



## IL SISTEMA OPERATIVO LINUX

*Introduzione all'uso di Linux*

---

Luigi Coppolino, Luigi Romano



## Prima di cominciare: preparazione dell'ambiente di lavoro

<https://www.vmware.com/products/player/playerpro-evaluation.html>



The screenshot shows the VMware website's product page for Workstation Player. The top navigation bar includes the VMware logo, a search icon, and links for 'US', 'Login', 'Training', 'Community', and 'Support'. The breadcrumb trail reads 'Products > Workstation Player > VMware Workstation Player - VMware Products'. The main heading is 'VMware Workstation Player - VMware Products', followed by a sub-heading 'Try VMware Workstation Player'. Below this is a large graphic for 'VMWARE WORKSTATION PLAYER 12.5' with a yellow and black color scheme. To the right of the graphic, there is text describing the product as the foundation of Workstation Player, mentioning its user interface for creating and applications in a virtual environment. It also notes that the free version is available for personal and home use, and that commercial organizations can benefit from the Workstation Player. A link is provided to 'Check out Workstation Player for Linux'.

## ***SCARICARE E INSTALLARE VMWARE WORKSTATION PLAYER (≥ v.12.5)***

# SCARICARE UN'IMMAGINE DI LUBUNTU

- LUBUNTU è una versione particolarmente leggera di Ubuntu
- Requisiti:
  - Intel **Pentium II**
  - **128 MB di RAM** (consigliato 256 MB)
- Un'immagine WMWare già pronta può essere scaricata da:  
<http://www.osboxes.org/lubuntu/#lubuntu-17-04-vmware>
  - Richiede 7-zip per un-zip <http://www.7-zip.org/download.html>



OSBOXES

VM IMAGES ▾

Home / Lubuntu

## Lubuntu

*Lubuntu*

Lubuntu is a fast and lightweight operating system developed by a community of Free and Open Source enthusiasts. Lubuntu uses the minimal desktop LXDE (Lightweight X11 Desktop Environment), and a selection of light applications. The core of the system is based on Linux and Ubuntu. Basically Lubuntu focus on speed and energy-efficiency. Because of this, Lubuntu has very low hardware requirements, it is intended to have low-resource system requirements and is designed primarily for netbooks, and older PCs. Tests show it can use half as much RAM as Xubuntu, making it an attractive choice for installing on older hardware being refurbished for charitable distribution.

### Lubuntu 17.04 Zesty

VirtualBox VMware

VMware (VMDK) 32bit **Download** Size: 708MB

Username: osboxes  
Password: osboxes.org

VB Guest Additions & VMware Tools: Not Installed

# SCARICARE UN'IMMAGINE DI LUBUNTU

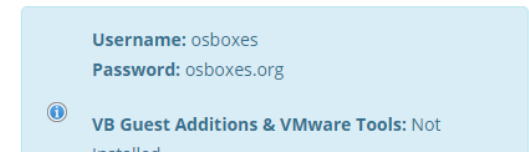
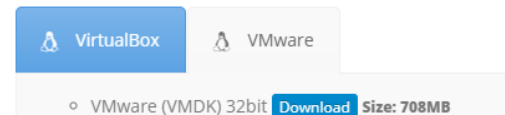
- LUBUNTU è una versione particolarmente leggera di Ubuntu
- Richiede **Più una serie di operazioni di configurazioni...in alternativa scaricare il pacchetto già pronto da:**
  - Intel Pentium II
  - 128 MB di RAM (consigliato 256 MB)
- Un'immagine WMWare già pronta può essere scaricata da:  
<http://www.osboxes.org/lubuntu/#lubuntu-17-04-vmware>
  - Richiede 7-zip per un-zip <http://www.7-zip.org/download.html>



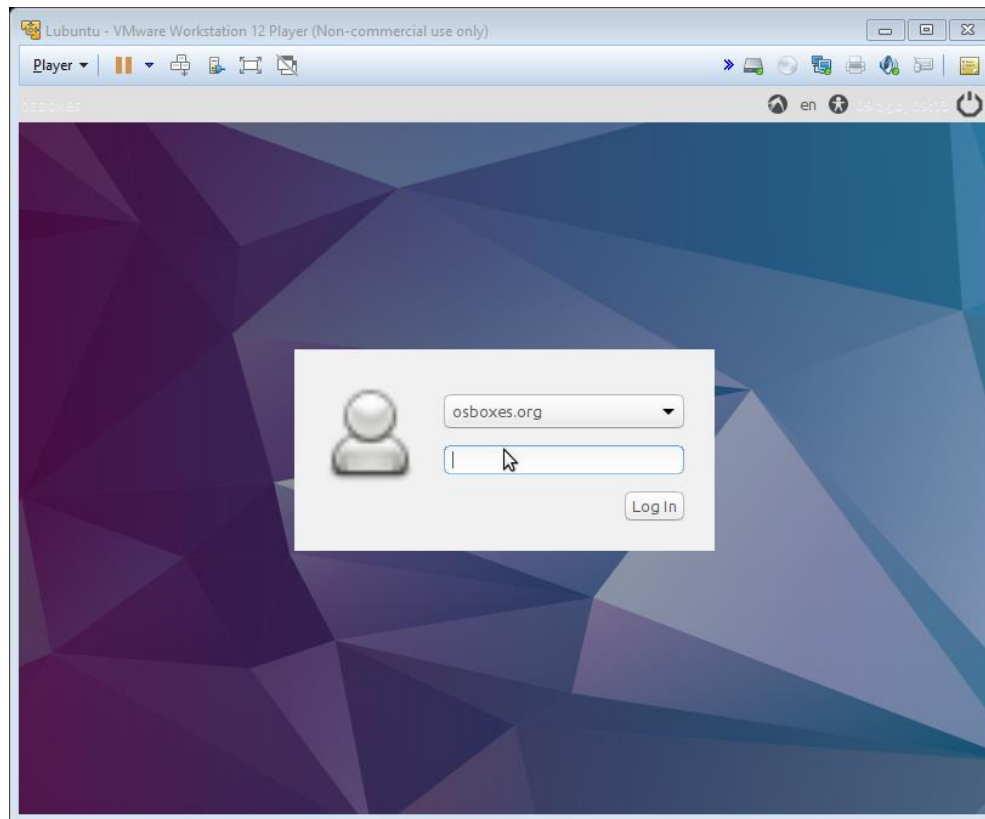
Lubuntu

Decomprimere e avviare il file **Other Linux 3.x kernel.vmx**

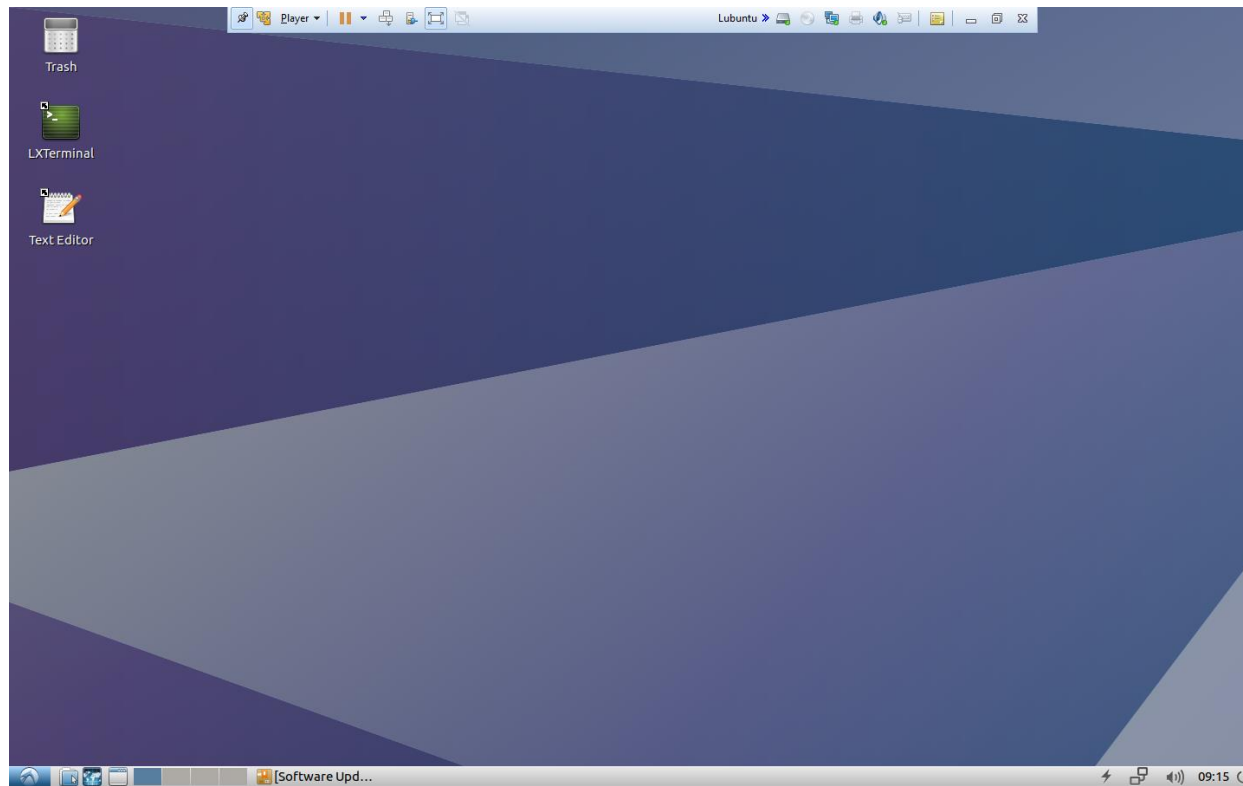
## Lubuntu 17.04 Zesty



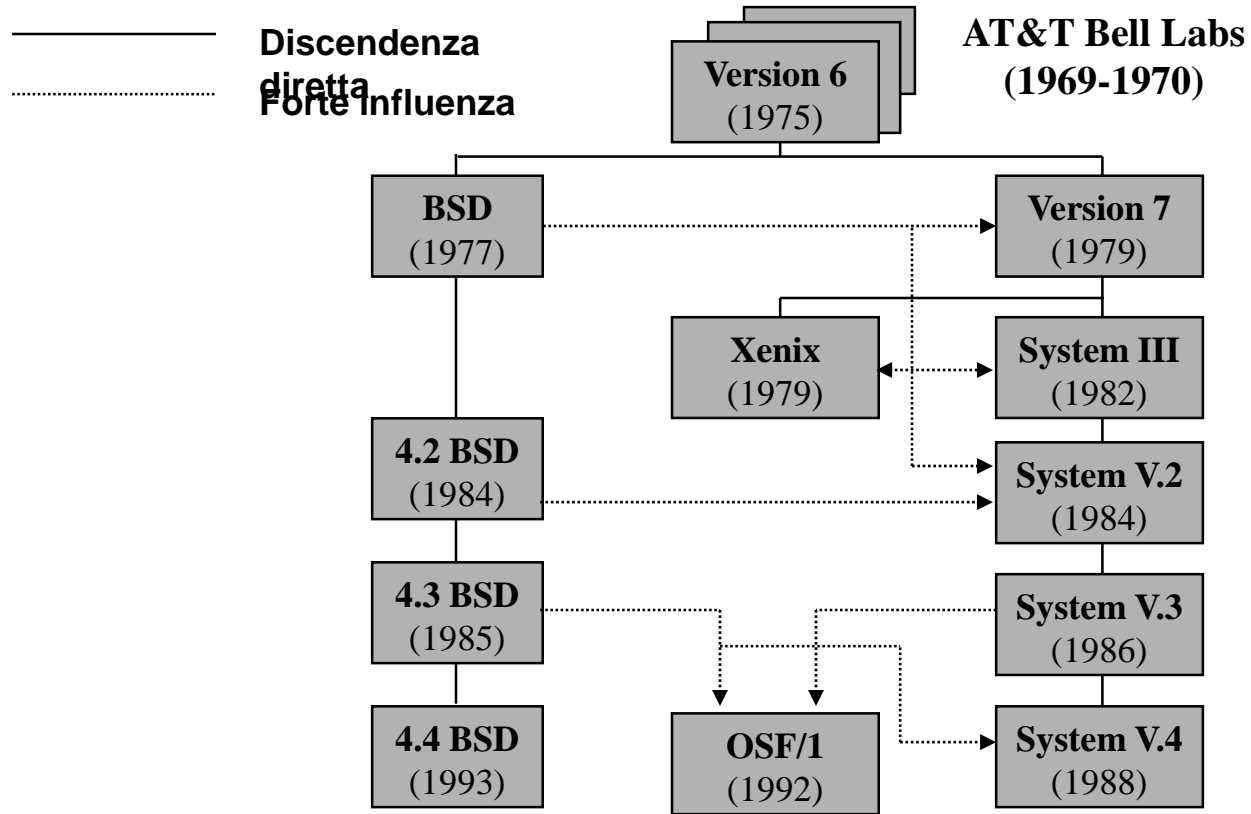
# INSERIRE LA PWD: OSBOXES.ORG



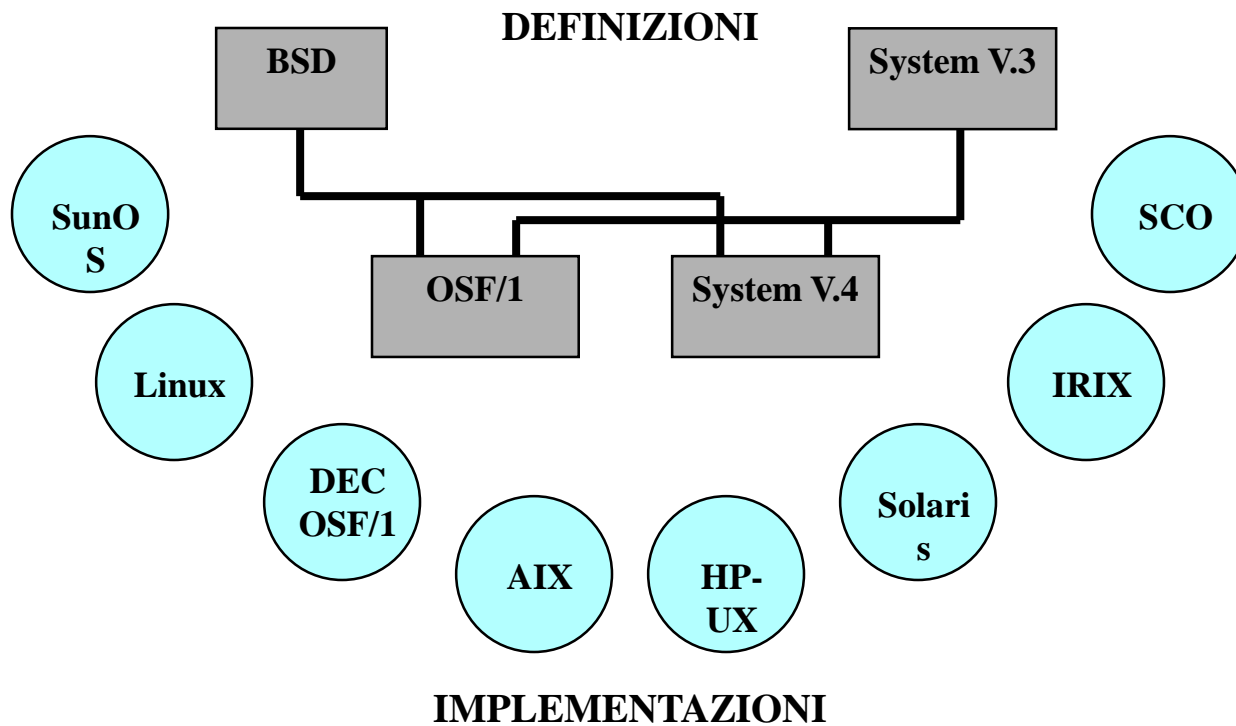
# BENVENUTI IN LINUX



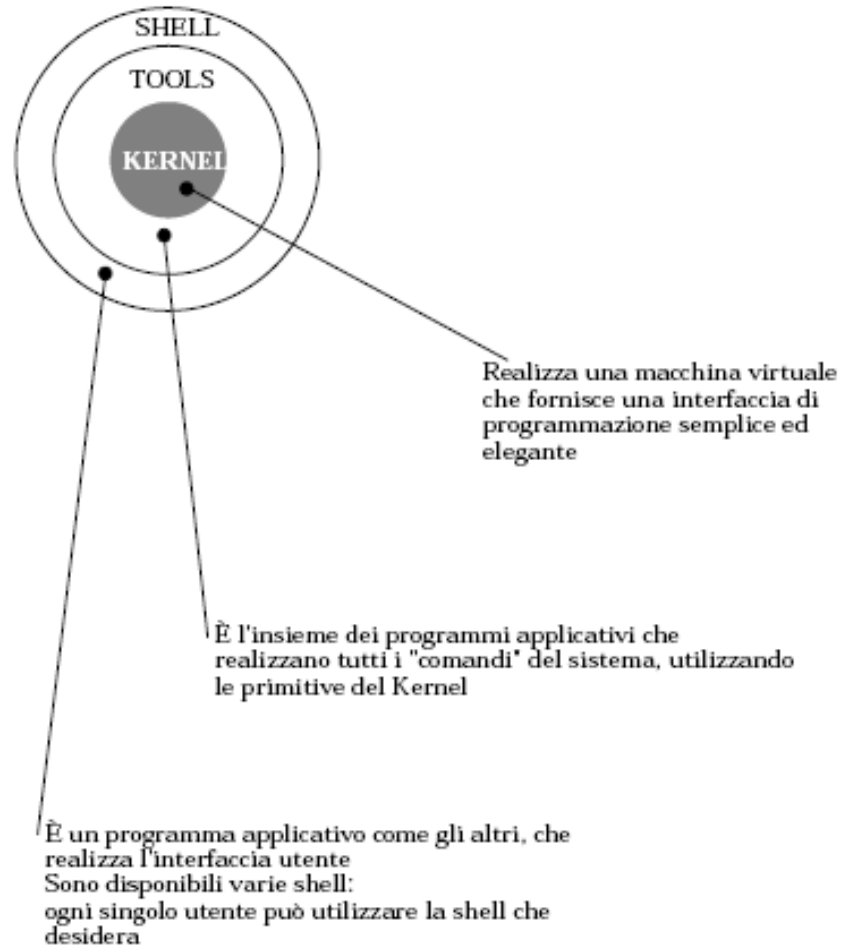
# GENEALOGIA DI UNIX (SEMPLIFICATA)





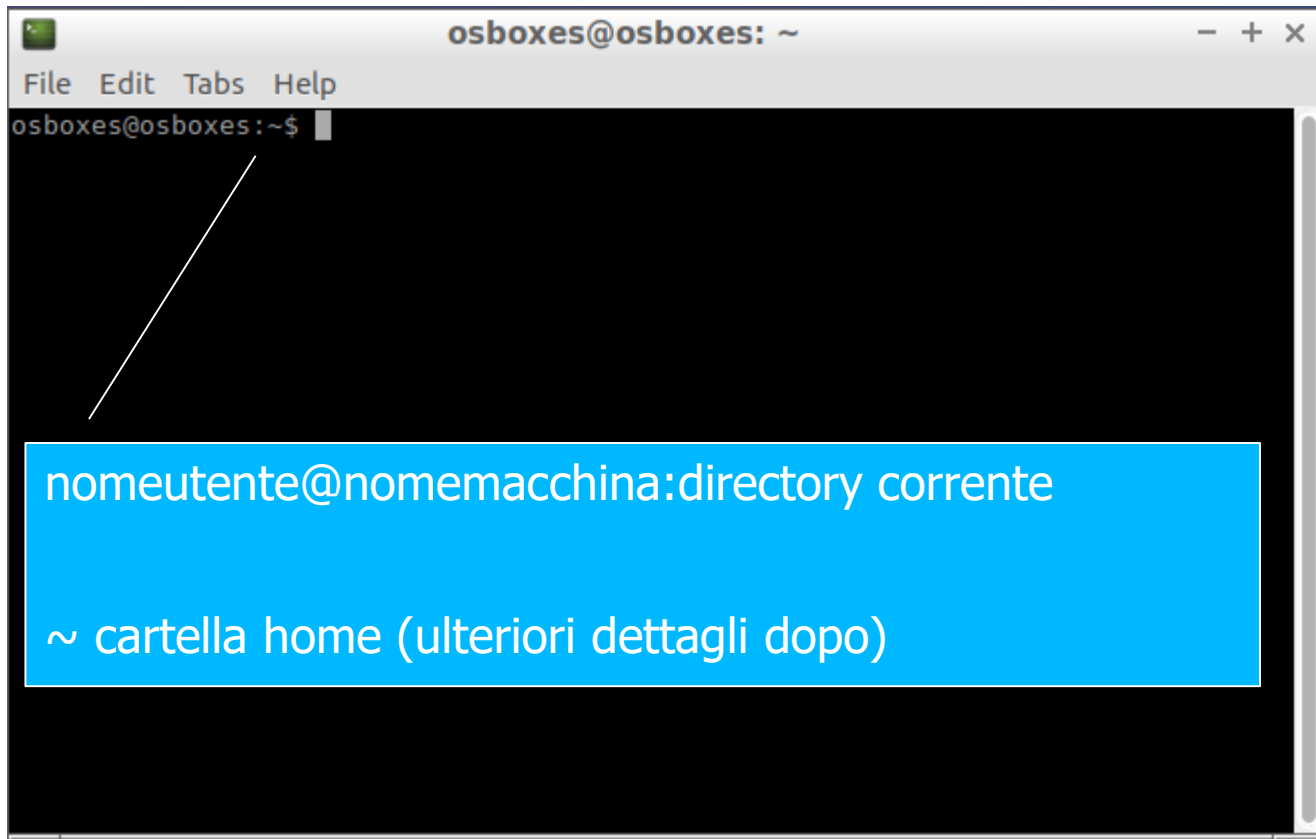


- Da Unix eredita il suo essere:
  - Multi-user
  - Multi-processing
  - Time-sharing



# LOGIN E PASSWORD

- Il login prompt
  - `fuore login:`
- Il login name
  - `fuore login: lrom`
- Il comando passwd
  - `passwd`
- Il file passwd
  - `lrom:nHJ7Gj4oFG.w6:184:100:Luigi`  
`Romano:/home/sorrento/lrom:/bin/tcsh`
- I processi coinvolti
  - Il login
  - La shell



- ❑ reperire informazioni su comandi, programmi, file...
- ❑ gestire filesystem
- ❑ operare su file e directory
- ❑ elaborare testi
- ❑ sviluppare software
- ❑ comunicare tra utenti (sullo stesso host)
- ❑ operare in remoto
- ❑ ...(comandi “di utilità” vari)
- ❑ amministrare il sistema
  - utenti e gruppi
  - dispositivi
  - software

## comando [ argomento ...]

Dove *comando* denota il nome o il path di un file eseguibile, mentre *argomento* può essere un'opzione o un parametro:

- le opzioni o flag servono a specificare il comportamento del comando; esse sono individuate dal carattere “-” oppure “- -” (comandi GNU );
- i parametri rappresentano l'input per il comando o per una delle sue opzioni;

Sia le opzioni che gli argomenti vanno separati da almeno un separatore (di default il carattere spazio); tuttavia, per quei comandi che prevedono solo opzioni di un'unico carattere, è spesso possibile raggruppare più opzioni in un'unica “stringa di opzioni”.

### Esempi d' uso

#### Uso delle opzioni separate:

```
ls -l -F file1 file2 file3
```

#### Per le opzioni senza parametri, l'ordine non è in generale rilevante:

```
ls -F -l file1 file2 file3
```

#### Talvolta è possibile fondere più opzioni in una singola stringa:

```
ls -Fl file1 file2 file3
```

## Note:

Le opzioni possono essere specificate in qualsiasi ordine, mentre per parametri l'ordine è importante.

### Esempi di comandi equivalenti:

```
ls -l -F file1 file2 file3
```

```
ls -Fl file1 file2 file3
```

```
ls -lF file1 file2 file3
```

### Esempio di comandi errati:

```
cat file1 > file2
```

```
cat file2 > file1
```

Per qualsiasi istruzione possiamo usare l'opzione `--help` che fornisce una guida in linea per l'istruzione

Es.

```
$ ls --help
```



# MESSAGGI D'ERRORE

- Tutti i comandi generano dei messaggi d'errore nel seguente formato:
  - nome-comando: azione tentata: messaggio
- Tutti i messaggi d'errore sono nel file:
  - `/usr/include/asm/errno.h`

# WILDCARDS

- \* – Qualsiasi stringa di caratteri di qualsiasi lunghezza
- ? – Un singolo carattere

# ESTENSIONI – 1

- .a – archivio statico
- .avi – video
- .au – audio
- .awk – programma sorgente `awk`
- .cc, .cxx, .C, .cpp – sorgenti C++
- .cf, .cfg – file di configurazione
- .cgi – eseguibile che produce pagine web
- .csh – `csh` shell script
- .Z – file compresso con `compress`
- .gz – file compresso con `gzip`
- .tgz – `tar` + `gzip`

## ESTENSIONI – 2

- .c – sorgenti C
- .h – header C/C++
- .gif, .giff – immagine giff
- .htm, .html – file html
- .jpeg – immagine jpeg
- .pfb – font file di XWindows
- .pdf – file Acrobat
- .ps – file PostScript
- .rpm – file della distribuzione RedHat
- .deb – file della distribuzione Debian

## ALCUNI COMANDI DI BASE

- bc
- cal
- cat
- clear
- date
- du
- df
- dircmp
- free
- echo
- expr
- hostname
- file
- less
- more
- sort
- strings
- split
- uname
- uniq
- wc
- who
- whoami

- ❑ *who*            Reperire info sugli utenti attivi  
                      lista il nome, terminale, e data/ora di login di tutti gli utenti correnti
- ❑ *who am I*        Come sopra ma solo per l'utente che lo esegue
  
- ❑ *date*            modifica o stampa data corrente

- Unix ha un manuale di riferimento molto completo, accessibile “in linea” mediante il comando *man*;

**man [options] [-M path] [-s section] name...**

**man [-M path] -k keyword**

**man [-M path] -f file**

\$ man ls

```
osboxes@osboxes: ~
File Edit Tabs Help
LS(1) User Commands LS(1)
NAME
  ls - list directory contents
SYNOPSIS
  ls [OPTION]... [FILE]...
DESCRIPTION
  List information about the FILES (the current directory by default).
  Sort entries alphabetically if none of -cftuvSUX nor --sort is specified.

  Mandatory arguments to long options are mandatory for short options too.

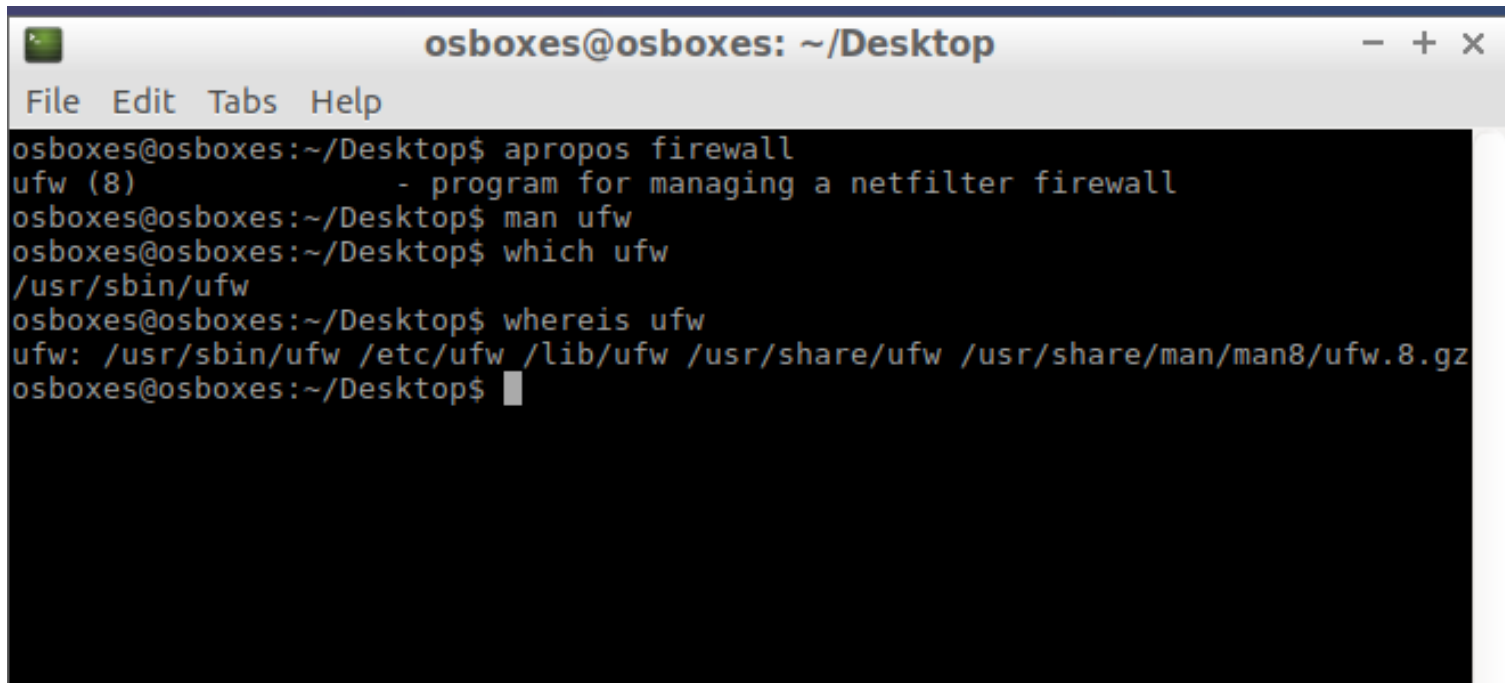
  -a, --all
      do not ignore entries starting with .

  -A, --almost-all
      do not list implied . and ..

  --author
Manual page ls(1) line 1 (press h for help or q to quit)
```

## REPERIRE INFORMAZIONI

- ❑ *man* Accede al manuale in linea di Unix
- ❑ *apropos* ricerca all'interno del manuale
- ❑ *which* Cerca il path di un file eseguibile
- ❑ *whereis* Determina il percorso di binari, sorgenti e documentazione per un programma



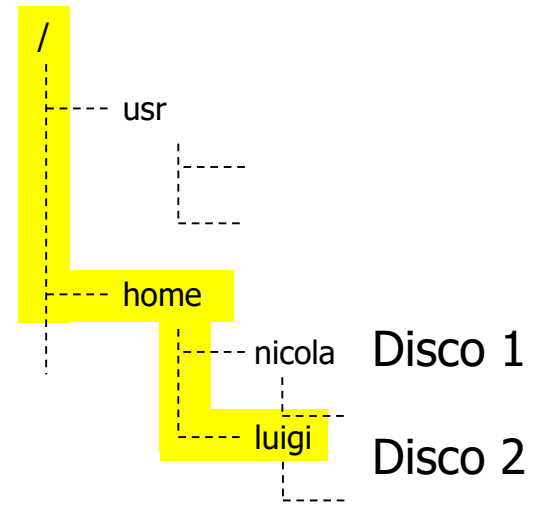
```
osboxes@osboxes: ~/Desktop
File Edit Tabs Help
osboxes@osboxes:~/Desktop$ apropos firewall
ufw (8)          - program for managing a netfilter firewall
osboxes@osboxes:~/Desktop$ man ufw
osboxes@osboxes:~/Desktop$ which ufw
/usr/sbin/ufw
osboxes@osboxes:~/Desktop$ whereis ufw
ufw: /usr/sbin/ufw /etc/ufw /lib/ufw /usr/share/ufw /usr/share/man/man8/ufw.8.gz
osboxes@osboxes:~/Desktop$
```





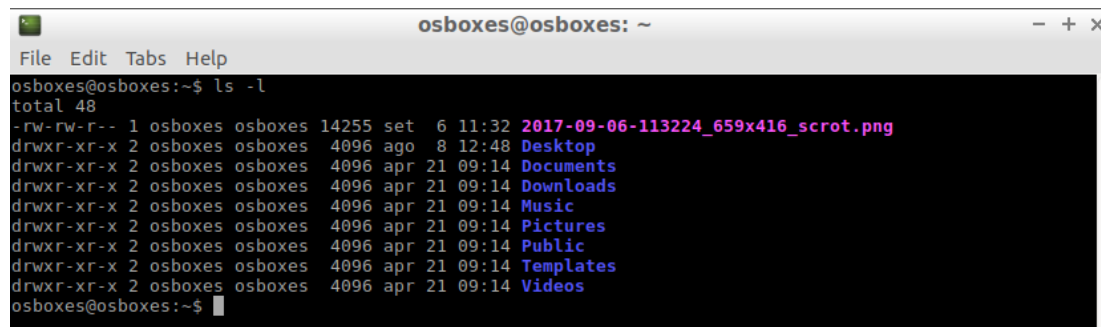
## Gestione del File System

- Il File System è organizzato in file e directory (cartelle)
- L'intero File System ha origine dalla directory root ( / )
- Ogni utente ha una directory home con il proprio nome all'interno della directory */home*
  - Nella cartella home dell'utente (es. */home/luigi*) la cartella Desktop/Scrivania (*/home/luigi/Desktop*) contiene tutti gli elementi presenti sul Desktop della GUI dell'utente
- Il simbolo *~* è un alias per la home dell'utente corrente  
es. utente osboxes sono equivalenti:  
*/home/osboxes/Desktop/nomefile*  
*~/Desktop/nomefile*
- Nei nomi dei file si distingue tra maiuscole e minuscole
  - "MIOFILE" ≠ "MioFile"
- Per spostarsi nel file system si usa *cd*
  - *cd /home/luigi*
  - *cd ..*
- Per conoscere la directory attuale *pwd* (print working directory)



- Definiscono le operazioni possibili su un file/directory da parte di un utente
- I permessi su una risorsa sono definiti e influenzano tre categorie:
  - **User:** è il proprietario del file (**u**)
  - **Group:** sono tutti gli utenti appartenenti allo stesso gruppo di cui fa parte lo User (**g**)
  - **Other:** sono tutti gli utenti che non appartengono ad una delle precedenti categorie (**o**)
- Per ogni file/directory possono essere definiti i seguenti permessi:
  - **Letture (r)** : consente di accedere al contenuto del file/directory (listing)
  - **Scrittura (w)** : consente di modificare il contenuto del file/directory (delete/create)
  - **Esecuzione (x)** : consente di eseguire il file come comando/accedere alla directory
- Per visualizzare i permessi su una risorsa eseguiamo: **ls -l**

```
-rw-rw-r-- 1 osboxes osboxes 14255 set  6 11:32 2017-09-06-113224_659x416_scrot.png
drwxr-xr-x 2 osboxes osboxes  4096 ago  8 12:48 Desktop
```



## MODIFICARE I PERMESSI

- Dato il file «pippo»

```
$ chown <nuovoproprietario> pippo
```

```
$ chgrp <nuovogrupo> pippo
```

```
$ chmod [ugoa][+ -=][rwxXst][,...] pippo
```

- Esempi:

```
$ cd /qualche/posizione/
```

```
$ chown -R pippo:pluto . (tutte le sottodirectory a partire dalla corrente owner=pippo  
gruppo=pluto)
```

```
$ chmod -R ug+rwX,o=rX . (per tutti i file a partire dalla directory corrente, per u e g  
aggiungiamo rwX, per o permessi solo di r ed X)
```

Anche

```
chmod 644 pippo (rw-r--r--)
```



- CP: copiare file da una directory source a una destination

```
cp [OPTION]... [-T] SOURCE DEST  
cp ~/Desktop/luigi.txt ~/Documents/
```

- MV: sposta un file da una directory source a una destination (rimuove il file dall'origine)

```
mv [OPTION]... [-T] SOURCE DEST  
mv ~/Documents/luigi.txt ~/Documents/myDocuments
```

- RM: elimina un file (o directory)

```
rm [OPTION]... [FILE]  
rm -R ~/Documents/* (Elimina tutto il contenuto della  
directory Documents)
```

- TOUCH: aggiorna il tempo di ultima modifica di un file al tempo attuale (se il file non esiste lo crea)

```
touch [OPTION]... FILE  
touch luigi.txt
```

- MKDIR: crea una directory

```
mkdir [OPTION]... DIRECTORY.
```

```
mkdir ~/Documents/myDocuments (Crea la sottodirectory  
myDocuments in Documents)
```

- RMDIR : elimina una directory

```
rmdir [OPTION]... DIRECTORY
```

```
rmdir ~/Documents/myDocuments (Cancella la sottodirectory  
myDocuments in Documents)
```

- compress
- gzip
  - Compressione:
    - `gzip <filename>`
  - Decompressione:
    - `gzip -d <filename>`
    - `gunzip <filename>`
  - Lettura:
    - `zcat <filename>`
- bzip2
  - Molto simile a gzip ma:
    - > tempo di esecuzione
    - > compressione

# ARCHIVI E BACKUP

- Archivio:
  - file unico contenente diversi files
- Creazione di un archivio:
  - `tar -cf <filename> <directory>`
- Estrazione di un archivio:
  - `tar -xf <filename>`
- Compressione di un archivio :
  - `gzip <filename>`



- ❑ Creare nella propria home, sotto la cartella Documents una sottocartella myDocuments
  - Verificare che sia stata creata la cartella myDocuments
- ❑ Creare un file luigi.txt
  - Verificare che sia stato creato il file luigi.txt
- ❑ Spostare il file luigi.txt nella cartella myDocuments
  - Verificare che sia stato spostato il file
- ❑ Cancellare la cartella myDocuments e i file/sottocartelle in essa contenuti
  - Verificare che sia stata cancellata la sottocartella e tutti i file

- Creare nella propria home, sotto la cartella Documents una sottocartella myDocuments

```
mkdir ~/Documents/myDocuments
```

  - Verificare che sia stata creata la cartella myDocuments

```
ls -al ~/Documents
```

 (Verificare l'attributo d)
  
- Creare un file luigi.txt

```
touch luigi.txt
```

  - Verificare che sia stato creato il file luigi.txt

```
ls
```
  
- Spostare il file luigi.txt nella cartella myDocuments

```
mv ./luigi.txt ~/Documents/myDocuments
```

  - Verificare che sia stato spostato il file

```
ls
```

```
ls ~/Documents/myDocuments
```
  
- Cancellare la cartella myDocuments e i file/sottocartelle in essa contenuti

```
rm -R ~/Documents/*
```

  - Verificare che sia stata cancellata la sottocartella e tutti i file

```
ls ~/Documents
```

## ESERCIZIO

- Creare il file `sorgente1.txt` usando il comando `cat`
- Verificare l'operazione con il comando `ls`
- Creare la directory `mydir` con il comando `mkdir`
- Creare nella directory i file `sorgente1.txt` e `sorgente2.txt` (copia del primo)
- Creare il file `destinazione.txt`, concatenazione dei file `sorgentex.txt`
- Eseguire un `tar` della directory e comprimere
- Copiare il tutto su di un'altra directory

- ❑ È un programma in esecuzione
- ❑ Ha un proprio identificatore, chiamato pid
- ❑ Ha un proprietario - generalmente il creatore – e va in esecuzione tipicamente con i diritti del proprietario
- ❑ Quando tenta di aprire un file vengono controllati i diritti
- ❑ Siccome in Unix anche i device sono dei file, questa strategia è applicabile praticamente a qualsiasi tentativo di accesso alle risorse del sistema
- ❑ L'esecuzione dei processi lanciati da una shell può essere controllata mediante le facility del job control
- ❑ I processi in esecuzione sono visibili mediante il comando `ps`

- Creare un file `processiattivi.txt` contenente una lista dei processi in esecuzione dell'utente `loginname` ordinata per pid decrescente

# SOLUZIONE

```
Romano@DESKTOP-GTD31RB MINGW64 /d/lrom/teaching/SSI/20180927/LinuxIntro  
$ ps | sort -r > processiattivi.txt
```

```
Romano@DESKTOP-GTD31RB MINGW64 /d/lrom/teaching/SSI/20180927/LinuxIntro  
$ cat processiattivi.txt
```

15356	8796	15356	11464	pty0	197609	06:12:52	/usr/bin/ps
12300	8796	15356	12300	pty0	197609	06:12:52	/usr/bin/bash
8796	7284	8796	2712	pty0	197609	05:51:20	/usr/bin/bash
7284	1	7284	7284	?	197609	05:51:20	/usr/bin/mintty
PID	PPID	PGID	WINPID	TTY	UID	STIME	COMMAND

- Creare uno script che esegua l'esercizio precedente (creare un file `processiattivi.txt` contenente una lista dei processi in esecuzione dell'utente `loginname` ordinata per pid decrescente)

# UCCIDERE UN PROCESSO - SIGNALS

- Due passi:

- `ps processname`
- `kill -9 pid`

- Segnali più comuni:

- SIGHUP
- SIGINT
- SIGQUIT
- SIGFPE
- SIGKILL
- SIGSEGV
- SIGPIPE
- SIGTERM



## □ Autocompletamento

- Mentre si sta scrivendo un comando o un nome di file/cartella, digitando il tasto TAB il nome, se univoco, sarà automaticamente completato, se non univoco, al secondo TAB sarà presentata una lista dei nomi in conflitto (così da digitare altre lettere e riprovare l'autocompletamento)

- `cd ~/De <TAB> ==> cd ~/Desktop`
- `cd ~/D <TAB><TAB> ==>`  
`Desktop/        DIRECTORY./ Documents/ Downloads/`

## □ Autocompletamento

- Mentre si sta scrivendo un comando o un nome di file/cartella, digitando il tasto TAB il nome, se univoco, sarà automaticamente completato, se non univoco, al secondo TAB sarà presentata una lista dei nomi in conflitto (così da digitare altre lettere e riprovare l'autocompletamento)

- `cd ~/De <TAB> ==> cd ~/Desktop`
- `cd ~/D <T AB><TAB> ==>`  
Desktop/      DIRECTORY./ Documents/ Downloads/

## □ History

- Cliccando la freccia in alto, viene ripetuto l'ultima istruzione
- Cliccando ancora, si scorre a ritroso la storia delle istruzioni usate
- Digitando "history" si ha una lista numerata di istruzioni usate
  - Con il comando: `!XX` si ripete l'istruzione numero XX (XX è uno dei numeri che compare nella lista)

## □ Autocompletamento

- Mentre si sta scrivendo un comando o un nome di file/cartella, digitando il tasto TAB il nome, se univoco, sarà automaticamente completato, se non univoco, al secondo TAB sarà presentata una lista dei nomi in conflitto (così da digitare altre lettere e riprovare l'autocompletamento)
  - `cd ~/De <TAB> ==> cd ~/Desktop`
  - `cd ~/D <TAB><TAB> ==>`  
Desktop/      DIRECTORY./ Documents/ Downloads/

## □ History

- Cliccando la freccia in alto, viene ripetuto l'ultima istruzione
- Cliccando ancora, si scorre a ritroso la storia delle istruzioni usate
- Digitando "history" si ha una lista numerata di istruzioni usate
  - Con il comando: `!XX` si ripete l'istruzione numero XX (XX è uno dei numeri che compare nella lista)

## □ Copia e Incolla

- Lasciando il cursore nel punto in cui si vuole scrivere, con il mouse selezioniamo un testo (click tasto sinistro e trascinamento sul testo da selezionare), cliccando con il tasto centrale del mouse (solitamente la rotellina o se non presete click contemporaneo dei due tasti sx e dx) il testo selezionato verrà copiato alla posizione del cursore
  - Attenzione, la combinazione `Ctrl+C` usata in Windows, in Linux provoca la terminazione del processo corrente (chiusura del programma attivo)

- Un programma tipicamente riceve dati dallo «standard input» (tipicamente la tastiera) e restituisce output su «standard output» (tipicamente schermo);
  - Esiste anche uno «standard error output» (tipicamente schermo) dove inviare i messaggi di errore
- È possibile redirigere gli standard input ed output in modo tale che il programma riceva input da sorgenti diverse dalla tastiera e fornisca output a destinazioni diverse dallo schermo
- Redirezione dell'output
  - `programma > nomefile` redirige l'output di programma memorizzandolo nel file `nomefile`

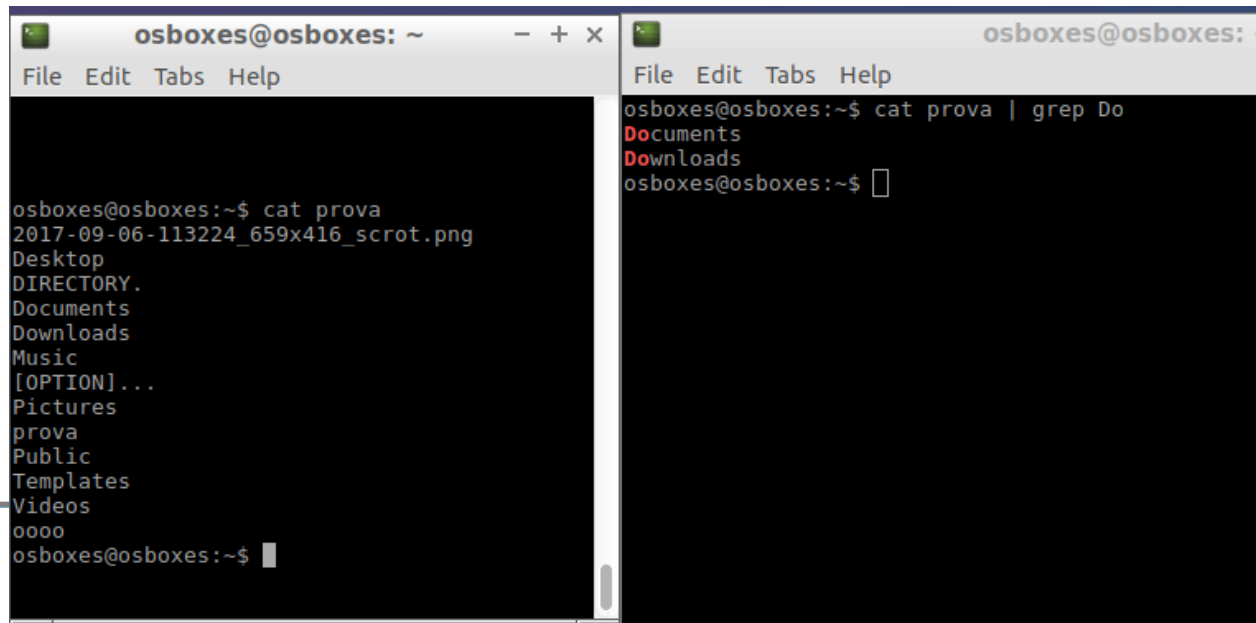
es.

`ls > prova`      l'output del programma ls inviato al file `prova` (se non esiste viene creato)

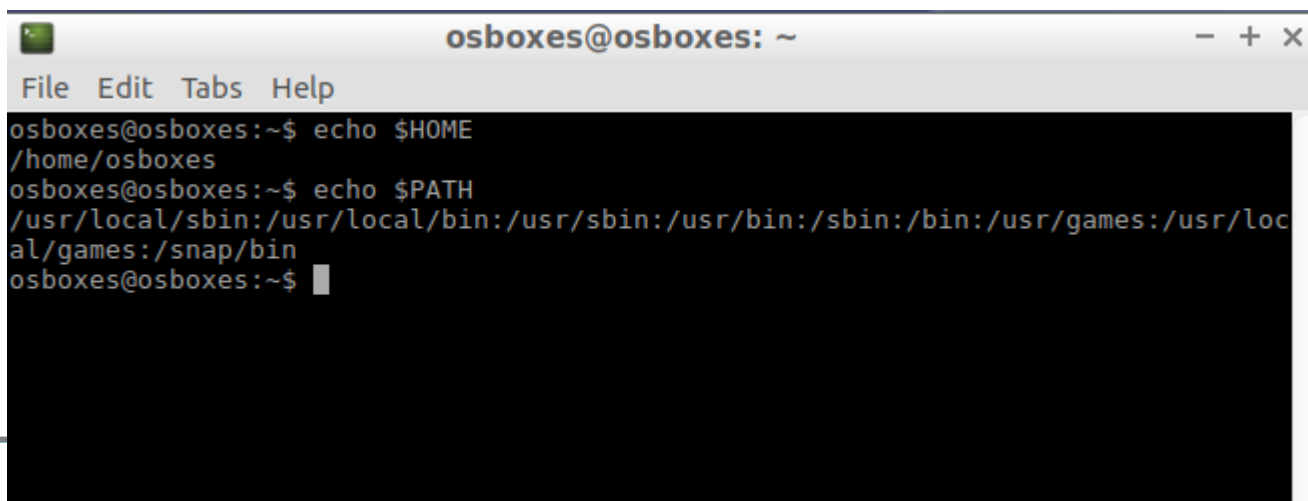
`cat prova`      mostra su standard output il contenuto del file `prova`

- Utilizzando l'operatore `>>` l'output viene aggiunto in «append» al contenuto del file destinazione

- Redirezione dell'Input
  - `programma < file`
  
- Una pipe consiste nell'incanalare l'output di un programma nell'input dell'altro
  - **Sindassi:** `programma 1 | programma 2`  
L'output di programma 1 è inviato in input a programma 2
  - **Es.**  
`ls > prova`  
`cat prova | grep Do` (ritorna solo le righe che contengono «Do»)
    - `grep PATTERN nomefile` cerca il PATTERN specificato nel file nomefile



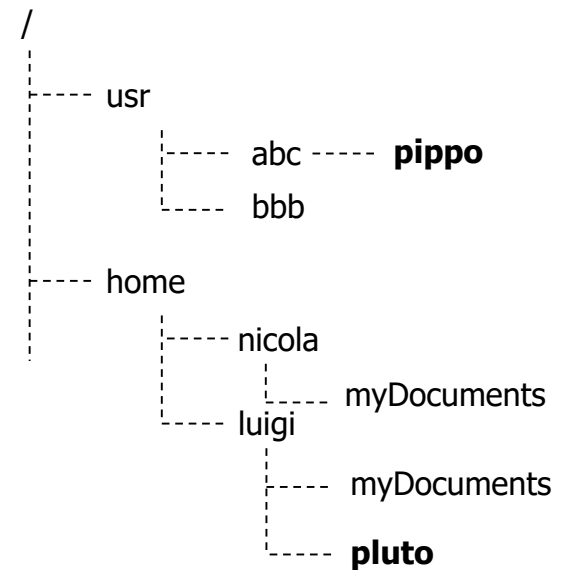
- Sono dei parametri di configurazione dell'ambiente (shell) rese disponibili ai processi
- Sono definite nella forma `NOME_VAR=VALORE` e il loro valore può essere riferito mediante `$`
- Per visualizzare il valore della variabile d'ambiente possiamo eseguire il comando
  - `echo $NOME_VARIABILE`
- Due importanti variabili d'ambiente sono: **HOME** e **PATH**
  - **HOME**: è inizializzata con la home directory dell'utente corrente
  - **PATH**: è il percorso, o l'insieme dei percorsi, in cui la shell cerca i comandi da eseguire (se non è fornito l'intero percorso del comando)



```
osboxes@osboxes: ~  
File Edit Tabs Help  
osboxes@osboxes:~$ echo $HOME  
/home/osboxes  
osboxes@osboxes:~$ echo $PATH  
/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin:/usr/games:/usr/local/games:/snap/bin  
osboxes@osboxes:~$
```

□ Si consideri l'albero di file system a lato (in grassetto i file eseguibili) e supponiamo di essere nella directory home, utente luigi

1. Eseguire il programma pippo
2. Eseguire il programma pluto
3. Spostarsi nella cartella myDocuments usando la cartella speciale «..»
4. Spostarsi nella cartella abc
5. Modificare i permessi del file pippo in modo da renderlo non eseguibile e in sola lettura
6. Spostarsi nella «root»
7. Cancellare la sottocartella usr con tutti i sottoelementi



PATH=/usr/abc/



## *The Magic of SUID*



- Ogni utente ha uno Username ed uno User ID

```
colui@COLUI-SURFACE:~$ id -u colui
1000
colui@COLUI-SURFACE:~$ whoami
colui
colui@COLUI-SURFACE:~$ id -u `whoami`
1000
colui@COLUI-SURFACE:~$ id -u `sudo whoami`
0
```

- /etc/passwd contiene informazioni su utenti, uid, ecc...

```
colui@COLUI-SURFACE:~$ cat /etc/passwd
root:x:0:0:root:/root:/bin/bash
daemon:x:1:1:daemon:/usr/sbin:/usr/sbin/nologin
bin:x:2:2:bin:/bin:/usr/sbin/nologin
sys:x:3:3:sys:/dev:/usr/sbin/nologin
sync:x:4:65534:sync:/bin:/bin/sync
games:x:5:60:games:/usr/games:/usr/sbin/nologin
man:x:6:12:man:/var/cache/man:/usr/sbin/nologin
lp:x:7:7:lp:/var/spool/lpd:/usr/sbin/nologin
mail:x:8:8:mail:/var/mail:/usr/sbin/nologin
news:x:9:9:news:/var/spool/news:/usr/sbin/nologin
uucp:x:10:10:uucp:/var/spool/uucp:/usr/sbin/nologin
wwwrun:x:11:11:wwwrun:/bin:/usr/sbin/nologin
```

- Uno User può appartenere ad uno o più gruppi
  - Il concetto di Gruppo consente all'amministratore di gestire contemporaneamente i permessi per più utenti

□ Alla luce di quanto detto, rivediamo come sono gestiti i permessi in Linux

Per visualizzare i permessi su una risorsa eseguiamo: ***ls -l***

```
-rw-rw-r-- 1 osboxes osboxes 14255 set 6 11:32 scrot.png
drwxr-xr-x 2 osboxes osboxes 4096 ago 8 12:48 Desktop
```

	owner	group							
-rw-rw-r--	1	osboxes	osboxes	14255	set	6	11:32	scrot.png	
drwxr-xr-x	2	osboxes	osboxes	4096	ago	8	12:48	Desktop	

**Others:** gli altri utenti possono solo leggere il file scrot.png; leggere ed entrare nella directory Desktop

**Group:** Gli utenti appartenenti al Gruppo osboxes possono leggere e scrivere il file scrot.png; listare e entrare nella directory Desktop

**Owner:** Il proprietario, osboxes, può leggere e scrivere il file scrot.png che non può essere eseguito. Lo stesso utente è proprietario della directory Desktop che può essere letta (listing dei file), modificata (creazione/cancellazione di un file al suo interno), eseguita (cd /Desktop)

## STICKY BIT

- ❑ Partiamo dal problema...
- ❑ In linux più utenti possono avere lo stesso UID
- ❑ Nel file /etc/shadow sono conservati gli hash delle password degli utenti
- ❑ Se un utente fosse in grado di modificare il file potrebbe, ad esempio, cancellare la pwd di un altro utente
- ❑ Per questo motivo solo “root” deve poter modificare /etc/shadow
- ❑ Ma ogni utente deve poter modificare la propria password

```
colui@COLUI-SURFACE:~$ cat /etc/shadow
cat: /etc/shadow: Permission denied
colui@COLUI-SURFACE:~$ sudo cat /etc/shadow | grep colui
colui:$6$zCezfzvF$diGLwFrhhhfXGPFU/rR8LqbPkSI9osQzS/htwbYizuCRr2APWYZ14YuILNu2E92S9ao6x7wCG07Z4dcJ2pU2r/:17500:0:99999:7
:::
```

- ❑ La soluzione è lo STICKY BIT



- ❑ Per modificare la propria password si usa il comando:
  - `/usr/bin/passwd`
- ❑ Guardiamo i permessi del comando `passwd`

```
colui@COLUI-SURFACE:~$ ls -al `which passwd`  
-rwsr-xr-x 1 root root 54256 May 17 2017 /usr/bin/passwd
```

- ❑ Il programma appartiene a root:root
- ❑ Gli utenti OTHER possono eseguire il comando ... ma il comando in esecuzione avrebbe UID dell'utente che lo lancia...per cui non può accedere al file shadow
- ❑ Notiamo che il bit X dell'utente OWNER anziché avere una x ha una S (Sticky bit)
  - `chmod +S filename`
- ❑ In questo caso, il programma, lanciato da OTHER, verrà eseguito con i diritti dell'OWNER (root) così da poter modificare il file `passwd` (il programma avrà UID di OTHER e eUID – effective UID di OWNER)

- [https://www.debian.org/doc/manuals/debian-reference/ch01.it.html#\\_unix\\_like\\_filesystem](https://www.debian.org/doc/manuals/debian-reference/ch01.it.html#_unix_like_filesystem)
- <https://www.freebsd.org/doc/it/books/unix-introduction/special-features-file-redir.html>



## Approfondimenti

- ❑ Il comando `sudo` sta per "superuser do"
- ❑ Al prompt richiede l'immissione della tua password personale e conferma la richiesta di esecuzione del comando facendo un check in un file chiamato `sudoers`, che è configurato dal system administrator
- ❑ Nel file `sudoers` il system administrator può dare accesso a specifici utenti o gruppi ad alcuni o a tutti i comandi, senza che questi debbano conoscere la password di root
- ❑ Esegue anche il logging dei comandi e degli argomenti, affinché sia possibile sapere chi ha fatto cosa e quando
- ❑ Per usare `sudo`, al command prompt digitare:

```
sudo command
```

- ❑ Il comando `sudo` è un "setuid". I suoi permessi sono:  

```
---s---x--x    1 root    root    81644 Jan 14 15:36 /usr/bin/sudo
```
- ❑ Chiunque lo può eseguire e durante l'esecuzione "diventa" root

## IMPERSONATION: IL COMANDO «SU»

- ❑ Il comando su sta per "switch user"
- ❑ Per usare su, al command prompt digitare:

```
su user
```

- ❑ Se non si specifica nulla, viene impersonato l'utente root
- ❑ Impersonare l'utente root è una manovra potenzialmente molto pericolosa



- mount** Collegamento (innesto) di un file system in quello globale
- umount** Distacco di un file system da quello globale
- df** Utilizzo del disco

**mount -a [options]**

**mount [options] [device] [directory]**

Nella prima forma vengono montati tutti i file system elencati in un file di configurazione (/etc/fstab o /etc/vfstab) che non abbiano l'opzione "no auto".

Nella seconda forma, permette di attaccare il file system presente sul dispositivo device al file system globale di Unix, nella posizione specificata da directory. Senza argomenti permette di visualizzare tutti i file montati.

## Alcune opzioni

**-t(-F)** Specifica il tipo di file system. Se il nome del tipo di file system viene preceduto dalla sigla no, si intende che quel tipo deve essere escluso.

**-v** mostra diagnostica dettagliata

**-f(-V)** Simula solo il montaggio; utile per valutare gli effetti del comando, soprattutto se usata insieme all'opzione precedente

**-f(-p)** mostra i file system montati in un formato più semplice

**umount -a [options]**

**umount [options] [device | directory]**

Esegue l'operazione inversa di mount: smonta i file system.

Nella prima forma, smonta i file system elencati nell'apposito file di configurazione (/etc/mstab o /etc/mnttab), eventualmente in base alle opzioni, con l'eccezione di quelli necessari al funzionamento del sistema.

Nella seconda forma, smonta il file system individuato dal device device o dal punto di attacco *directory*.

L'operazione (a meno di usare l'opzione *-f*, valida solo per alcuni sistemi) avviene solo se non ci sono più attività in corso su quei file system.

Nota:

Le operazioni di (s)montaggio con (u)mount richiedono privilegi speciali.

## **Esempi di esecuzione**

### **Monta il floppy su un sistema Linux x86 in /tmp**

```
mount /dev/fd0 /tmp
```

### **Smontaggio del dispositivo precedente**

```
umount /tmp
```

### **Monta un CD nella directory /mnt/cdrom**

```
mount /dev/dsk/c0t0d0s0 /mnt/cdrom
```

### **Forza lo smontaggio del precedente dispositivo**

```
umount -f /mnt/cdrom
```

## df [options] [device...]

Permette di conoscere lo spazio a disposizione di una o di tutte le partizioni che risultano montate.

Se non vengono indicati i nomi dei dispositivi, si ottiene l'elenco completo di tutti i dispositivi attivi, altrimenti l'elenco si riduce a quelli specificati.

L'opzione "h"

-h Emette le informazioni su dimensione, spazio occupato e spazio libero in un formato facilmente leggibile

- `pwd` Emette il percorso della directory corrente
- `ls` Elenca il contenuto di uno o più directory
- `mkdir` Crea un directory
- `rmdir` Cancella un directory
- `cd` Cambia directory di lavoro
- `du` Calcola lo spazio utilizzato da una serie di directory e subdirectory

# PWD (*PRINT WORKING DIRECTORY*)

63

**pwd**

Stampa il path della directory di lavoro corrente

## Esempi di esecuzione

```
gio$ pwd  
/home/gio  
gio$
```

## ls [options][file...]

Lista, in una modalità dipendente dalle opzioni selezionate, informazioni sui file indicati. Se *file* è una directory, mostra i nomi dei file in essa contenuti ed eventuali informazioni aggiuntive. Se *file* non è una directory, ne mostra il nome ed eventuali informazioni aggiuntive. Se non è indicato alcun argomento, lista il contenuto della working directory.

### Alcune opzioni

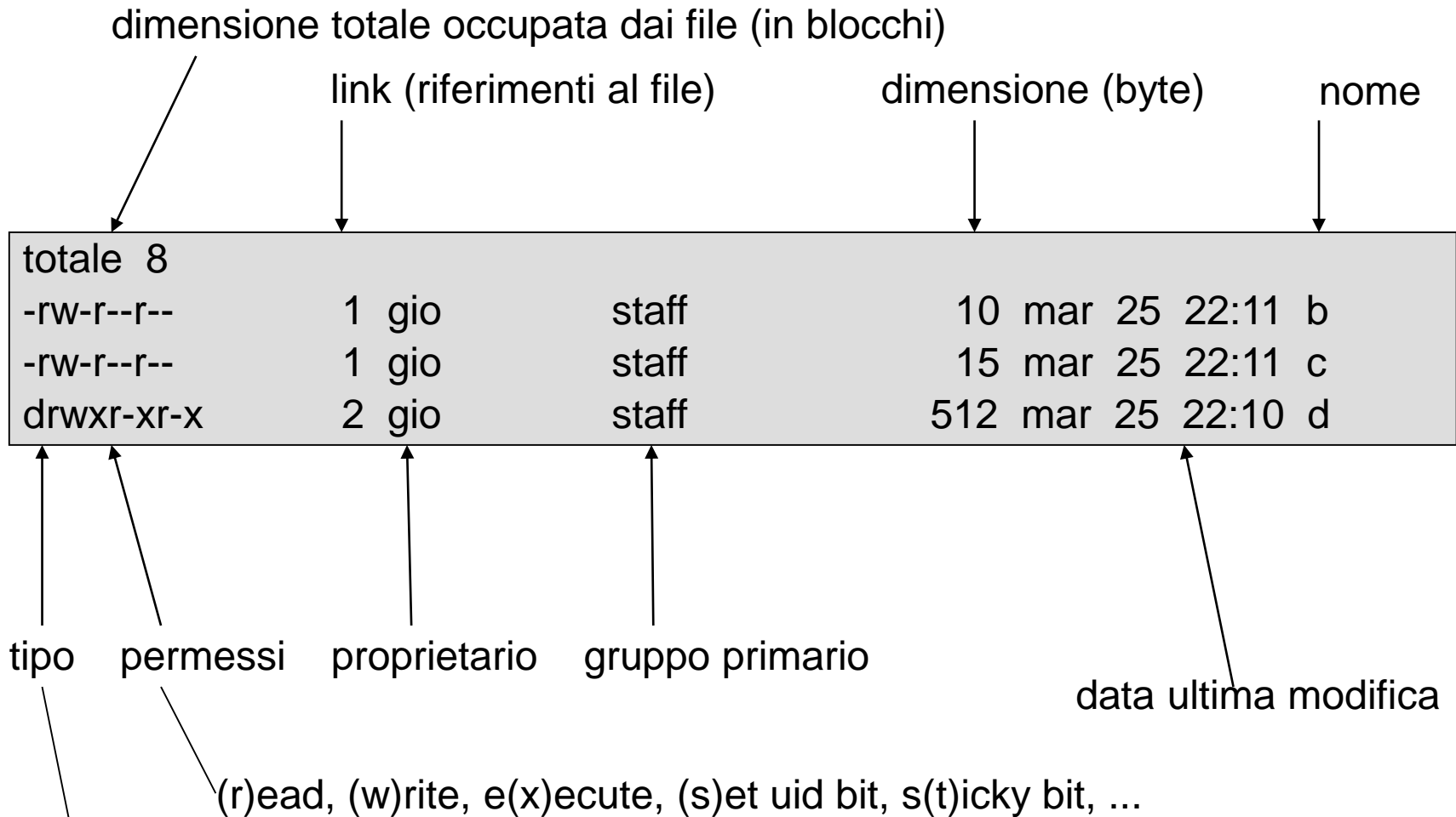
- a mostra anche i file nascosti (file il cui nome incomincia col punto)
- l formato esteso, con informazioni su modo, proprietario, dimensione, etc dei file
- s fornisce la dimensione in blocchi dei file
- t lista i file nell'ordine di modifica (prima il file modificato per ultimo) -1 elenca i file in una singola colonna
- F aggiunge / al nome delle directory e \* al nome dei file eseguibili
- R si chiama ricorsivamente su ogni subdirectory
- i fornisce l'i-number (numero di i-nodo) dei file



The screenshot shows a terminal window titled "savio@savio-desktop: ~/corso - Shell - Konsole". The terminal displays the following commands and their outputs:

```
savio@savio-desktop:~/corso$ ls
dir          Makefile    Makefile3   pargs       pargs.o     test1.h     test3.c
leggifile.c Makefile2   nuovadir    pargs.c     test1.c     test2.c
savio@savio-desktop:~/corso$ ls -la
totale 60
drwxr-xr-x  3 savio savio 4096 2008-03-04 23:27 .
drwxr-xr-x 12 savio savio 4096 2008-03-04 16:39 ..
drwxr-xr-x  2 savio savio 4096 2008-03-04 16:51 dir
-rwxr-xr-x  1 savio savio 1019 2007-03-14 16:06 leggifile.c
-rwxr-xr-x  1 savio savio  269 2007-03-20 19:33 Makefile
-rwxr-xr-x  1 savio savio  287 2007-03-14 16:15 Makefile2
-rwxr-xr-x  1 savio savio  268 2008-03-04 23:20 Makefile3
lrwxrwxrwx  1 savio savio    3 2008-03-04 16:52 nuovadir -> dir
-rwxr-xr-x  1 savio savio 7091 2007-03-20 19:34 pargs
-rwxr-xr-x  1 savio savio  223 2008-03-04 23:07 pargs.c
-rwxr-xr-x  1 savio savio  984 2007-03-20 19:33 pargs.o
-rwxr-xr-x  1 savio savio  273 2008-03-04 23:21 test1.c
-rw-r--r--  1 savio savio   18 2008-03-04 23:07 test1.h
-rwxr-xr-x  1 savio savio   91 2008-03-04 23:12 test2.c
-rwxr-xr-x  1 savio savio   92 2008-03-04 23:13 test3.c
savio@savio-desktop:~/corso$
```





(d)irectory, (l)ink, (c)haracter special file, (b)lock special file, (-) ordinary file,

# MKDIR (MAKE DIRECTORY)

67

**mkdir [options] directory...**

Crea le directory passate come argomento, secondo le opzioni specificate

Alcune opzioni

**-m mode** Specifica il modo `mode` per la directory da creare

**-p** Genera anche tutte le directory gerarchicamente superiori necessarie. Il modo per le directory "genitrici" sono impostati al valore della maschera di modo, modificato in modo da aggiungere i permessi `w` ed `x` per il proprietario

## Esempi d' uso

**Crea la directory `pippo` a partire da quella corrente con modo `755` :**

```
mkdir -m 755 pippo
```

**Crea le directory `esame`, `prove` ed `esercizi` a partire da quella corrente:**

```
mkdir -p esame/prove/esercizi
```



# RMDIR (REMOVE DIRECTORY)

68

**rmdir [options] directory...**

Rimuove le directory indicate secondo le opzioni specificate.  
Per poter essere rimosse, le directory devono essere vuote e la loro directory genitrice deve avere il permesso in scrittura

Alcune opzioni

-p rimuove ricorsivamente directory e subdirectory contenute nel path directory (ammesso che non contengano altri file)

(-s) non mostra eventuali messaggi di errore se è attiva l'opzione -p

## Esempi d' uso

**Rimuove la directory pippo a partire da quella corrente :**

```
rmdir pippo
```

**Rimuove le directory esame, prove ed esercizi a partire da quella corrente:**

```
rmdir -p esame/prove/esercizi
```

**cd [directory]**

La directory specificata diviene la working directory

Se nessuna directory è specificata, si viene posizionati nella home directory

## Esempi d' uso

```
gio$ cd /home/gio/esempio
gio$ pwd
/home/gio/esempio
gio$ cd
gio$ pwd
/home/gio
gio$ cd esempio
gio$ pwd
/home/gio/esempio
```

**du [options] file...**

Emette informazioni sull'utilizzo dello spazio disco da parte di file, che può rappresentare un file o una directory.

Se **file** è una directory, l'output standard prevede l'elencazione di tutti i file e le subdirectory in modo ricorsivo, con il calcolo della dimensione di ciascuno.

Alcune opzioni

-s invece dell'output standard, mostra solo lo spazio disco occupato complessivamente da file.

-k esprime le dimensioni in kibi byte (1024 byte), invece che in blocchi (512 byte)

- touch        Cambia il tempo di ultimo accesso ad un file o lo crea
- cp        Copia un file in un altro
- ln        Crea un collegamento
- mv        Rinomina un file
- rm        Cancella un file
- chown     Cambia proprietario e gruppo primario di un file
- chmod     Cambia i permessi di un file
- find     Ricerca file in base ad opportuni criteri
- tar        Archivia un file
- gzip,gunzip    Comprime (decomprime) un file

## **touch [options] [date\_time] file...**

Cambia sia il tempo di ultimo accesso che di ultimo aggiornamento dei file.

Se non viene specificato il parametro `date_time`, viene utilizzato il valore ottenuto dall'orologio del sistema nel momento in cui viene eseguito il comando.

Se si specificano file che non esistono, questi vengono creati vuoti. Il parametro `date_time` deve avere il formato `MMDDhhmm[YY]`

Alcune opzioni:

`-a` cambia solo il tempo di ultimo accesso `-c` se file non esiste, esso non viene creato `-m` cambia solo il tempo di ultima modifica



## cp [ options] source... target

Copia un file in un altro file oppure uno o più file in una directory

- se vengono specificati solo i nomi di due file, il primo viene copiato sul secondo
- se vengono specificati solo due nomi, e se il secondo nome indicato è una directory, source viene copiato con lo stesso nome nella directory target. Se source è una directory, la copia avviene solo con opzioni particolari.

Se vengono indicati più di due nomi, il file target deve essere una directory e vengono generate le copie dei source in target. In mancanza di opzioni particolari, le directory non vengono copiate.

### Alcune opzioni

- r se source e target sono directory, copia ricorsivamente source, i suoi file e le sue subdirectory in target
- i opera in modo interattivo, chiedendo una conferma se la copia comporta la cancellazione di un target preesistente

## Esempi d' uso

**Copia il file pippo nella directory tmp con il nome pippo.bck:**

```
cp pippo /tmp/pippo.bck
```

**Copia il file /tmp/pippo.bck e la directory dir nella directory nuovadir:**

```
cp -r /tmp/pippo.bck dir nuovadir
```

**Copia il file pippo nel file pippo1:**

```
cp pippo pippo1
```

## Nota:

**Una copia duplica i dati relativi ai source in target.**

**Se target rappresenta un nuovo file, viene impegnato un nuovo inode**

**mv [options] source... destination**

Rinomina (sposta) file o directory.

Se vengono specificati solo i nomi di due elementi, source viene rinominato in destination, oppure in destination/source, a seconda che destination indichi un file o una directory. Qualora destination denoti un file preesistente, questo non sarà più accessibile come tale, e non sarà più accessibile in alcun modo se destination era il suo unico nome.

Se vengono indicati più di due elementi, destination deve essere una directory, e source\_1...source\_n vengono rinominati come destination/source\_1...destination/source\_n.

Nel caso che source e destination appartengono a due diversi file system, il comando effettua un vero e proprio spostamento dati tra i due file system. In tal caso vengono spostati solo i file ordinari, quindi: né collegamenti, né directory.

Alcune opzioni:

- i il comando chiede conferma all'utente qualora destination è un file preesistente

**rm [options] file...**

**rm -R [options] dirname... [file... .]**

Elimina i file o le directory indicati come argomento. Nella prima forma elimina uno o più file.

Nella seconda forma elimina ricorsivamente una o più directory con i loro contenuti; inoltre, opzionalmente elimina uno o più file

Alcune opzioni:

-i chiede conferma prima di rimuovere ogni file

# CHOWN (CHANGE OWNER)

**chown [options] [user][:[group]] file...**

Cambia proprietario e/o gruppo primario per uno o più file.

Se dopo : non segue il nome del gruppo, viene attribuito il gruppo principale a cui appartiene user.

Se prima di :group non viene indicato il nome dell'utente, viene cambiato solo il gruppo primario.

Alcune opzioni:

-R opera ricorsivamente su directory e subdirectory

-h se file è un link simbolico, opera su di esso piuttosto che sul file da esso referenziato

Nota:

Alcuni sistemi operativi (es. Solaris) hanno una opzione di configurazione per permettere i cambi di proprietà solo all'utente root.

**chmod [options] mode file...**

Cambia la modalità dei permessi sui file indicati come argomento

Il parametro mode può essere espresso sia in forma ottale che simbolica. In quest' ultimo caso, la stringa dei beneficiari ((u)ser,(g)roup,(o)ther) è separata da quella dei permessi ((r)ead,(w)rite,e(x)ecute) tramite i segni + o -, a seconda che i permessi vadano aggiunti o sottratti; mentre stringhe relative a beneficiari/permessi diversi sono separate da virgole, come in: u+rwx,go+rx

Alcune opzioni:

-R opera ricorsivamente su directory e subdirectory

## Esempi d'uso:

**Attribuisce al file pippo il modo di permessi rwxr-xr-x**

```
chmod 755 pippo
```

**Elimina i permessi x per il gruppo ed rx per other per pippo**

```
chmod g-x,o-rx pippo
```

## `find [pathname...] [expression]`

Discende ricorsivamente le directory specificate (pathname...), cercando tutti i file che rendono vera expression.

Molto flessibile:

- ricerca file di specificati attributi (nome, tipo, permessi, proprietario, gruppo, numero di link, dimensione, data di ultima modifica/accesso ...)
- and, or, not di attributi
- può eseguire automaticamente, o previa conferma, uno o più comandi sui file individuati
- le espressioni si ottengono combinando flag, parametri e gli operatori booleani;
- le espressioni costituite solo da un flag e da un parametro (opzionale) si dicono espressioni elementari;

- ❑ -atime [+]-n: l'accesso al file è avvenuto [più di, meno di] n giorni prima
- ❑ -ctime [+]-n: il modo del file è stato cambiato [più di, meno di] n giorni prima
- ❑ -exec command: il comando restituisce 0 come stato di uscita
- ❑ -fstype type: il filesystem cui il file appartiene è i tipo type
- ❑ -links [+]-n: il file ha [più di, meno di] n link
- ❑ -mtime [+]-n: i dati del file sono stati modificati [più di, meno di] n giorni prima
- ❑ -name pattern: pattern coincide col nome del file
- ❑ -newer file: il file corrente è stato modificato più recentemente di file
- ❑ -perm onum: i permessi del file (in notazione ottale) coincidono con onum
- ❑ -size [+]-n[c]: la dimensione del file è [più di, meno di] n blocchi (512 byte) [byte]
- ❑ -type t: il file è di tipo t . t può assumere i valori b(lock), d(irectory), p(lain file), etc.
- ❑ -user uname: il file è di proprietà dell'utente uname



## Esempi d'uso:

**Ricerca in /home/gio di file la cui dimensione è maggiore di 100 blocchi**

```
find /home/gio -size +100
```

**Come prima, ma adesso la dimensione dei file è espressa in byte**

```
find /home/gio -size +100c
```

**Ricerca dei file modificati meno di due giorni fa e che non sono di root**

```
find / -mtime -2 -a ! -user root
```

**Rimozione di file il cui ultimo accesso risale a più di 7 giorni**

```
find /home/gio -atime +7 -exec rm {} \;
```

**Rimozione di file dell'utente gio e che hanno estensione .old oppure .back**

```
find . \( '*.old' -o '*.back' \) -user gio -ok rm {} \;
```

```
tar c [options] [f tar_file] [-C directory] file..
```

```
tar x [options] [f tar_file] [file..]
```

Crea archivi, vi aggiunge o vi estrae file.

Nella prima forma crea un archivio dei file passati come argomento sul dispositivo di default, oppure - se è presente l'opzione f - nel file tar\_file. Se file è una directory, viene archiviato tutto il suo contenuto. Se è presente l'opzione -C, il comando effettua dapprima il cambio di directory in directory, eppoi archivia file. Se file è uguale a "." viene archiviato tutto il contenuto della directory in cui opera tar.

Nella seconda forma, estrae i file passati come argomento dall'archivio di default, oppure - se è presente l'opzione f - dal file tar\_file. Se file è omissso, viene estratto tutto il contenuto dell'archivio.

Esistono altre forme del comando, rispettivamente per sostituire file in un archivio (tar r), aggiornare un archivio (tar u) e mostrare il (o parte del) contenuto di un archivio (tar t).

Alcune opzioni:

-v elenca i file che vengono elaborati

**Archivia il contenuto della directory corrente sul dispositivo /dev/rmt/0**

```
tar cf /dev/rmt/0 .
```

**Archivia nel file docs.tar il contenuto della directory /home/gio/docs**

```
tar cf docs.tar -C /home/gio/docs .
```

**Visiona il contenuto dell'archivio docs.tar**

```
tar tf docs.tar
```

**Estrae il file y2004/jan.pdf dall'archivio docs.tar**

```
tar xf docs.tar y2004/jan.pdf
```

**Estrae in modalità "verbose" tutto il contenuto dell'archivio docs.tar**

```
tar xfv docs.tar
```

```
gzip [options] [file...]  
gunzip [options] [file...]
```

Vengono usati , rispettivamente, per comprimere e decomprimere i file indicati negli argomenti.

gunzip può attualmente decomprimere file compressi creati con zip, compress e pack, oltre naturalmente a quelli generati da gzip.

Alcune opzioni:

- h mostra una breve documentazione sul comando
- l per ogni file compresso, mostra informazioni circa il fattore di compressione, la dimensione del file compresso e non compresso, il nome del file non compresso, ecc.
- r se file è una directory opera ricorsivamente su ogni file e subdirectory in esso contenuti

- cat** Concatena file
- wc** Conat caratteri, parole e linee in un file
- cut** Taglie delle colonne opportune da un file di testo
- paste** Compone più file di testo
- sort** Ordina le linee di un file
- diff** Mostra le differenze di contenuto tra due file
- cmp** Confronta due file byte per byte
- grep** Cerca espressioni regolari all'interno di un file
- passwd** Cambia o visualizza la password di un utente

**cat [options] [file...]**

Concatena i file indicati come argomento, visualizzandoli attraverso lo standard output

Alcune opzioni:

- n fa precedere ogni linea di output dal numero progressivo che identifica la posizione della linea nel file concatenato
- b come l'opzione precedente, ma omette la numerazione delle linee bianche
- v mostra anche i caratteri non stampabili, ad eccezione dei caratteri di tabulazione, nuova linea e ritorno a capo

**wc [options] [file...]**

Fornisce il numero dei codici di interruzione di riga (in pratica il numero delle righe), delle parole o dei caratteri contenuti in file. Senza opzioni fornisce, nell'ordine suddetto, ciascuna delle precedenti informazioni.

Alcune opzioni:

- c emette solo il numero complessivo di caratteri di file.
- w emette solo il numero complessivo di parole in file.
- l emette solo il numero di righe in file.

## Esempi d'uso:

```
gio$ wc which_manpage
132      239      2083      which_manpage
gio$ wc -c which_manpage
2083      which_manpage
gio$
```

**cut [options] [file...]**

Estrae delle colonne specifiche dalle linee di testo che compongono file.

Alcune opzioni:

**-c char\_list** definisce gli intervalli da estrarre espressi in caratteri.

**-f field\_list** definisce gli intervalli da estrarre espressi in campi. I campi sono distinti in base a un certo carattere usato come delimitatore. Quello predefinito è il carattere di tabulazione.

**-d delimiter** definisce un delimitatore alternativo al carattere di tabulazione.

## Esempi d'uso:

**Estrae la prima colonna del file /etc/passwd**

```
cut -d: -f1 /etc/passwd
```

**Estrae i primi dieci caratteri da ogni riga del file /etc/passwd**

```
cut -c1-10 /etc/passwd
```



**paste [options] file...**

Concatena le righe corrispondenti dei file inseriti come argomento, inserendo tra di esse un delimitatore. Il delimitatore di default è il carattere di tabulazione.

Alcune opzioni:

**-d delimiter** definisce un delimitatore alternativo al carattere di tabulazione.

## Esempi d'uso:

**Concatena ciascuna riga di pippo con la corrispondente di prova**

```
paste pippo prova
```

**Come prima, ma in luogo di <TAB> è inserito ":" come delimitatore**

```
paste -d: pippo prova
```

## **sort [options] [file...]**

Permette di (ri)ordinare o fondere insieme il contenuto dei file passati come parametri, oppure di (ri)ordinare le linee passategli in input. L'ordinamento è la modalita predefinita, ed è effettuato tenendo conto delle opzioni inserite dall'utente. Il comando tratta ogni linea come un insieme ordinato di campi, separati da opportuni caratteri (di default i caratteri di separazione sono la tabulazione e lo spazio). In assenza di opzioni che definiscano diversi criteri di ordinamento, quest'ultimo avviene in base al primo campo ed è alfabetico.

Alcune opzioni:

- f ignora le differenze tra lettere minuscole e maiuscole
- n considera numerica anzichè testuale la chiave di ordinamento
- r ordina in senso decrescente anzichè crescente
- o fileout invia l'output a fileout anzichè sull'output standard
- t s usa s come separatore di campo
- k s1,s2 usa i campi da s1 a s2-1 come chiavi di ordinamento

**Ordina le linee del file /etc/passwd in base al valore del terzo campo (UID)**

```
sort -t: -k3,4 /etc/passwd
```

**Come prima, solo che ora l'ordinamento è numerico anziché alfabetico**

```
sort -t: -n -k3,4 /etc/passwd
```

**Come prima, ma seguendo l'ordinamento inverso (prima l'UID maggiore)**

```
sort -t: -n -k3,4 -r /etc/passwd
```

**Come prima, ma ora l'output è memorizzato in passwd\_reordered**

```
sort -t: -n -k3,4 -r /etc/passwd -o passwd_reordered
```

**diff [options] file\_1 file\_2**

Può funzionare con diverse modalità, stabilite in base alle opzioni, per determinare semplicemente i due file passati come parametri sono identici o meno, oppure per indicare le differenze che ci sono tra i due, con maggiore o minore dettaglio di informazioni al riguardo. Il risultato del confronto dei file viene emesso attraverso lo standard output, e di default mostra la lista di cambiamenti da apportare a file\_1 per renderlo uguale a file\_2.

Alcune opzioni:

- l produce l'output in formato esteso
- s segnala se i due file sono identici (per default non è emesso output in tal caso) -b ignora i “caratteri bianchi”, quali TAB e spazi, e considera stringhe di tali caratteri come equivalenti
- i considera equivalenti le lettere maiuscole e minuscole -r opera ricorsivamente su directory e subdirectory

```
cmp [options] file_1 file_2 [skip_1] [skip_2]
```

Confronta i due file passati come parametri, scrivendo sullo standard output solo qualora i due file presentano delle differenze. A meno che le opzioni specifichino diversamente, scrive sullo stdout il byte ed il numero di linea in cui occorre la prima differenza. I byte e le linee sono numerati a partire da 1. I parametri skip\_1 e skip\_2 sono byte offset iniziali relativi, rispettivamente, a file\_1 e file\_2. possono essere espressi sia in forma decimale che ottale, ed in quest'ultimo caso va utilizzato "0" come prima cifra.

Alcune opzioni:

- l scrive il numero di linea (in decimale) ed il byte differente (in ottale) per ogni differenza riscontrata;
- s non scrive nulla nel caso di file differenti, restituendo solo lo stato di uscita.

**grep [options] modello [file...]**

**grep [options] -e modello [file...] grep [options] -f file\_modello [file...]**

Esegue una ricerca all'interno dei file indicati come argomento oppure, nel caso che non sia specificato alcuno di tali file, all'interno dello standard input.

Il modello di ricerca può essere semplicemente il primo degli argomenti che seguono le opzioni (prima versione del comando), oppure può essere indicato precisamente come argomento dell'opzione -e, oppure ancora può essere contenuto in un file che viene indicato attraverso l'opzione -f.

Le opzioni options sono relative a modalità di visualizzazione dell'output e degli eventuali messaggi di errore.

Alcune opzioni

- v restituisce le linee che non contengono modello
- w restituisce solo le linee che hanno modello come parola completa
- x restituisce solo le linee che coincidono con modello

- ^ Inizio della linea
- \$ Fine della linea
- \< Inizio di una parola
- \> Fine di una parola
- . Un singolo carattere qualsiasi
- [stringa] Un qualunque carattere in stringa
- [~stringa] Un qualunque carattere non in stringa
- [c0-c1] Un qualunque carattere compreso tra il carattere c0 ed il carattere c1
- \ Inibisce l'interpretazione del metacarattere seguente

**Fornisce le linee del file /etc/passwd che contengono la parola gio**

```
grep -w gio /etc/passwd
```

**Come prima, ma ora sono restituite tutte le linee che hanno gio come (sub) stringa**

```
grep gio /etc/passwd
```

**Restituisce le linee che finiscono con la lettera f**

```
grep /home/gio/pippo f$
```

**Restituisce tutti i file nella directory corrente che non sono directory e che hanno il permesso in esecuzione per il proprietario**

```
ls -al | grep ^-..x
```



## **passwd [options] [name]**

Modifica o mostra la password ed i suoi attributi per l'utente corrente, oppure per l'utente il cui nome di login è name.

Solo un utente con privilegi speciali (ad es. root) può cambiare la password relativa ad utenti diversi e gli attributi di questa.

Le regole relative agli attributi di una password (lunghezza, periodo di validità, ecc.) sono stabilite in base ad opportuni file di configurazione, e sono sotto l'esclusivo controllo di utenti muniti di credenziali opportune (es. root).

La scelta di una nuova password non conforme alle suddette regole provoca il rigetto della password e la richiesta da parte del programma di una password adeguata.

Una volta lanciato, il programma richiede l'inserimento della vecchia password (se definita) ed in caso tale inserimento sia corretto, l'inserimento per due volte della nuova password.

- vi Editor di testo

## vi [-r filename] [filename...]

Editor a video presente su tutti i sistemi UNIX, che consente di richiamare l'editor di linea ex, costituendo un ambiente flessibile e potente.

Utilizza un buffer di memorizzazione temporaneo relativo a filename, modificando quest'ultimo solo quando esplicitamente richiesto dall'utente.

Con l'opzione -r consente di recuperare il contenuto del buffer, qualora l'applicativo o il sistema siano andati in crash.

Dispone di due modalità operative: modalità comando e modalità inserimento testo.

La modalità comando consente sia di spostarsi in filename, sia di modificare/cancellare singoli caratteri, sia di richiamare un comando di ex (grazie all'uso del prefisso ":").

La modalità di default è quella di comando.

La modalità inserimento permette di inserire nuovo testo e si ottiene digitando uno dei caratteri i,a,o,O (i=prima del cursore, a=dopo il cursore, o=inizio riga successiva, O=inizio riga precedente).

# Modalità di funzionamento

L'editor *vi* ha tre modalità di funzionamento:

- *command mode* è lo stato iniziale nel quale possono essere dati i comandi monocarattere in modo invisibile
- *line command mode* è lo stato in cui i comandi vengono scritti su una linea, l'ultima del video
- *insert mode* è lo stato nel quale vengono inseriti nuovi caratteri e nuove parole di testo

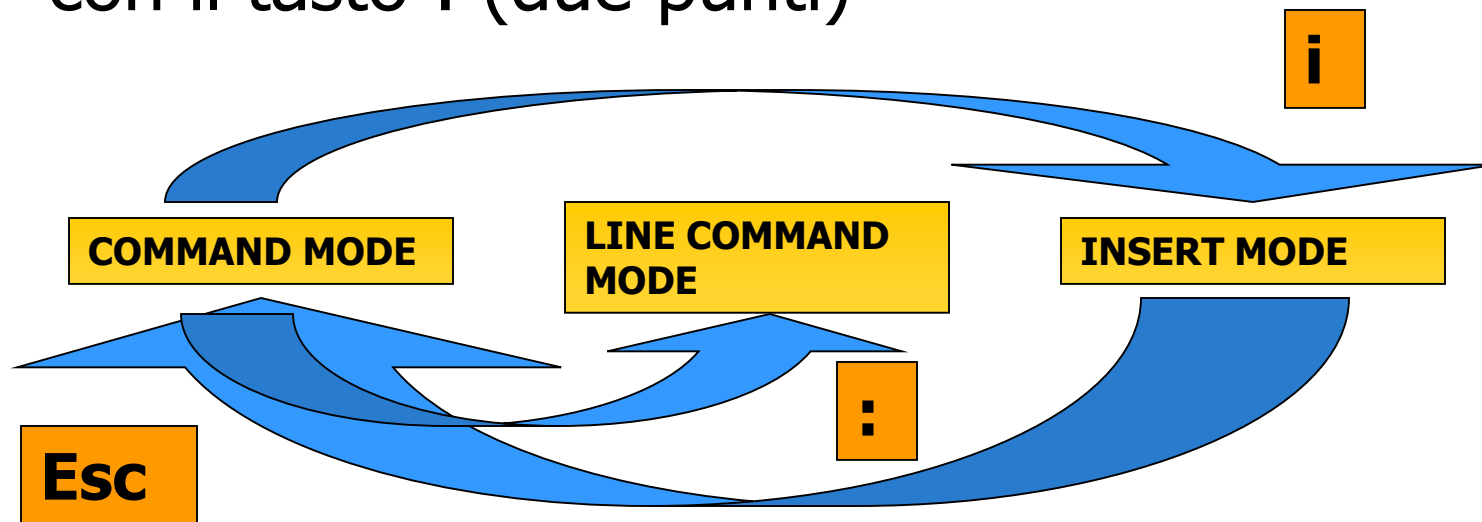
# Modalità di funzionamento

All'inizio il programma si trova in *command mode*

Per passare all'*insert mode* si deve dare il comando *i*

Per tornare al *command mode* si preme il tasto *ESC*

Dal *command mode* si passa al *line command mode* con il tasto *:* (due punti)



# Per uscire da VI

Per uscire da *vi* il comando è `:q` se si è in *command mode*, se il file è stato modificato verrà inviato un messaggio di segnalazione.

Se si vuole uscire senza modificare il file il comando è `:q!`

Per salvare le modifiche il comando è `:w`

Per salvare le modifiche ed uscire il comando è `:wq`

# Command Mode : movimento del cursore

Testo

<b>1G</b>	Inizio testo
<b>G</b>	Fine testo

Pagina

<b>H</b>	Inizio pagina
<b>L</b>	Fine pagina
<b>Ctrl B</b>	Pagina su
<b>Ctrl F</b>	Pagina giù

Riga

<b>k</b>	Riga su
<b>j</b>	Riga giù
<b>0</b>	Inizio riga
<b>\$</b>	Fine riga

Parola

<b>l</b>	A destra di un carattere
<b>h</b>	A sinistra di un carattere
<b>b</b>	Parola precedente
<b>w</b>	Parola successiva

# Command Mode : modifica

## Inserimento

<b>i</b>	Carattere a sinistra
<b>a</b>	Carattere a destra
<b>A</b>	Carattere a fine riga
<b>o</b>	Riga sotto la linea corrente
<b>O</b>	Riga sopra la linea corrente

## Cancellazione

<b>x</b>	Carattere sotto il cursore
<b>dw</b>	Parola a destra
<b>db</b>	Parola a sinistra
<b>dd</b>	Riga corrente
<b>D</b>	Dal carattere a fine riga
<b>dL</b>	Dal carattere a fine schermo
<b>dG</b>	Dal carattere a fine testo



# Command Mode : modifica

## Correzioni

<b>r</b>	Carattere sotto all cursore
<b>R</b>	Dal carattere in poi fino alla pressione del tasto ESC
<b>cw</b>	Parola
<b>cc</b>	Riga

Premendo il carattere *u* (undo) si elimina l'effetto dell'ultimo comando.

Tutti i comandi possono essere preceduti dalla cardinalità del comando, cioè da un numero che indica quante volte il comando deve essere ripetuto, per esempio:

```
10dd
```

cancella 10 linee dalla riga corrente.

# Command Mode: copia e incolla

## Per copiare una riga:

- premere **Y** sulla riga corrente
- muovere il cursore nel punto dove copiare al riga (senza premere ESC)
- premere **p**

## Per copiare un blocco di righe:

- premere **nY** (dove n indica il numero di righe da copiare a partire dalla riga corrente)
- muovere il cursore nel punto dove copiare al riga (senza premere ESC)
- premere p

# Command Mode: copia e incolla

Per spostare una riga:

- premere ***dd*** sulla riga corrente
- muovere il cursore nel punto dove copiare al riga (senza premere ESC)
- premere ***p***

Per spostare un blocco di righe:

- premere ***ndd*** (dove n indica il numero di righe da copiare a partire dalla riga corrente)
- muovere il cursore nel punto dove copiare al riga (senza premere ESC)
- premere ***p***

Per salvare su disco un blocco di righe:

```
:x,y w filename
```

- ***x*** e ***y*** indicano i numeri della prima e dell'ultima riga del blocco da scrivere
- ***w*** indica l'operazione di scrittura
- ***filename*** indica il nome del file

Per inserire un blocco di righe da file su disco:

- muovere il cursore nel punto del testo dove si vuole inserire il blocco di righe
- in line command mode scrivere

```
:r filename
```

# Line Command Mode: ricerca stringhe

Per ricerca una stringa in avanti:

`:/stringa`

Per ricerca una stringa all'indietro:

`:?stringa`

Per ripetere la ricerca nella stessa direzione :

`n` (in command mode)

Per ripetere la ricerca nella direzione opposta:

`N` (in command mode)

# Line Command Mode: ricerca stringhe

## Caratteri speciali:

<code>^</code>	Il pattern di ricerca deve trovarsi ad inizio riga
<code>.</code>	Il pattern deve contenere un'occorrenza del carattere
<code>*</code>	Il pattern deve contenere zero o più occorrenze del carattere precedente
<code>\$</code>	Il pattern di ricerca deve trovarsi a fine riga
<code>[]</code>	Racchiudono un set di caratteri che il pattern deve contenere o non contenere. Per es. <code>/f[iae]t</code> ricerca pattern tipo fit, fat o fet <code>/a[^bcd]</code> ricerca pattern che iniziano per a e come secondo carattere non contengono b,c e d
<code>&lt;</code>	Preceduto dal carattere di escape <code>\</code> indica che il pattern deve trovarsi all'inizio di una parola. Per. Es. <code>/\&lt;the</code> ricerca tutte le parole che iniziano con the
<code>&gt;</code>	Preceduto dal carattere di escape <code>\</code> indica che il pattern deve trovarsi alla fine di una parola. Per. Es. <code>/the\&gt;</code> ricerca tutte le parole che finiscono con the

Per sostituire una stringa con un'altra:

```
:n1,n2 s/vecchia/nuova
```

dove *n1,n2* indicano il numero della prima riga e il numero dell'ultima riga per la sostituzione su una parte di testo

Per la sostituzione di tutto il testo occorre specificare *1,\$*

*vecchia* indica la parola o parte di testo da sostituire con *nuova*.