> <file_name|pattern>

Ricerca all'interno di file di testo, in modalità case insensitive, e ricorsivo da una certa posizione su tutti gli elementi nelle sottocartelle

grep -i -r <pattern> <file name|pattern>



Utilizzo di caratteri Jolly



di Ingegneria Gestionale, DI BERGAMO dell'Informazione e della Produzione



Ricerca di file

find <path> -iname <pattern>



Dipartimento
 di Ingegneria Gestionale,
 dell'Informazione e della Produzione

Ordinamento di valori in un file

Supponiamo di avere un file «values», di tipo csv, e volerne vedere il contenuto:

\$ cat values a;12 c;1 b;500



Ordinamento di valori in un file

Ordiniamo, utilizzando la seconda colonna (-k2) di un file di testo con separatore (-t ';'), sfruttando l'ordinamento numerico (n)

Ordiniamo utilizzando la prima colonna del file



Ordinamento di valori in un file

Ordiniamo in ordine decrescente (-r)

sort -n -r -k2 -t ';' values

Ordinamento con rimozione di duplicati

sort -u values



Dipartimento
 di Ingegneria Gestionale,
 dell'Informazione e della Produzione

Sostituzione di caratteri in un file

Eseguiamo (-e) uno script di sostituzione (s) i delimitatori «;» presenti nel file values con « - », utilizzando il comando che permette di svolgere diverse operazioni su testi

sed -e 's/;/ - /' values



Dipartimento
 di Ingegneria Gestionale,
 dell'Informazione e della Produzione

Rimozione di righe duplicate da un file

uniq values

O in alternativa è meglio usare

sort values | uniq



A | Dipartimento
 Di di Ingegneria Gestionale,
 O | dell'Informazione e della Produzione

Programmazione Batch



DEGLI STUDI di Ingegneria Gestionale, DI BERGAMO dell'Informazione e della Produzione



La shell Linux, della quale abbiamo appena visto alcuni comandi, permette di fare una semplice programmazione:

- Si raccolgono i comandi in sequenza dentro ad un file batch eseguibile, detto script di shell
- Nello script di shell è possibile anche utilizzare aspetti più «complessi»: cicli, variabili, istruzioni condizionali, etc.



Dipartimento
 di Ingegneria Gestionale,
 dell'Informazione e della Produzione

A cosa serve?

Utilizzare la programmazione della Shell di Linux è molto utile per:

- Building di applicazione
- Distribuzione ed esecuzione da remoto di programmi ed artefatti SW
- Utilizzare le Cloud Shell
- ...



A | Dipartimento I | di Ingegneria Gestionale, O | dell'Informazione e della Produzione

Costrutti base

Dichiarazione di variabili

VAR = valore

Accesso al valore di una variabile (ad esempio in una espressione)

\$VAR



Dipartimento
 di Ingegneria Gestionale,
 dell'Informazione e della Produzione

Costrutti base

Istruzione condizionale

```
if [[ $VAR -gt 10 ]]
then
    echo "The variable is greater than 10."
else
    echo "The variable is smaller than 10."
fi
```



Costrutti base

Cicli enumerativi (simil for-each)

for [VAR] [in [LIST]] do [COMANDI] done



Dipartimento di Ingegneria Gestionale, dell'Informazione e della Produzione

Demo ed Esercizi



UNIVERSITÀ DEGLI STUDI DI BERGAMO DI BERGAMO
Demo – std_out_err.sh

Prendiamo ora il file bash std_out_err.sh e proviamo a:

- Redirigere *stderr* sullo *stdout*
- Redirigere *stdout* su *file*
- Redirigere stdout su file in modalità append
- Redirigere stderr su file
- Redirigere *stderr* su file in modalità append
- Redirigere stderr e stdout su file
- Redirigere *stderr* e *stdout* su file in modalità append



Demo – classifica

Il file *artists_overall* contiene una lista di artisti, a ciascuno dei quali è associato il numero di ascolti mensili.

Vogliamo redigere la classifica degli artisti in base al numero di ascolti, inserendo il risultato nel file ranking.txt



A | Dipartimento D | di Ingegneria Gestionale, O | dell'Informazione e della Produzione

Demo – posizione in classifica

Sfruttando ciò che abbiamo realizzato in precedenza, scriviamo ora un semplice script in Bash che, dato il nome di un artista, ci dice la sua posizione in classifica



A Dipartimento
I di Ingegneria Gestionale,
O dell'Informazione e della Produzione

Demo – Creazione cartelle

Scriviamo uno script creaDir.sh che crea X cartelle, denominate con i parametri che sono stati passati in input.

Ad esempio, il comando

creaDir.sh dir1 dir2 dir3

Deve creare 3 cartelle, denominate dir1, dir2 e dir3



Dipartimento di Ingegneria Gestionale, dell'Informazione e della Produzione

Demo – Creazione cartelle nazioni

Il file *world_countries* contiene la lista delle nazioni del mondo. Scrivere uno script bash che crea, sotto una sottocartella (della cartella corrente) «countries» una cartella per ogni nazione.



A Dipartimento
I di Ingegneria Gestionale,
O dell'Informazione e della Produzione

Link Utili



UNIVERSITÀ DEGLI STUDI DI BERGAMO DI BERGAMO



Guida avanzata di scripting Bash:

https://diraimondo.dmi.unict.it/wp-content/uploads/classes/so/mirror-stuff/absguide.pdf

Lista di comandi per terminale:

http://www.coresis.com/extra/linuxcorsobase/11-3.htm



A | Dipartimento
Di di Ingegneria Gestionale,
O dell'Informazione e della Produzione