

Shell Linux-Giorno 1

ESERCIZIO:

ESERCIZI SHELL

cd	Cambia Directory
Mkdir	Make Directory (fa una directory)
rmdir	rimuove una directory (se vuota)
mv	sposta un file - directory
cp	Copia un file (directory)
rm	Cancella un file
ls	visualizza il contenuto della cartella
pwd	print working directory (stampa il percorso assoluto dove mi trovo)
man argomento	Visualizza il manuale di un comando

Collegarsi al sistema con utenza e password

Esercizio 1

Come prima cosa creare le seguenti cartelle e sottocartelle (usando i comandi "terminale" mkdir cd rmdir ... a partire dalla propria HOME e visualizzarle a video:
(Per "Propria home" si intende il posto dove vi posiziona quando aprite il terminale!)



Ti trovi nella directory **lavoro** (sotto nicola), scrivere il comando per passare alla directory **casa** (sotto anna) con percorso relativo e percorso assoluto.

- Copia il file compito.doc (dalla directory scuola) nella directory corrente (casa).
- Sposta il file relazione.doc nella directory corrente (casa).
- Cancella la cartella 'tmp'
- Creare il file pippo.txt nella cartella lavoro
- Cambiare gli attributi del file pippo.txt e renderlo scrivibile e leggibile solo per il proprietario, mentre per tutti gli altri solo leggibile...
- Nascondere il contenuto della cartella anna
- Spostarsi nella cartella lavoro e visualizzare il contenuto del file pippo.txt
- Rimuovere la cartella amici
- Rimuovere tutte le cartelle precedentemente create

Esercizi - processi:

1. Aprire un terminale
2. leggere il manuale del comando job, ps e kill
3. lanciare il comando vi pippo
4. aprire un nuovo terminale e visualizzare tutti i propri processi...
5. cercare di terminare (killare) il processo vi per sbloccare il terminale precedente
6. lanciare il comando firefox in background
7. portarlo in background
8. cercare di terminare il processo firefox
9. verificare quanto spazio si sta occupando su disco

Creo nuove cartelle e sottocartelle con il comando **mkdir**

```
kali@kali: ~  
File Actions Edit View Help  
(kali@kali)-[~]  
$ id  
uid=1000(kali) gid=1000(kali) groups=1000(kali),4(adm),20(dialout),24(cdrom),  
25(floppy),27(sudo),29(audio),30(dip),44(video),46(plugdev),100(users),106(ne  
tdev),111(bluetooth),117(scanner),140(wireshark),142(kaboxer),143(vboxsf)  
(kali@kali)-[~]  
$ mkdir dos  
(kali@kali)-[~]  
$ mkdir studenti  
(kali@kali)-[~]  
$ mkdir windows  
(kali@kali)-[~]  
$ mkdir tmp
```

```
kali@kali: ~  
File Actions Edit View Help  
(kali@kali)-[~]  
$ mkdir tmp  
(kali@kali)-[~]  
$ mkdir studenti/nicola  
(kali@kali)-[~]  
$ mkdir studenti/nicola/scuola  
(kali@kali)-[~]  
$ mkdir studenti/nicola/lavoro  
(kali@kali)-[~]  
$ mkdir studenti/anna  
(kali@kali)-[~]  
$ mkdir studenti/anna/casa  
(kali@kali)-[~]  
$ mkdir studenti/matteo  
(kali@kali)-[~]  
$ mkdir studenti/matteo/amici  
(kali@kali)-[~]  
$
```

Con il comando **touch** aggiungo i contenuti compito.doc, relazione.doc e gli altri e con **ls** verifico che siano stati creati (verifico contenuto della cartella)

Copio compito.doc nella sottocartella anna con il comando **cp**

```
(kali㉿kali)-[~/studenti/nicola/scuola]
$ cp compito.doc /home/kali/studenti/anna/casa

(kali㉿kali)-[~/studenti/nicola/scuola]
$ cd /home/kali/studenti/anna/casa

(kali㉿kali)-[~/studenti/anna/casa]
$ ls
compito.doc

(kali㉿kali)-[~/studenti/anna/casa]
$
```

Sposto con il comando **mv**, relazione.doc nella cartella corrente (anna)

```
(kali㉿kali)-[~/studenti/anna/casa]
$ mv /home/kali/studenti/nicola/scuola/relazione.doc ./

(kali㉿kali)-[~/studenti/anna/casa]
$ ls
compito.doc  relazione.doc

(kali㉿kali)-[~/studenti/anna/casa]
$
```

Elimino file tmp con il comando **rm -r**

```
(kali㉿kali)-[~]  
$ rm -r tmp  
  
(kali㉿kali)-[~]  
$ ls -l  
anna  
Desktop  
Documents  
dos  
Downloads  
Music  
Pictures  
Public  
studenti  
Templates  
Videos  
windows
```

creo pippo.txt nella directory lavoro con **touch**

```
kali@kali: ~/studenti/nicola/lavoro  
File Actions Edit View Help  
  
(kali㉿kali)-[~]  
$ cd studenti/nicola/lavoro  
  
(kali㉿kali)-[~/studenti/nicola/lavoro]  
$ touch pippo.txt  
  
(kali㉿kali)-[~/studenti/nicola/lavoro]  
$ ls  
pippo.txt  
  
(kali㉿kali)-[~/studenti/nicola/lavoro]  
$
```

Cambio gli attributi del file pippo.txt e lo rendo scrivibile e leggibile al proprietario e leggibile agli altri con

```
(kali㉿kali)-[~/studenti/nicola/lavoro]
$ chmod u+x pippo.txt

(kali㉿kali)-[~/studenti/nicola/lavoro]
$ ls -l
total 0
-rwxr--r-- 1 kali kali 0 Nov 21 14:59 pippo.txt

(kali㉿kali)-[~/studenti/nicola/lavoro]
$
```

Nascondo la cartella casa

```
(kali㉿kali)-[~]
$ cd studenti/anna

(kali㉿kali)-[~/studenti/anna]
$ ls
casa

(kali㉿kali)-[~/studenti/anna]
$ .casa
.casa: command not found

(kali㉿kali)-[~/studenti/anna]
$ mv casa .casa

(kali㉿kali)-[~/studenti/anna]
$
```

Cancello la cartella amici

```
(kali㉿kali)-[~/studenti/matteo]
$ rm -r amici

(kali㉿kali)-[~/studenti/matteo]
$ ls

(kali㉿kali)-[~/studenti/matteo]
$ cd ../

(kali㉿kali)-[~/studenti]
$ ls
anna  matteo  nicola

(kali㉿kali)-[~/studenti]
$
```

Infine, cancello tutte le cartelle create

```
(kali㉿kali)-[~]
$ rm -r studenti dos windows
```

Apri i manuali per kill, PS e Jobs (che si trova all'interno di builtins) con il comando **man**

```
kali@kali: ~  
File Actions Edit View Help  
KILL(1) User Commands KILL(1)  
  
NAME  
kill - send a signal to a process  
  
SYNOPSIS  
kill [options] <pid> [...]  
  
DESCRIPTION  
The default signal for kill is TERM. Use -l or -L to list available  
signals. Particularly useful signals include HUP, INT, KILL, STOP,  
CONT, and 0. Alternate signals may be specified in three ways: -9,  
-SIGKILL or -KILL. Negative PID values may be used to choose whole  
process groups; see the PGID column in ps command output. A PID of  
-1 is special; it indicates all processes except the kill process  
itself and init.  
  
OPTIONS  
<pid> [...]  
Send signal to every <pid> listed.  
  
-<signal>  
-s <signal>  
--signal <signal>  
Specify the signal to be sent. The signal can be specified  
by using name or number. The behavior of signals is ex-  
Manual page kill(1) line 1 (press h for help or q to quit)
```

```
kali@kali: ~  
File Actions Edit View Help  
BASH-BUILTINS(7) Miscellaneous Information Manual BASH-BUILTINS(7)  
  
NAME  
bash-builtins - bash built-in commands, see bash(1)  
  
SYNOPSIS  
bash defines the following built-in commands: :, ., [, alias, bg,  
bind, break, builtin, case, cd, command, compgen, complete, con-  
tinue, declare, dirs, disown, echo, enable, eval, exec, exit, ex-  
port, fc, fg, getopts, hash, help, history, if, jobs, kill, let, lo-  
cal, logout, popd, printf, pushd, pwd, read, readonly, return, set,  
shift, shopt, source, suspend, test, times, trap, type, typeset,  
ulimit, umask, unalias, unset, until, wait, while.  
  
BASH BUILTIN COMMANDS  
Unless otherwise noted, each builtin command documented in this sec-  
tion as accepting options preceded by - accepts -- to signify the  
end of the options. The :, true, false, and test/[ builtins do not  
accept options and do not treat -- specially. The exit, logout, re-  
turn, break, continue, let, and shift builtins accept and process  
arguments beginning with - without requiring --. Other builtins  
that accept arguments but are not specified as accepting options in-  
terpret arguments beginning with - as invalid options and require --  
to prevent this interpretation.  
: [arguments]  
No effect; the command does nothing beyond expanding argu-  
Manual page builtins(7) line 1 (press h for help or q to quit)
```

```
kali@kali: ~  
File Actions Edit View Help  
PS(1) User Commands PS(1)  
  
NAME  
ps - report a snapshot of the current processes.  
  
SYNOPSIS  
ps [options]  
  
DESCRIPTION  
ps displays information about a selection of the active processes.  
If you want a repetitive update of the selection and the displayed  
information, use top instead.  
  
This version of ps accepts several kinds of options:  
  
1 UNIX options, which may be grouped and must be preceded by a  
dash.  
2 BSD options, which may be grouped and must not be used with a  
dash.  
3 GNU long options, which are preceded by two dashes.  
  
Options of different types may be freely mixed, but conflicts can  
appear. There are some synonymous options, which are functionally  
identical, due to the many standards and ps implementations that  
this ps is compatible with.  
  
Manual page ps(1) line 1 (press h for help or q to quit)
```

lancio **vi pippo** e apro altro tab e lancio comando **ps aux**. Successivamente chiudo **vi pippo** con il comando **Kill** e il numero PID.

```
kali@kali: ~  
File Actions Edit View Help  
kali@kali: ~ x kali@kali: ~ x  
kali      1294  0.0  0.5 307296 10340 ?        Ssl  13:49   0:00 /usr/libex  
kali      1295  0.0  2.5 448324 52152 ?        Sl   13:49   0:00 /usr/bin/p  
kali      1314  0.0  0.4 386272  9920 ?        Ssl  13:49   0:00 /usr/libex  
kali      1344  0.0  0.5 308252 11020 ?        Ssl  13:49   0:00 /usr/libex  
kali      1388  0.0  0.5 385768 10724 ?        Sl   13:49   0:00 /usr/libex  
kali      1396  0.0  0.4 233768  8340 ?        Ssl  13:49   0:00 /usr/libex  
kali      1432  0.0  0.4  48888  8064 ?        Ss   13:49   0:00 /usr/libex  
root      6184  0.0  0.0      0      0 ?        I    13:58   0:01 [kworker/1  
root      9624  0.0  0.0      0      0 ?        I    14:05   0:00 [kworker/u  
root      9625  0.0  0.0      0      0 ?        I    14:05   0:00 [kworker/u  
root     56498 0.0  0.0      0      0 ?        I    15:39   0:00 [kworker/1  
root     69007 0.0  0.0      0      0 ?        I    16:04   0:00 [kworker/0  
root     71602 0.0  0.0      0      0 ?        I    16:09   0:00 [kworker/0  
kali     72112 0.3  5.1 447416 104712 ?        Sl   16:10   0:01 /usr/bin/q  
kali     72120 0.0  0.3  10180  6364 pts/0    Ss   16:10   0:00 /usr/bin/z  
kali     72163 0.0  0.4  14876  9600 pts/0    Sl+  16:10   0:00 vi pippo  
kali     72501 0.0  0.3  10180  6248 pts/1    Ss   16:11   0:00 /usr/bin/z  
root     74139 0.0  0.0      0      0 ?        I    16:14   0:00 [kworker/0  
kali     74749 100  0.2  11296  4736 pts/1    R+   16:15   0:00 ps aux  
  
(kali@kali)-[~]  
$ kill 72163  
  
(kali@kali)-[~]  
$
```


Infine lancio Firefox in background con il comando **firefox &**
e lo chiudo con il comando **pkill firefox**

```
kali@kali: ~  
File Actions Edit View Help  
(kali@kali)-[~]  
$ firefox &  
[1] 85507  
  
(kali@kali)-[~]  
$ pkill firefox  
  
Exiting due to channel error.  
(kali@kali)-[~]  
$  
Exiting due to channel error.  
Exiting due to channel error.  
Exiting due to channel error.  
Exiting due to channel error.  
  
[1] + terminated firefox  
(kali@kali)-[~]  
$
```

controllo spazio che sto occupando su disco con comando **df**

```
(kali@kali)-[~]  
$ df  
Filesystem      1K-blocks    Used Available Use% Mounted on  
udev             966784         0    966784   0% /dev  
tmpfs            201428         988    200440   1% /run  
/dev/sda1       82083148 14982840  62884760  20% /  
tmpfs            1007136         0    1007136   0% /dev/shm  
tmpfs             5120          0         5120   0% /run/lock  
tmpfs            201424        112    201312   1% /run/user/1000  
  
(kali@kali)-[~]  
$
```