

# 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

*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



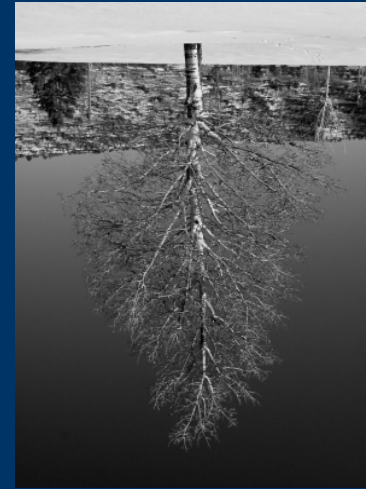
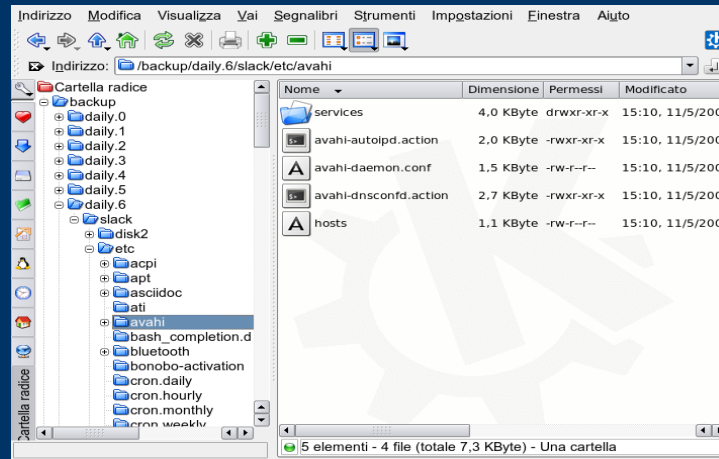
# *Uso della CLI editor testuali*

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



# 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 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)
- Il significato 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 !

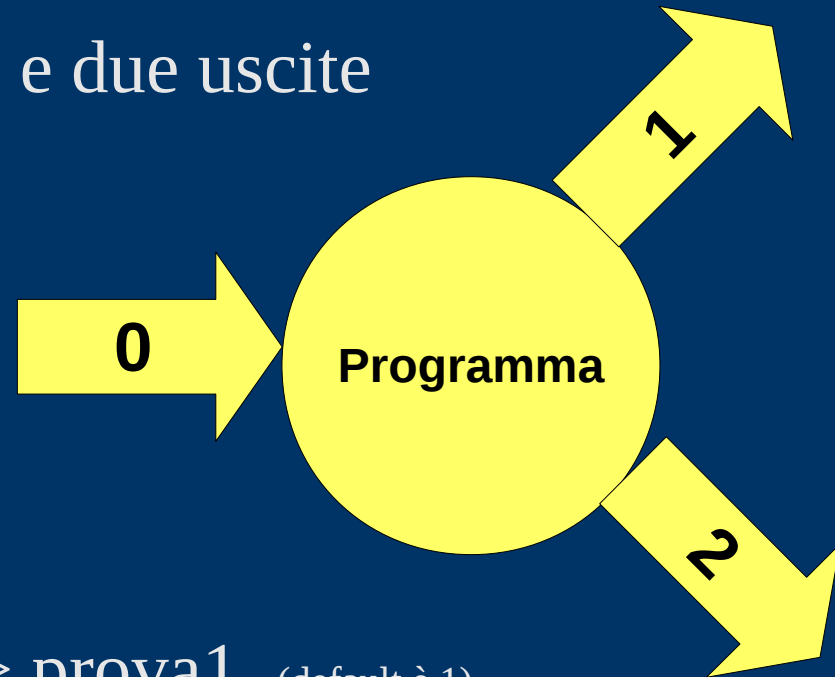
## Primi giochi --- le ridirezioni

- 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



# *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 /)
- Una esempio il programma bc - `bc <<< "9^3+5"`



# *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*

## ***esempi***

- PATH dove li cerco i programmi che mi dici di eseguire ? (lista separata da :)
- LANG=it\_IT (locale -a)
- PS1=\u:\w\\$\
- LS\_COLORS
- HTTP\_PROXY
- HTTPS\_PROXY



# *Bash* i 3 tipi di virgolette

- “”

echo ;

echo “.”

echo “\$LANG”

- ‘ ’

echo '\$LANG'

- ` `

echo “ti ho detto ciao alle ore : `date +%H:%M:%S.%N` “



# *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.





# *Bash* esercizi

- Musica in “sottofondo” comandi mpg321 o mpg123
- Tutta e sola la musica di pluto .  
Suggerimento `/usr/bin/ls -1Q`

