

Consegna:

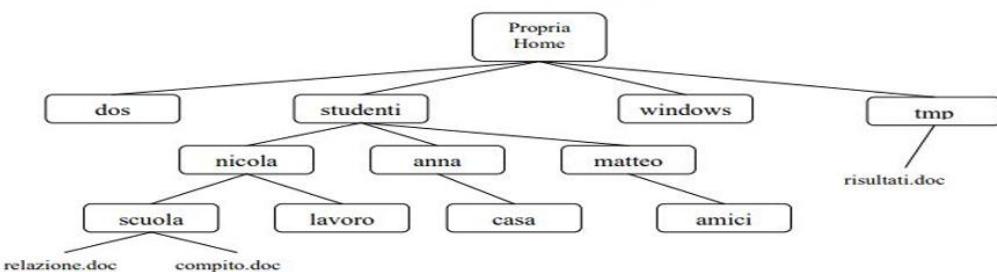
ESERCIZI SHELL

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 quanto 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

Soluzione:

Esercizio:

Il primo esercizio richiede la creazione di una serie di cartelle e sottocartelle, nel fare ciò abbiamo utilizzato i comandi ‘mkdir’ per la creazione delle singole cartelle ed il comando ‘touch’ per la creazione dei file all’interno delle varie cartelle (come da consegna).

In allegato screenshot della creazione delle singole cartelle.

```
(kali㉿kali)-[~]
$ cd home

(kali㉿kali)-[~/home]
$ mkdir dos

(kali㉿kali)-[~/home]
$ mkdir studenti

(kali㉿kali)-[~/home]
$ mkdir windows

(kali㉿kali)-[~/home]
$ mkdir tmp
      Home
(kali㉿kali)-[~/home]
$ ls
dos  studenti  tmp  windows

(kali㉿kali)-[~/home]
$ cd studenti
eth0.pcapng
(kali㉿kali)-[~/home/studenti]
$ mkdir anna

(kali㉿kali)-[~/home/studenti]
$ mkdir matteo

(kali㉿kali)-[~/home/studenti]
$ mkdir nicola

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

(kali㉿kali)-[~/home/studenti/nicola]
$ mkdir scuola

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

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

(kali㉿kali)-[~/home/studenti/nicola/scuola]
$ touch relazione.doc

(kali㉿kali)-[~/home/studenti/nicola/scuola]
$ touch compito.doc

(kali㉿kali)-[~/home/studenti/nicola/scuola]
$ █
```

Per spostarsi da una directory all'altra tramite percorso relativo utilizzeremo il comando ‘cd ..’, invece per spostarci tramite percorso assoluto dalla directory lavoro alla directory casa utilizzeremo il comando ‘cd /percorso/della/directory’

- a) Nel punto a viene richiesto di copiare il file ‘compito.doc’ dalla cartella scuola alla cartella casa, nel fare ciò abbiamo utilizzato il comando ‘cp /percorso della cartella di origine del file/nome del file /percorso della directory di destinazione’.
- b) Nel punto b viene richiesto di spostare il file ‘relazione.doc’ nella cartella corrente cioè casa a partire dalla directory nicola (scuola), nel fare ciò abbiamo utilizzato il comando ‘ mv /percorso della cartella di origine del file/nome del file/ /percorso della cartella di destinazione/’

```

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

└─(kali㉿kali)-[~/home/studenti/nicola/scuola]
$ ls
compito.doc  relazione.doc

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

└─(kali㉿kali)-[~/home/studenti/anna]
$ cd casa

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

└─(kali㉿kali)-[~/home/studenti/anna/casa]
$ █

```

```

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

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

└─(kali㉿kali)-[~/home/studenti/anna/casa]
$ █

```

- c) Nel punto c viene richiesta la cancellazione della directory (piena) tmp, nel fare ciò abbiamo utilizzato il comando ‘rm –r nome della directory’ mettendo ‘il comando –r’ abbiamo il privilegio amministratori per l’eliminazione definita della directory contenente il file risultati.doc.
- d) Al punto d viene richiesta la creazione del file ‘pippo.txt’ dentro la directory lavoro, nel fare ciò abbiamo utilizzato il comando ‘touch nome del file’.
- e) Nel punto e viene richiesto di cambiare gli attributi del file ‘pippo.txt’ in modo tale da renderlo leggibile, scrivibile ed eseguibile solo da noi e solo leggibile dagli altri utenti, nel fare ciò abbiamo utilizzato il comando ‘chmod 744 pippo.txt’ poiché seguendo i permessi numerici il nr. 7 si configura come il permesso per tutte e 3 le operazioni, mentre il nr 4 si configura come il permesso di sola lettura.

```

└─(kali㉿kali)-[~/home/studenti/nicola/lavoro]
$ chmod 744 pippo.txt

└─(kali㉿kali)-[~/home/studenti/nicola/lavoro]
$ ls -l pippo.txt
-rwxr--r-- 1 kali kali 0 Nov 24 12:21 pippo.txt

└─(kali㉿kali)-[~/home/studenti/nicola/lavoro]
$ █

```

- f) Questo punto richiede di nascondere il contenuto all’interno della cartella anna, nel fare ciò abbiamo utilizzato il comando ‘mv /percorso della directory/ /percorso della/.directory’ ; il punto messo davanti il nome della directory la rende invisibile.

```
(kali㉿kali)-[~/home/studenti/anna]
$ cd ..
(kali㉿kali)-[~/home/studenti]
$ mv /home/kali/home/studenti/anna /home/kali/home/studenti/.anna
(kali㉿kali)-[~/home/studenti]
$ ls System
matteo nicola
(kali㉿kali)-[~/home/studenti]
$
```

- g) Al punto g veniva richiesta l'apertura del file pippo.txt, nel farlo abbiamo utilizzato il comando 'cat nome del file'
- h) Nel punto h viene richiesta l'eliminazione della directory amici presente all'interno della directory matteo, per effettuare questa operazione utilizziamo il comando 'rmdir nome della directory' come allegato.

```
Home
(kali㉿kali)-[~/home/studenti]
$ cd matteo
(kali㉿kali)-[~/home/studenti/matteo]
$ ls
amici
eth0.pcapng
(kali㉿kali)-[~/home/studenti/matteo]
$ rmdir amici
(kali㉿kali)-[~/home/studenti/matteo]
$ ls
(kali㉿kali)-[~/home/studenti/matteo]
$
```

- i) All'ultimo punto viene richiesta l'eliminazione di tutti i file creati fino ad ora, effettueremo l'operazione richiesta con il comando 'rm -r nome della directory principale', in questo modo abbiamo provveduto alla cancellazione di tutte le cartelle e sottocartelle precedentemente create.

Seconda consegna:

who	lista utenti collegati
who am i	chi sono io
jobs	elenco lavori sul terminale
&	apre processo in background
fg	metti in foreground
bg	metti in background
ps	elenco processi
kill	termina processo

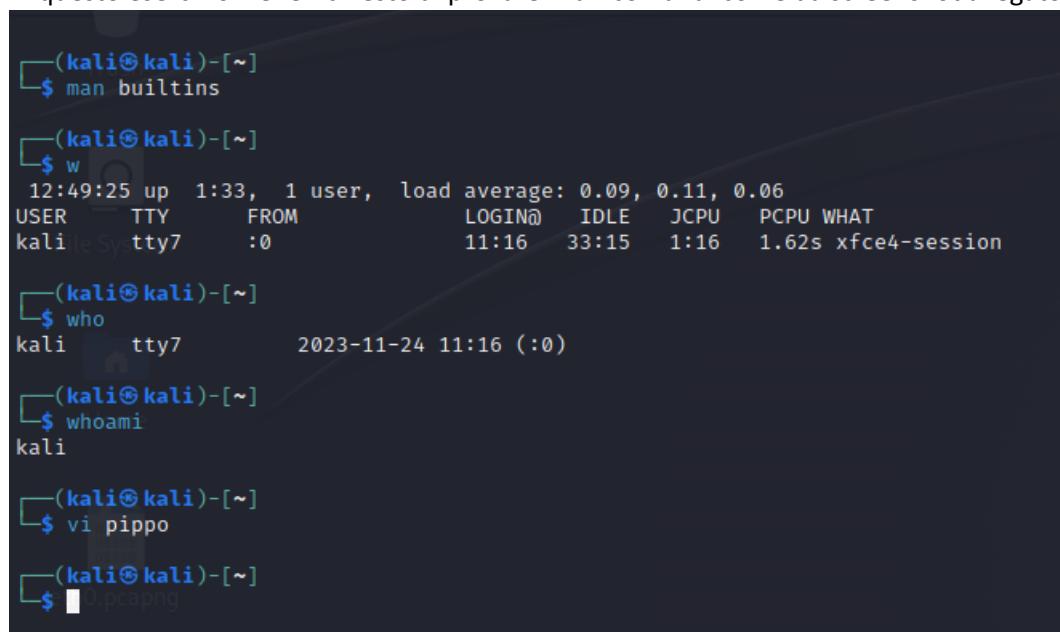
Provare i comandi:

```
w
who
who am i
```

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

In questo esercizio viene richiesto di provare i vari comandi come da screenshot allegato:



```
(kali㉿kali)-[~]
$ man builtins

(kali㉿kali)-[~]
$ w
12:49:25 up 1:33, 1 user,  load average: 0.09, 0.11, 0.06
USER   TTY      FROM          LOGIN@    IDLE      JCPU     PCPU WHAT
kali   tty7     :0          11:16   33:15   1:16   1.62s xfce4-session

(kali㉿kali)-[~]
$ who
kali    tty7        2023-11-24 11:16 (:0)

(kali㉿kali)-[~]
$ whoami
kali

(kali㉿kali)-[~]
$ vi pippo

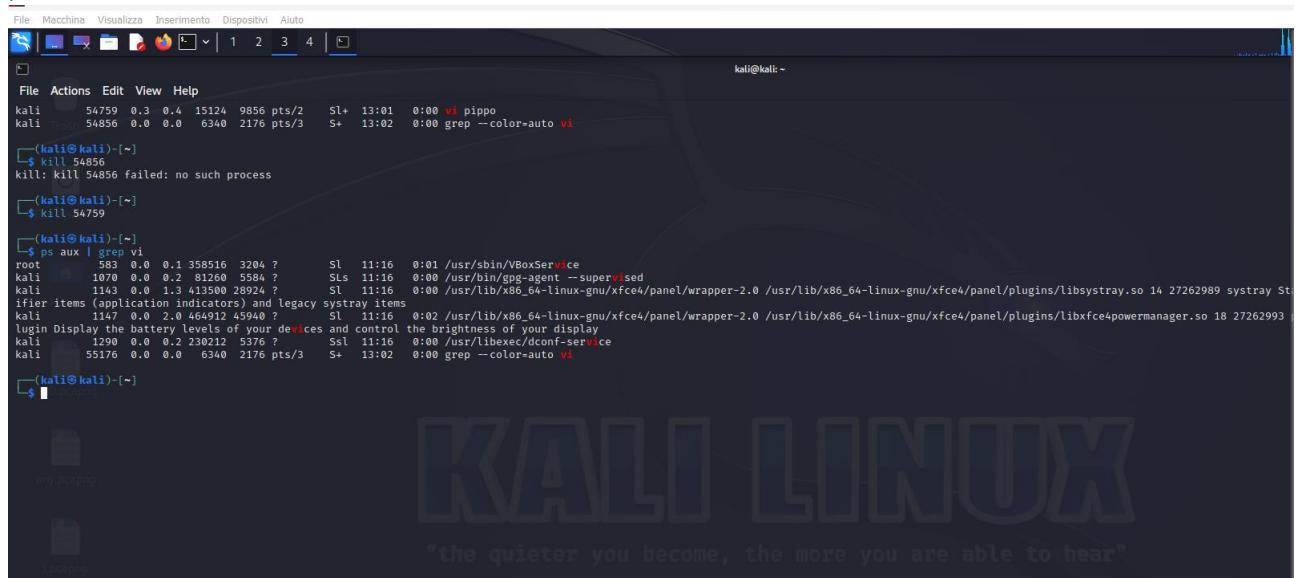
(kali㉿kali)-[~]
$ █ 0.pcapng
```

2) Viene richiesta la lettura del manuale dei comandi ‘jobs, ps, kill’ nel fare ciò abbiamo utilizzato il comando ‘ man builtins’ e tramite questo abbiamo avuto modo di accedere al manuale di Shell e leggere i suddetti comandi.

```
jobs [-l[pids] | -j[obspec ... ]  
jobs -x [command [ args ... ]]  
The first form lists the active jobs. The options have the following meanings:  
-l List process IDs in addition to the normal information.  
-n List display information only about jobs that have changed status since the user was last notified of their status.  
-p List only the process ID of the job's process group leader.  
-r Display only running jobs.  
-s Display only stopped jobs.  
  
If the -x option is supplied, jobs replaces any jobspec found in command or args with the corresponding process group ID, and executes command passing it args, returning its exit status.  
  
kill [-s sigspec | -n signum | --sigspec] [pid | jobspec] ...  
kill -l[-l [sigspec | exit_status]  
Send the signal named by sigspec or signum to the processes named by pid or jobspec. sigspec is either a case-insensitive signal name such as SIGKILL (with or without the SIG prefix) or a signal number; signum is a signal number. If sigspec or signum is not present, then SIGTERM is assumed. An argument of -l lists the signal names. If any arguments are supplied when -l is given, the names of the signals corresponding to the arguments are listed, and the return status is 0. The exit_status argument to -l is a number specifying either a signal number or the exit status of a process terminated by a signal. The -L option is equivalent to -l. kill returns true if at least one signal was successfully sent, or false if an error occurs or an invalid option is encountered.
```

3) Qui viene chiesto di lanciare il comando ‘vi pippo’, nel fare ciò si apre il processo ‘pippo’.

4) Viene richiesto di aprire il processo ‘vi pippo’ ed aprire un nuovo terminale per visualizzare i processi in corso, nel fare ciò abbiamo lanciato il comando ‘ ps aux |grep vi pippo’ per vedere i processi in corso.



```
File Macchina Visualizza Inserimento Dispositivi Aiuto  
kali@kali:~$ ps aux | grep vi  
root      583  0.0  0.1 358516 3206 ?        Sl   11:16  0:01 /usr/sbin/VBoxService  
kali     1070  0.0  0.2  81260 5584 ?       Sls  11:16  0:00 /usr/bin/png-agent --supervised  
kali    1143  0.0  1.3 413500 28924 ?      Sl   11:16  0:00 /usr/lib/x86_64-linux-gnu/xfce4/panel/wrapper-2.0 /usr/lib/x86_64-linux-gnu/xfce4/panel/plugins/libsystray.so 14 27262989 systray St  
ifier items (application indicators) and legacy systray items  
kali    1147  0.0  2.0 464912 45940 ?      Sl   11:16  0:02 /usr/lib/x86_64-linux-gnu/xfce4/panel/wrapper-2.0 /usr/lib/x86_64-linux-gnu/xfce4/panel/plugins/libxfce4powermanager.so 18 27262993  
login Display the battery levels of your devices and control the brightness of your display  
kali   1290  0.0  0.2 230212 5376 ?      Ssl  11:16  0:00 /usr/libexec/dconf-service  
kali    55176 0.0  0.0  6340 2176 pts/3      S+  13:02  0:00 grep --color=auto vi  
kali@kali:~$
```

5) Qui chiede di cercare di terminare il processo ‘vi pippo’ nel fare ciò abbiamo lanciato il comando ‘kill codice processo’ (vedi foto sopra)

6) Poi viene richiesto di lanciare firefox in background, nel fare ciò abbiamo lanciato il comando ‘firefox &’ e abbiamo visto i processi attivi tramite il comando ‘ps aux | grep firefox’

```

└─(kali㉿kali)-[~]
└─$ ps aux | grep firefox
kali      58610 60.3 15.1 11240432 333020 pts/3 SNl 13:08 0:08 firefox-esr
kali      58684 1.0 1.6 217472 37040 pts/3 SNl 13:08 0:00 /usr/lib/firefox-esr/firefox-esr -cont
b9ab-c3c6a12786a7} 58610 true socket
kali      58716 5.3 4.3 2417124 96592 pts/3 SNl 13:08 0:00 /usr/lib/firefox-esr/firefox-esr -cont
ib/firefox-esr/browser {b0b8c08a-d06d-4602-a8c2-4e59d985f62d} 58610 true tab
kali      58757 11.5 5.3 10833548 118760 pts/3 SNl 13:08 0:01 /usr/lib/firefox-esr/firefox-esr -cont
ib/firefox-esr/browser {e0c14156-05c0-4f72-9328-649feac1179b} 58610 true tab
kali      58800 14.8 4.7 2433508 105376 pts/3 SNl 13:08 0:01 /usr/lib/firefox-esr/firefox-esr -cont
ib/firefox-esr/browser {5969c97c-48f8-4ddd-9aa8-a7dbb2395d83} 58610 true tab
kali      58862 4.5 3.2 2401768 71360 pts/3 SNl 13:08 0:00 /usr/lib/firefox-esr/firefox-esr -cont
ib/firefox-esr/browser {ae6b2217-68eb-4775-9adb-d1b83f5a8601} 58610 true tab
kali      58864 4.6 3.2 2401764 71552 pts/3 SNl 13:08 0:00 /usr/lib/firefox-esr/firefox-esr -cont
ib/firefox-esr/browser {4d3a9ca1-3703-4e5d-8b3f-cbf1e0950ef6} 58610 true tab
kali      58907 5.0 3.2 2401768 70976 pts/3 SNl 13:08 0:00 /usr/lib/firefox-esr/firefox-esr -cont
ib/firefox-esr/browser {986241c4-b8c6-44f5-b879-861d45bfc68d} 58610 true tab
kali      58978 0.0 0.0 6340 2176 pts/3 S+ 13:08 0:00 grep --color=auto firefox

└─(kali㉿kali)-[~]
└─$ kill 58978
kill: kill 58978 failed: no such process

└─(kali㉿kali)-[~]
└─$ kill 58610

└─(kali㉿kali)-[~]
└─$ Exiting due to channel error.

[1] + terminated firefox
└─(kali㉿kali)-[~]
└─$ █

```

8) Nel punto 8 viene richiesto di terminare firefox e nel fare ciò abbiamo inserito il comando ‘kill 58610’

9) Qui viene richiesto quanto spazio stiamo occupando su disco, nel fare ciò abbiamo utilizzato il comando ‘df –h’

```

└─(kali㉿kali)-[~]
└─$ df -h
Filesystem      Size  Used Avail Use% Mounted on
udev            1.1G    0  1.1G   0% /dev
tmpfs           216M  1004K 215M   1% /run
/dev/sda1        79G   15G   60G  20% /
tmpfs           1.1G    0  1.1G   0% /dev/shm
tmpfs            5.0M    0  5.0M   0% /run/lock
tmpfs           216M   112K 216M   1% /run/user/1000
tmpfs           216M   108K 216M   1% /run/user/110

└─(kali㉿kali)-[~]
└─$ █

```