

W5D1 - Shell Linux

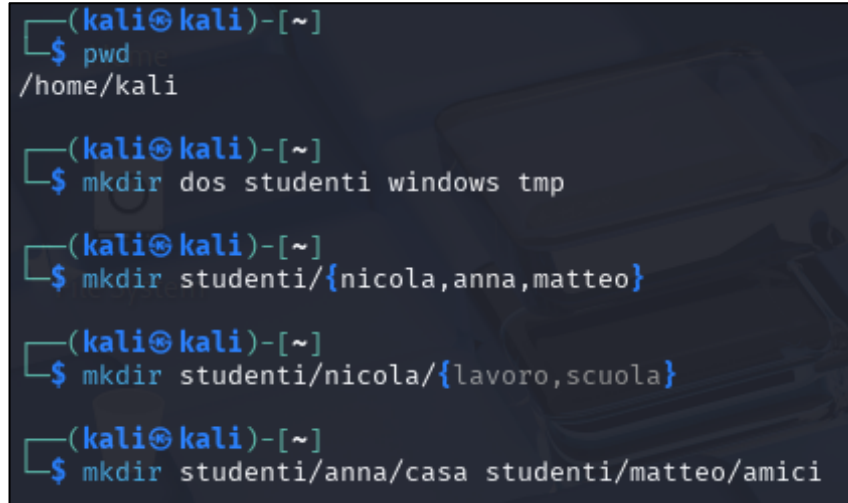
Obiettivo dell'esercitazione

Questo report documenta l'esercizio pratico sull'uso dei comandi Linux per la gestione di file e cartelle. Ho creato una struttura di directory, manipolato file, gestito permessi e processi, seguendo le istruzioni dell'esercizio.

Creazione struttura cartelle

Creazione delle cartelle principali e sottocartelle come richiesto dall'esercizio.

1. Uso del comando "pwd" che mostra il percorso assoluto della cartella in cui ci troviamo. Utile per confermare la propria posizione nell'albero delle directory prima di eseguire operazioni critiche.
2. Uso del comando "mkdir" per creare le cartelle principali (dos, studenti, windows, tmp) e le sottocartelle richieste:



```
(kali㉿kali)-[~]  
$ pwd  
/home/kali  
  
(kali㉿kali)-[~]  
$ mkdir dos studenti windows tmp  
  
(kali㉿kali)-[~]  
$ mkdir studenti/{nicola,anna,matteo}  
  
(kali㉿kali)-[~]  
$ mkdir studenti/nicola/{lavoro,scuola}  
  
(kali㉿kali)-[~]  
$ mkdir studenti/anna/casa studenti/matteo/amici
```

Creazione cartelle principali e sottocartelle

3. Conferma della struttura creata usando "ls -R", che mostra ricorsivamente tutte le sottocartelle:

Verifica struttura cartelle

Esempio di modifica e salvataggio del file risultati.doc con vi, usando “wq” per salvare:



Creazione dei file risultati.doc, compito.doc e relazione .doc con l'editor vi. Ogni file è stato posizionato nella cartella corretta:

```
(kali㉿kali)-[~]  
$ vi tmp/risultati.doc  
  
(kali㉿kali)-[~]  
$ vi studenti/nicola/scuola/compito.doc  
  
(kali㉿kali)-[~]  
$ vi studenti/nicola/scuola/relazione.doc  
  
(kali㉿kali)-[~]  
$ ls studenti/nicola/scuola/  
compito.doc  relazione.doc
```

Verifica dei doc creati con vi

Spostamento tra Directory

Spostamento nella cartella lavoro con "cd studenti/nicola/lavoro". Verifica del percorso con pwd.

```
(kali㉿kali)-[~]  
$ cd studenti/nicola/lavoro  
  
(kali㉿kali)-[~/studenti/nicola/lavoro]  
$ pwd  
/home/kali/studenti/nicola/lavoro
```

Accesso alla cartella lavoro

1. Uso del percorso relativo "../.. /anna/casa" per spostarsi da lavoro a casa. Ogni ../ sale di un livello:

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

Percorso relativo per spostarsi

2. Uso del percorso assoluto “~/studenti/anna/casa” per raggiungere la stessa cartella. Parte sempre dalla home (~)
3. Entrambi i metodi portano alla stessa destinazione:

```
(kali㉿kali)-[~/studenti/anna/casa]
$ cd ~/studenti/anna/casa

(kali㉿kali)-[~/studenti/anna/casa]
$ pwd
/home/kali/studenti/anna/casa
```

Percorso assoluto per spostarsi

Operazioni su file

1. Copia del file compito.doc dalla cartella scuola a casa con “cp” e percorso relativo:

```
(kali㉿kali)-[~/studenti/anna/casa]
$ cp ../ ../nicola/scuola/compito.doc .

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

Copia di compito.doc

2. Spostamento del file relazione.doc con “mv”. Verifica con “ls” che i file siano presenti:

```
(kali㉿kali)-[~/studenti/anna/casa]
$ mv ../ ../nicola/scuola/relazione.doc .

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

Spostamento relazione.doc

Gestione permessi

1. Creazione del file pippo.txt nella cartella lavoro usando vi. Il file contiene il testo "Contenuto di prova Pippo":

```
(kali㉿kali)-[~/studenti/anna/casa]
$ cd ~/studenti/nicola/lavoro
(kali㉿kali)-[~/studenti/nicola/lavoro]
$ vi pippo.txt
```

Creazione di pippo.txt

2. Modifica dei permessi con "chmod u=rw,go=r" per impostare: amministratore (lettura+scrittura), gruppo/altri (solo lettura)

```
(kali㉿kali)-[~/studenti/nicola/lavoro]
$ chmod u=rw,go=r pippo.txt
(kali㉿kali)-[~/studenti/nicola/lavoro]
$ ls -l pippo.txt
-rw-r--r-- 1 kali kali 25 Mar 26 15:14 pippo.txt
```

Cambio permessi di pippo.txt

3. Lettura del contenuto di pippo.txt con "cat". Mostra il testo inserito in precedenza:

```
(kali㉿kali)-[~/studenti/nicola/lavoro]
$ cat pippo.txt
Contenuto di prova Pippo
```

Visualizzazione di pippo.txt

Gestione Cartelle

1. Impostazione dei permessi con "chmod u=rwx, go= ~/studenti/anna " per la cartella anna. Solo l'amministratore può accedere:

```
(kali㉿kali)-[~/studenti/nicola/lavoro]
$ chmod u=rwx,go= ~/studenti/anna

(kali㉿kali)-[~/studenti/nicola/lavoro]
$ ls -ld ~/studenti/anna
drwx----- 3 kali kali 4096 Mar 26 14:49 /home/kali/studenti/anna
```

Nascondere la cartella anna

2. Eliminazione della cartella amici con "rmdir". Verifica con "ls" che sia stata rimossa:

```
(kali㉿kali)-[~/studenti/nicola/lavoro]
$ rmdir ~/studenti/matteo/amici

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

Eliminazione cartella amici

3. Eliminazione di tutte le cartelle create usando "rm -r" e verifica con "ls" che sia stato tutto rimosso:

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

(kali㉿kali)-[~]
$ rm -r dos studenti windows tmp

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

Pulizia finale

Esercizio Facoltativo:

Questa sezione completa il report con l'esercizio facoltativo sulla gestione dei processi e utenti in Linux, utilizzando i comandi richiesti.

Monitoraggio Utenti

Uso del comando "w" per visualizzare gli utenti collegati e i loro processi attivi. Mostra anche il carico del sistema:

```
(kali㉿kali)-[~]  
$ w  
16:48:59 up 4 min,  2 users,  load average: 0.15, 0.14, 0.05  
USER      TTY      FROM          LOGIN@   IDLE   JCPU   PCPU   WHAT  
kali      -        -             16:46    4:13   0.00s  0.01s  lightdm --ses  
kali      -        -             16:46    4:13   0.00s  0.11s  /usr/lib/syst
```

Output del comando w

Gestione Processi

1. Gestione del processo vi: lancio in background, sospensione e ripresa "fg". Il comando "jobs" mostra lo stato:

```
(kali㉿kali)-[~]  
$ vi pippo &  
[1] 9839  
  
(kali㉿kali)-[~]  
$  
[1] + suspended (tty output) vi pippo  
(kali㉿kali)-[~]  
$ fg %1  
[1] + continued vi pippo  
  
zsh: suspended vi pippo  
  
(kali㉿kali)-[~]  
$ jobs  
[1] + suspended vi pippo
```

Gestione processo vi

2. Identificazione del PID di vi con "ps" e terminazione forzata con "kill -9":

```
(kali㉿kali)-[~]
$ ps -u $USER | grep vi
1138 ?          00:00:00 dconf-service
9839 pts/0      00:00:00 vi

(kali㉿kali)-[~]
$ kill -9 9839
```

Terminazione del processo

Gestione Firefox

1. Lancio di Firefox in background (&) e gestione con "bg". Il comando jobs conferma l'esecuzione in background:

```
(kali㉿kali)-[~]
$ firefox &
[1] 13429

(kali㉿kali)-[~]
$ bg %1
bg: job already in background

(kali㉿kali)-[~]
$ jobs
[1] + running  firefox
```

Firefox background

2. Terminazione di Firefox con "kill -9". L'avviso "channel error" è normale per processi grafici chiusi forzatamente:

```
(kali㉿kali)-[~]
$ ps -u $USER | grep firefox
13429 pts/0      00:00:03 firefox-esr

(kali㉿kali)-[~]
$ kill -9 13429

(kali㉿kali)-[~]
$
$ 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.
Exiting due to channel error.

[1] + killed  firefox

(kali㉿kali)-[~]
$ jobs
```

Terminazione di Firefox

Verifica spazio disco

Verifica dello spazio disco con “df -h” (partizione utente) e “du -sh” (home utente):

```
(kali㉿kali)-[~]  
$ df -h /  
Filesystem      Size  Used Avail Use% Mounted on  
/dev/sda1        79G   15G   60G   20% /  
  
(kali㉿kali)-[~]  
$ du -sh ~  
31M    /home/kali
```

Verifica spazio disco

Conclusioni

Esercizio principale:

Cosa ho imparato:

- Creazione e organizzazione di cartelle e sottocartelle con mkdir;
- Gestione di file tramite vi, cp, mv e rm;
- Differenza tra percorsi relativi e assoluti;
- Impostazione dei permessi con chmod;
- Importanza del comando pwd per verificare la posizione corrente.

Competenze acquisite:

- Padronanza nella navigazione del filesystem;
- Capacità di manipolare file e cartelle in modo sicuro;
- Comprensione dei permessi per la sicurezza dei dati.

Esercizio facoltativo:

Cosa ho imparato:

- Monitoraggio utenti;
- Gestione dei processi;
- Terminazione di processi;
- Verifica dello spazio su disco.

Competenze acquisite:

- Controllo dei processi attivi e delle risorse di sistema;
- Ottimizzazione dell'uso della CPU e memoria.