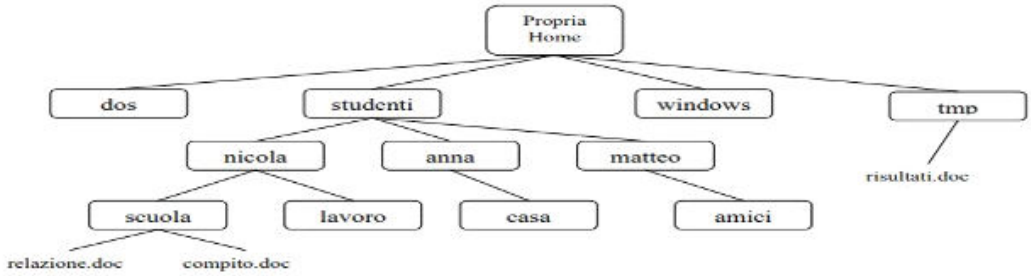
W5D1 esercitazione 1

Di Florin Eugen Peticaru

Nell'esercizio di oggi si richiedeva:

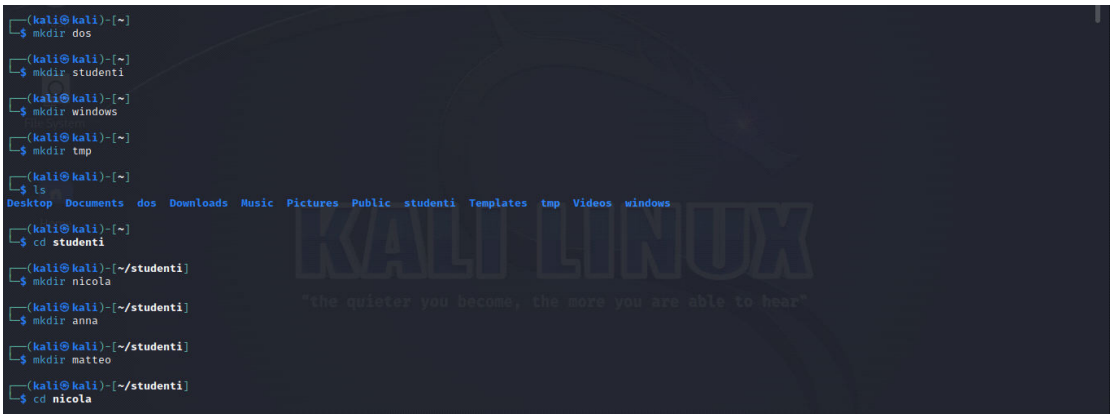
- La creazione/modifica ed eliminazione di alcuni file e cartelle tramite la shell di linux
- provare i comandi W - Who - Who am I
- l'avvio e l'interruzione dialcuni processi tramite i comandi Jobs - Ps - Kill

Iniziamo con la creazione di alcuni file e cartelle seguendo la tabella che ci è stata data

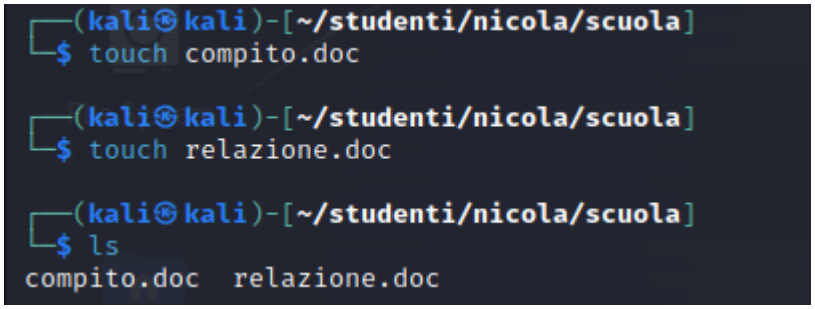


Utilizziamo da shell di comando i comandi:

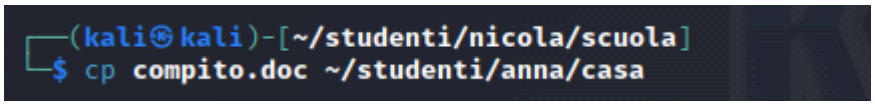
- *mkdir* - per creare una nuova cartella nella directory in cui ci troviamo
- *cd* - per spostarci da una directory all'altra
- *ls* - per vedere la lista di elementi (cartelle e file) presenti nella directory



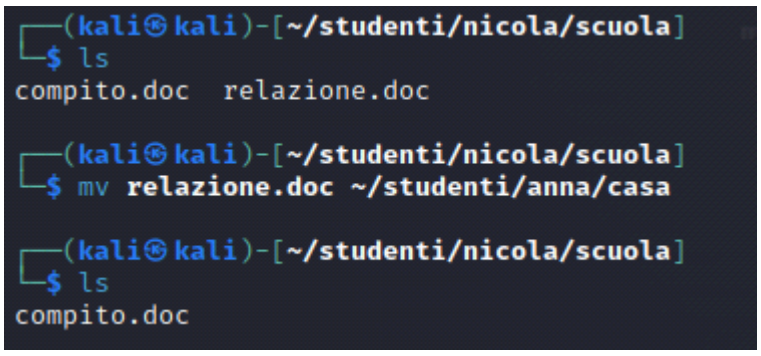
Per creare dei file utilizziamo il comando *Touch*



In seguito come richiesto copiamo il file *compito.doc* nella cartella **Casa** con il comando *cp*



e spostiamo il file *Relazione.doc* sempre nella cartella **Casa** con il comando *mv*



Adesso proseguiamo con l'eliminazione della cartella */tmp*

```
┌──(kali㉿kali)-[~/tmp]
└─$ cd ~

┌──(kali㉿kali)-[~]
└─$ rmdir tmp
rmdir: failed to remove 'tmp': Directory not empty

┌──(kali㉿kali)-[~]
└─$ cd tmp

┌──(kali㉿kali)-[~/tmp]
└─$ rm risultati.doc

┌──(kali㉿kali)-[~/tmp]
└─$ cd ~

┌──(kali㉿kali)-[~]
└─$ rmdir tmp

┌──(kali㉿kali)-[~]
└─$ ls
Desktop  Documents  dos  Downloads  Music  Pictures  Public  studenti  Templates  Videos  windows
```

Come da consegna creiamo il file *pippo.txt*

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

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

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

E cambiamo gli attributi come richiesto dalla consegna con il comando *chmod*

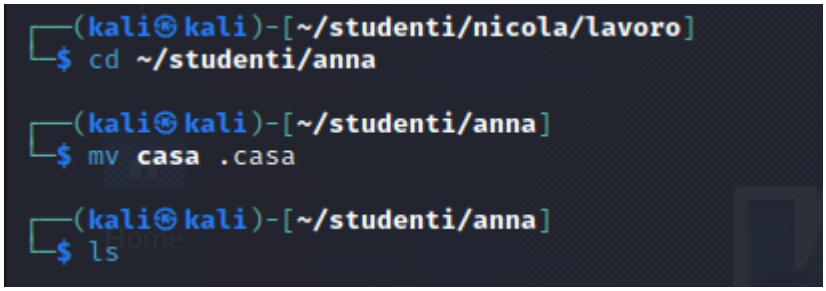
```
┌──(kali㉿kali)-[~/studenti/nicola/lavoro]
└─$ chmod u+w pippo.txt

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

┌──(kali㉿kali)-[~/studenti/nicola/lavoro]
└─$ chmod g+x pippo.txt

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

E quindi possiamo vedere gli attributi modificati

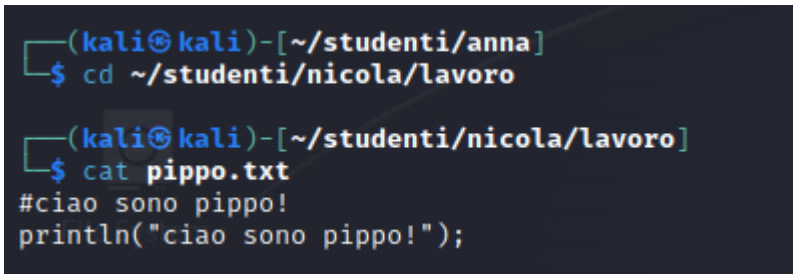
```
┌──(kali㉿kali)-[~/studenti/nicola/lavoro]
└─$ ls -l
total 4
-rwxr-xr-x 1 kali kali 47 Nov 20 14:28 pippo.txt
```

Adesso sempre seguendo la consegna nascondiamo il contenuto della cartella

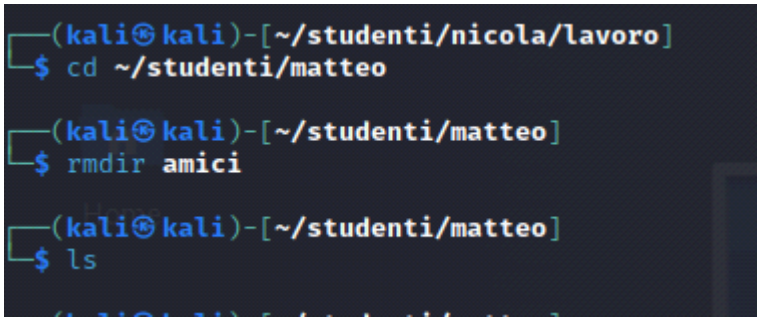
Anna sfruttando un errore di Linux che dice che se rinominiamo una directory o un file mettendo un punto all'inizio del nome diventa invisibile

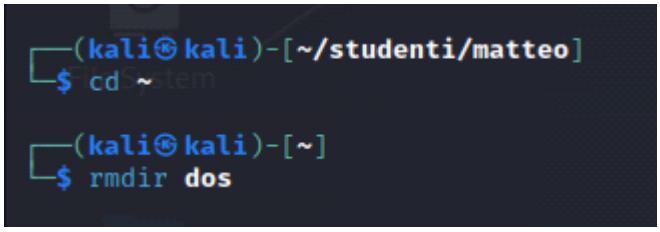


Il prossimo punto è tornare alla cartella **lavoro** e visualizzare il contenuto el file *pippo.txt* con il comando *cat*



Infine rimuovere prima la cartella **Amici** e poi tutte le cartelle che abbiamo creato





```
┌──(kali㉿kali)-[~/studenti/nicola/scuola]
└─$ rm compito.doc

┌──(kali㉿kali)-[~/studenti/nicola/scuola]
└─$ rm relazione.doc

┌──(kali㉿kali)-[~/studenti/nicola/scuola]
└─$ ls

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

┌──(kali㉿kali)-[~/studenti/nicola]
└─$ rmdir scuola

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

┌──(kali㉿kali)-[~/studenti/nicola/lavoro]
└─$ rm pippo.txt

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

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

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

┌──(kali㉿kali)-[~/studenti]
└─$ ls
anna  matteo
```

```
┌──(kali㉿kali)-[~/studenti/anna/casa]
└─$ rm compito.doc

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

┌──(kali㉿kali)-[~/studenti/anna]
└─$ rmdir casa

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

┌──(kali㉿kali)-[~/studenti]
└─$ rmdir anna

┌──(kali㉿kali)-[~/studenti]
└─$ ls
matteo

┌──(kali㉿kali)-[~/studenti]
└─$ rmdir matteo

┌──(kali㉿kali)-[~/studenti]
└─$ cd ~

┌──(kali㉿kali)-[~]
└─$ rmdir studenti
```

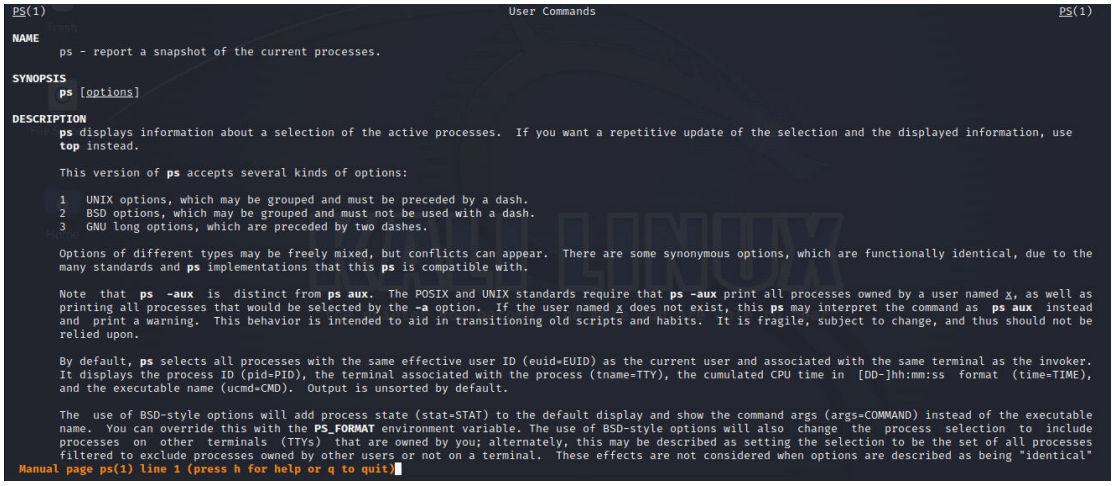
Così da poter vedere infine che tutte le cartelle sono state rimosse

```
┌──(kali㉿kali)-[~]
└─$ rmdir windows

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

Adesso eseguiamo il secondo punto della consegna che ci chiedeva di eseguire i 3 comandi *W - Who - Who am I*

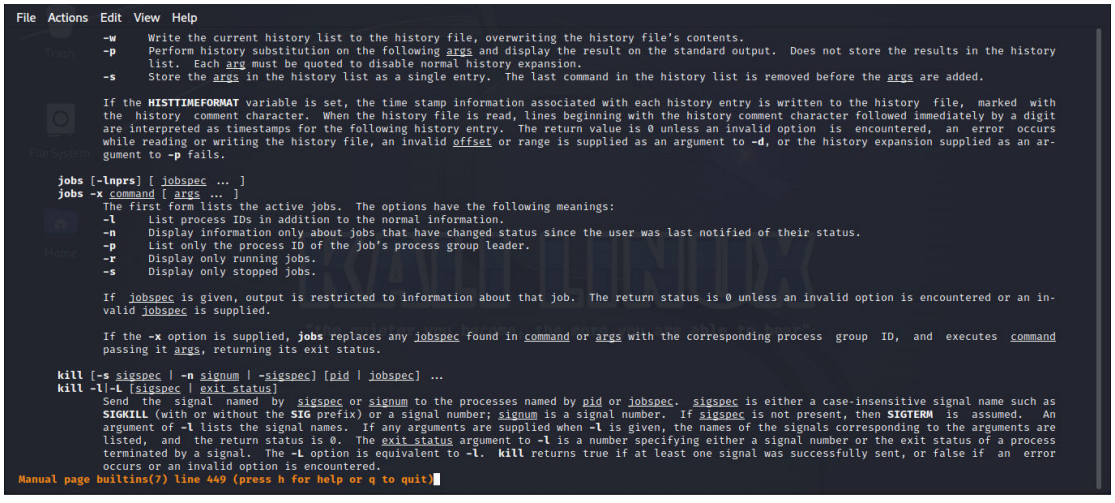
```
┌──(kali㉿kali)-[~]
└─$ w
 15:46:19 up  1:49,  1 user,  load average: 0.14, 0.30, 0.35
USER     TTY      FROM             LOGIN@   IDLE   JCPU   PCPU WHAT
kali     tty7     :0               14:10    1:48m  6:49   4.81s xfce4-session

┌──(kali㉿kali)-[~]
└─$ who
kali     tty7         2023-11-20 14:10 (:0)

┌──(kali㉿kali)-[~]
└─$ sudo who am i
[sudo] password for kali:
kali     pts/1        2023-11-20 15:46
```

Per iniziare a svolgere gli ultimi punti della consegna aprimo un nuovo terminale e come da prima richiesta leggiamo i manuali di *job, ps e kill* con il comando *man*



```
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.

       Note  that  ps  -aux  is  distinct from ps aux.  The POSIX and UNIX standards require that ps -aux print all processes owned by a user named x, as well as
       printing all processes that would be selected by the -a option.  If the user named x does not exist, this ps may interpret the command as  ps aux  instead
       and  print a warning.  This behavior is intended to aid in transitioning old scripts and habits.  It is fragile, subject to change, and thus should not be
       relied upon.

       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 executable 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 args (args=COMMAND) instead of the executable
       name.  You can override this with the PS_FORMAT environment variable. The use of BSD-style options will also  change  the  process  selection  to  include
       processes  on  other  terminals  (TTYs)  that are owned by you; alternately, this may be described as setting the selection to be the set of all processes
       filtered to exclude processes owned by other users or not on a terminal.  These effects are not considered when options are described as being "identical"
Manual page ps(1) line 1 (press h for help or q to quit)
```

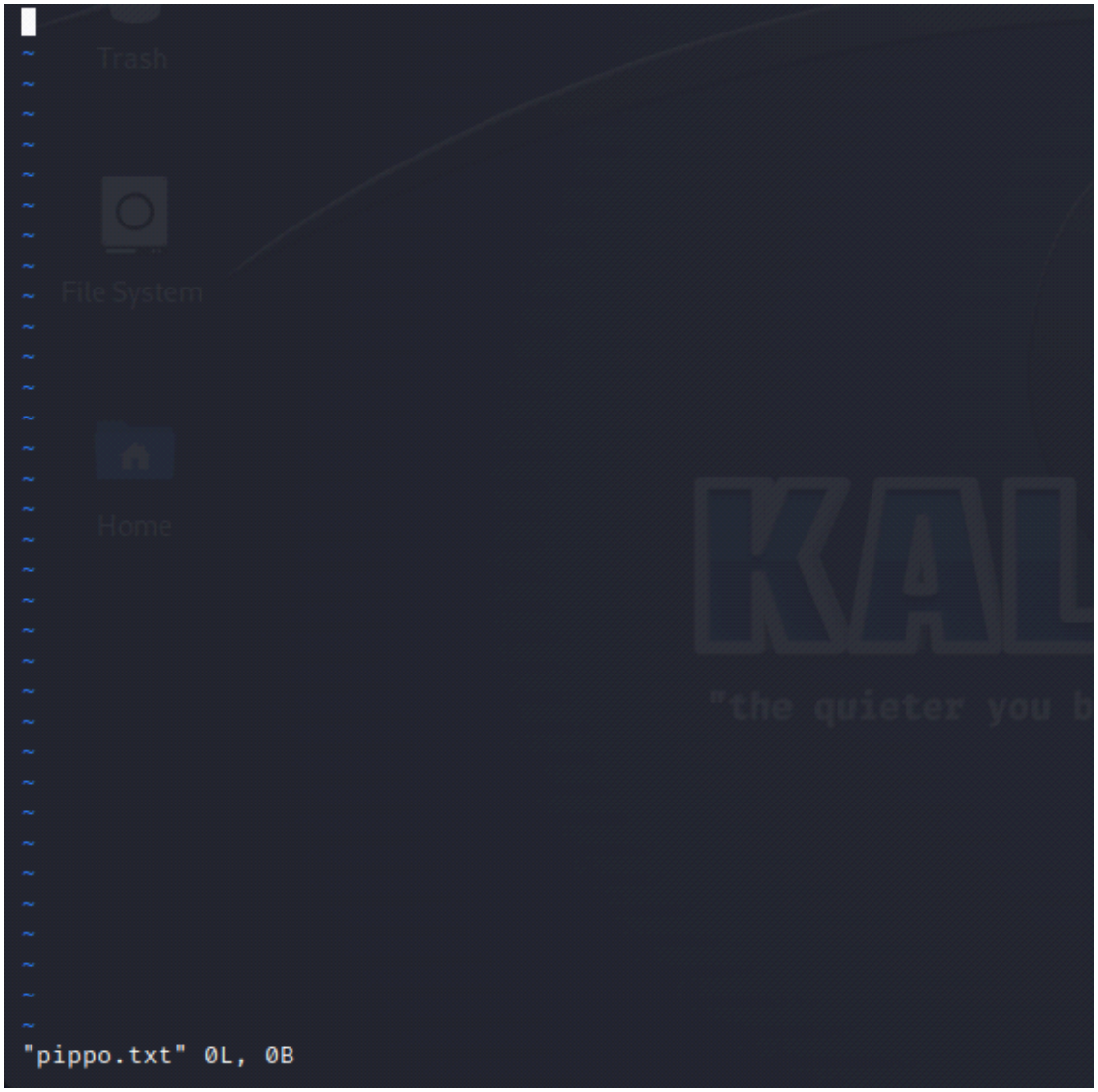
```
KILL(1)                                 User Commands                                 KILL(1)

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.  Al-
       ternate  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 col-
       umn 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  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 sigaction(2), then it can obtain this data via the si_value field of the siginfo_t
              structure.

       -l, --list [signal]
              List signal names.  This option has optional argument, which will convert signal number to signal name, or other way round.

       -L, --table
              List signal names in a nice table.

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



```
File  Actions  Edit  View  Help
              -w     Write the current history list to the history file, overwriting the history file's contents.
              -p     Perform history substitution on the following args and display the result on the standard output.  Does not store the results in the history
                     list.  Each arg must be quoted to disable normal history expansion.
              -s     Store the args in the history list as a single entry.  The last command in the history list is removed before the args are added.

              If the HISTTIMEFORMAT variable is set, the time stamp information associated with each history entry is written to the history  file,  marked  with
              the  history  comment character.  When the history file is read, lines beginning with the history comment character followed immediately by a digit
              are interpreted as timestamps for the following history entry.  The return value is 0 unless an invalid option  is  encountered,  an  error  occurs
              while reading or writing the history file, an invalid offset or range is supplied as an argument to -d, or the history expansion supplied as an ar-
              gument to -p fails.

       jobs [-lnprs] [ jobspec  ... ]
       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     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  jobspec is given, output is restricted to information about that job.  The return status is 0 unless an invalid option is encountered or an in-
              valid jobspec is supplied.

              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 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.
Manual page builtins(7) line 449 (press h for help or q to quit)
```

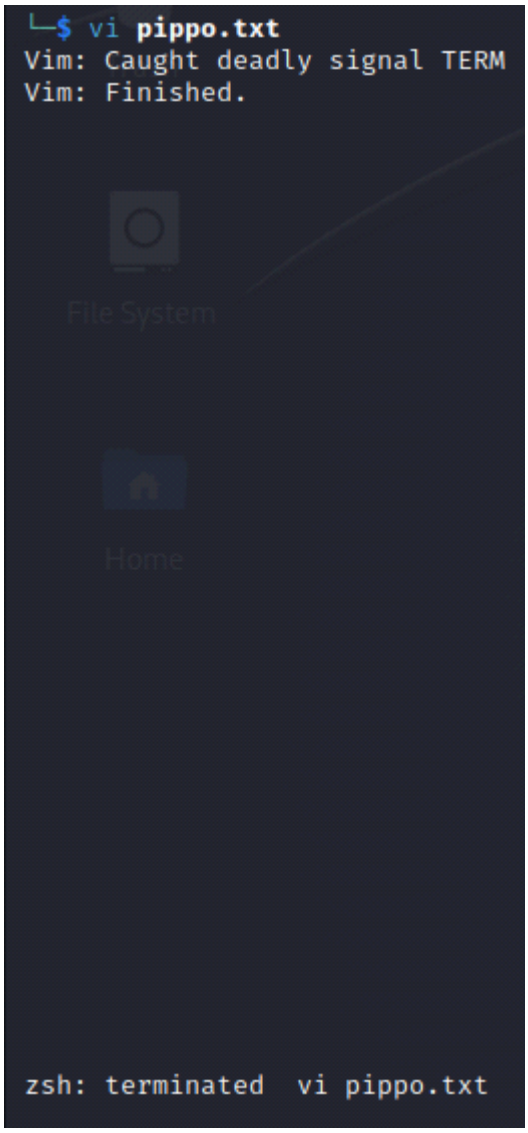
Nel caso del comando job però non riusciamo a leggerne il manuale con il normale comando *man job* ma dobbiamo utilizzare *man builtins* poichè job è un comando interno alla shell e non appartiene a linux.

Continuiamo lanciando il comando *vi pippo*

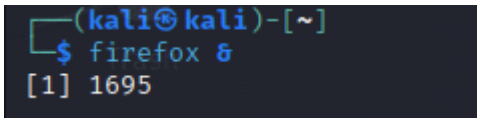


continuiamo aprendo una nuova shell, visualizziamo tutti i processi avviati e killiamo il processo *vi* per sbloccare di nuovo la shell iniziale

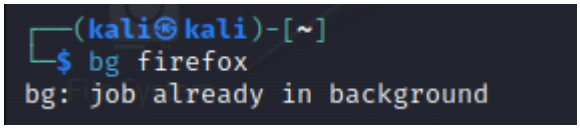
```
┌──(kali㉿kali)-[~]
└─$ ps -all
F S   UID     PID    PPID  C PRI  NI ADDR SZ WCHAN  TTY          TIME CMD
0 S  1000  165910  155280  2  80   0 -  3740 do_sel pts/0    00:00:00 vi
0 R  1000  165970  157691 99  80   0 -  2824 -      pts/1    00:00:00 ps

┌──(kali㉿kali)-[~]
└─$ kill 165910
```



Adesso lanciamo firefox in background con il comando *firefox &*



Ora proviamo a mettere firefox in background con il comando *bg firefox* e possiamo notare che ci apparirà il messaggio che firefox è già attivo in background



Continuiamo terminando il processo di firefox con il comando *kill* seguito dal PID della task di firefox

```
┌──(kali㉿kali)-[~]
└─$ kill 156659
```

```
┌──(kali㉿kali)-[~]
└─$ fg firefox
[1]  + running    firefox


Exiting due to channel error.
Exiting due to channel error.
Exiting due to channel error.
Exiting due to channel error.
Exiting due to channel error.
zsh: terminated  firefox

Exiting due to channel error.
```

Come ultimo punto della consegna ci chiedeva di verificare lo spazio utilizzato sul disco con il comando *df -h*

```
┌──(kali㉿kali)-[~]
└─$ df -h
Filesystem      Size  Used Avail Use% Mounted on
udev            698M     0  698M   0% /dev
tmpfs           148M  980K  147M   1% /run
/dev/sda1        79G   15G   61G  19% /
tmpfs           737M     0  737M   0% /dev/shm
tmpfs           5.0M     0  5.0M   0% /run/lock
tmpfs           148M  112K  148M   1% /run/user/1000
```