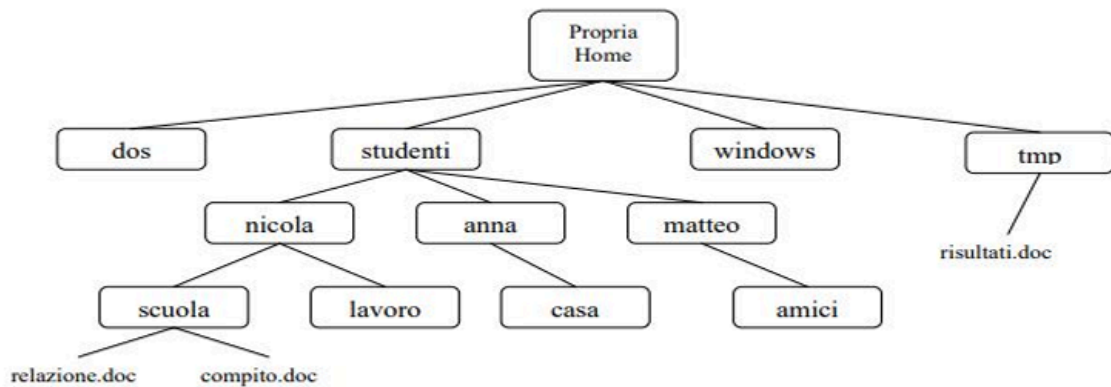


W5D1 - Shell Linux

Esercizio 1

Come prima cosa creare le seguenti cartelle e sottocartelle (usando i comandi “terminale” `mkdir` `cd` `rmdir` ... a partire dalla propria HOME e visualizzarle a video:

(Per “Propria home” si intende il posto dove vi posiziona quanto aprite il terminale!)



Come propria HOME ho voluto creare una directory a parte chiamata “esercizio” (solo per comodità)

quindi con il comando:

`mkdir esercizio`

si crea una directory

prepariamo le cartelle e sottocartelle con il comando `mkdir`, per fare più veloce, ad esempio per le sottocartelle della directory studenti, si può fare, entrando con il comando `cd studenti`, `mkdir {nicola,anna,matteo}`, così almeno per creare 3 cartelle dentro studenti, stessa cosa si può fare anche per la cartella “nicola”.

per invece creare i file.doc dentro le cartelle, basta usare il comando `touch`, quindi:

`cd studenti/nicola/scuola`

`touch relazione.doc compito.doc`

il comando `touch` quindi crea file.

usando questi comandi riusciamo a creare la struttura come in figura sopra.

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

12 directories, 3 files

con il comando `tree` si può vedere meglio la struttura di “esercizio” a “albero”.

- a) Per copiare un file si usa il comando “cp” - copy, quindi per copiare il file compito.doc, che si trova su scuola, nella directory casa basta fare:

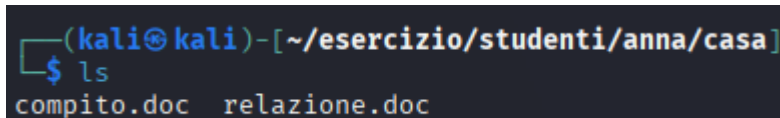
```
cd //cd solo torna nella directory principale in questo caso “esercizio”  
cd studenti/anna/casa
```

`cp studenti/nicola/scuola/compito.doc compito.doc` //il secondo “compito.doc” è per rinominare il file (volendo si poteva mettere “.” per dire copiare qui o altro nome per la rinomina.

- b) per invece spostare un file si usa il comando “mv” - move, quindi per spostare il file relazione.doc nella cartella casa basta fare:

```
mv studenti/nicola/scuola/relazione.doc .
```

se vogliamo fare un check si può fare con il comando “ls” list- fa vedere il contenuto della cartella



```
(kali@kali)-[~/esercizio/studenti/anna/casa]  
$ ls  
compito.doc  relazione.doc
```

- c) per cancellare la directory “tmp” basta usare il comando `rmdir`, ma prima bisogna eliminare il contenuto della cartella con il comando `rm` (rm non può eliminare tmp perché è una directory), quindi:

```
rm tmp/risultati.doc  
rmdir tmp
```

- d) per creare il file pippo.txt nella cartella lavoro si fa:

```
cd studenti/nicola/lavoro  
touch pippo.txt
```

- e) Per cambiare gli attributi del file pippi.txt e renderlo scrivibile e leggibile solo per il proprietario e per tutti gli altri solo leggibile basta fare:

```
ls -l //per visualizzare gli attributi del file
```

```
chmod 644 pippo.txt //questo per cambiare gli attributi di un file, i numeri sono i bit dato che gli attributi sono 3 blocchi di byte
```

(es. `-rwxrwxrwx` il verde è per l'owner, il giallo è per i gruppi, il rosso è per gli altri (others) e sono basati su bit, 777 per attivare tutto 000 per disattivare, poi ci sono varie combinazioni seguendo la figura sotto) in questo caso 6 per attivare lettura e scrittura (owner), 4 per attivare solo lettura (gruppi), 4 solo lettura (others)

Octal	Binary	File Mode
0	000	- - -
1	001	- - x
2	010	- w -
3	011	- wx
4	100	r - -
5	101	r - x
6	110	rw -
7	111	rwx

- f) per nascondere il contenuto della cartella anna basta fare:
`cd studenti/anna`
`mv *.casa` //in questo spostato tutto il contenuto della cartella anna su .casa, il punto all'inizio è per renderlo nascosto e per visualizzare cartelle nascoste si usa il comando `ls -a`
- g) per visualizzare il contenuto di un file si usa il comando `cat`

```
(kali㉿kali)-[~/esercizio/studenti/anna]
$ cd ../nicola/lavoro

(kali㉿kali)-[~/esercizio/studenti/nicola/lavoro]
$ ls

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

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

(kali㉿kali)-[~/esercizio/studenti/nicola/lavoro]
$ echo "ciao sono pippo" > pippo.txt

(kali㉿kali)-[~/esercizio/studenti/nicola/lavoro]
$ cat pippo.txt
ciao sono pippo
```

il comando `echo` ripete ciò che si scrive e ">" lo manda dentro il file `pippo.txt`

- h) per rimuovere la cartella amici basta fare:
`rmdir studenti/matteo/amici`
- i) per rimuovere tutte le cartelle create si usa il comando `rm -rf`, `-rf` serve appunto per eliminare sia file che cartelle

ESERCIZIO FACOLTATIVO

1. aprire terminale
2. per leggere i manuali si usa il comando `man`:
`man ps` : `ps` fornisce un riepilogo di tutti i processi in corso
`man kill` : `kill` termina un processo (in realtà manda segnali al processo)
`man jobs` : `jobs` visualizza i processi in esecuzione in background
3. lanciando `vi pippo` mi fa modificare il contenuto di `pippo`, infatti `vi` è un editor di testo
4. aprendo nuovo terminale e eseguendo il comando `ps aux` (`aux` per vedere tutti i processi dell'utente) mi fa vedere tutti i processi in esecuzione come appunto c'è anche "vi pippo".
5. per terminare il processo `vi pippo` si usa il comando `kill`:

```
(kali㉿kali)-[~/esercizio]
$ kill 76944
```

il numero è il PID (un numero di identificazione dei processi)

appunto nell'altro terminale viene questo messaggio:

```
(kali㉿kali)-[~/esercizio]
$ vi pippo
Vim: Caught deadly signal TERM
Vim: Finished.
zsh: terminated vi pippo lo
```

6. per avviare firefox in background si usa il comando *firefox &*
7. per invece portarlo in background si usa *bg* dopo averlo avviato
8. per terminare firefox basta fare come in figura, *kill %1* è per killare il processo e l'1 è il numero del processo di firefox come si vede appena dopo averlo messo in bg

```
(kali㉿kali)-[~/esercizio]
$ firefox
^Z
zsh: suspended firefox

(kali㉿kali)-[~/esercizio]
$ bg
[1] + continued firefox

(kali㉿kali)-[~/esercizio]
$ kill %1

(kali㉿kali)-[~/esercizio]
$ 
[1] + terminated firefox
(kali㉿kali)-[~/esercizio]
$
```

9. per verificare quanto spazio si sta occupando su disco si può usare il comando *df -h* (-h per human readable così almeno è più leggibile)

```
(kali㉿kali)-[~]
$ df -h
```

Filesystem	Size	Used	Avail	Use%	Mounted on
udev	921M	0	921M	0%	/dev
tmpfs	198M	968K	197M	1%	/run
/dev/sda1	79G	16G	59G	22%	/
tmpfs	987M	4.0K	987M	1%	/dev/shm
tmpfs	5.0M	0	5.0M	0%	/run/lock
tmpfs	1.0M	0	1.0M	0%	/run/credentials/systemd-journald.service
tmpfs	987M	284K	987M	1%	/tmp
tmpfs	1.0M	0	1.0M	0%	/run/credentials/getty@tty1.service
tmpfs	198M	124K	198M	1%	/run/user/1000