

ESERCITAZIONE WEEK 5 DAY 2

ESERCIZIO 2

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

Comandi **w**, **who**, **who am i**:

```
lisa@lisa-VirtualBox: ~  
lisa@lisa-VirtualBox:~/Scrivania$ cd ..  
lisa@lisa-VirtualBox:~$ w  
20:31:46 up 1:43, 1 user, load average: 0,12, 0,10, 0,09  
UTENTE TTY FROM LOGIN@ IDLE JCPU PCPU WHAT  
lisa tty2 tty2 18:52 1:43m 0.05s 0.05s /usr/libexec/gn  
lisa@lisa-VirtualBox:~$ who  
lisa tty2 2023-11-21 18:52 (tty2)  
lisa@lisa-VirtualBox:~$ who am i  
lisa@lisa-VirtualBox:~$ sudo who am i  
[sudo] password di lisa:  
lisa pts/1 2023-11-21 20:32  
lisa@lisa-VirtualBox:~$
```

Tutti e 3 i comandi danno informazioni sugli utenti connessi al sistema ma lo fanno in modo diverso:

- **w**: oltre ad indicare chi sia connesso al sistema, da informazioni anche dettagliate sull'orario e il tempo trascorso dall'inizio della connessione e cosa l'utente fa, ad esempio l'orario di login. Tty2 è il terminale di connessione;
- **who**: mostra meno dettagli del comando precedente, focalizzandosi su utente, terminale, data e ora di login. Può eventualmente anche specificare un indirizzo IP;
- **who am i**: mentre gli altri comando danno informazioni sugli utenti connessi, questo comando dà informazioni sull'utente che richiede tale comando;

1 e 2) Aprendo un terminale, sono stati letti i manuali di **ps**, **kill** e **jobs**:

```

lisa@lisa-VirtualBox: ~/Scrivania
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.

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

```

```

lisa@lisa-VirtualBox: ~/Scrivania
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.  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 special; it indicates all processes except the kill process itself
    and init.

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

    -<signal>
    -s <signal>

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

```

```
lisa@lisa-VirtualBox: ~/Scrivanian$ cd ..
lisa@lisa-VirtualBox:~$ man jobs -l
Non c'è il manuale per jobs
Non c'è il manuale per -l
lisa@lisa-VirtualBox:~$ help jobs
jobs: jobs [-lnprs] [specjob ...] oppure jobs -x comando [argomenti]
Visualizza lo stato dei lavori.

Elenca i lavori attivi. JOBSPEC limita l'output a quel lavoro.
Senza opzioni, viene visualizzato lo stato di tutti i lavori attivi.

Opzioni:
  -l      elenca gli ID di processo oltre alle normali informazioni
  -n      elenca solo i processi che hanno cambiato stato dall'ultimo
[scheda] notifica
  -p      elenca solo gli ID di processo
  -r      limita l'output ai lavori in esecuzione
  -s      limita l'output ai lavori interrotti

Se viene fornito -x, COMANDO viene eseguito dopo tutte le specifiche del lavoro che
appaiono in ARGS sono stati sostituiti con l'ID processo di quel lavoro
capogruppo di processo.
```

Mentre per **ps** e **kill** che sono comandi standard di Linux, il manuale associato esiste e quindi si può usare il comando **man**, per quando riguarda **jobs** in questa distribuzione di Linux, essa è una funzione integrata della shell, invece di essere un programma esterno con un manuale dedicato. Per avere informazioni si usa allora il comando **help**.

3 e 4) Lanciare il comando **vi pippo** e visualizzare l'elenco dei processi con il comando **ps -ef**:

[illegible]

5) Killare il processo:

```
lisa@lisa-VirtualBox: ~  
lisa@lisa-VirtualBox:~$ vi pippo  
vim: Caught deadly signal TERM  
vim: Finished.  
  
Terminato  
lisa@lisa-VirtualBox:~$ ps -ef  
root      3654      2   0 20:02 ?        00:00:00 [kworker/2:2-events]  
root      3815      2   0 20:09 ?        00:00:00 [kworker/3:1-cgroup_destroy]  
root      3991      2   0 20:17 ?        00:00:00 [kworker/u8:1-events_unbound]  
root      4121      2   0 20:25 ?        00:00:00 [kworker/1:1-cgroup_destroy]  
root      4151      2   0 20:26 ?        00:00:00 [kworker/2:0-cgroup_destroy]  
root      4191      2   0 20:27 ?        00:00:00 [kworker/0:2-cgroup_destroy]  
root      4362      2   0 20:33 ?        00:00:00 [kworker/u8:0-events_power_e  
root      4364      2   0 20:33 ?        00:00:00 [kworker/3:0]  
lisa@lisa-Virt root      4446      2   0 20:34 ?        00:00:00 [kworker/1:0-mm_percpu_wq]  
root      4523      2   0 20:34 ?        00:00:00 [kworker/2:1-cgroup_destroy]  
root      4542      2   0 20:35 ?        00:00:00 [kworker/0:1-mm_percpu_wq]  
lisa      4547    2812 0 20:35 ?        00:00:00 /usr/bin/python3 /usr/bin/gn  
lisa      4548    4547 0 20:35 ?        00:00:00 /usr/bin/gnome-terminal.real  
lisa      4553   1324 1 20:35 ?        00:00:00 /usr/libexec/gnome-terminal-  
lisa      4571    4553 0 20:35 pts/0    00:00:00 bash  
lisa      4577    4571 0 20:35 pts/0    00:00:00 vi pippo  
lisa      4579    2812 0 20:36 ?        00:00:00 /usr/bin/python3 /usr/bin/gn  
lisa      4580    4579 0 20:36 ?        00:00:00 /usr/bin/gnome-terminal.real  
lisa      4585    4553 0 20:36 pts/1    00:00:00 bash  
lisa      4593    4585 0 20:36 pts/1    00:00:00 ps -ef  
lisa@lisa-VirtualBox:~$ kill vi  
bash: kill: vi: gli argomenti devono essere ID di processo o di job  
lisa@lisa-VirtualBox:~$ kill 4577  
lisa@lisa-VirtualBox:~$
```

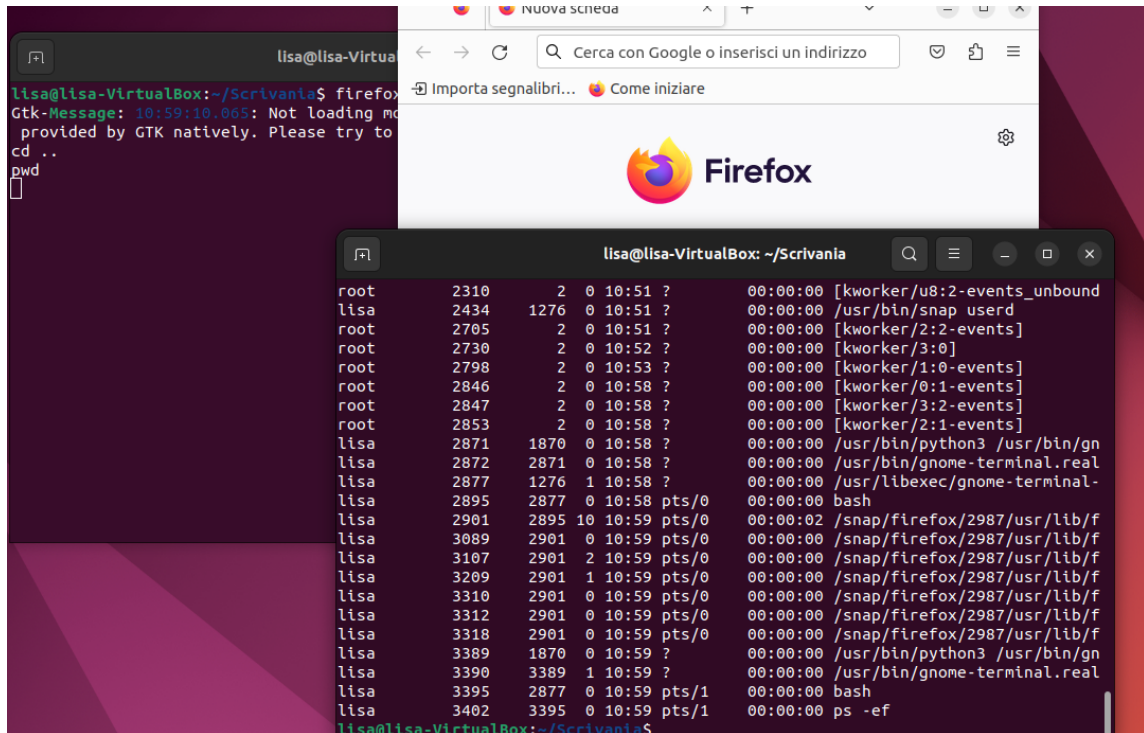
Per terminare il processo si manda il comando **kill** seguito dall'ID del processo visibile nella seconda colonna. In tal modo il terminale viene sbloccato.

6) Lanciare il comando **firefox** in background. Di fatto significa che il programma viene eseguito indipendentemente dalla shell che lo ha avviato. Per farlo si invia il comando **firefox &**.

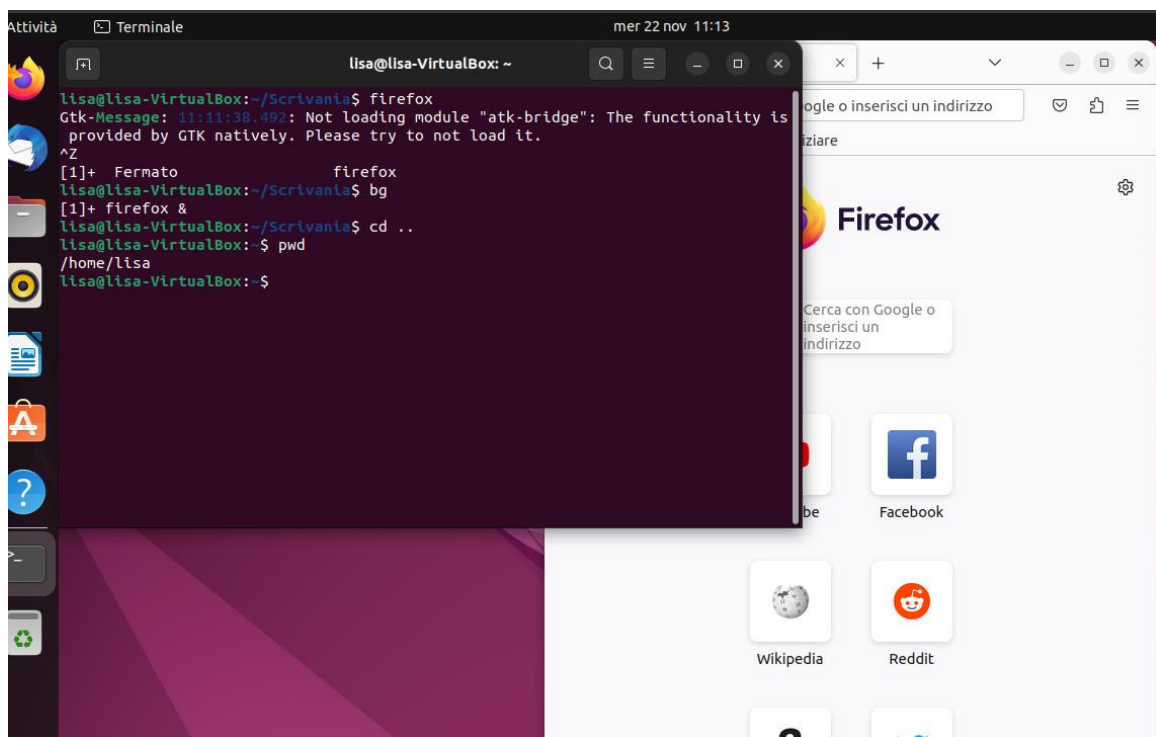
Si può notare infatti che anche dopo l'avvio del programma, la shell è utilizzabile dall'utente per mandare altri comandi.

```
lisa@lisa-VirtualBox: ~  
[1] 2139  
lisa@lisa-VirtualBox:~/Scrivania$ update.go:85: cannot chan  
cording to change mount (/var/lib/snapd/hostfs/usr/share/gi  
e/gimp/2.0/help none bind,ro 0 0): cannot open directory "/  
denied  
update.go:85: cannot change mount namespace according to ch  
snapd/hostfs/usr/share/xubuntu-docs /usr/share/xubuntu-docs  
cannot open directory "/var/lib": permission denied  
Gtk-Message: 10:51:08.149: Not loading module "atk-bridge":  
provided by GTK natively. Please try to not load it.  
cd ..  
lisa@lisa-VirtualBox:~$ pwd  
/home/lisa  
lisa@lisa-VirtualBox:~$ touch x.txt  
lisa@lisa-VirtualB  
totale 36  
drwxr-xr-x 2 lisa lisa      2108    1870   0 10:50 ?        00:00:00 /usr/bin/python3 /usr/bin/gn  
drwxr-xr-x 2 lisa lisa      2109    2108   0 10:50 ?        00:00:00 /usr/bin/gnome-terminal.real  
drwxr-xr-x 2 lisa lisa      2114    1276   0 10:50 ?        00:00:00 /usr/libexec/gnome-terminal-  
drwxr-xr-x 2 lisa lisa      2132    2114   0 10:50 pts/0    00:00:00 bash  
drwxr-xr-x 2 lisa lisa      2139    2132   1 10:51 pts/0    00:00:04 /snap/firefox/2987/usr/lib/f  
drwxr-xr-x 2 lisa lisa      2260    1442   0 10:51 ?        00:00:00 /usr/bin/Xwayland :0 -rootle  
drwxr-xr-x 2 lisa lisa      2263    1276   0 10:51 ?        00:00:00 /usr/libexec/gsd-xsettings  
drwx----- 4 lisa lisa      2289    1276   0 10:51 ?        00:00:00 /usr/libexec/ibus-x11  
root      2310      2   0 10:51 ?        00:00:00 [kworker/u8:2-events_unbound]  
lisa      2375    2139 0 10:51 pts/0    00:00:00 /snap/firefox/2987/usr/lib/f  
lisa      2397    2139 0 10:51 pts/0    00:00:00 /snap/firefox/2987/usr/lib/f  
lisa      2434    1276   0 10:51 ?        00:00:00 /usr/bin/snap userd  
lisa      2514    2139 0 10:51 pts/0    00:00:00 /snap/firefox/2987/usr/lib/f  
lisa      2616    2139 0 10:51 pts/0    00:00:00 /snap/firefox/2987/usr/lib/f  
lisa      2620    2139 0 10:51 pts/0    00:00:00 /snap/firefox/2987/usr/lib/f  
lisa      2624    2139 0 10:51 pts/0    00:00:00 /snap/firefox/2987/usr/lib/f  
root      2705      2   0 10:51 ?        00:00:00 [kworker/2:2-events]  
root      2730      2   0 10:52 ?        00:00:00 [kworker/3:0]  
root      2798      2   0 10:53 ?        00:00:00 [kworker/1:0-events]  
lisa      2813    1870   1 10:54 ?        00:00:00 /usr/bin/python3 /usr/bin/gn  
lisa      2814    2813   1 10:54 ?        00:00:00 /usr/bin/gnome-terminal.real  
lisa      2820    2114   0 10:54 pts/1    00:00:00 bash  
lisa      2827    2820   0 10:54 pts/1    00:00:00 ps -ef
```

7) Chiudere il processo precedente. Lanciare **firefox** in foreground e poi portarlo in background. In foreground significa che è attualmente in esecuzione e sta interagendo direttamente con l'utente. La shell risulta bloccata fino a che non si termina il processo (CTRL +C). Si invia dunque il comando **firefox**:



Per riportarlo in background, si preme CTRL + Z per mettere in pausa il processo e si invia il comando **bg**. La shell risulta sbloccata.



8) Per terminare il processo in foreground occorre premere CTRL+C. Per terminare invece un processo in background si può usare il comando **kill** seguito dal PID.

9) Verificare lo spazio occupato su disco:

- comando **df -h**:

```
lisa@lisa-VirtualBox: ~/Scrivania
lisa@lisa-VirtualBox:~/Scrivania$ df -h
File system      Dim. Usati Dispon. Uso% Montato su
tmpfs            794M  1,5M   792M   1% /run
/dev/sda3        49G   11G   36G   23% /
tmpfs            3,9G    0   3,9G   0% /dev/shm
tmpfs            5,0M  4,0K   5,0M   1% /run/lock
/dev/sda2        512M  6,1M   506M   2% /boot/efi
tmpfs            794M  96K   794M   1% /run/user/1000
lisa@lisa-VirtualBox:~/Scrivania$
```

- comando **du -h**:

```
lisa@lisa-VirtualBox:~/Scrivania$ du -h
4,0K .
lisa@lisa-VirtualBox:~/Scrivania$
```

In breve, **df -h** mostra informazioni sullo spazio su disco disponibile e utilizzato per ciascun filesystem, mentre **du -h** fornisce informazioni sulla dimensione dell'utilizzo del disco per file e directory specifici nel sistema. Entrambi i comandi sono utili in contesti diversi, a seconda delle informazioni richieste.

Altro esempio nella directory `/home/lisa`:

```
lisa@lisa-VirtualBox:~$ df -h
File system      Dim. Usati Dispon. Uso% Montato su
tmpfs            794M  1,5M   792M   1% /run
/dev/sda3        49G   11G   36G   23% /
tmpfs            3,9G    0   3,9G   0% /dev/shm
tmpfs            5,0M  4,0K   5,0M   1% /run/lock
/dev/sda2        512M  6,1M   506M   2% /boot/efi
tmpfs            794M  96K   794M   1% /run/user/1000
lisa@lisa-VirtualBox:~$ du -h
4,0K ./Modelli
28K  ./snap/snapd-desktop-integration/common/.cache/gio-modules
72K  ./snap/snapd-desktop-integration/common/.cache/immodules
108K ./snap/snapd-desktop-integration/common/.cache
112K ./snap/snapd-desktop-integration/common
4,0K ./snap/snapd-desktop-integration/83/.local/share/glib-2.0/schemas
8,0K ./snap/snapd-desktop-integration/83/.local/share/glib-2.0
4,0K ./snap/snapd-desktop-integration/83/.local/share/icons
16K  ./snap/snapd-desktop-integration/83/.local/share
20K  ./snap/snapd-desktop-integration/83/.local
4,0K ./snap/snapd-desktop-integration/83/.config/gtk-3.0
4,0K ./snap/snapd-desktop-integration/83/.config/dconf
```