

Shell di Linux

Parte 2

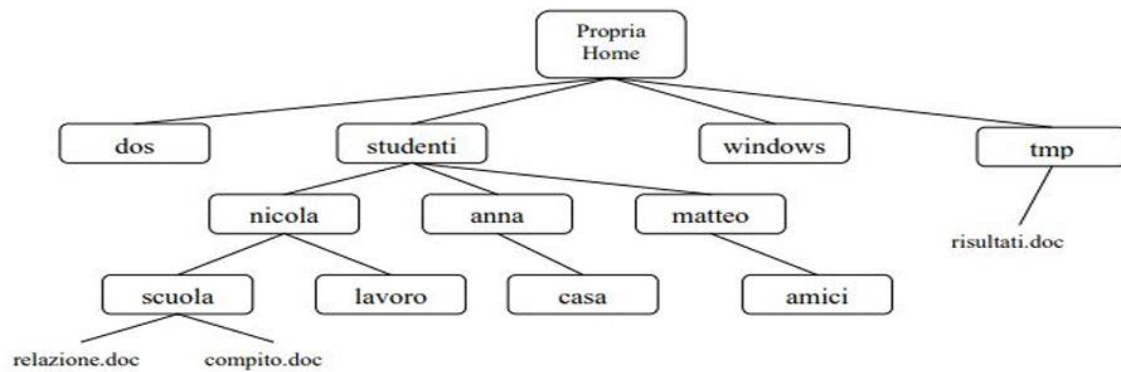


Yilei Wu

24 Luglio 2024

Sommario

Consegna Esercizio 1	3
Legenda comandi terminale sulla gestione file & cartelle	4
Legenda del terminale sulla gestione dei processi.....	5
Cartelle	6
Gerarchia e ramificazione delle cartelle	6
Creazione delle cartelle	6
Esercizi sulla gestione directory e file	7
j) Cambiare gli attributi della directory lavoro facendo in modo che il suo contenuto non sia leggibile ma consentendo solo a te di entrarvi	7
k) Entra nella cartella lavoro e crea un file (emacs – ricordati di salvarlo).....	7
l) Prova a visualizzare il contenuto della directory dove ti trovi dopo di che modifica gli attributi della directory '.' Facendo in modo che siano nuovamente leggibile il contenuto e prova nuovamente a leggere il contenuto	8
m) Spostati nella cartella scuola usando il percorso relativo (a dove ti trovi)	8
n) Nella directory scuola crea una directory .mia (punto mia) e fai in modo che sia leggibile ed eseguibile solo da te	9
o) Prova a eseguire il comando ls e successivamente ls -al	9
Esercizi sulla gestione processi	10
10. lascia il comando emacs &.....	10
11. esegui il comando jobs	10
12. lancia il comando firefox e successivamente sul terminale premi CTRL+Z	10
13. manda il processo firefox in background (bg) e lancia il comando jobs.....	10
15. manda in foreground (fg) il programma emacs	11
16. termina emacs con i tasti CTRL+C	11
17. verificare quanto spazio si sta occupando sul disco	11



Ti trovi nella directory **nicola** (sotto studenti),

- j) Cambiare gli attributi della directory lavoro facendo in modo che il suo contenuto non sia leggibile ma consentendo solo a te di entrarvi
- k) Entra nella cartella lavoro e crea un file (emacs – ricordati di salvarlo)
- l) Prova a visualizzare il contenuto della directory dove ti trovi dopo di che modifica gli attributi della directory '.' Facendo in modo che siano nuovamente leggibile il contenuto e prova nuovamente a leggere il contenuto
- m) Spostati nella cartella scuola usando il percorso relativo (a dove ti trovi)
- n) Nella directory scuola crea una directory .mia (punto mia) e fai in modo che sia leggibile scrivibile ed eseguibile solo da te
- o) Prova a eseguire il comando ls e successivamente ls -al

Esercizi:

10. lancia il comando emacs &
11. esegui il comando jobs
12. lancia il comando firefox e successivamente sul terminale premi ^z
13. manda il processo firefox in background (bg)
14. lancia il comando jobs
15. manda in foreground (fg) il programma emacs
16. termina emacs con il tasti ^c
17. verificare quanto spazio si sta occupando su disco

Legenda comandi terminale sulla gestione file & cartelle

Lista comandi utili per l'esercizio:

- ✓ **sudo + comando** il comando viene avviato con privilegi di superuser
- ✓ **cd** cambia directory (cd + percorso di destinazione)
 1. **cd-** riporta alla directory da cui sei arrivato immediatamente prima;
 2. **cd ..** porta alla directory padre di quella corrente;
 3. **cd ../..** porta a due livelli più in alto nella struttura delle directory.
- 2. **ls** visualizza il contenuto della cartella
 1. **ls -a** elenca tutto i file e cartelle, incluse quelle nascoste;
 2. **ls -l nome_file** verifica gli attuali permessi del file.
 3. **ls -al** combina le due funzioni sopraelencate quindi anche file nascosti con gli attuali permessi
- 3. **pwd** stampa il percorso assoluto di dove mi trovo
- 4. **chmod [opzioni] permessi file**
 1. **Simbolico:**

r (read) Lettura, **w** (write) Scrittura, **x** (execute) Esecuzione;
u (user) Proprietario, **g** (group) Gruppo, **o** (others) Altri utenti.
 2. **Esempi:**
 - Aggiungere permesso di esecuzione per il proprietario: **chmod u+x file.txt**
 - Rimuovere il permesso di scrittura per il gruppo: **chmod g-w file.txt**
 - Impostare i permessi di lettura e scrittura per il proprietario e solo lettura per gli altri: **chmod u=rw,go=r file.txt**
- ✓ **mkdir** creare una cartella
- ✓ **rmdir** rimuovere una cartella vuota
- ✓ **rm** remove
- ✓ **mv** sposta
- ✓ **cp** copia
- ✓ **man argomento** visualizza manuale di un comando
- ✓ editor di testo
 1. **nano: nano nomefile.txt**
Per salvare e uscire **CTRL+O** + **invio** e **CTRL+X**
 2. **vi o vim: vi nomefile.txt o vim nomefile.txt**
Per salvare e uscire, premi **CTRL+C**, poi digita **:wq** e premi Invio. Questo comando scrive le modifiche (**w**) e chiude l'editor (**q**).
Per uscire senza salvare, digita **:q!** e premi Invio.
- ✓ **cat nome_file.txt** per leggere l'intero contenuto nel terminale
- ✓ **less nome_file.txt** per scorrere il file avanti e indietro, per uscire premere **q**.

Nascondere i file singolarmente

Rinominando ogni file e cartella all'interno di una directory per aggiungere un punto (.) all'inizio del nome.

Esempio:

```
mv file1.txt .file1.txt
```

```
mv file2.txt .file2.txt
```

```
mv sottocartella .sottocartella
```

Legenda del terminale sulla gestione dei processi

Lista comandi utili per l'esercizio:

- ✓ **sudo su + comando** il comando viene avviato con privilegi di superuser assoluto
- ✓ **who** lista utenti collegati
- ✓ **who am i** chi sono io
- ✓ **jobs** elenco lavori sul terminale
- ✓ **&** apre processo in background [**nome_processo &**]
- ✓ **fg nome_processo** metti in foreground
- ✓ **bg nome_processo** metti in background
- ✓ **ps** elenco processi

ps aux questo comando mostra tutti i processi in esecuzione sul sistema con dettagli come l'utente, l'ID del processo (PID), la percentuale di utilizzo della CPU e della memoria, e il comando che ha avviato il processo.
- ✓ **top** è un comando interattivo che fornisce una visualizzazione in tempo reale dei processi in esecuzione, inclusi l'utilizzo della CPU e della memoria
- ✓ **kill + PID** termina processo

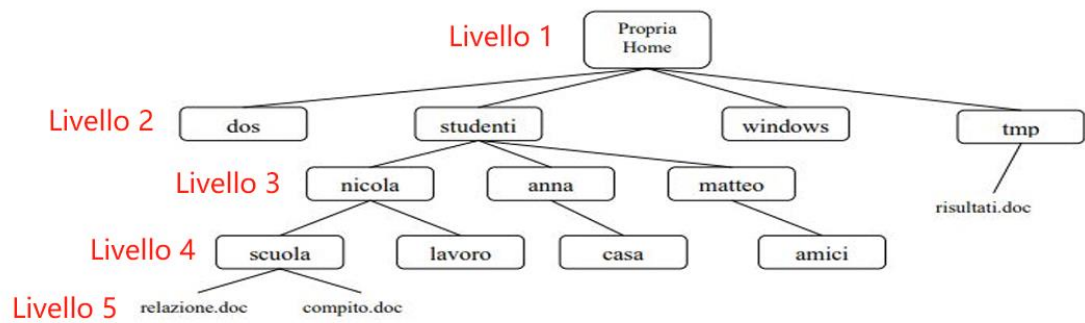
opzione **kill -9 + PID** indica il segnale "SIGKILL" per forzare l'interruzione immediata senza possibilità di pulizia o salvare lo stato
- ✓ **killall nome_processo** serve per killare tutti i processi con quel nome
- ✓ **egrep** extended grep serve per filtrare più file

esempio **egrep 'error|warning' server.log**: Cerca tutte le righe nel file server.log che contengono "error" o "warning". Il simbolo | è utilizzato come operatore OR nelle espressioni regolari estese.
- ✓ **pgrep + nome processo** Trova gli ID dei processi che corrispondono a "nome_processo".
- ✓ **df**: Mostra l'utilizzo del disco per tutti i file system montati, con i valori in blocchi (in base a 1K).
 - a. **-h**: Mostra l'uso del disco in un formato leggibile dall'uomo (ad es. GB, MB).
 - b. **-T**: Mostra anche il tipo di file system.
- 2. **du** Mostra l'uso del disco per la directory corrente e le sue sottodirectory in blocchi
 - a. **-h**: Mostra l'uso del disco in un formato leggibile dall'uomo.

Cartelle

Gerarchia e ramificazione delle cartelle

Dividiamo la gerarchia delle cartelle e file per una migliore comprensione della ramificazione.



I livelli sono

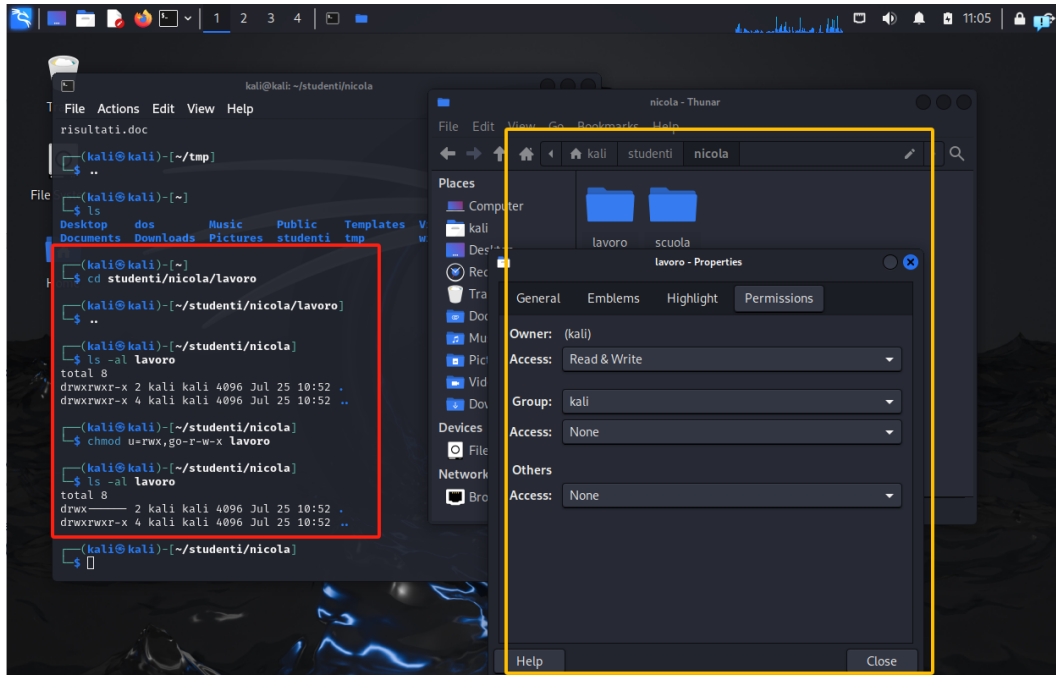
1. Home
2. Dos – Studenti – Windows – tmp
3. nicola – anna – matteo – risultati.doc
4. scuola – lavoro – casa – amici
5. relazione.doc – compito.doc

Creazione delle cartelle

Si rimanda al report M2\W5\D2 parte 1 dove la creazione delle cartelle è stata spiegata dettagliatamente.

Esercizi sulla gestione directory e file

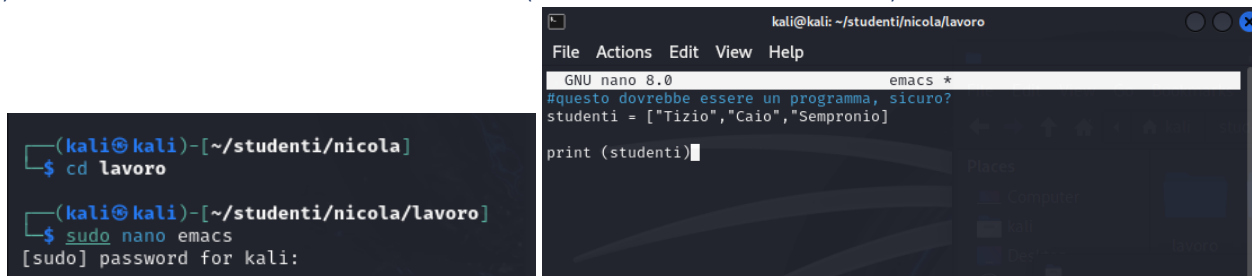
j) Cambiare gli attributi della directory lavoro facendo in modo che il suo contenuto non sia leggibile ma consentendo solo a te di entrarvi



1. Recarsi al livello appena superiore alla cartella lavoro con il comando **cd**.
2. Lanciare il comando **ls -al** per visualizzare gli attuali attributi della cartella.
3. Utilizzare il comando **chmod u=rwx,go-r-w-x lavoro** tradotto vuol dire utente attribuisce permessi di lettura (read), scrittura (write) e esecuzione (execute), mentre group (gruppo) e others (altri) toglie i permessi di rwx (lettura, scrittura ed esecuzione).
4. Verificare di aver apportato le modifiche utilizzando nuovamente il comando **ls -al**.

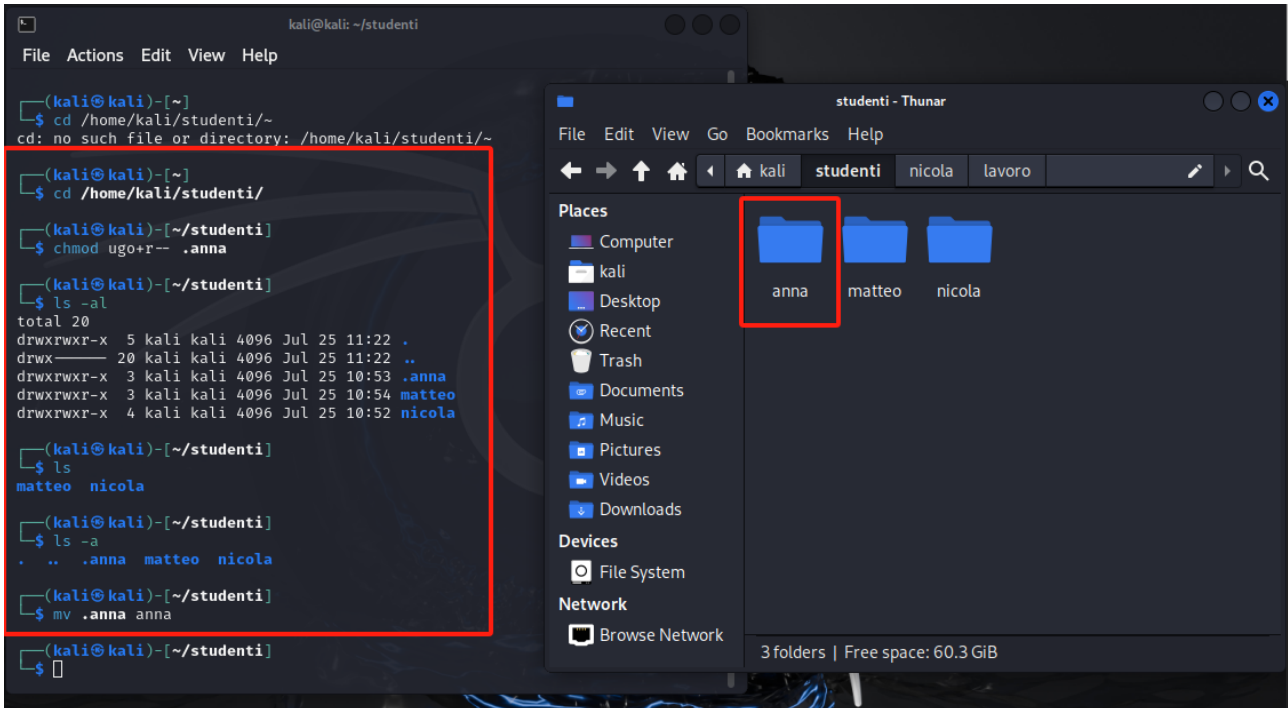
Come da immagine, anche l'interfaccia grafica in arancione, conferma l'avvenuta modifica.

k) Entra nella cartella lavoro e crea un file (emacs – ricordati di salvarlo)



Spostarsi nella cartella lavoro con il comando **cd** e usare l'editor nano (comando **sudo nano emacs**) per scrivere il contenuto del programma **emacs**

l) Prova a visualizzare il contenuto della directory dove ti trovi dopo di che modifica gli attributi della directory “. Facendo in modo che siano nuovamente leggibile il contenuto e prova nuovamente a leggere il contenuto



1. Utilizzando il comando **cd** recarsi alla cartella studenti
2. Cambiare i permessi con il comando **chmod ugo+r-- .anna** significa che la cartella nascosta anna per tutti user+group+others possa essere letta +read e – vuol dire che il resto rimane invariato
3. Verificare il contenuto con i comandi **ls** e/o **ls -a**
4. Rendere nuovamente non nascosto la cartella anna con il comando **mv .anna anna**

m) Spostati nella cartella scuola usando il percorso relativo (a dove ti trovi)

```

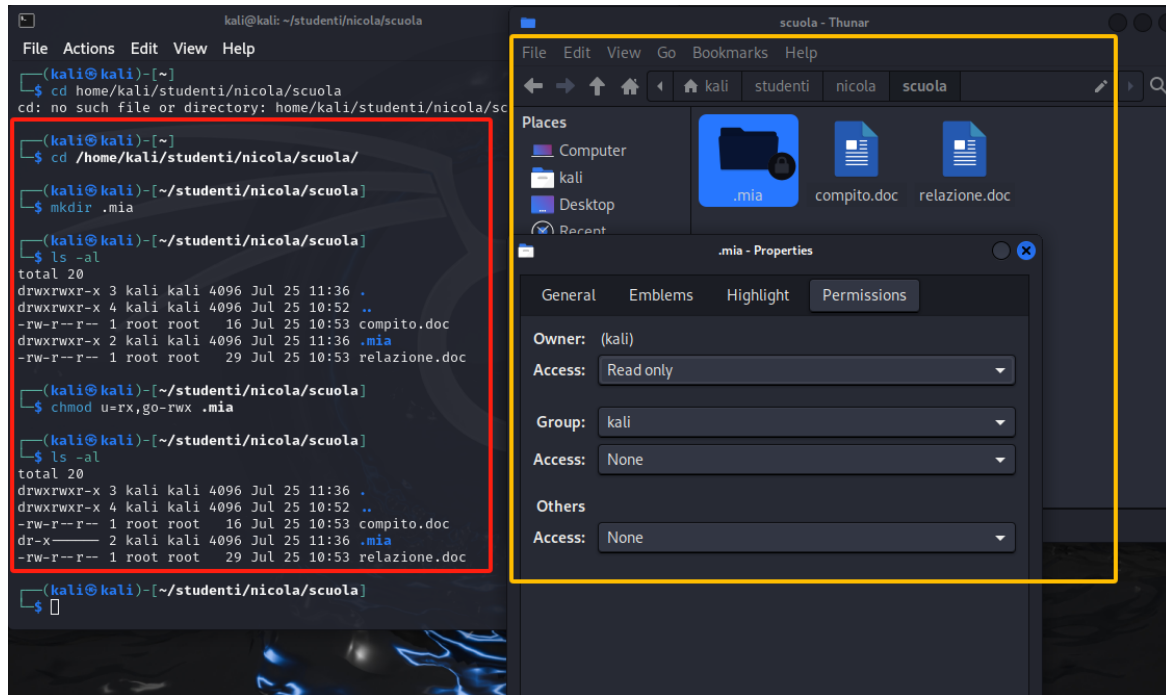
(kali@kali)-[~/studenti]
$ cd nicola

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

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

Come da immagine utilizzando il comando **cd**

n) Nella directory scuola crea una directory .mia (punto mia) e fai in modo che sia leggibile ed eseguibile solo da te



1. Recarsi con il comando **cd** nella cartella scuola
2. Creare la cartella .mia quindi **mkdir .mia**
3. Ottenere la lista del contenuto, anche nascosto della cartella e i permessi attuali comando **ls -al**
4. Modificare i permessi **chmod u=rwx, go-rwx .mia** che vuol dire utente = read write execute mentre group & others togli i permessi read write execute alla cartella .mia
5. Verificare nuovamente delle modifiche apportate con **ls -al**
6. Modifiche apportate con successo: non ci sono permessi nella colonna group e others

o) Prova a eseguire il comando **ls** e successivamente **ls -al**

```

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

(kali@kali)-[~/studenti/nicola/scuola]
$ ls -al
total 20
drwxrwxr-x 3 kali kali 4096 Jul 25 11:36 .
drwxrwxr-x 4 kali kali 4096 Jul 25 10:52 ..
-rw-r--r-- 1 root root 16 Jul 25 10:53 compito.doc
dr-x----- 2 kali kali 4096 Jul 25 11:36 .mia
-rw-r--r-- 1 root root 29 Jul 25 10:53 relazione.doc
  
```

La differenza sta nella mole di dati che si ottengono con **ls -al** in più: soprattutto file nascosti e attributi.

Esercizi sulla gestione processi

10. lascia il comando emacs &

```
kali@kali: ~/studenti/nicola/lavoro
File Actions Edit View Help

(kali@kali)-[~/studenti/nicola/lavoro]
$ emacs &
[1] 169380

(kali@kali)-[~/studenti/nicola/lavoro]
$ Command 'emacs' not found, did you mean:
  command 'jmacs' from deb joe
  command 'jmacs' from deb joe-jupp
Try: sudo apt install <deb name>

[1] + exit 127  emacs
(kali@kali)-[~/studenti/nicola/lavoro]
$
```

1. Recarsi alla cartella lavoro con il comando **cd**
2. Avviare emacs in background **emacs &**
3. Si evince che il PID è 169380 tuttavia è stato chiuso

11. esegui il comando jobs

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

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

Al momento non funziona in quanto non ci sono processi di competenza del comando attivi.

12. lancia il comando firefox e successivamente sul terminale premi CTRL+Z

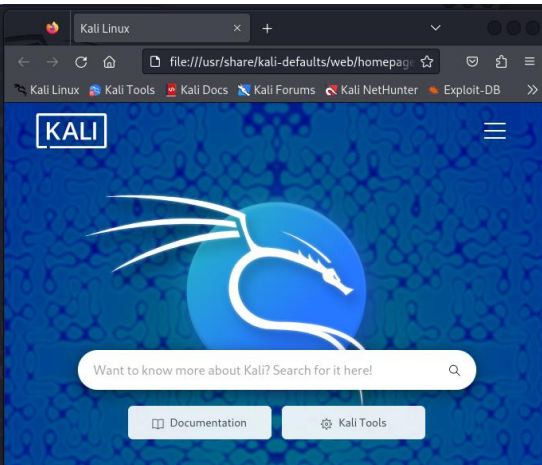
```
kali@kali: ~/studenti/nicola/lavoro
File Actions Edit View Help

(kali@kali)-[~/studenti/nicola/lavoro]
$ emacs &
[1] 169380

(kali@kali)-[~/studenti/nicola/lavoro]
$ Command 'emacs' not found, did you mean:
  command 'jmacs' from deb joe
  command 'jmacs' from deb joe-jupp
Try: sudo apt install <deb name>

[1] + exit 127  emacs
(kali@kali)-[~/studenti/nicola/lavoro]
$ jobs

(kali@kali)-[~/studenti/nicola/lavoro]
$ firefox
zsh: suspended firefox
(kali@kali)-[~/studenti/nicola/lavoro]
$
```



13. manda il processo firefox in background (bg) | 14. lancia il comando jobs

```
(kali@kali)-[~/studenti/nicola/lavoro]
$ bg firefox
[1] + continued firefox

(kali@kali)-[~/studenti/nicola/lavoro]
$ jobs
[1] + running  firefox
```

1. Utilizzare comando **bg firefox** per portarlo in background
2. Ora si può usare il comando **jobs** che riporta il processo di firefox in background

15. manda in foreground (fg) il programma emacs

```
(kali㉿kali)-[~/studenti/nicola/lavoro]
$ fg emacs
fg: job not found: emacs
```

Operazione non possibile perché il programma emacs è stato già terminato dal sistema precedentemente. Vedi punto 10.

16. termina emacs con i tasti CTRL+C

Operazione non possibile perché il programma emacs è stato già terminato dal sistema precedentemente. Vedi punto 10.

17. verificare quanto spazio si sta occupando sul disco

```
(kali㉿kali)-[~/studenti/nicola/lavoro]
$ du -h
8.0K .

(kali㉿kali)-[~/studenti/nicola/lavoro]
$ du
8 .
```

Utilizzando il comando **du -h** riporta che la directory occupa 8 kilobyte sul disco.