

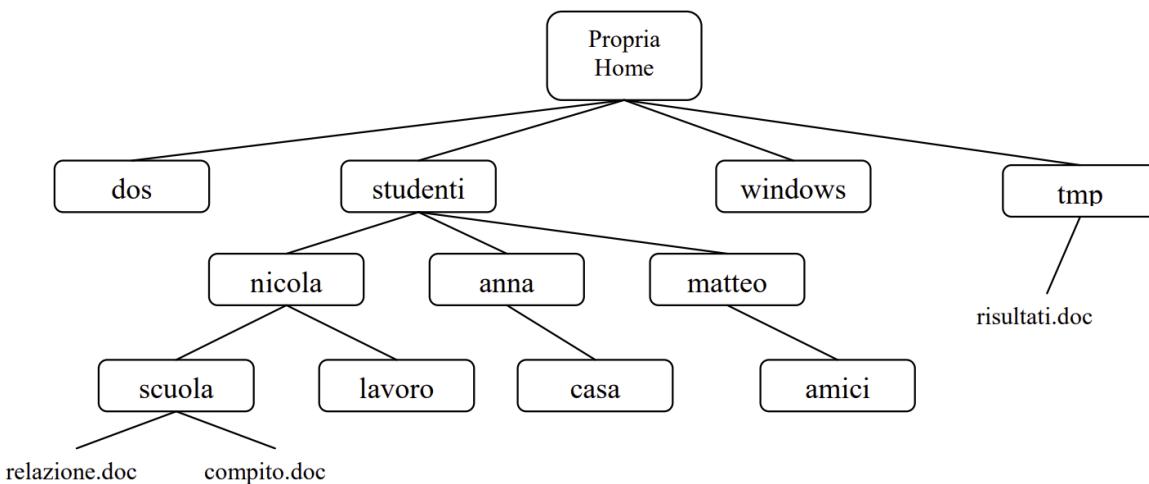
Esercitazione W5D1 - Pratica 1

Comandi Linux

Fabio Benevento - 21/11/2023

Traccia 1

Ricreare l'alberatura di file e cartelle seguente a partire dalla propria home.



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

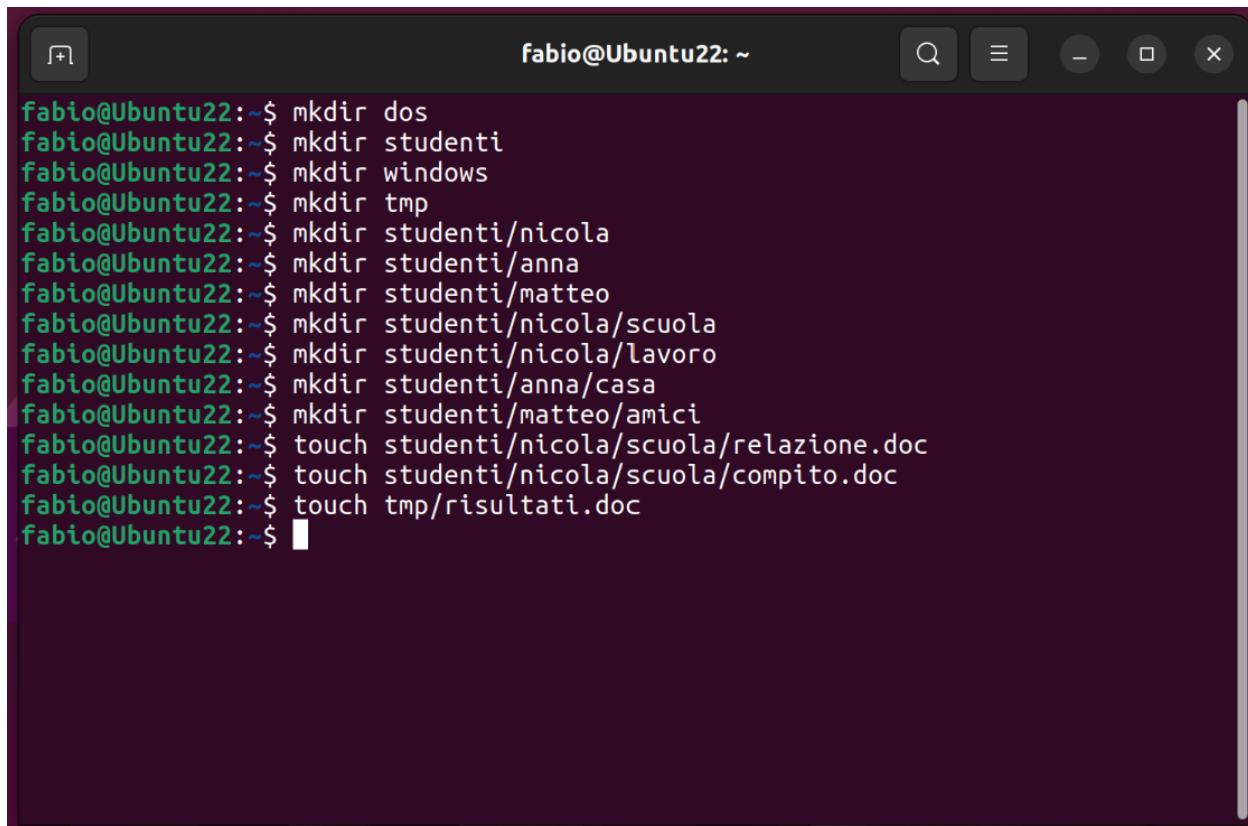
A partire dalla directory corrente (casa), eseguire i seguenti passaggi:

- 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

- h) Rimuovere la cartella amici
- i) Rimuovere tutte le cartelle precedentemente create

Implementazione

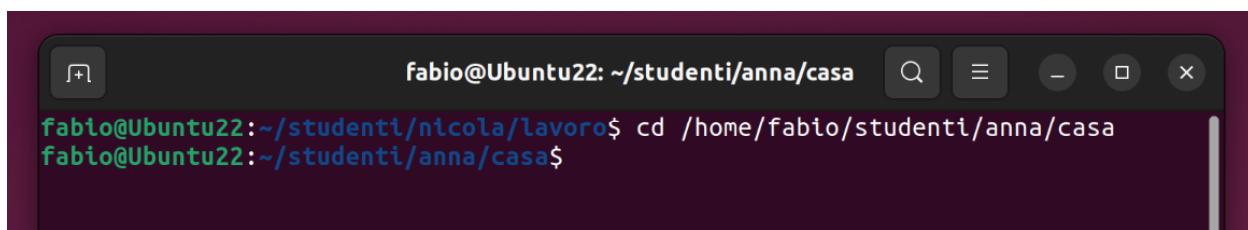
Tramite la lista dei comandi in figura ho ricreato l'alberatura richiesta, comprensiva di file e cartelle.



A screenshot of a terminal window titled "fabio@Ubuntu22: ~". The terminal displays a series of commands used to create a directory structure. The commands are:

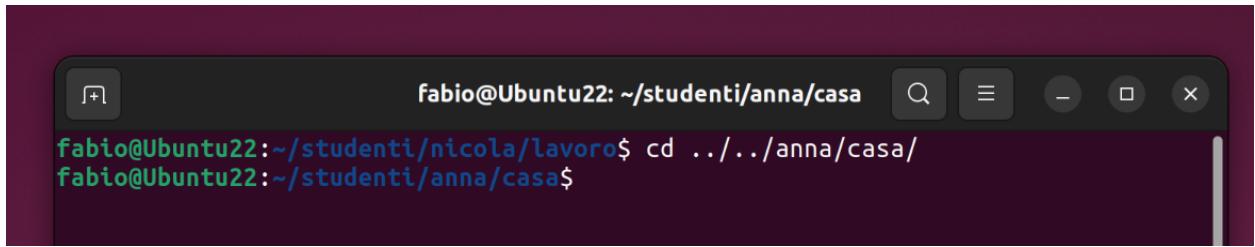
```
fabio@Ubuntu22:~$ mkdir dos
fabio@Ubuntu22:~$ mkdir studenti
fabio@Ubuntu22:~$ mkdir windows
fabio@Ubuntu22:~$ mkdir tmp
fabio@Ubuntu22:~$ mkdir studenti/nicola
fabio@Ubuntu22:~$ mkdir studenti/anna
fabio@Ubuntu22:~$ mkdir studenti/matteo
fabio@Ubuntu22:~$ mkdir studenti/nicola/scuola
fabio@Ubuntu22:~$ mkdir studenti/nicola/lavoro
fabio@Ubuntu22:~$ mkdir studenti/anna/casa
fabio@Ubuntu22:~$ mkdir studenti/matteo/amici
fabio@Ubuntu22:~$ touch studenti/nicola/scuola/relazione.doc
fabio@Ubuntu22:~$ touch studenti/nicola/scuola/compito.doc
fabio@Ubuntu22:~$ touch tmp/risultati.doc
fabio@Ubuntu22:~$
```

Di seguito sono invece illustrati i comandi per passare dalla directory nicola alla directory casa usando prima path assoluti e poi path relativi



A screenshot of a terminal window titled "fabio@Ubuntu22: ~/studenti/anna/casa". The terminal shows the user navigating to the directory "/home/fabio/studenti/anna/casa" from their current working directory. The commands are:

```
fabio@Ubuntu22:~/studenti/nicola/lavoro$ cd /home/fabio/studenti/anna/casa
fabio@Ubuntu22:~/studenti/anna/casa$
```

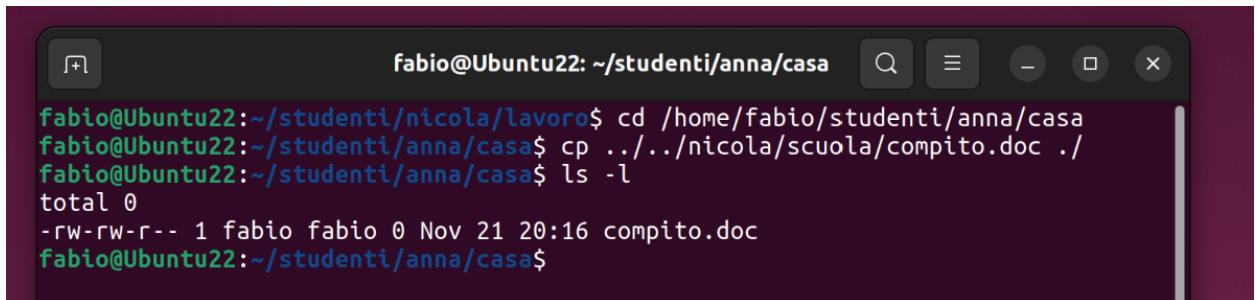


```
fabio@Ubuntu22: ~/studenti/anna/casa$ cd ../../anna/casa/
fabio@Ubuntu22:~/studenti/anna/casa$
```

A partire dalla directory corrente ho quindi svolto i passaggi richiesti come di seguito:

a) Copia il file compito.doc (dalla directory scuola) nella directory corrente (casa)

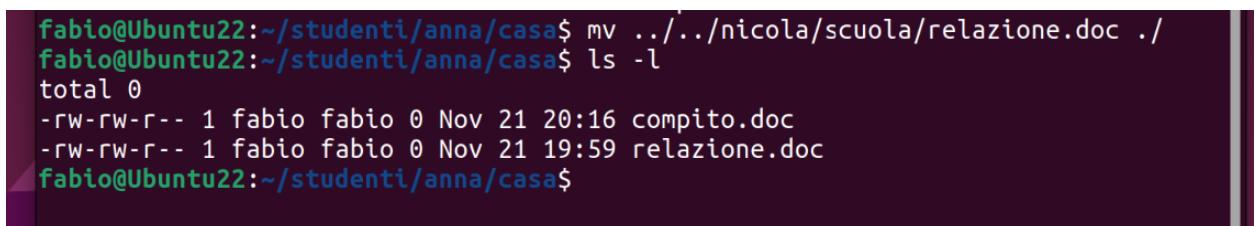
Per copiare il file compito.doc presente all'interno della directory scuola a partire dalla direcotry corrente casa, ho utilizzato il comando cp ../../nicola/scuola/compito.doc ./, facendo quindi ricorso a path relativi. Ho quindi verificato la presenza del file nella cartella corrente casa, mediante il comando ls



```
fabio@Ubuntu22:~/studenti/nicola/lavoro$ cd /home/fabio/studenti/anna/casa
fabio@Ubuntu22:~/studenti/anna/casa$ cp ../../nicola/scuola/compito.doc ./
fabio@Ubuntu22:~/studenti/anna/casa$ ls -l
total 0
-rw-rw-r-- 1 fabio fabio 0 Nov 21 20:16 compito.doc
fabio@Ubuntu22:~/studenti/anna/casa$
```

b) Sposta il file relazione.doc nella directory corrente (casa).

Per spostare il file relazione (sempre sotto scuola) all'interno della directory casa ho usato il comando mv ../../nicola/scuola/relazione.doc ./ facendo quindi ricorso a path relativi. Ho quindi verificato la presenza del file nella cartella corrente casa, mediante il comando ls



```
fabio@Ubuntu22:~/studenti/anna/casa$ mv ../../nicola/scuola/relazione.doc ./
fabio@Ubuntu22:~/studenti/anna/casa$ ls -l
total 0
-rw-rw-r-- 1 fabio fabio 0 Nov 21 20:16 compito.doc
-rw-rw-r-- 1 fabio fabio 0 Nov 21 19:59 relazione.doc
fabio@Ubuntu22:~/studenti/anna/casa$
```

c) Cancella la cartella \tmp

Per cancellare la cartella temp ho utilizzato il comando rm -r /home/fabio/tmp in maniera da cancellare la cartella e ogni eventuale file presente al suo interno (nel caso in esame, il file risultati.doc)

```
fabio@Ubuntu22:~/studenti/anna/casa$ rm -r /home/fabio/tmp/
fabio@Ubuntu22:~/studenti/anna/casa$
```

d) Creare il file pippo.txt nella cartella lavoro

Per creare il file pippo.txt nella cartella lavoro, ho utilizzato a partire dalla directory corrente casa, il comando touch ../../nicola/lavoro/pippo.txt. Ho quindi verificato la presenza del file nella cartella corrente casa, mediante il comando ls

```
fabio@Ubuntu22:~/studenti/anna/casa$ touch ../../nicola/lavoro/pippo.txt
fabio@Ubuntu22:~/studenti/anna/casa$ ls -l ../../nicola/lavoro
total 0
-rw-rw-r-- 1 fabio fabio 0 Nov 21 20:23 pippo.txt
fabio@Ubuntu22:~/studenti/anna/casa$
```

e) Cambiare gli attributi del file pippo.txt e renderlo scrivibile e leggibile solo per il proprietario, mentre per tutti gli altri solo leggibile...

Ho cambiato gli attributi al file pippo.txt, rendendolo leggibile per tutti e scrivibile solo per il proprietario facendo ricorso al comando chmod 644 ../../nicola/pippo.txt- Ho verificato l'applicazione di questa proprietà tramite il comando ls.

```
fabio@Ubuntu22:~/studenti/anna/casa$ chmod 644 ../../nicola/lavoro/pippo.txt
fabio@Ubuntu22:~/studenti/anna/casa$ ls -l ../../nicola/lavoro
total 0
-rw-r--r-- 1 fabio fabio 0 Nov 21 20:23 pippo.txt
fabio@Ubuntu22:~/studenti/anna/casa$
```

f) Nascondere il contenuto della cartella anna

Per nascondere il contenuto della cartella anna, ho rinominato quest'ultima in .anna. Questo perchè in Linux, i file che iniziano con “.” sono per l'appunto nascosti e non visibili. Infatti utilizzando il comando ls -l sotto studenti, la cartella anna non risulta più visibile.

```
fabio@Ubuntu22:~/studenti$ ls -l
total 8
drwxrwxr-x 3 fabio fabio 4096 Nov 21 19:57 matteo
drwxrwxr-x 4 fabio fabio 4096 Nov 21 19:57 nicola
fabio@Ubuntu22:~/studenti$
```

Per visualizzarla occorre utilizzare il comando ls -al, che visualizza tutti i file, anche quelli nascosti

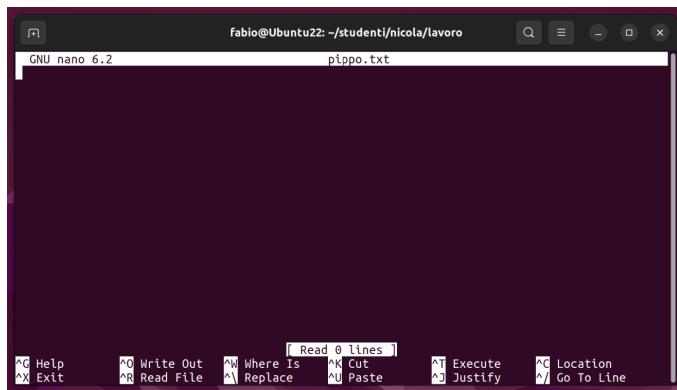
```
fabio@Ubuntu22:~/studenti$ ls -al
total 20
drwxrwxr-x 5 fabio fabio 4096 Nov 21 20:37 .
drwxr-x--- 19 fabio fabio 4096 Nov 21 20:20 ..
drwxrwxr-x 3 fabio fabio 4096 Nov 21 19:57 .anna
drwxrwxr-x 3 fabio fabio 4096 Nov 21 19:57 matteo
drwxrwxr-x 4 fabio fabio 4096 Nov 21 19:57 nicola
fabio@Ubuntu22:~/studenti$
```

g) Spostarsi nella cartella lavoro e visualizzare il contenuto del file pippo.txt

Per spostarmi all'interno della cartella lavoro ho utilizzato il comando cd nicola/lavoro, utilizzando quindi un path relativo.

```
fabio@Ubuntu22:~/studenti$ cd nicola/lavoro/
fabio@Ubuntu22:~/nicola/lavoro$ nano pippo.txt
fabio@Ubuntu22:~/nicola/lavoro$
```

Ho quindi aperto il file pippo.txt tramite il comando nano (alternativamente si sarebbero potuti usare il comando cat). Il file risulta naturalmente vuoto non essendo stato in precedenza editato



h) Rimuovere la cartella amici

Per rimuovere la cartella amici ho utilizzato il comando rm -r /home/fabio/studenti/matteo/amici, utilizzando in questo caso un path assoluto.

```
fabio@Ubuntu22:~/studenti/nicola/lavoro$ nano pippo.txt
fabio@Ubuntu22:~/studenti/nicola/lavoro$ rm -r /home/fabio/studenti/matteo/amici/
fabio@Ubuntu22:~/studenti/nicola/lavoro$ ls -l /home/fabio/studenti/matteo
total 0
fabio@Ubuntu22:~/studenti/nicola/lavoro$
```

i) Rimuovere tutte le cartelle precedentemente create

Per rimuovere tutte le cartelle precedentemente create ho usato infine il comando rm -r doc studenti windows. Distruggendo le cartelle con l'opzione -r vengo eliminati anche i file e le directory in esse contenute, andando quindi a rimuovere l'alberatura precedentemente creata.

```
fabio@Ubuntu22:~$ rm -r doc studenti windows
```

Ho quindi verificato di aver rimosso le cartelle create tramite il comando ls

```
fabio@Ubuntu22:~$ ls
Desktop Documents Downloads Music Pictures Public snap Templates Videos
```

Traccia 2

Provare i comandi:

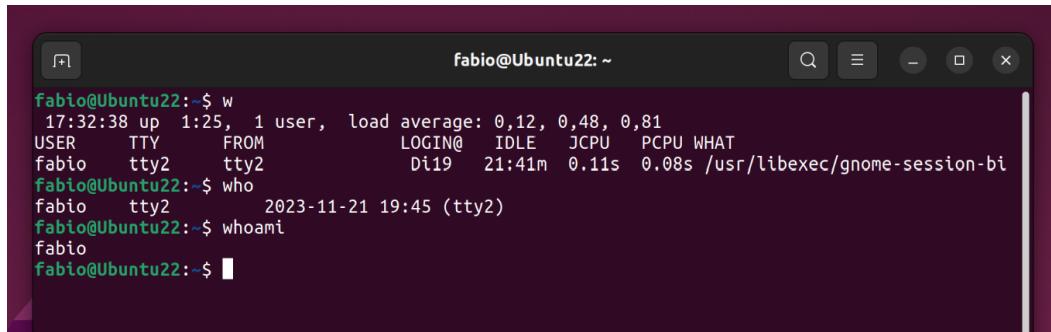
- w
- who
- who am i

Eseguire quindi i seguenti passaggi i:

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

Ho in primo luogo eseguito i comandi w, who e who am i ed analizzato l'output restituito



```
fabio@Ubuntu22: ~
fabio@Ubuntu22: $ w
17:32:38 up 1:25, 1 user,  load average: 0,12, 0,48, 0,81
USER   TTY   FROM          LOGIN@    IDLE   JCPU   PCPU WHAT
fabio   tty2   tty2        Di19   21:41m  0.11s  0.08s /usr/libexec/gnome-session-bi
fabio@Ubuntu22: $ who
fabio   tty2   2023-11-21 19:45 (tty2)
fabio@Ubuntu22: $ whoami
fabio
fabio@Ubuntu22: $
```

Tutti e tre i comandi permettono di ottenere informazioni sugli utenti attualmente connessi e sul proprio account. Nello specifico

w: Mostra informazioni dettagliate sugli utenti attualmente connessi al sistema, inclusi il tempo di attività, i processi in esecuzione e il carico di lavoro del sistema. È molto utile per ottenere una panoramica delle attività in corso sul sistema.

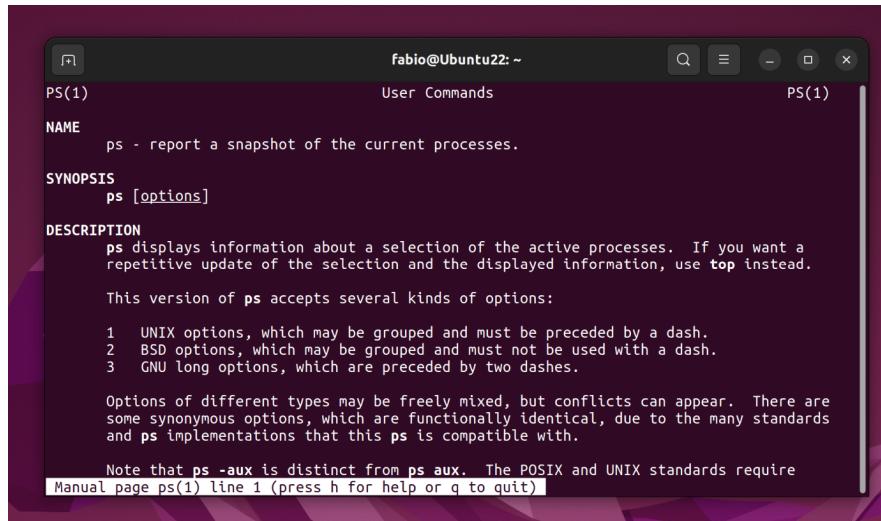
who: Fornisce un elenco degli utenti attualmente connessi al sistema, mostrando il loro username, l'orario di accesso, l'indirizzo IP e il terminale da cui sono connessi.

whoami: Restituisce semplicemente il nome dell'utente che ha eseguito il comando. È utile quando si desidera confermare rapidamente l'identità dell'utente loggato senza dover esaminare l'output dettagliato di altri comandi.

Sono passato quindi all'esecuzione dei passi richiesti.

1. Aprire un terminale e 2. leggere il manuale del comando job, ps e kill

Una volta aperto il terminale ho utilizzato il comando `man [nome comando]` per visualizzare il manuale del comando ricercato



```
PS(1) User Commands PS(1)

NAME
ps - report a snapshot of the current processes.

SYNOPSIS
ps [options]

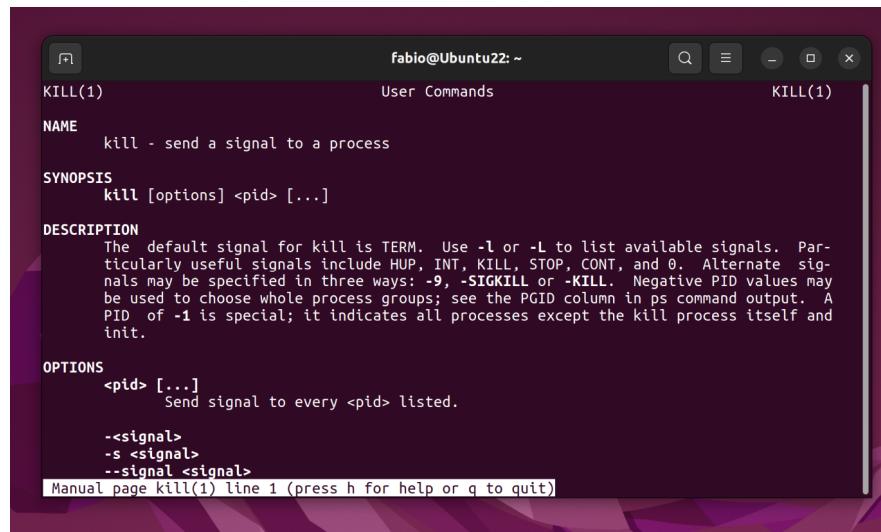
DESCRIPTION
ps displays information about a selection of the active processes. If you want a
repetitive update of the selection and the displayed information, use top instead.

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
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

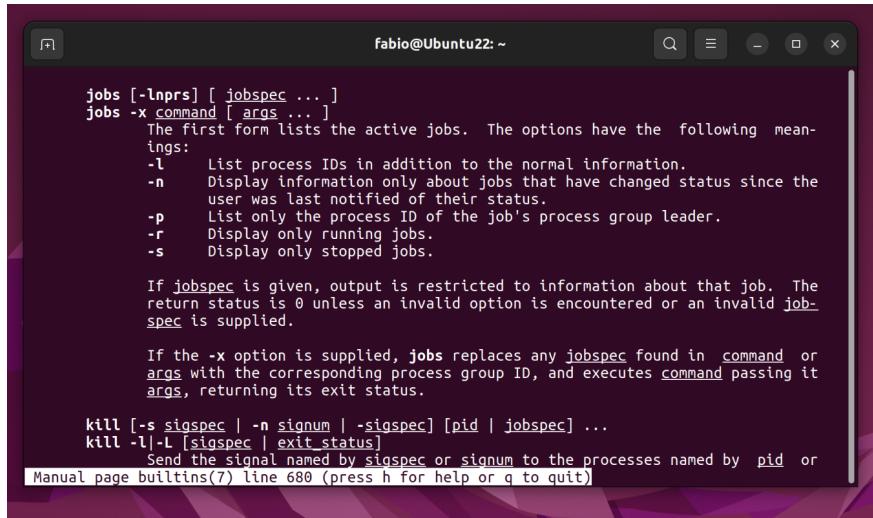
SYNOPSIS
kill [options] <pid> [...]

DESCRIPTION
The default signal for kill is TERM. Use -l or -L to list available signals. Par-
ticularly useful signals include HUP, INT, KILL, STOP, CONT, and QUIT. Alternate sig-
nals 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>
Manual page kill(1) line 1 (press h for help or q to quit)
```

Per quanto riguarda il comando jobs, questo è un comando built-in, ovvero un comando integrato nella shell, per cui non è richiamabile direttamente da terminale ma occorre avviare una sessione di shell interattiva come bash. Il suo manuale è visualizzabile scorrendo quindi il manuale dei built-ins tramite il comando man builtins



```
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 invalid 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.

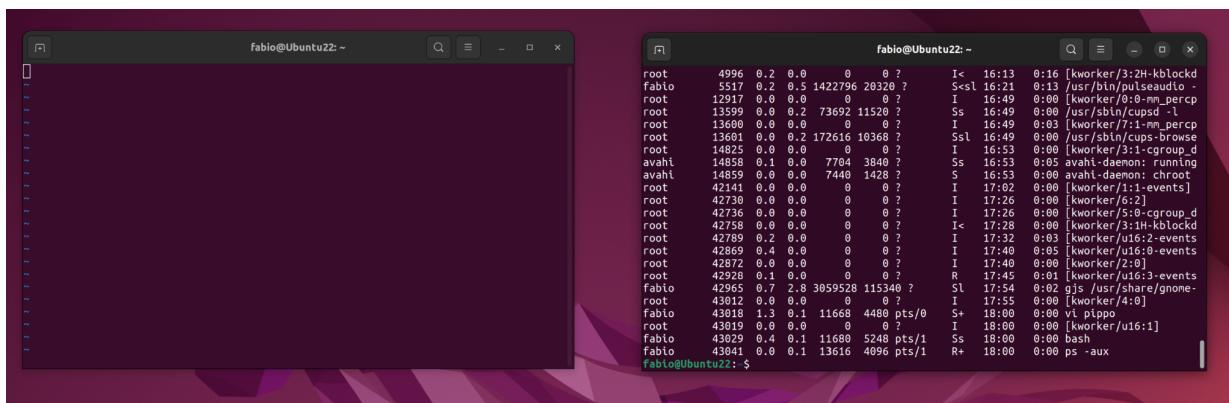
kill [-s sigspec | -n signum | -sigspec] [pid | jobspec] ...
kill -l|-L [sigspec | exit_status]
      Send the signal named by sigspec or signum to the processes named by pid or jobspec.
```

Manual page builtins(7) line 680 (press h for help or q to quit)

3. lanciare il comando vi pippo e 4. aprire un nuovo terminale e visualizzare tutti i propri processi

Una volta lanciato il comando vi pippo, ho aperto un nuovo terminale e visualizzato tutti i processi tramite il comando ps -aux.

Come è possibile notare, tra la lista è presente il processo vi pippo, avente PID 43018 e owner fabio



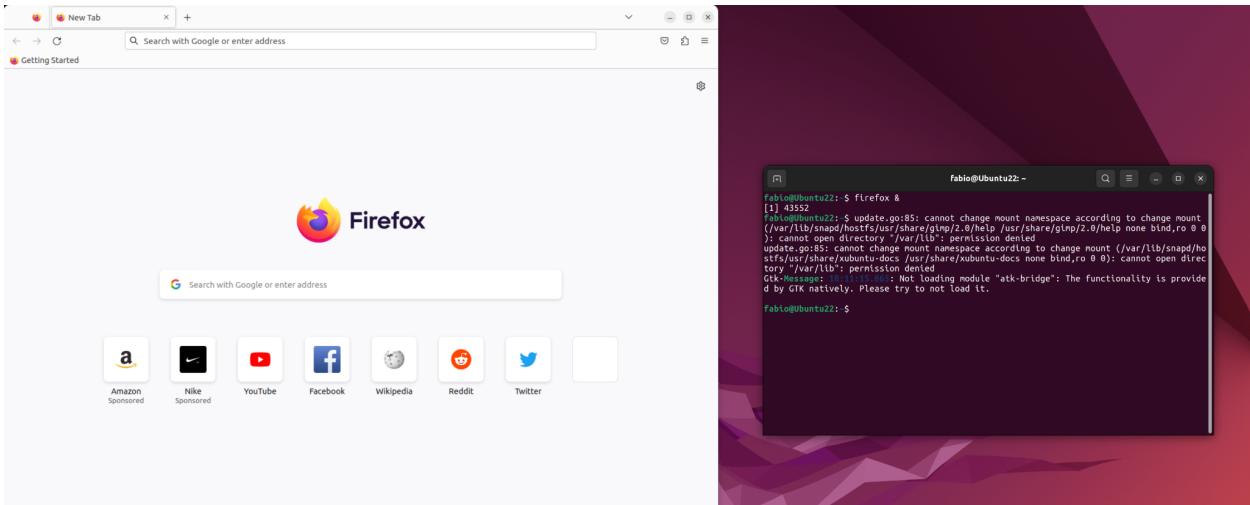
Process	User	PID	Time	TTY	%CPU	%MEM	VSZ	RSS	State	Command
root		4996	0.2	0.0	0	0	I<	16:13	0:16	[kworker/3:2-H-kblockd
fabio		5517	0.2	0.5	1422796	20320	?	16:21	0:13	/usr/bin/pulseaudio -
root		12917	0.0	0.0	0	0	I	16:49	0:00	[kworker/0:0-mm_percp
root		13599	0.0	0.2	73692	11520	?	16:49	0:00	/usr/sbin/cupsd -l
root		13600	0.0	0.0	0	0	I	16:49	0:03	[kworker/7:1-mm_percp
root		13601	0.0	0.2	172616	10368	?	16:49	0:00	/usr/sbin/cups-brows
root		14825	0.0	0.0	0	0	I	16:53	0:00	[kworker/3:1-cgroup_d
avahi		4858	0.0	0.0	7704	3849	?	16:53	0:00	/usr/lib/avahi/bin/avahi-daemon
avahi		14859	0.0	0.0	7440	1428	?	16:53	0:00	avahi-daemon: chroot
root		42141	0.0	0.0	0	0	I	17:02	0:00	[kworker/1:1-events]
root		42730	0.0	0.0	0	0	?	17:26	0:00	[kworker/6:2]
root		42736	0.0	0.0	0	0	I	17:26	0:00	[kworker/5:0-cgroup_d
root		42758	0.0	0.0	0	0	?	17:28	0:00	[kworker/3:1-H-kblockd
root		42789	0.2	0.0	0	0	I	17:32	0:03	[kworker/u16:2-events
root		42869	0.4	0.0	0	0	?	17:40	0:05	[kworker/u16:0-events
root		42872	0.0	0.0	0	0	I	17:40	0:00	[kworker/2:0]
root		42965	0.7	2.8	3059528	115340	?	17:54	0:02	gjs /usr/share/gnome-
root		43012	0.0	0.0	0	0	I	17:55	0:00	[kworker/4:0]
fabio		43018	1.3	0.1	11668	4480	pts/0	18:00	0:00	vi pippo
root		43019	0.0	0.0	0	0	I	18:00	0:00	[kworker/u16:1]
fabio		43029	0.4	0.1	11680	5248	pts/1	18:00	0:00	bash
fabio		43041	0.0	0.1	13616	4096	pts/1	R+	18:00	0:00 ps -aux

5. cercare di terminare (killare) il processo vi per sbloccare il terminale precedente

Avendo individuato il PID del processo vi, è possibile ucciderlo mediante il comando kill -9 43018, dove 43018 è il PID del processo individuato in precedenza

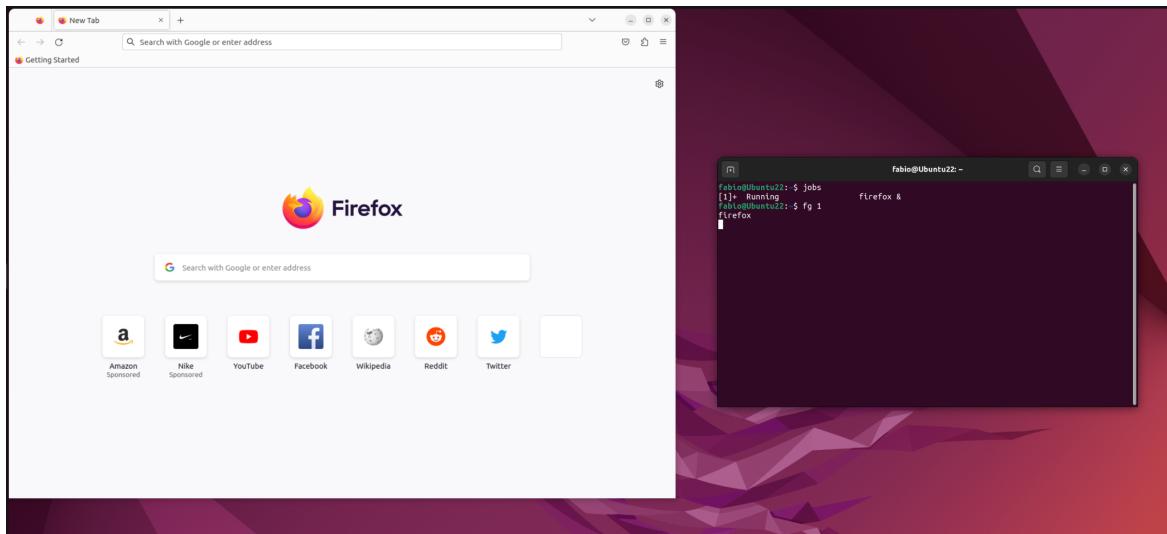
6. lanciare il comando firefox in background

Per lanciare il programma firefox in background è possibile eseguire il comando `firefox &`, dove & sta ad indicare proprio di far partire l'applicazione in background, rendendo subito disponibile il terminale al chiamante come visualizzato in figura



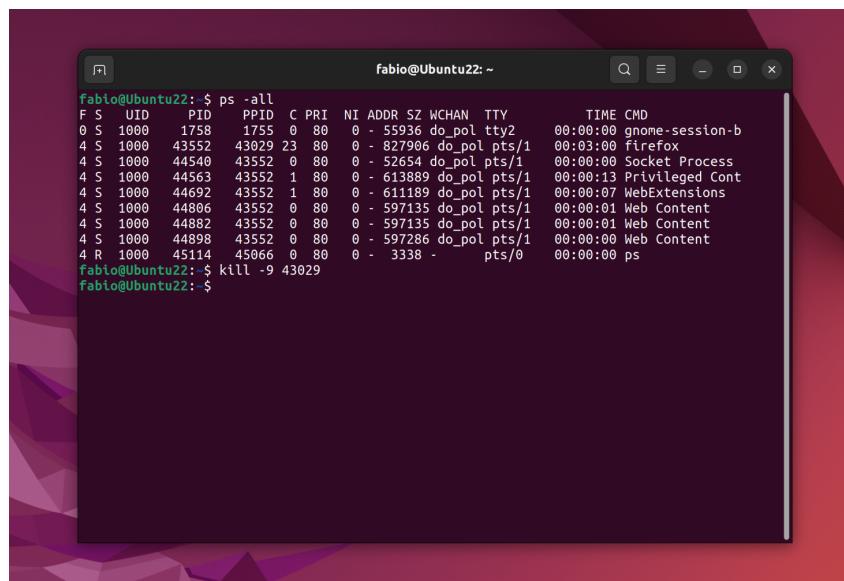
7. portarlo in foreground

Per portare l'applicazione firefox in foreground ho utilizzato il comando fg come evidenziato in figura



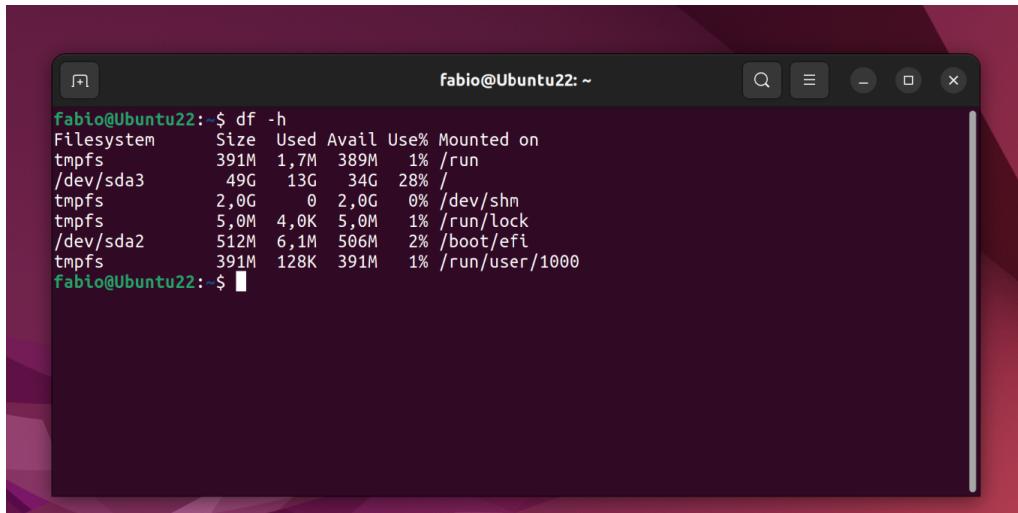
8. cercare di terminare il processo firefox

Per terminare il processo firefox, ho proceduto come in precedenza, ovvero ho individuato il pid del processo mediante il comando ps -all e l'ho ucciso tramite il comando kill -9 [PID]



9. verificare quanto spazio si sta occupando su disco

Per verificare lo spazio su disco è possibile utilizzare il comando df -h che restituisce lo spazio di ciascuna partizione su disco



```
fabio@Ubuntu22: ~$ df -h
Filesystem      Size  Used Avail Use% Mounted on
tmpfs          391M  1,7M  389M  1% /run
/dev/sda3        49G   13G   34G  28% /
tmpfs          2,0G     0  2,0G  0% /dev/shm
tmpfs          5,0M  4,0K  5,0M  1% /run/lock
/dev/sda2       512M  6,1M  506M  2% /boot/efi
tmpfs          391M  128K  391M  1% /run/user/1000
fabio@Ubuntu22: ~$
```