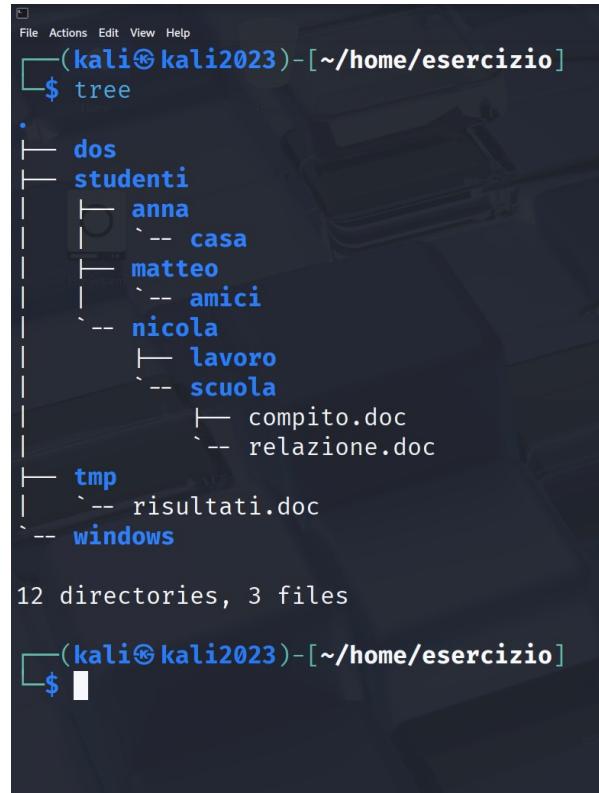


ESERCIZIO W5D1

Parte uno

IMMAGINE 1



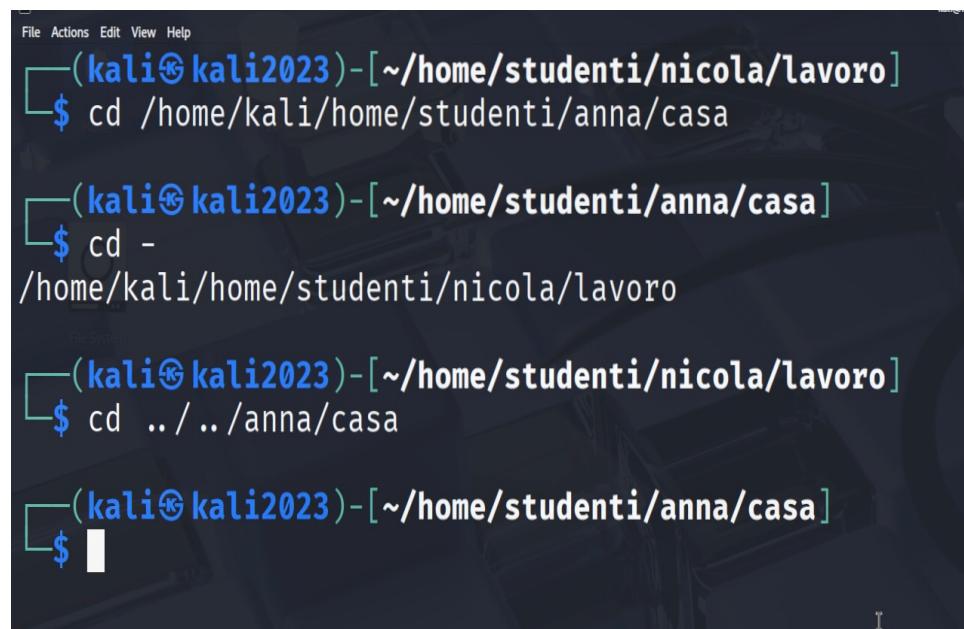
```
(kali㉿kali2023)-[~/home/esercizio]
$ tree
.
├── dos
├── studenti
│   ├── anna
│   │   ├── casa
│   │   ├── matteo
│   │   │   ├── amici
│   │   └── nicola
│   │       ├── lavoro
│   │       └── scuola
│   │           ├── compito.doc
│   │           └── relazione.doc
└── tmp
    └── risultati.doc
└── windows

12 directories, 3 files

(kali㉿kali2023)-[~/home/esercizio]
$
```

- Nell'immagine 1 abbiamo preparato tutto il necessario per l'esercizio, abbiamo creato le varei directory e i vari file
- Con il comando mkdir abbiamo creato (dos,studenti,tmp,windows) dentro studenti abbiamo inserito (anna/casa .. matteo/amici .. nicola/{lavoro,scuola}) , sotto scuola abbiamo creato due file con il comando "touch" (completo.doc;relazione.doc), sempre con il comando "touch" entriamo nella directory tmp e aggiungiamo il file " risultati.doc"

IMMAGINE 2



```
(kali㉿kali2023)-[~/home/studenti/nicola/lavoro]
$ cd /home/kali/home/studenti/anna/casa

(kali㉿kali2023)-[~/home/studenti/anna/casa]
$ cd -
/home/kali/home/studenti/nicola/lavoro

(kali㉿kali2023)-[~/home/studenti/nicola/lavoro]
$ cd ../../anna/casa

(kali㉿kali2023)-[~/home/studenti/anna/casa]
$
```

- Nell'immagine 2 ci siamo spostati dalla directorynicola/lavoro alla directoryanna/casa con il comando cd con i due percorsi (assoluto e relativo)come si può vedere nell'iimmagine
- nell'immagine 2 si puo notare il comando "cd -" usato per tornare nella directory precedente

```

File Actions Edit View Help
└─( kali㉿kali2023 )-[ ~/home/studenti/anna/casa ]
$ cp .. / .. /nicola/scuola/compito.doc .
└─( kali㉿kali2023 )-[ ~/home/studenti/anna/casa ]
$ ls
compito.doc
└─( kali㉿kali2023 )-[ ~/home/studenti/anna/casa ]
$ █

```

- Con il comando "cp" e il percorso di cosa vogliamo copiare e mettendo il punto alla fine, copiamo il file scritto, in questo caso "compito.doc" , avviando il comando incolliamo il file anche nella directory in cui siamo
- Usando il comando "ls" possiamo vedere che il file è stato copiato correttamente

IMMAGINE 4

```

└─( kali㉿kali2023 )-[ ~/home/studenti/anna/casa ]
$ mv .. / .. /nicola/scuola/relazione.doc .
└─( kali㉿kali2023 )-[ ~/home/studenti/anna/casa ]
$ ls
compito.doc  relazione.doc
└─( kali㉿kali2023 )-[ ~/home/studenti/anna/casa ]
$ █

```

- Nell'immagine 4 abbiamo spostato il file "relazione.doc" con il comando "mv" dalla directory/nicola/scuola alla directory/anna/casa
- Sempre con il comando "ls" possiamo vedere come il file è stato spostato correttamente

```
File Actions Edit View Help
└──(kali㉿kali2023)-[~/home]
    $ ls
    dos studenti tmp windows

└──(kali㉿kali2023)-[~/home]
    $ rm -rf tmp
    File System
└──(kali㉿kali2023)-[~/home]
    $ ls
    dos studenti windows

└──(kali㉿kali2023)-[~/home]
    $ ┌─[
```

- Nell'immagine 5 vediamo come, con il comando "rm" (serve per rimuovere un file) e aggiungendo -rf (cancelliamo anche i documenti dentro la directory) , seguito dal nome della nostra directory "tmp".. possiamo vedere come abbiamo cancellato tutto, usando sempre il comando "ls" nella directory "home" vediamo che la directory tmp e i file contenuti all'interno siano stati cancellati

IMMAGINE 6

```
File Actions Edit View Help
└──(kali㉿kali2023)-[~/home]
    $ cd studenti/nicola/lavoro

└──(kali㉿kali2023)-[~/home/studenti/nicola/lavoro]
    $ touch pippo.txt

└──(kali㉿kali2023)-[~/home/studenti/nicola/lavoro]
    $ ls
    pippo.txt

└──(kali㉿kali2023)-[~/home/studenti/nicola/lavoro]
    $ ┌─[
```

- Nell'immagine 6 ci siamo spostati nella directory "lavoro" e con il comando "touch" abbiamo creato il file "pippo.txt"

IMMAGINE 7

```
(kali㉿kali2023)-[~/home/studenti/lavoro]
└─$ ls -l
total 0
-rw-rw-r-- 1 kali kali 0 Jul 24 14:15 pippo.txt

(kali㉿kali2023)-[~/home/studenti/lavoro]
└─$ chmod g-w
chmod: missing operand after 'g-w'
Try 'chmod --help' for more information.

(kali㉿kali2023)-[~/home/studenti/lavoro]
└─$ chmod g-w pippo.txt

(kali㉿kali2023)-[~/home/studenti/lavoro]
└─$ ls -l
total 0
-rw-r--r-- 1 kali kali 0 Jul 24 14:15 pippo.txt

(kali㉿kali2023)-[~/home/studenti/lavoro]
└─$ █
```

- Nell'immagine 7 abbiamo reso il file "pippo.txt" leggibile e scrivibile solo per l'utente..
- Con il comando "chmod g-w pippo.txt" abbiamo tolto la possibilità di scrivere al gruppo
- Con il comando "ls -l" possiamo vedere i permessi dell'utente, del gruppo e di tutti gli altri

IMMAGINE 8

```
(kali㉿kali2023)-[~/home]
└─$ cd studenti

(kali㉿kali2023)-[~/home/studenti]
└─$ ls
anna matteo nicola

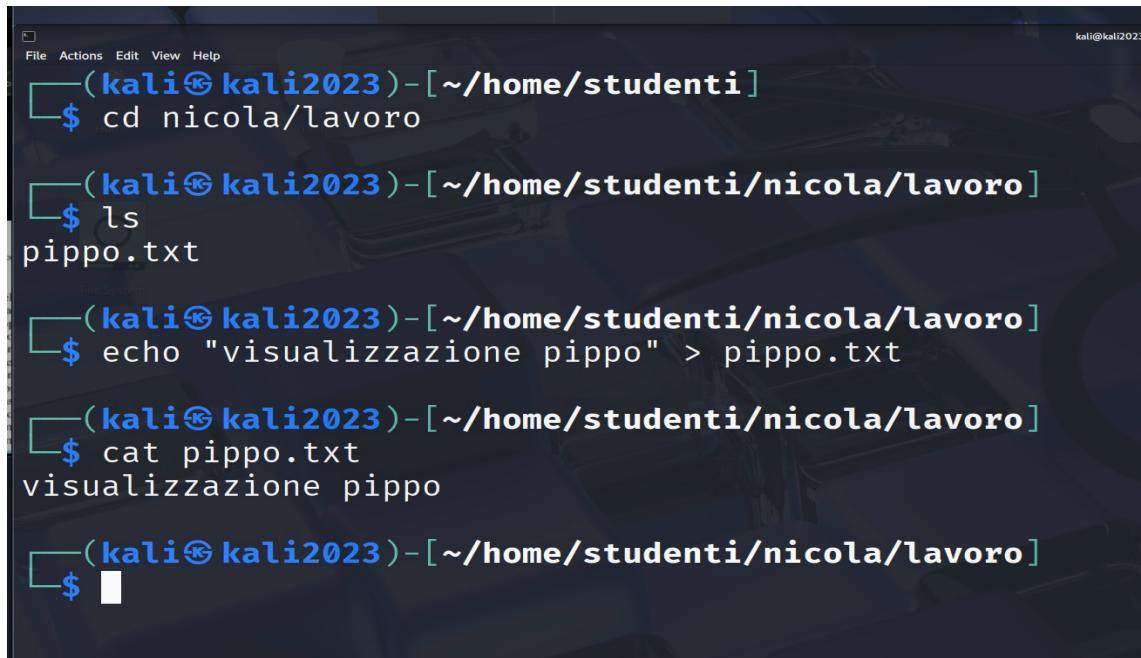
(kali㉿kali2023)-[~/home/studenti]
└─$ mv anna/.anna

(kali㉿kali2023)-[~/home/studenti]
└─$ ls
matteo nicola

(kali㉿kali2023)-[~/home/studenti]
└─$ ls -a
. .. .anna matteo nicola

(kali㉿kali2023)-[~/home/studenti]
└─$ █
```

- Nell'immagine 8 abbiamo nascosto la cartella "anna" con il comando "mv anna/.anna" abbiamo spostato il file, se facciamo il comando "ls" non riusciamo a vederlo..
- Se usiamo il comando "ls -a" possiamo vedere tutte le cartelle nascoste della directory "studenti", così riusciamo a dimostrare che la cartella è stata nascosta correttamente



```
(kali㉿kali2023)-[~/home/studenti/nicola/lavoro]
$ cd nicola/lavoro

(kali㉿kali2023)-[~/home/studenti/nicola/lavoro]
$ ls
pippo.txt

(kali㉿kali2023)-[~/home/studenti/nicola/lavoro]
$ echo "visualizzazione pippo" > pippo.txt

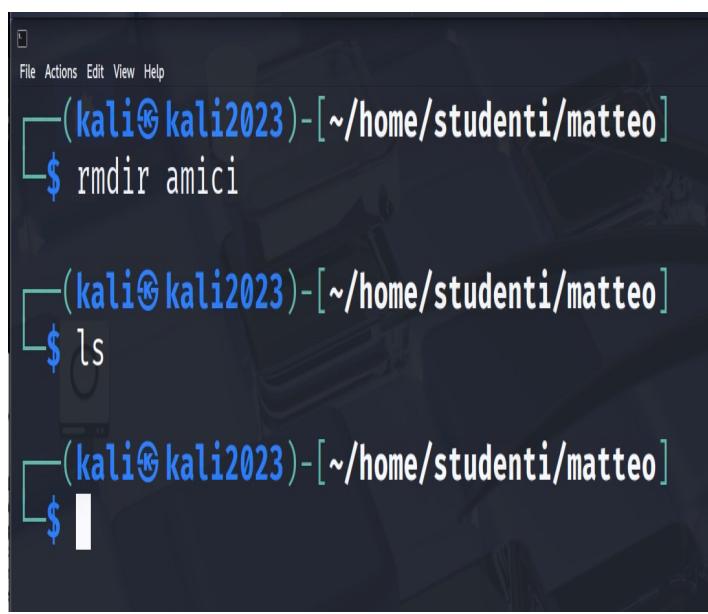
(kali㉿kali2023)-[~/home/studenti/nicola/lavoro]
$ cat pippo.txt
visualizzazione pippo

(kali㉿kali2023)-[~/home/studenti/nicola/lavoro]
$
```

- Nell'immagine 9 abbiamo visualizzato il contenuto del file "pippo.txt"
- Prima però abbiamo scritto "visualizzazione pippo" dentro il file per essere sicuri di riuscire a vederlo
- Abbiamo usato il comando "echo "visualizzazione pippo" > pippo.txt"; con il comando "echo" abbiamo scritto "visualizzazione pippo" e con > (serve per inserire dentro il file quello che lo precede) lo abbiamo inserito dentro il file "pippo.txt"
- Non ci resta altro che usare il comando "cat pippo.txt" per visualizzare il contenuto del file e verificare che ci sia scritto "visualizzazione pippo"

IMMAGINE 10

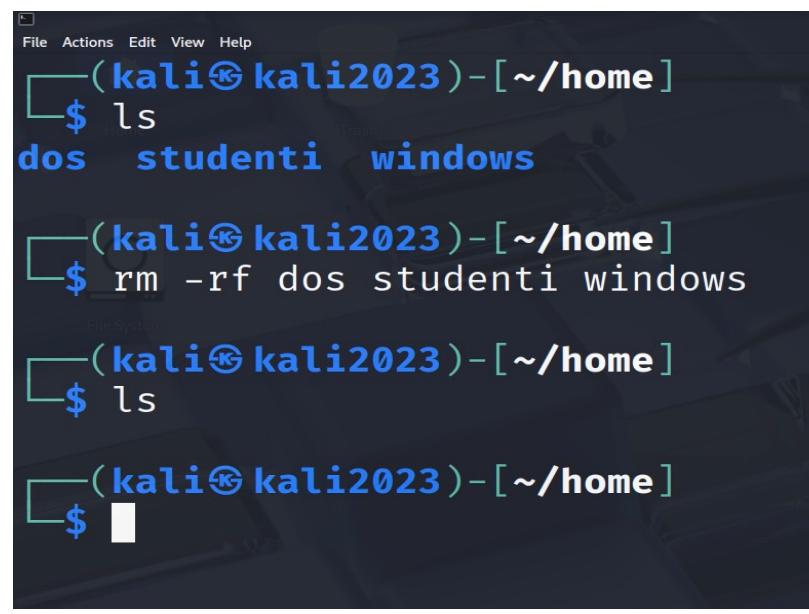
IMMAGINE 11



```
File Actions Edit View Help
(kali㉿kali2023)-[~/home/studenti/matteo]
$ rmdir amici

(kali㉿kali2023)-[~/home/studenti/matteo]
$ ls
dos studenti windows

(kali㉿kali2023)-[~/home/studenti/matteo]
$
```



```
File Actions Edit View Help
(kali㉿kali2023)-[~/home]
$ ls
dos studenti windows

(kali㉿kali2023)-[~/home]
$ rm -rf dos studenti windows

(kali㉿kali2023)-[~/home]
$ ls

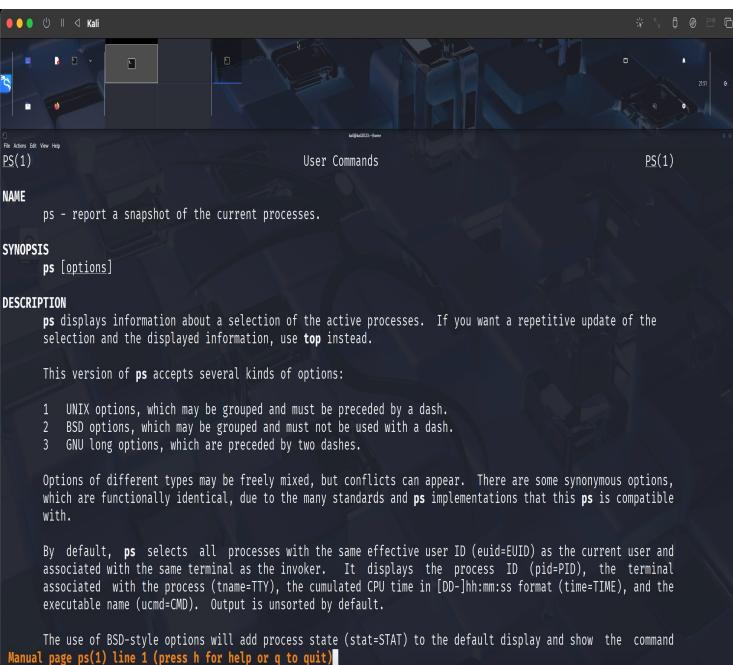
(kali㉿kali2023)-[~/home]
$
```

- Nell'immagine 10 ci viene chiesto di rimuovere la directory "amici" dalla directory "matteo", facendo il comando "rmdir amici" dentro la directory "matteo"
- Una volta rimossa la directory "amici" torniamo alla directory "home" (ci spostiamo nell'immagine 11), facciamo il comando "ls" e vediamo che ci sono 3 directory rimanenti (dos,studenti,windows).. l'esercizio ci chiede di rimuoverle tutte, quindi con il comando "rm -rf" visto anche prima, aggiungendo le directory da rimuovere, possiamo avviare il comando..
- facciamo "ls" e vediamo come le directory siano state cancellate correttamente

Fine Prima Parte

Inizio Seconda parte

IMMAGINE 1



```

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.

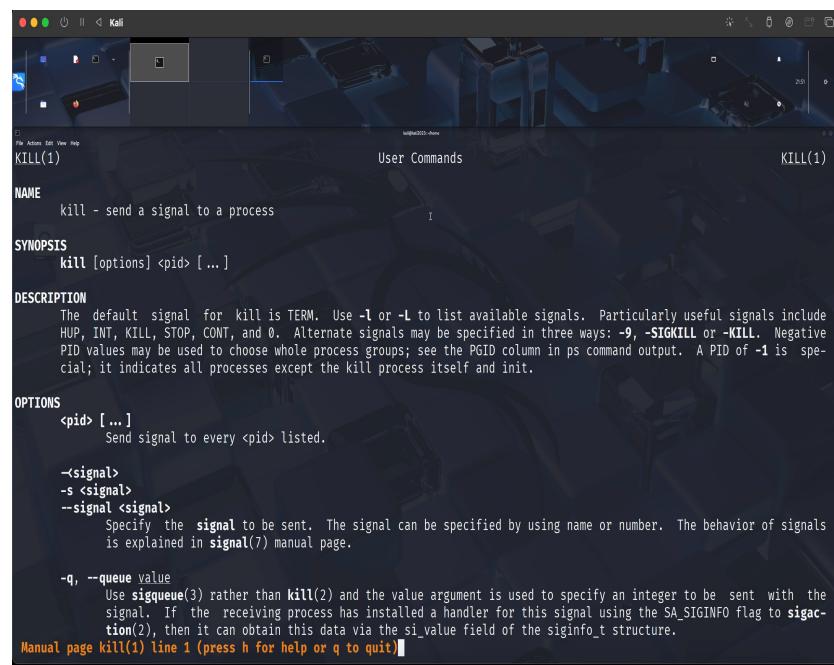
Options of different types may be freely mixed, but conflicts can appear. There are some synonymous options,
which are functionally identical, due to the many standards and ps implementations that this ps is compatible
with.

By default, ps selects all processes with the same effective user ID (euid=EUID) as the current user and
associated with the same terminal as the invoker. It displays the process ID (pid=PID), the terminal
associated with the process (tname=TTY), the cumulated CPU time in [DD-]hh:mm:ss format (time=TIME), and the
executive name (ucmd=CMD). Output is unsorted by default.

The use of BSD-style options will add process state (stat=STAT) to the default display and show the command
Manual page ps(1) line 1 (press h for help or q to quit)

```

IMMAGINE 2



```

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 QUIT. 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 spe-
cial; it indicates all processes except the kill process itself and init.

OPTIONS
<pid> [ ... ]
    Send signal to every <pid> listed.

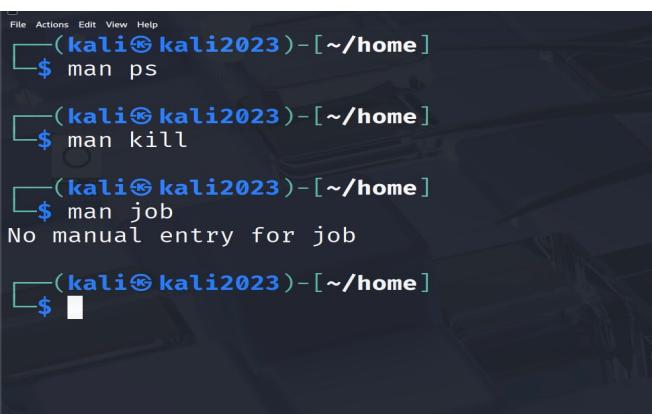
<signal>
-s <signal>
-signal <signal>
    Specify the signal to be sent. The signal can be specified by using name or number. The behavior of signals
    is explained in signal(7) manual page.

-q, --queue value
    Use sigqueue(3) rather than kill(2) and the value argument is used to specify an integer to be sent with the
    signal. If the receiving process has installed a handler for this signal using the SA_SIGINFO flag to sigac-
    tion(2), then it can obtain this data via the si_value field of the siginfo_t structure.

Manual page kill(1) line 1 (press h for help or q to quit)

```

IMMAGINE 3



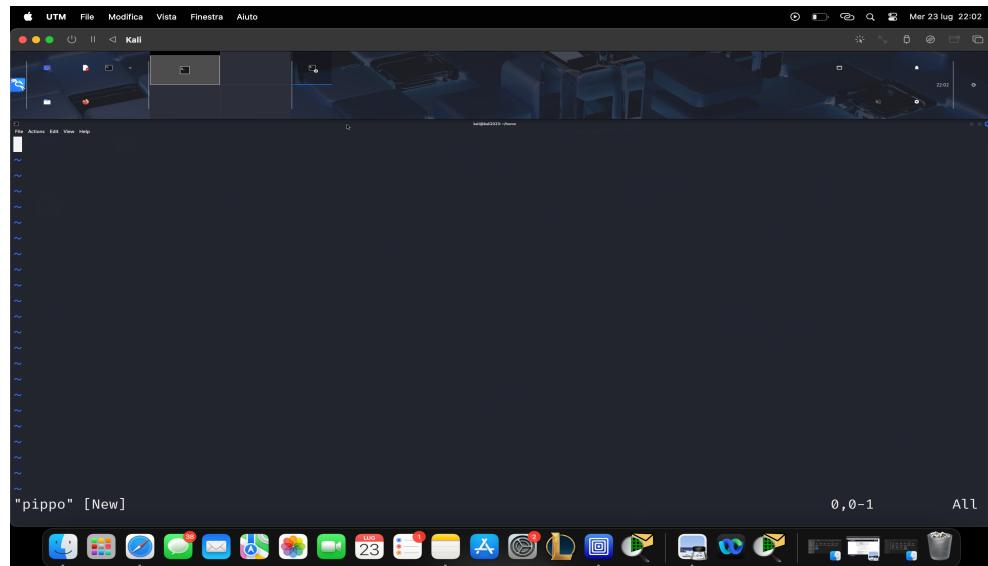
```

File Actions Edit View Help
(kali㉿kali2023)-[~/home]
$ man ps
(kali㉿kali2023)-[~/home]
$ man kill
(kali㉿kali2023)-[~/home]
$ man job
No manual entry for job
(kali㉿kali2023)-[~/home]
$ 

```

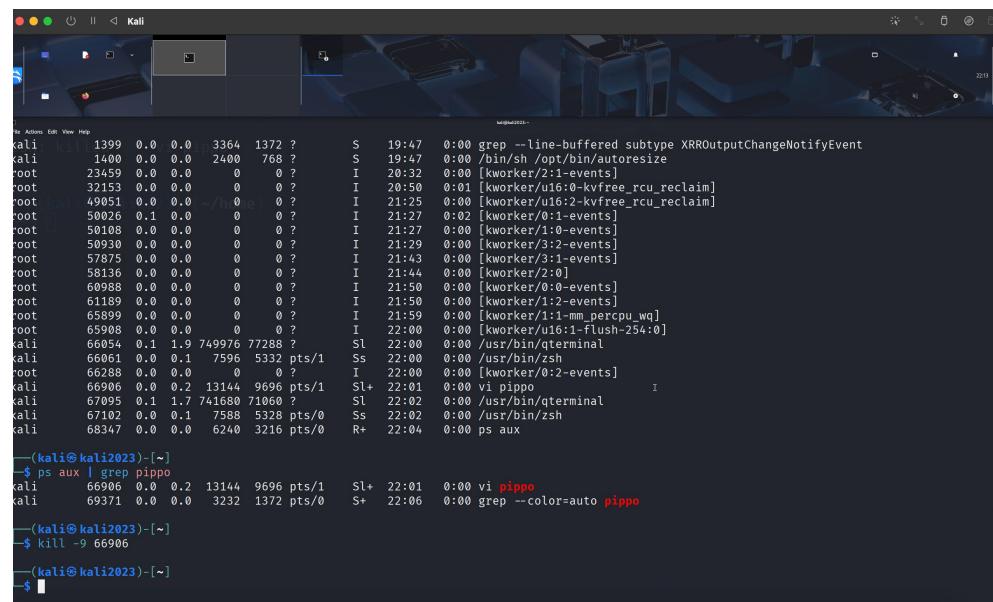
- Nelle immagine 1/2 possiamo vedere che abbiamo visualizzato il manuale e letto cosa vuol dire "ps" e "kill".. Nell'immagine 3 invece il manuale per il comando "jobs" non lo trova

IMMAGINE 4



- Abbiamo aperto un terminale e abbiamo eseguito il comando "vi pippo", come si può vedere nell'immagine 4

IMMAGINE 5



```

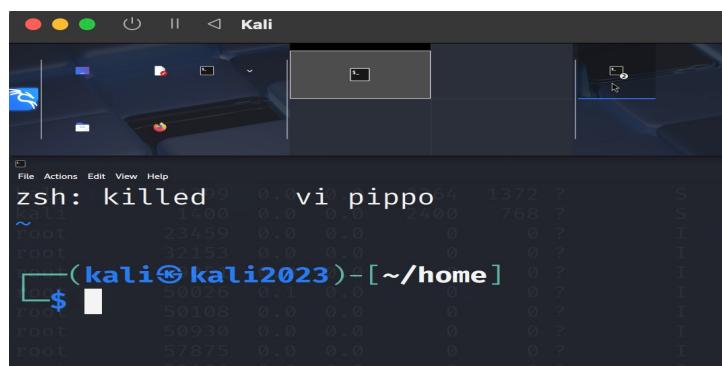
kali 1399 0.0 0.0 3364 1372 ? S 19:47 0:00 grep --line-buffered subtype XRROutputChangeNotifyEvent
kali 1400 0.0 0.0 2400 768 ? S 19:47 0:00 /bin/sh /opt/bin/autoresize
root 23459 0.0 0.0 0 0 ? I 20:32 0:00 [kworker/2:1-events]
root 32153 0.0 0.0 0 0 ? I 20:50 0:00 [kworker/u16:0-kvfree_rcu_reclaim]
root 49051 0.0 0.0 0 0 ? I 21:25 0:00 [kworker/u16:2-kvfree_rcu_reclaim]
root 50026 0.1 0.0 0 0 ? I 21:27 0:02 [kworker/0:1-events]
root 50108 0.0 0.0 0 0 ? I 21:27 0:02 [kworker/1:0-events]
root 50930 0.0 0.0 0 0 ? I 21:29 0:00 [kworker/3:2-events]
root 57875 0.0 0.0 0 0 ? I 21:43 0:00 [kworker/3:1-events]
root 58136 0.0 0.0 0 0 ? I 21:44 0:00 [kworker/2:0]
root 60988 0.0 0.0 0 0 ? I 21:50 0:00 [kworker/0:0-events]
root 61189 0.0 0.0 0 0 ? I 21:50 0:00 [kworker/1:2-events]
root 65899 0.0 0.0 0 0 ? I 21:59 0:00 [kworker/1:1-mm_percpu_wq]
root 65908 0.0 0.0 0 0 ? I 22:00 0:00 [kworker/u16:1-flush-254:0]
kali 66054 0.1 1.9 749976 77288 ? SL 22:00 0:00 /usr/bin/qterminal
kali 66061 0.0 0.1 7596 5332 pts/1 Ss 22:00 0:00 /usr/bin/zsh
root 66288 0.0 0.0 0 0 ? I 22:00 0:00 [kworker/0:2-events]
kali 66906 0.0 0.2 13144 9696 pts/1 SL+ 22:01 0:00 vi pippo
kali 67095 0.1 1.7 741680 71060 ? SL 22:02 0:00 /usr/bin/qterminal
kali 67102 0.0 0.1 7588 5328 pts/0 Ss 22:02 0:00 /usr/bin/zsh
kali 68347 0.0 0.0 6240 3216 pts/0 R+ 22:04 0:00 ps aux

---(kali㉿kalilinux)-[~]
-$ ps aux | grep pippo
kali 66906 0.0 0.2 13144 9696 pts/1 SL+ 22:01 0:00 vi pippo
kali 69371 0.0 0.0 3232 1372 pts/0 S+ 22:06 0:00 grep -color=auto pippo

---(kali㉿kalilinux)-[~]
-$ kill -9 66906
---(kali㉿kalilinux)-[~]
-$

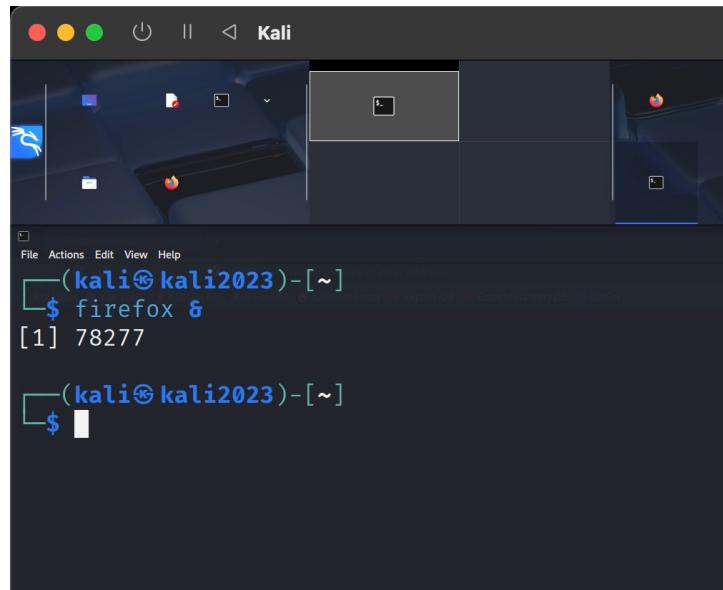
```

IMMAGINE 6



- Nell'immagine 5 abbiamo aperto un altro terminale e abbiamo fatto partire il comando "ps aux" per vedere i processi, successivamente abbiamo usato "ps aux | grep pippo" per visualizzare il processo pippo
- Dopodichè abbiamo usato il comando "kill(comando per terminare un processo) -9 (serve per terminare il processo senza risposta) 66906(questo numero indica il PID che indica il nostro processo trovato grazie al comando "ps aux | grep pippo"... quindi il comando è "kill -9 66906" nel nostro caso.
- Come possiamo vedere nell'immagine 6 il processo è stato terminato correttamente

IMMAGINE 7



- Nell'immagine 7 abbiamo lanciato firefox con il comando "firefox &"
- Come si può vedere nelle pagine di kali in alto a destra dell'immagine, firefox è partito
- Qui sotto un immagine di prova

IMMAGINE DI PROVA

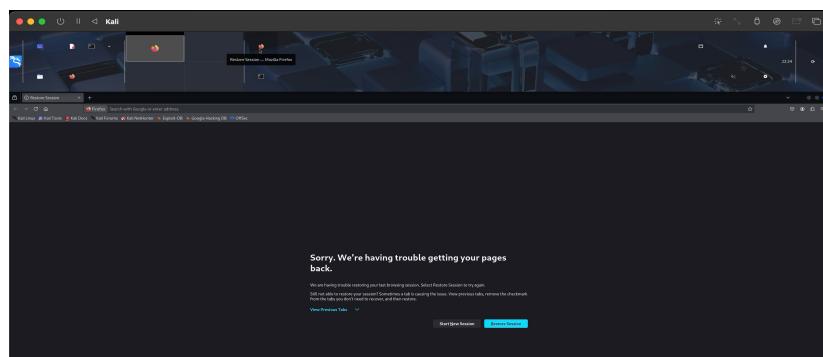
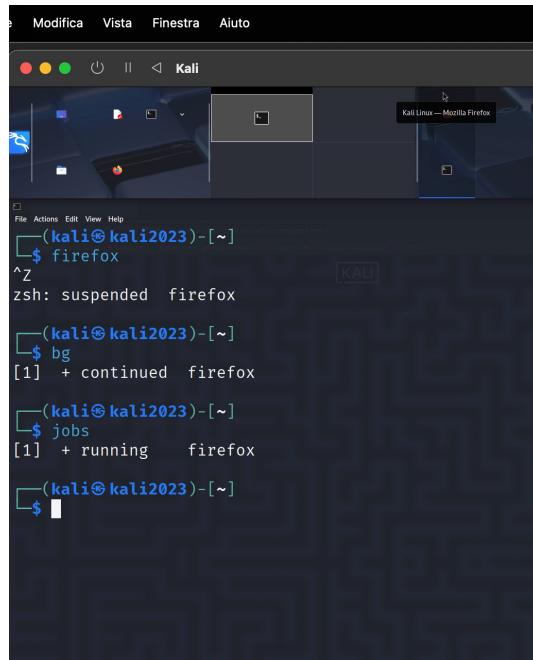


IMMAGINE 8



```
(kali㉿kali2023)~]$ firefox
^Z
zsh: suspended  firefox

(kali㉿kali2023)~]$ bg
[1] + continued  firefox

(kali㉿kali2023)~]$ jobs
[1] + running    firefox

(kali㉿kali2023)~]$
```

- Analizziamo l'immagine 8, come prima cosa abbiamo avviato firefox, premendo "control+z" sospendiamo il processo in esecuzione,cioè firefox, e digitiamo "bg" che sta per "background" e con il comando "jobs" vediamo che è in esecuzione

IMMAGINE 9



```
(kali㉿kali2023)~]$ firefox
^Z
zsh: suspended  firefox

(kali㉿kali2023)~]$ ps
  PID TTY          TIME CMD
 1439 pts/0        00:00:00 zsh
 2840 pts/0        00:00:01 firefox-esr
 2908 pts/0        00:00:00 Socket Process
 2931 pts/0        00:00:00 Privileged Cont
 2978 pts/0        00:00:00 WebExtensions
 3038 pts/0        00:00:00 Utility Process
 3050 pts/0        00:00:00 Web Content
 3053 pts/0        00:00:00 Web Content
 3058 pts/0        00:00:00 Web Content
 3162 pts/0        00:00:00 ps

(kali㉿kali2023)~]$ kill -9 2840
[1] + killed      firefox

(kali㉿kali2023)~]$
```

- In questa immagine numero 9, abbiamo avviato firefox, lo abbiamo sospeso con "control+z"
- Abbiamo usato il comando "ps" per trovare il PID di firefox (evidenziato nell'immagine)
- Dopodichè abbiamo usato il comando "kill -9 2840" e abbiamo fatto terminare il processo correttamente

File Modifica Vista Finestra Aiuto

Mer 23 lug 22:35

Kali

```
(kali㉿kali2023) [~]
$ sudo fdisk -l
Disk /dev/vda: 64 GiB, 68719476736 bytes, 134217728 sectors
Units: sectors of 1 * 512 = 512 bytes
Sector size (logical/physical): 512 bytes / 512 bytes
I/O size (minimum/optimal): 512 bytes / 512 bytes
Disklabel type: gpt
Disk identifier: E0133FB3-C6D1-4BFC-8FD2-32466FD19302

Device      Start     End   Sectors  Size Type
/dev/vda1    2048 1050623 1048576 512M EFI System
/dev/vda2 1050624 132216831 131166208 62.5G Linux filesystem
/dev/vda3 132216832 134215679 1998848 976M Linux swap

(kali㉿kali2023) [~]
$ df -h
Filesystem  Size  Used Avail Use% Mounted on
udev        1.9G    0  1.9G  0% /dev
tmpfs       392M  1.3M 390M  1% /run
/dev/vda2    62G  20G  39G 34% /
tmpfs       2.0G  4.0K  2.0G  1% /dev/shm
efivarsfs   256K  30K 227K 12% /sys/firmware/efi/efivars
tmpfs       5.0M    0  5.0M  0% /run/lock
tmpfs       1.0M    0  1.0M  0% /run/credentials/systemd-journald.service
tmpfs       2.0G   48K  2.0G  1% /tmp
/dev/vda1    512M 168K 512M  1% /boot/efi
tmpfs       1.0M    0  1.0M  0% /run/credentials/serial-getty@ttyAMA0.service
tmpfs       1.0M    0  1.0M  0% /run/credentials/getty@tty1.service
tmpfs       392M 128K 391M  1% /run/user/1000
```



- Con i comandi "sudo fdisk -l" e "df -h" possiamo vedere come richiesto dall'esercizio la nostra memoria nel disco

FINE SECONDA PARTE