

Shell di linux

ESERCIZI SHELL

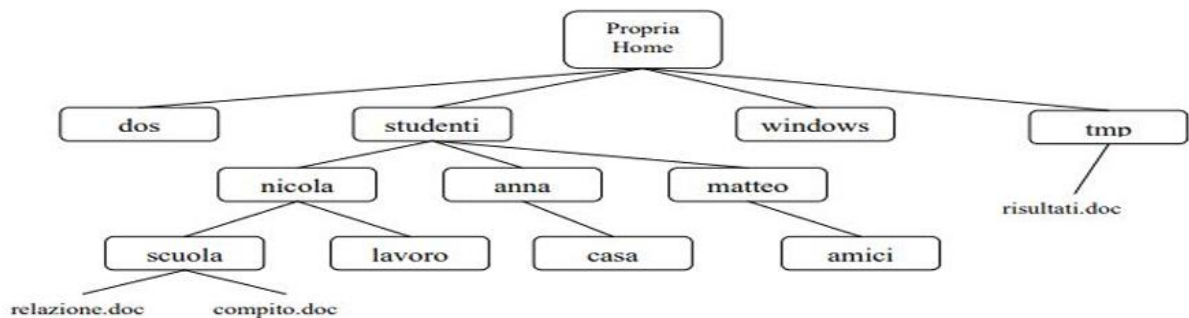
cd	Cambia Directory
Mkdir	Make Directory (fa una directory)
rmdir	rimuove una directory (se vuota)
mv	sposta un file - directory
cp	Copia un file (directory)
rm	Cancella un file
ls	visualizza il contenuto della cartella
pwd	print working directory (stampa il percorso assoluto dove mi trovo)
man argomento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

Comandi per l'esercizio

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!)



Costruiamo la nostra rete di **Direcrory**:

```
File Actions Edit View Help
(kali@kali)-[~]
$ mkdir dos
(kali@kali)-[~]
$ mkdir studenti
(kali@kali)-[~]
$ mkdir windows
(kali@kali)-[~]
$ mkdir tmp
(kali@kali)-[~]
$ ls
Desktop Documents dos Downloads Music Pictures Public studenti Templates tmp Videos windows
(kali@kali)-[~]
$
```

torniamo poi tramite il comando **CD** nella nostra directory principale e tramite il comando **ls** controlliamo l'effettiva creazione delle cartelle.

Ora come nella mappa delle directory all'interno della dir. **Scuola** creiamo dei file **nano** denominati *relazione.doc* e *compito.doc* mentre nella dir. **Tmp** creeremo un file **nano** *risultati.doc*

```
(kali@kali)-[~]
$ cd tmp

(kali@kali)-[~/tmp]
$ ls
..  .risultati.doc.swp

(kali@kali)-[~/tmp]
$ ls -al
.  ..  .risultati.doc.swp

(kali@kali)-[~/tmp]
$ sudo rm .risultati.doc.swp
[sudo] password for kali:

(kali@kali)-[~/tmp]
$ sudo nano risultati.doc

(kali@kali)-[~/tmp]
$ sudo nano risultati.doc

(kali@kali)-[~/tmp]
$ ls
risultati.doc

(kali@kali)-[~]
$ cd studenti

(kali@kali)-[~/studenti]
$ mkdir Nicola

(kali@kali)-[~/studenti]
$ cd nicola
cd: no such file or directory: nicola

(kali@kali)-[~/studenti]
$ cd Nicola

(kali@kali)-[~/studenti/Nicola]
$ mkdir scuola

(kali@kali)-[~/studenti/Nicola]
$ cd scuola

(kali@kali)-[~/studenti/Nicola/scuola]
$ sudo nano

(kali@kali)-[~/studenti/Nicola/scuola]
$ sudo nano

(kali@kali)-[~/studenti/Nicola/scuola]
$ ls
compito.doc  relazione.doc

(kali@kali)-[~/studenti/Nicola/scuola]
$
```

Creiamo le dir. Rimaste

```
(kali@kali)-[~]
$ cd studenti

(kali@kali)-[~/studenti]
$ mkdir anna

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

(kali@kali)-[~/studenti/anna]
$ mkdir casa

(kali@kali)-[~/studenti/anna]
$ cd casa

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

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

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

(kali@kali)-[~/studenti/matteo]
$ mkdir amici

(kali@kali)-[~/studenti/matteo]
$ cd amici

(kali@kali)-[~/studenti/matteo/amici]
$
```

Ora eseguiremo le seguenti consegne :

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

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

Percorso relativo :

il percorso relativo è correlato alla directory corrente ed è preceduto da ..

es

```
File Actions Edit View Help
(kali@kali)-[~/studenti/Nicola/lavoro]
$ cd ../ ../anna/casa
(kali@kali)-[~/studenti/anna/casa]
$ relativo
```

Percorso assoluto :

specifica il percorso dalla directory principale e inizia con un carattere di delimitazione come /

es

```
(kali@kali)-[~/studenti/Nicola/lavoro]
$ cd ../ ../anna/casa
(kali@kali)-[~/studenti/anna/casa]
$ cd ../ ../Nicola/lavoro
(kali@kali)-[~/studenti/Nicola/lavoro]
$ cd /home/kali/studenti/anna/casa
(kali@kali)-[~/studenti/anna/casa]
$ assoluto
```

percorso relativo

percorso assoluto

A) Eseguiremo la copia del file utilizzando il comando cp (nome file) (percorso directory di destinazione)

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

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

(kali㉿kali)-[~]
$ studenti

(kali㉿kali)-[~/studenti]
$ anna

(kali㉿kali)-[~/studenti/anna]
$ casa

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

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

B) sposteremo il file desiderato con il comando mv (nome file) (percorso directory di destinazione)

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

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

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

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

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

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

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

C) per cancellare un directory contenente un file (non vuota) andremo ad utilizzare il comando rm -r (nome directory)

```
File Actions Edit View Help

(kali@kali)-[~]
$ rm -r tmp
rm: remove write-protected regular file 'tmp/risultati.doc'? y

(kali@kali)-[~]
$ ls
Desktop Documents dos Downloads Music Pictures Public studenti Templates Videos windows

(kali@kali)-[~]
$
```

D)creeremo un file (pippo.txt) nella directory lavoro tramite nano , il comando sara: sudo nano pippo.txt

```
File Actions Edit View Help

(kali@kali)-[~]
$ cd studenti/Nicola/lavoro

(kali@kali)-[~/studenti/Nicola/lavoro]
$ sudo nano pippo.txt

(kali@kali)-[~/studenti/Nicola/lavoro]
$ ls
pippo.txt

(kali@kali)-[~/studenti/Nicola/lavoro]
$
```

E) osserviamo come pippo abbia gia i permessi preimpostati secondo la comanda Dove U (legge scrive) G (legge) O (legge)

```
(kali@kali)-[~/studenti/Nicola/lavoro]
$ ls -al
total 8
drwxr-xr-x 2 kali kali 4096 Jan  9 12:27 .
drwxr-xr-x 4 kali kali 4096 Jan  9 11:36 ..
-rw-r--r-- 1 root root    0 Jan  9 12:27 pippo.txt

(kali@kali)-[~/studenti/Nicola/lavoro]
$
```

F)per nascondere il contenuto di una directory bisogna trovarsi all interno di essa e cambiare il nome al file che si vuole nascondere in questo modo (nome file) dovra essere cambiato in (.nome file) .

Il comando quindi sara: sudo mv (nome file) (.nome file)

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

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

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

(kali㉿kali)-[~/studenti/anna]
$ ls

(kali㉿kali)-[~/studenti/anna]
$ ls -al
total 12
drwxr-xr-x 3 kali kali 4096 Jan  9 12:43 .
drwxr-xr-x 5 kali kali 4096 Jan  9 10:48 ..
drwxr-xr-x 2 kali kali 4096 Jan  9 12:15 .casa

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

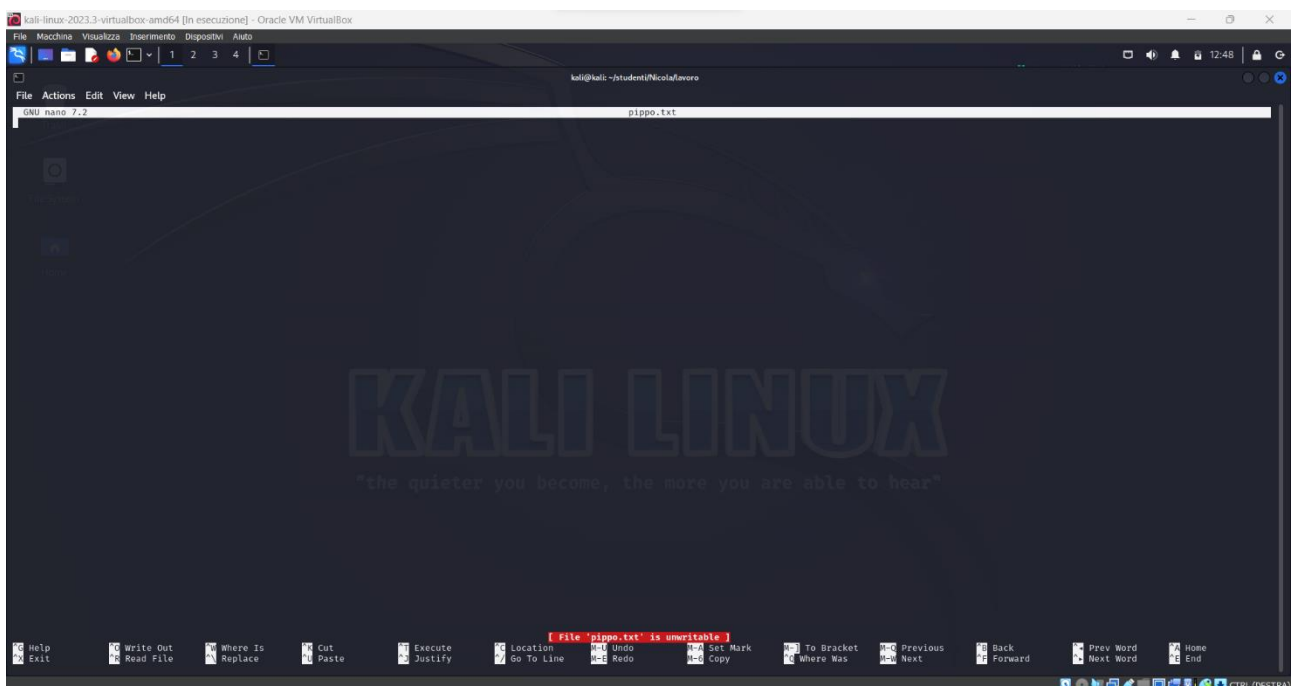
(kali㉿kali)-[~/studenti/anna]
$ ls
.  ..  .casa

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

(kali㉿kali)-[~/studenti/anna]
$
```

vediamo infatti come la cartella **casa** dopo essere diventata **.casa** non e piu visibile con il semplice ls

G) vediamo ora come il file pippo.txt se non si usa il comando sudo sara solo ed esclusivamente in modalita lettura



(scusi ma non riesco a capire perche e in bassa risoluzione)

H) il comando e il seguente rm -r amici

```
File Actions Edit View Help
(kali㉿kali)-[~]
$ cd studenti/matteo
(kali㉿kali)-[~/studenti/matteo]
$ rmdir amici
(kali㉿kali)-[~/studenti/matteo]
$ ls system
(kali㉿kali)-[~/studenti/matteo]
$
```

- l) il comando per questa operazione e sempre il seguente `rm -r (directory)`

```
(kali㉿kali)-[~]
$ rm -r dos

(kali㉿kali)-[~]
$ rm -r studenti
rm: remove write-protected regular file 'studenti/anna/.casa/relazione.doc'? y
rm: remove write-protected regular file 'studenti/Nicola/scuola/compito.doc'? y
rm: remove write-protected regular empty file 'studenti/Nicola/lavoro/pippo.txt'? y

(kali㉿kali)-[~]
$ rm -r windows

(kali㉿kali)-[~]
$ ls
Desktop Documents Downloads Music Pictures Public Templates Videos

(kali㉿kali)-[~]
$
```

Vediamo infatti che le cartelle create in precedenza non esistono piu.

Seconda parte

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

Prova dei comandi

```

kali-linux-2023.3-virtualbox-amd64 [In esecuzione] - Oracle VM VirtualBox
File Macchina Visualizza Inserimento Dispositivi Aiuto
File Actions Edit View Help
(kali@kali)-[~]
$ w
13:25:19 up 3:24, 1 user, load average: 0.10, 0.17, 0.10
USER TTY FROM LOGIN@ IDLE JCPU PCPU WHAT
kali - 10:00 3:24m 0.00s 0.02s lightdm --session-child 13 24

(kali@kali)-[~]
$ who
kali tty7 2024-01-09 10:00 (:0)

(kali@kali)-[~]
$ who am i

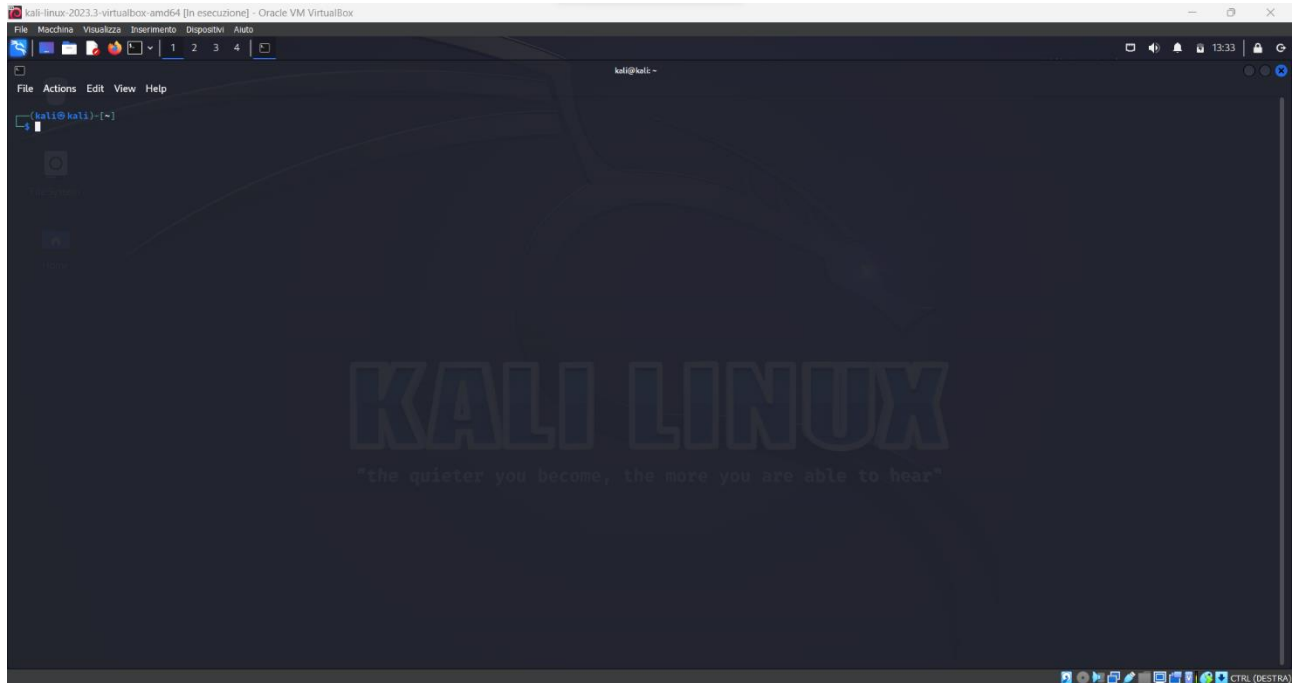
(kali@kali)-[~]
$ who am i

(kali@kali)-[~]
$ sudo who am i
[sudo] password for kali:
kali pts/1 2024-01-09 13:29

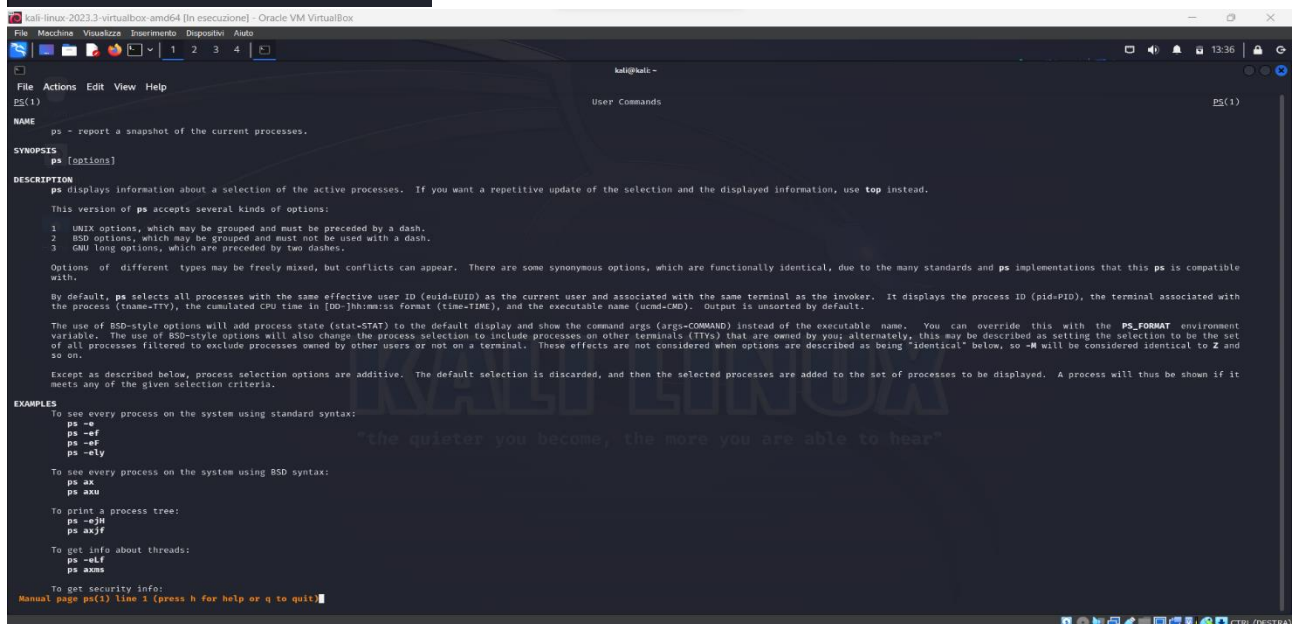
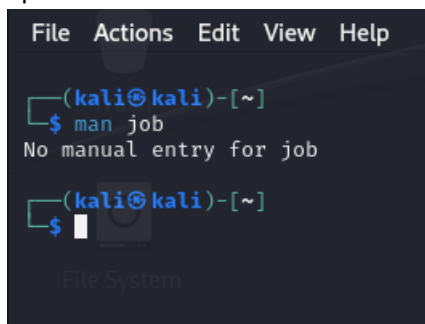
(kali@kali)-[~]
$

```

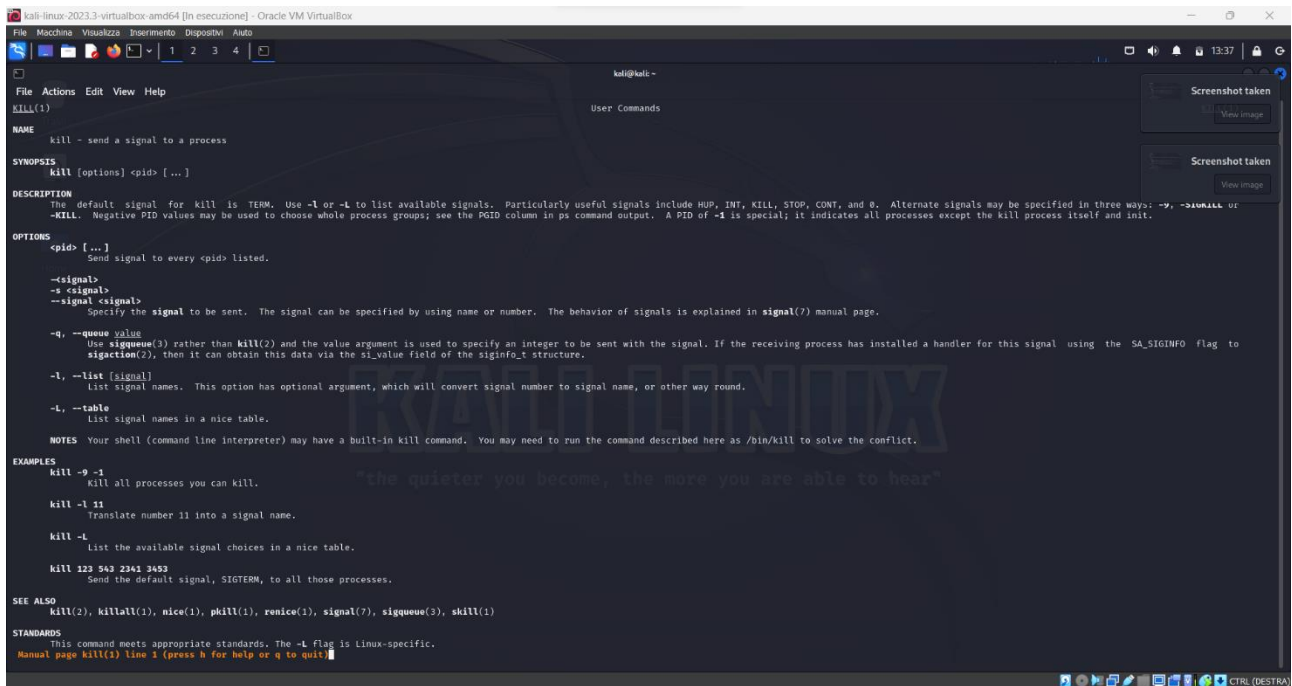

1) apro il terminale di comando



2) apro i manuali richiesti

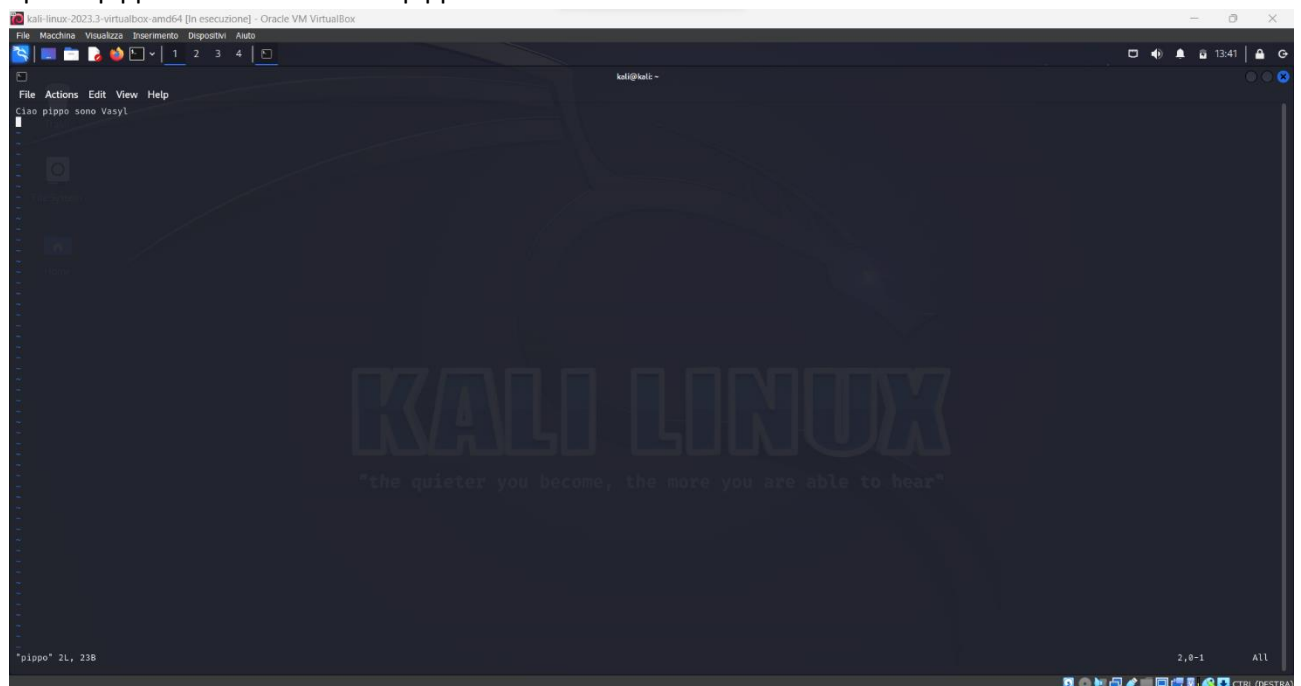


Ps questa in alto

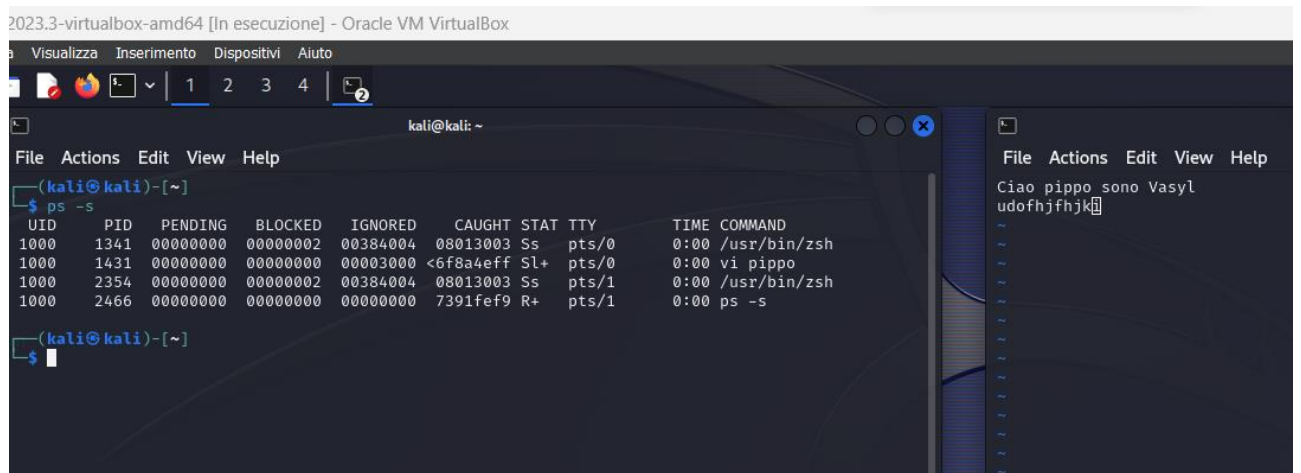


Kill questa in alto

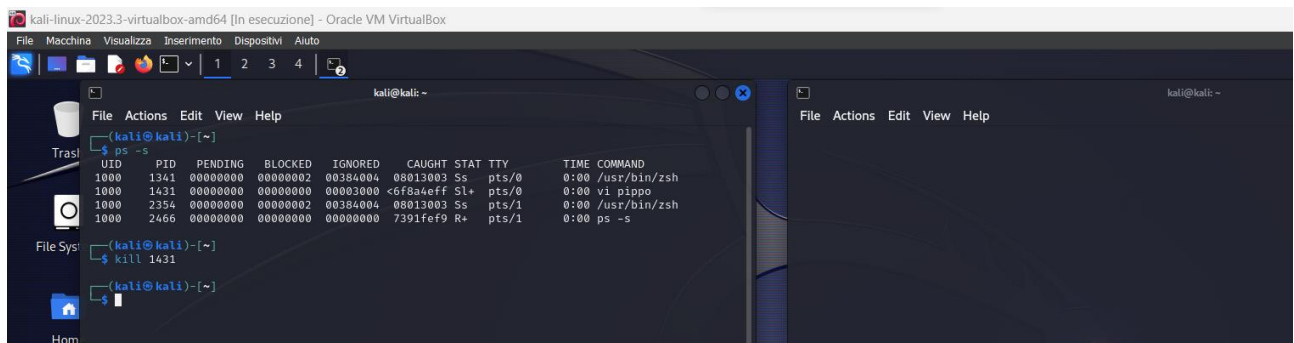
3) apro vi pippo con il comando : vi pippo



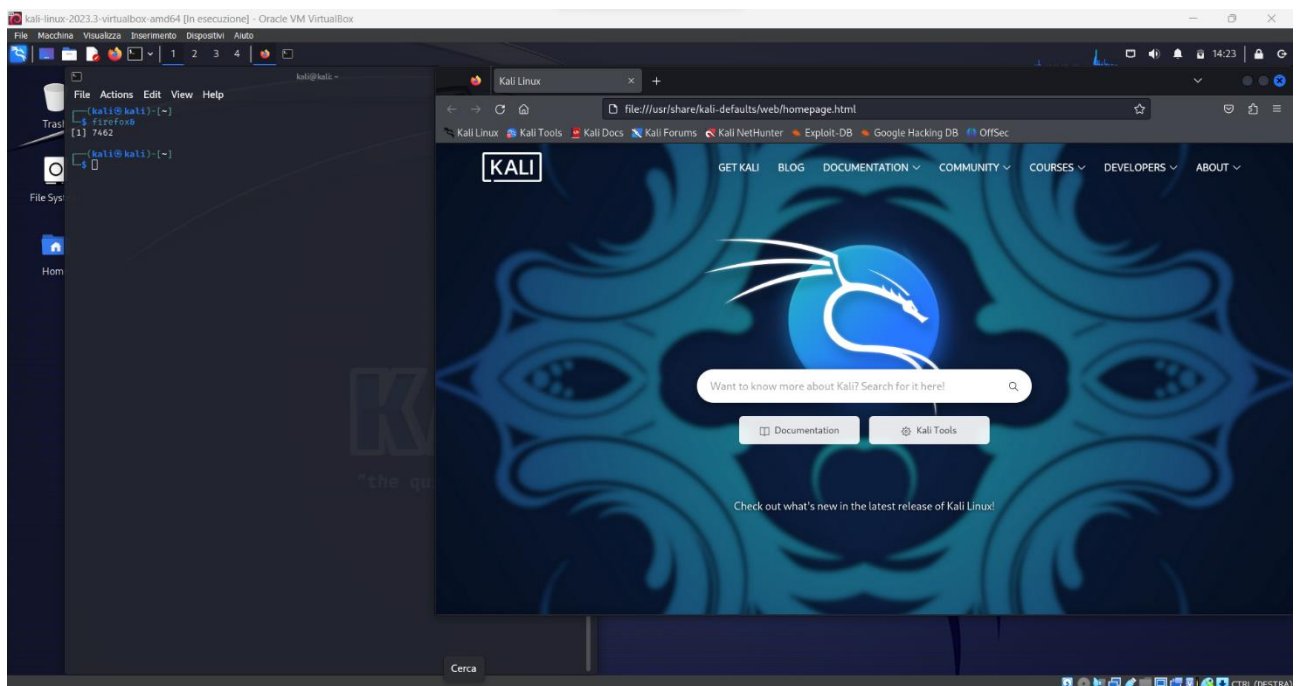
4) aprendo un secondo terminale e inserendo il comando ps -s riesco a vedere i processi attualmente attivi tra i quali (PIPPPO)



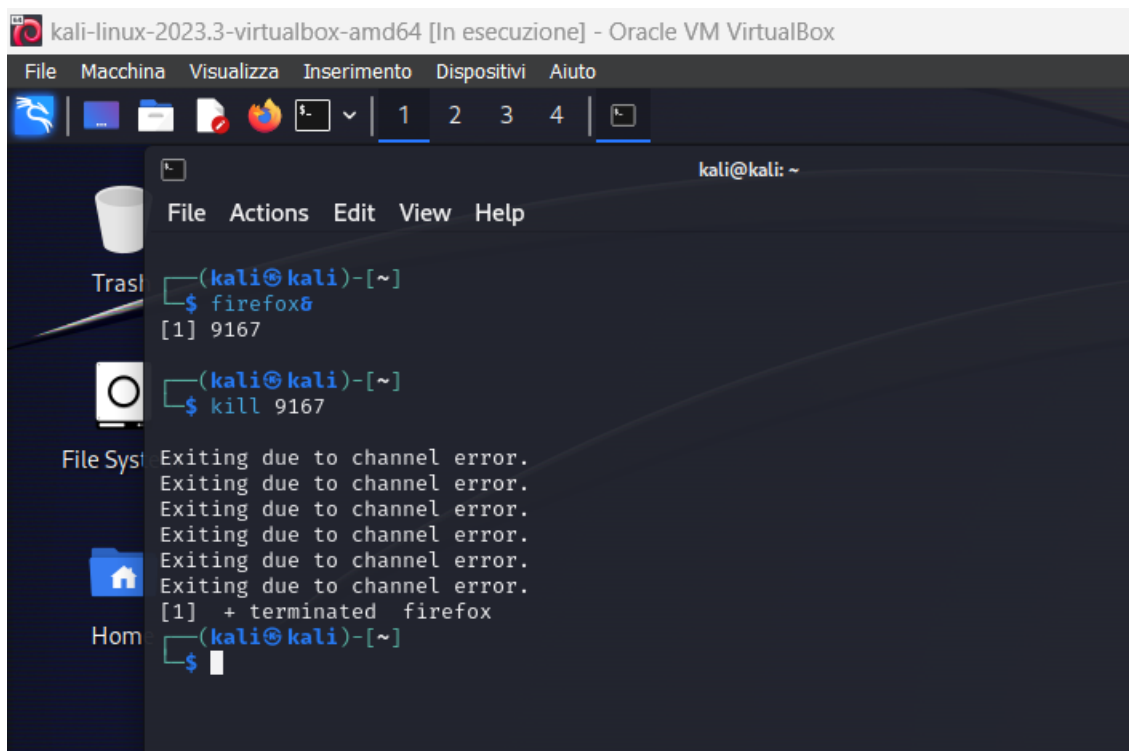
5) identificato il PID di vi PIPPO con il comando kill (PID del processo da spegnere),riusciamo a far terminare il processo Vi PIPPO nell altro terminale



6) per lanciare firefox il background il comando da usare e (nome programma)&



7/8) visto che firefox e gia in background ora terminiamolo con il comando kill + PID di firefox



9) ora non ci resta altro che controllare lo spazio che stiamo occupando utilizziamo quindi il comando

