

Uso della CLI

- Cosa è una Shell

ambiente che permette di impartire dei comandi da tastiera. (Command Line Interface)
interpreta i comandi immessi e quindi in genere manda in esecuzione altri programmi.

- Bash (bourne again shell) default in linux
derivata da Bourne shell .
- Altre Korn shell, C shell , tcsh (T shell) ,
Z shell ...



Uso della CLI

- Come si interagisce con bash ?
- Si usa un terminale

(ctrl+alt+F1) tty1

(ctrl+alt+F2) tty2

....

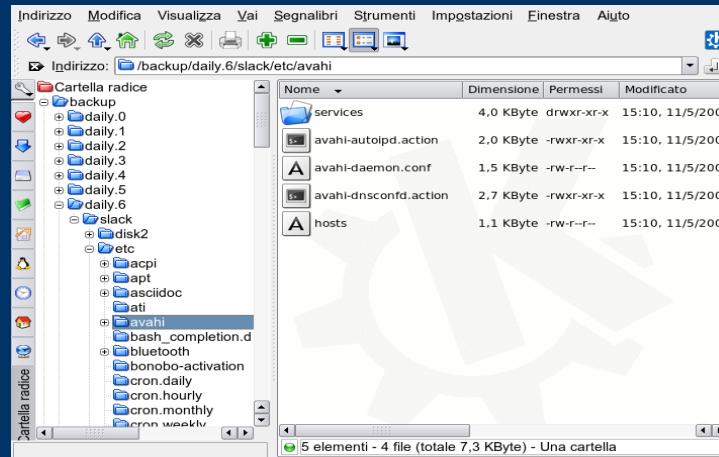
(ctrl+alt+F7) ritorna nella modalità grafica

- Oppure si usa un emulatore di terminale
xterm -konsole ...



Uso della CLI Concetti base

- Struttura del file system



- Differenza fra windows e Linux
A: B: C: D: (tanti alberi separati) - Unico albero
- Prima cartella : root indicata con /



Uso della CLI Concetti base

- Mount Point (punto di montaggio)

```
alberto:/disk2/prog$ tree
.
|-- level1
|   |-- README.Xmodmap
|   |-- level2
|   |   |-- README
|   |   |-- level3
|   |   |-- xinitrc
|   |   |-- xinitrc.blackbox
|   |   |-- xinitrc.fluxbox
|   |   |-- xinitrc.fvwm2
|   |   |-- xinitrc.kde
|   |   |-- xinitrc.twm
|   |   |-- xinitrc.xfce
|   |-- xorg.conf
--
slackware
3 directories, 10 files
```

- Comando mount su cartella vuota



Uso della CLI spostarsi fra le cartelle

- Si parte sempre da casa. (tilde ~)
- Cartella corrente . (per sapere quale è pwd =print working directory)
- Cartella superiore ..
- Come indicare una certa cartella o file
 - Percorso assoluto inizia con /
/home/alberto/Scrivania/miofile.txt
 - Percorso relativo (a dove siamo attualmente)
Scrivania/miofile.txt
- Comando per spostarsi di cartella cd (change directory)
cd =cd ~ va a casa
- mkdir crea nuova cartella



Uso della CLI

Proviamo dei semplici comandi

- echo ciao a tutti
- Comando cat visualizza su video il contenuto di un file.
- cat /etc/shells (usare il tab per aiutarvi)
- cat /etc/shells /etc/resolv.conf (2 files)
- Copiare i file cp origine destinazione
- Eliminare file rm
- Rinominare spostare mv



Uso della CLI

Proviamo dei semplici comandi

- echo ciao a tutti
- Comando cat visualizza su video il contenuto di un file.
- cat /etc/shells (usare il tab per aiutarvi)
- cat /etc/shells /etc/resolv.conf (2 files)



Uso della CLI facilitazioni di bash

- Tab completamento automatico (set show-all-if-ambiguous on nel file /etc/inputrc)
- history
- Freccie su e giù
- !cat !"numero" ctrl+r ... non quello giusto allora ancora ctrl+r
- Muoversi sulla linea end e home. ctrl+w cancella parola a sinistra del cursore....alt+b alt+f -alt.



Uso della CLI editor testuali

- nano pico
- vi emacs NO !! non indicati per principianti



Uso della CLI *quali file ci sono ?*

- Comando ls
- Parametri voglio vedere i dettagli ls -l
- Un file ha un proprietario e un gruppo. (un utente molti gruppi)
 - Diritto di scrittura W (Write)
 - Diritto di lettura R (Read)
 - Diritto di esecuzione X (eXecute)

Il proprietario può cambiare i diritti.

- Provare ls -a -l . File nascosto inizia con un punto. Uguale a ls -al (vero per la maggior parte dei programmi)
- Caratteri jolly Il significato di * e di ?



Uso della CLI *il file è mio e lo gestisco io ?*

- Chi puo fare cosa ... `chmod`
 chi :
 - proprietario u (user)
 - quelli del gruppo g (group)
 - tutti gli atri o (other)
 - Tutti a (all) cosa : + mette - toglie
 - R read leggere
 - W write scrivere
 - X eseguire per cartelle accedere
- voglio che il gruppo non possa scrivere :
 `chmod g-w ./miofile.txt`



Uso della CLI un “animale” silenzioso

- Se tutto tace tutto va bene .
- Scrittura su schermo operazione dispendiosa.
- Parametro -v verbose (in genere)
- Avrà capito cosa volevo dire meglio usarlo all'inizio.



Uso della CLI come trovare aiuto ?

- Cosa può fare ls ?? man ls oppure man chmod
sezioni di man :
 - 1 comandi in genere
 - 8 comandi di amministrazione
 - 5 descrizione dei file di configurazione
 - 4 “driver” come sono viste le periferiche
- Ma se voglio fare una cosa e non so come ??
(prima google) poi apropos
- apropos copy
aiuto troppi risultati !!-- come fare ??



Bash giocare con i tubi !

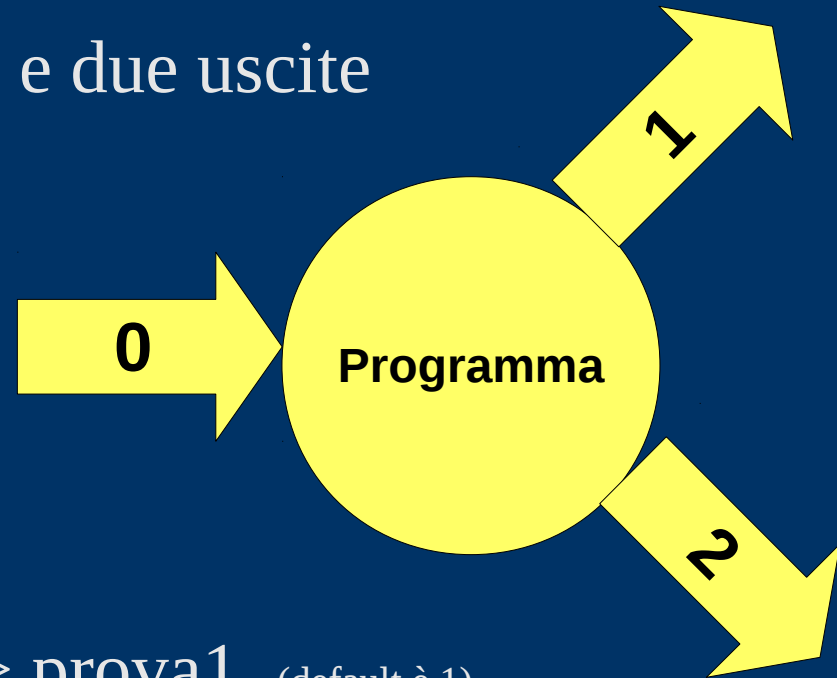
Primi giochi --- le ridirezioni

- Un programma ha un ingresso e due uscite

standard input tastiera 0

standard error video 2

standard output video 1



- Ridirezione >
- Provare ls “file inesistente” > prova1 (default è 1)
- Provare ls “file inesistente” 2 > prova2
- Provare ls “file inesistente” &> prova3
> crea sostituisce >> accoda
- find /etc -iname '*pass*' 2>/dev/null



Bash giocare con i tubi !

Primi giochi --- le ridirezioni

- Esempio di ridirezione standar input (cat da solo)
- Programma < “file”
- Programma << “EOF” il mio input finisce a...
- Programma <<< fino a fine riga (anche se sono più di una ... carattere speciale /)
- Un esempio il programma bc - `bc <<<"9^3+5"`
- at now +2 minutes <<<'play /home/gulp /Scrivania/lezioneBash/esempi/allarmeRosso.wav'



Bash giocare con i tubi !

Cosa è una pipe-line -mettiamo il primo tubo

- Non solo file ...anche programmi
- La pippe |il mio output diventa il tuo input
- La pipeline : ognuno fa una cosa sola ma la fa bene.. tutti insieme fanno tante cose.
- Un “paginatore “ : less .
- Un “cercatore” : grep . grep stringa nomefile
- apropos copy |grep -i file



***Bash** giocare con i tubi !*

Altri esempi di pipe

- ps lista processi attivi ps aux
- interattivo top -O seleziona ordinamento R ascendente discendente.
- Trovare e uccidere un processo “cattivo” kill -9
- Il programma wc word count
- ls -l *.txt | wc -l
- Altri programmi: sort tail head lsof netstat
- xargs cambia le righe in colonne ...



Bash gli alias i soprannomi che belli !!

- alias – quelli che già esistono
- creare un nuovo alias

```
alias mio_nome="comando"
```

```
alias iochisono="whoami"
```

```
alias dormi='sudo /sbin/shutdown -t 8 -h now'
```

```
alias cancellaCD='alias cancellaCD="/usr/bin/cdrecord -v /
```

```
gracetime=2 dev=ATAPI:/dev/hdd speed=12 /
```

```
-tao driveropts=burnfree blank=all -force"
```

- alias fine=" tail -n15 "
- eliminare un alias unalis "nome_alias"
- Fare un alias "tel" che cerca un numero telefonico nel file tel.txt



Bash *una ambiente su misura !*

- Shell di login –login
legge ed esegue : /etc/profile
e poi il primo che trova leggibile fra :
 - ~/.bash_profile,
 - ~/.bash_login,
 - ~/.profile
- Shell interattiva.
 - ~/.bashrc
- Alla fine (comando exit).
 - ~/.bash_logout



Bash *uff !! che noia automatizziamo!*

- Uno script : una sequenza di comandi
 - # cancelletto commento --riga ignorata
 - #! chi è in grado di capire i comandi
#!/bin/bash
 - chmod +x per eseguirlo direttamente



Bash *variabile di ambiente condividiamo delle informazioni*

- Associa ad un nome un valore.

In che lingua preferisci i messaggi di errore ???

LANG

- Creare una nuova variabile VAR = "valore"
senza spazi prima e dopo =
- Leggere una variabile echo \$VAR
- Variabile Globale -locale
comando export nome_variabile
- Comando env visualizza le variabili globali. Set anche quelle locali.



Bash *variabile di ambiente condividiamo delle informazioni*

- Associa ad un nome un valore.

In che lingua preferisci i messaggi di errore ???

LANG

- Creare una nuova variabile VAR = "valore"
senza spazi prima e dopo =
- Leggere una variabile echo \$VAR
- Variabile Globale -locale
comando export nome_variabile
- Comando env visualizza le variabili globali. Set anche quelle locali.



Bash *variabile di ambiente*

esempi

- PATH dove li cerco i programmi che mi dici di eseguire ? (lista separata da :)
- LANG=it_IT (locale -a)
- PS1=\u:\w\\$\n
- LS_COLORS
- HTTP_PROXY
- HTTPS_PROXY



Bash i 3 tipi di virgolette

- “” doppi apici - provare i seguenti comandi

```
echo ;
```

```
echo “;”
```

```
echo “$LANG”
```

doppi apici fanno stampare caratteri che per bash hanno un significato speciale ma comunque esaminano il valore delle variabili

- ' ' apici singoli - provare

```
echo '$LANG'
```

apici singoli bash non interagisce con quello incluso negli apici singoli echo stampa la stringa passata.

- `` apici inclinati (altgr+apice singolo) - provare

```
echo “ora sono le ore : `date +%H:%M:%S.%N`”
```

tra apici inclinati viene scritto un comando ..l'output del comando sarà messo nel messaggio .



Bash *job control*

- CTRL +c termina programma
- CTRL+z mettere in pausa un programma
yes test | nl
- fg Foreground
yes test |nl > ./testbg
- bg background
- & in fondo: esecuzione da subito in background
- comando jobs
- Kill %numjob



Bash più istruzioni sulla stessa riga

- Più istruzioni ma non nella stessa pipeline
- ; eseguo sempre e comunque
- && se il comando precedente è stato eseguito correttamente allora lo eseguo altrimenti mi fermo.

wget <http://www.sito.org/file.tgz> && tar -xf file.tgz

- || se il comando precedente è terminato in errore allora lo eseguo altrimenti mi fermo.



Esempi

```
# ordina le linee di lista.txt in ordine alfabetico
sort lista.txt
#Fa vedere solo i warning della partenza dell'ambiente grafico
grep 'WW' /var/log/Xorg.0.log
#Fa vedere solo gli errori della partenza dell'ambiente grafico
grep 'EE' /var/log/Xorg.0.log
#trova tutte le righe che contengono mario nel file telefoni.txt
grep mario telefoni.txt
#aggiunge infondo al file una nuova riga senza dover aprire il file in un editor
#ATTENZIONE a usare >> se si usa solo > si sovrascrive
echo 'mario ultimo 05062111111111' >>telefoni.txt
#emette un suono fra 2 minuti si può scolare la pasta :-)
at now +2 minutes <<<'play /home/gulp /Scrivania/lezioneBash/esempi/allarmeRosso.wav'
wall <<FINE
# cerca tutti i file che hanno la stringa pass nel nome e non fa vedere errori
find /etc -iname '*pass*' 2>/dev/null
#aggiorna il sistema e poi spegne il pc
sudo apt-get update;sudo apt-get upgrade 2>errori.log;sudo halt
#Realizzazione di un fortune grafico usa pacchetti fortune e libnotify
#per nomi icone guardare http://standards.freedesktop.org/icon-naming-spec/icon-naming-spec-latest.html
notify-send --urgency=low --expire-time=20000 --icon='face-wink' 'La sorte dice:' "$(fortune -a )"
```