

## INDICE

Indice.....	1
INTRODUZIONE ED OBIETTIVO.....	2
VERIFICA DEL CONTESTO OPERATIVO.....	3
CREAZIONE DELLA STRUTTURA PRINCIPALE DI DIRECTORY.....	4
CREAZIONE SOTTOCARTELLE.....	5-6
CREAZIONE DEI FILE NELLE DIRECTORY.....	7-8
COPIA DEL FILE.....	9
MODIFICA ALLOCAZIONE DEL FILE.....	10
ELIMINAZIONE DELLA DIRECTORY "TMP" .....	11-12
CREAZIONE FILE PIPPO.....	13
MODIFICA DEI PERMESSI DEL FILE PIPPO.....	14
RINOMINA DIRECTORY.....	15
VISUALIZZAZIONE DEL CONTENUTO DEL FILE PIPPO.TXT.....	16
RIMOZIONE DIRECTORY AMICI.....	17
RIMOZIONE DIRECTORY PRINCIPALI.....	18-19
CONCLUSIONE.....	20

## INTRODUZIONE ED OBIETTIVO

Nel presente esercizio, svolto nel terminale di Kali Linux, si è lavorato alla gestione di file e directory impiegando esclusivamente alcuni comandi della shell (*cd*, *mkdir*, *rmdir*, *mv*, *cp*, *rm*, *ls*, *pwd*, *man*).

L'obiettivo consisteva nel ricreare, a partire dalla Home, una struttura gerarchica di cartelle comprendente le directory principali *"dos"*, *"studenti"*, *"windows"* e *"tmp"* e, all'interno di *"studenti"*, le sottodirectory *"nicola"*, *"anna"* e *"matteo"* con le rispettive dir secondarie.

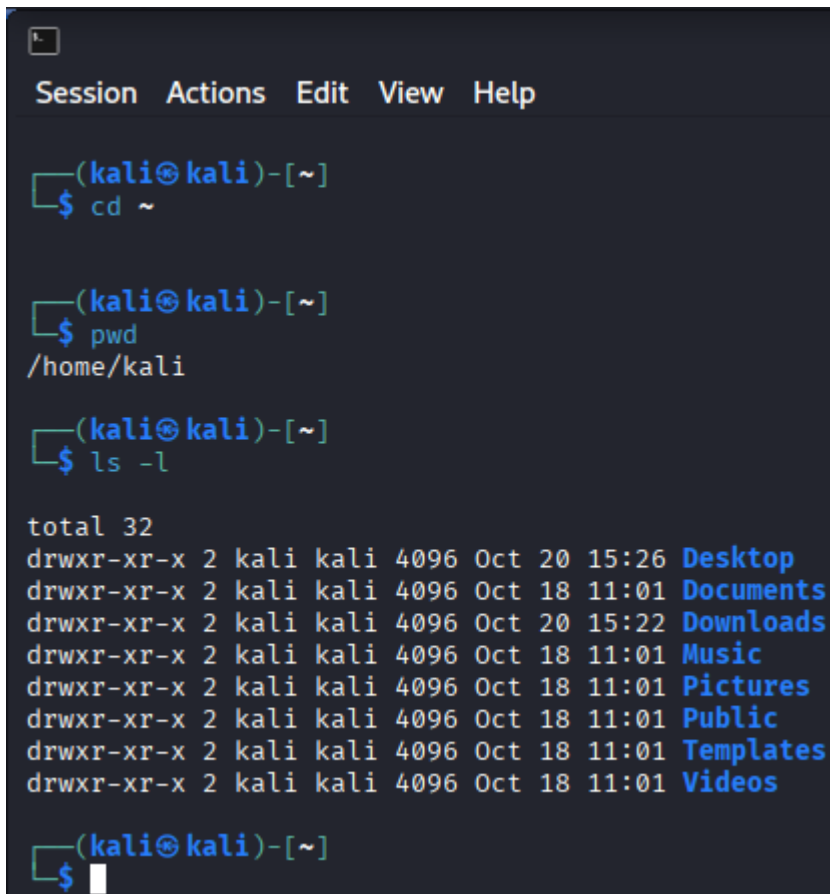
Sono stati creati file dimostrativi (*"relazione.doc"*, *"compito.doc"*, *"risultati.doc"*) e sono state eseguite operazioni di copia e spostamento con percorsi sia relativi sia assoluti al fine di consolidare l'indirizzamento nel file system.

È stata inoltre applicata la modifica dei permessi su un file per ottenere lettura/scrittura al proprietario e sola lettura per gruppo e altri.

Si è mostrato, altresì, come nascondere una directory antepoendo un punto al nome.

Le verifiche finali sono state svolte con le varianti di *ls*, seguite dalla rimozione della struttura creata per ripristinare l'ambiente.

## VERIFICA DEL CONTESTO OPERATIVO



```
Session  Actions  Edit  View  Help

(kali@kali)-[~]
$ cd ~

(kali@kali)-[~]
$ pwd
/home/kali

(kali@kali)-[~]
$ ls -l

total 32
drwxr-xr-x 2 kali kali 4096 Oct 20 15:26 Desktop
drwxr-xr-x 2 kali kali 4096 Oct 18 11:01 Documents
drwxr-xr-x 2 kali kali 4096 Oct 20 15:22 Downloads
drwxr-xr-x 2 kali kali 4096 Oct 18 11:01 Music
drwxr-xr-x 2 kali kali 4096 Oct 18 11:01 Pictures
drwxr-xr-x 2 kali kali 4096 Oct 18 11:01 Public
drwxr-xr-x 2 kali kali 4096 Oct 18 11:01 Templates
drwxr-xr-x 2 kali kali 4096 Oct 18 11:01 Videos

(kali@kali)-[~]
$
```

All'inizio del lavoro ci si porta nella directory personale dell'utente mediante il comando `cd ~`.

La tilde indica in modo chiaro la Home corrente e consente di partire da un punto noto, evitando equivoci sui percorsi.

Subito dopo, `pwd` conferma la posizione mostrando il percorso assoluto (ad esempio `/home/kali`): è la verifica rapida che l'ambiente di lavoro è quello giusto.

A questo punto si esegue `ls -l` per ottenere una vista ordinata e dettagliata del contenuto iniziale della Home.

Il formato esteso elenca, per ogni elemento, tipo e permessi (con il primo carattere che distingue file regolari - da directory `d`), proprietario, gruppo, dimensione in byte, data/ora dell'ultima modifica e nome.

Questo passaggio serve da base di confronto per i passaggi successivi: documenta lo stato di partenza, permette di notare eventuali residui di esercitazioni precedenti e offre un quadro preciso prima di creare la nuova struttura.

## CREAZIONE DELLA STRUTTURA PRINCIPALE DI DIRECTORY

```
(kali㉿kali)-[~]
$ mkdir dos studenti windows tmp

(kali㉿kali)-[~]
$ ls -l

total 48
drwxr-xr-x 2 kali kali 4096 Oct 20 15:26 Desktop
drwxr-xr-x 2 kali kali 4096 Oct 18 11:01 Documents
drwxrwxr-x 2 kali kali 4096 Oct 21 08:53 dos
drwxr-xr-x 2 kali kali 4096 Oct 20 15:22 Downloads
drwxr-xr-x 2 kali kali 4096 Oct 18 11:01 Music
drwxr-xr-x 2 kali kali 4096 Oct 18 11:01 Pictures
drwxr-xr-x 2 kali kali 4096 Oct 18 11:01 Public
drwxrwxr-x 2 kali kali 4096 Oct 21 08:53 studenti
drwxr-xr-x 2 kali kali 4096 Oct 18 11:01 Templates
drwxrwxr-x 2 kali kali 4096 Oct 21 08:53 tmp
drwxr-xr-x 2 kali kali 4096 Oct 18 11:01 Videos
drwxrwxr-x 2 kali kali 4096 Oct 21 08:53 windows
```

Dopo aver verificato la posizione nella directory Home è stata avviata la creazione della struttura principale richiesta dall'esercizio, composta dalle cartelle "dos", "studenti", "windows" e "tmp".

L'operazione è stata eseguita tramite il comando `mkdir dos studenti windows tmp`, che consente di generare più directory in un'unica istruzione, riducendo il numero di comandi e rendendo più pratica la procedura di configurazione iniziale.

Per confermare la corretta creazione delle directory, è stato utilizzato il comando `ls -l`, al fine di visualizzare l'elenco dei file e delle cartelle presenti nella directory corrente, includendo informazioni aggiuntive come permessi, proprietario, gruppo, dimensioni e data di ultima modifica.

Ogni directory è contrassegnata dalla lettera iniziale "**d**", seguita dai permessi in forma simbolica (es. `drwxr-xr-x`). In questo caso, i permessi sono strutturati in modo che l'utente proprietario abbia pieni diritti di lettura, scrittura ed esecuzione (**rw****x**), mentre il gruppo e gli altri utenti dispongano soltanto dei permessi di lettura ed esecuzione (**r****-x**).

## CREAZIONE SOTTOCARTELLE

```
(kali㉿kali)-[~]
$ cd studenti

(kali㉿kali)-[~/studenti]
$ mkdir nicola anna matteo

(kali㉿kali)-[~/studenti]
$ mkdir nicola/scuola nicola/lavoro anna/casa matteo/amici

(kali㉿kali)-[~/studenti]
$ ls -R
.:
anna matteo nicola

./anna:
casa

./anna/casa:

./matteo:
amici

./matteo/amici:

./nicola:
lavoro scuola

./nicola/lavoro:

./nicola/scuola:
```

In questa fase dell'esercitazione è stata definita la struttura gerarchica interna della directory *studenti* e , dopo essersi posizionati all'interno della directory tramite il comando 'cd studenti', è stato utilizzato il comando 'mkdir nicola anna matteo' per creare simultaneamente tre sottocartelle principali, corrispondenti ai nominativi dei tre utenti.

Successivamente, attraverso il comando:

'mkdir nicola/scuola nicola/lavoro anna/casa matteo/amici' sono state create le rispettive sottocartelle di secondo livello. Nello specifico:

- all'interno della cartella *nicola* sono state generate le directory *scuola* e *lavoro*;
- all'interno della cartella *anna* la directory *casa*;
- all'interno della cartella *matteo* la directory *amici*.

Per verificare la corretta creazione della struttura completa, è stato eseguito il comando `'ls -R'`, che elenca in modo ricorsivo il contenuto di tutte le directory e sottodirectory presenti.

Dall'output si osserva che ogni cartella risulta correttamente posizionata e vuota, confermando la coerenza con l'albero logico previsto.

## CREAZIONE DEI FILE NELLE DIRECTORY

```
└─$ cd ~

(kali㉿kali)-[~]
└─$ touch studenti/nicola/scuola/relazione.doc studenti/nicola/lavoro/compito.doc tmp/risultati.doc

(kali㉿kali)-[~]
└─$ ls -R
.:
Desktop Documents dos Downloads Music Pictures Public studenti Templates tmp Videos windows

./Desktop:
code.desktop nm-connection-editor.desktop org.wireshark.Wireshark.desktop

./Documents:

./dos:

./Downloads:
code_1.105.1-1760482543_amd64.deb

./Music:

./Pictures:

./Public:

./studenti:
anna matteo nicola

./studenti/anna:
casa

./studenti/anna/casa:

./studenti/matteo:
amici

./studenti/matteo/amici:

./studenti/nicola:
lavoro scuola

./studenti/nicola/lavoro:
compito.doc

./studenti/nicola/scuola:
relazione.doc

./Templates:

./tmp:
risultati.doc

./Videos:

./windows:
```

Una volta completata la struttura delle directory, è stato necessario popolare le sottocartelle con i file richiesti, al fine di simulare la presenza di documenti di lavoro all'interno della gerarchia appena creata. Dalla directory *home*, è stato eseguito il comando: '*touch studenti/nicola/scuola/relazione.doc studenti/nicola/lavoro/compito.doc tmp/risultati.doc*'. Il comando *touch* viene utilizzato per creare file vuoti oppure per aggiornare la data di ultima modifica di file già esistenti. In questo caso, ha permesso di generare tre file distinti nelle posizioni indicate:

- *relazione.doc* all'interno di *studenti/nicola/scuola*;
- *compito.doc* all'interno di *studenti/nicola/lavoro*;
- *risultati.doc* all'interno della directory *tmp*

Successivamente, è stato impiegato il comando `ls -R` per visualizzare in maniera ricorsiva l'intera struttura delle directory e verificare la presenza effettiva dei file.



## CREAZIONE DEL FILE

```
(kali㉿kali)-[~]  
$ cd ~/studenti/nicola/lavoro  
  
(kali㉿kali)-[~/studenti/nicola/lavoro]  
$ pwd  
/home/kali/studenti/nicola/lavoro  
  
(kali㉿kali)-[~/studenti/nicola/lavoro]  
$ cp ../scuola/relazione.doc .  
  
(kali㉿kali)-[~/studenti/nicola/lavoro]  
$ cp /home/kali/studenti/nicola/scuola/relazione.doc /home/kali/studenti/nicola/lavoro/  
  
(kali㉿kali)-[~/studenti/nicola/lavoro]  
$ ls -l  
total 0  
-rw-rw-r-- 1 kali kali 0 Oct 21 09:10 compito.doc  
-rw-rw-r-- 1 kali kali 0 Oct 21 09:18 relazione.doc
```

È stato necessario copiare il file *relazione.doc* dalla sottocartella *scuola* alla sottocartella *lavoro*, entrambe appartenenti all'utente *nicola*.

L'operazione è servita per simulare lo spostamento di documenti tra due aree di lavoro differenti, mantenendo però una copia del file originale.

Per prima cosa, ci si è posizionati nella directory di destinazione tramite il comando: *cd ~/studenti/nicola/lavoro.*

La corretta posizione è stata verificata utilizzando il comando *pwd*, che ha restituito il percorso assoluto */home/kali/studenti/nicola/lavoro* ; successivamente è stato utilizzato il comando *cp* per copiare il file.

Il comando *ls -l* ha confermato la presenza dei due file nella directory di destinazione.

## MODIFICA ALLOCAZIONE DEL FILE

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

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

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

Completata la fase di copia, è stato eseguito lo spostamento del documento *relazione.doc* dall'area di lavoro di *nicola* (sottocartella *scuola*) all'area di *anna* (sottocartella *casa*).

L'operazione ha trasferito fisicamente il file nella nuova destinazione, senza lasciarne copie nella posizione di origine, a differenza della precedente duplicazione.

Al termine, una verifica del contenuto della cartella di destinazione ha confermato la presenza di *relazione.doc*, attestando la riuscita dello spostamento e la corretta gestione dei percorsi all'interno della gerarchia di directory.

## ELIMINAZIONE DELLA DIRECTORY "TMP"

```
(kali㉿kali)-[~]  
$ cd ~  
  
(kali㉿kali)-[~]  
$ ls -l  
total 48  
drwxr-xr-x 2 kali kali 4096 Oct 20 15:26 Desktop  
drwxr-xr-x 2 kali kali 4096 Oct 18 11:01 Documents  
drwxrwxr-x 2 kali kali 4096 Oct 21 08:53 dos  
drwxr-xr-x 2 kali kali 4096 Oct 20 15:22 Downloads  
drwxr-xr-x 2 kali kali 4096 Oct 18 11:01 Music  
drwxr-xr-x 2 kali kali 4096 Oct 18 11:01 Pictures  
drwxr-xr-x 2 kali kali 4096 Oct 18 11:01 Public  
drwxrwxr-x 5 kali kali 4096 Oct 21 08:58 studenti  
drwxr-xr-x 2 kali kali 4096 Oct 18 11:01 Templates  
drwxrwxr-x 2 kali kali 4096 Oct 21 09:10 tmp  
drwxr-xr-x 2 kali kali 4096 Oct 18 11:01 Videos  
drwxrwxr-x 2 kali kali 4096 Oct 21 08:53 windows  
  
(kali㉿kali)-[~]  
$ rm -r tmp  
  
(kali㉿kali)-[~]  
$ ls -l  
total 44  
drwxr-xr-x 2 kali kali 4096 Oct 20 15:26 Desktop  
drwxr-xr-x 2 kali kali 4096 Oct 18 11:01 Documents  
drwxrwxr-x 2 kali kali 4096 Oct 21 08:53 dos  
drwxr-xr-x 2 kali kali 4096 Oct 20 15:22 Downloads  
drwxr-xr-x 2 kali kali 4096 Oct 18 11:01 Music  
drwxr-xr-x 2 kali kali 4096 Oct 18 11:01 Pictures  
drwxr-xr-x 2 kali kali 4096 Oct 18 11:01 Public  
drwxrwxr-x 5 kali kali 4096 Oct 21 08:58 studenti  
drwxr-xr-x 2 kali kali 4096 Oct 18 11:01 Templates  
drwxr-xr-x 2 kali kali 4096 Oct 18 11:01 Videos  
drwxrwxr-x 2 kali kali 4096 Oct 21 08:53 windows
```

In questo passaggio è stata rimossa la directory *tmp*, precedentemente creata nella directory principale.

L'operazione aveva lo scopo di verificare l'utilizzo corretto del comando di eliminazione ricorsiva, indispensabile quando una directory contiene al suo interno file o sottocartelle.

Dopo aver visualizzato l'elenco delle directory presenti tramite *ls -l*, è stata individuata *tmp* come cartella da eliminare. L'operazione di rimozione è stata quindi eseguita con il comando appropriato, che ha consentito di eliminare la directory insieme al suo contenuto.

Una successiva verifica mediante ls -l ha confermato la scomparsa della cartella *tmp* dall'elenco delle directory della Home, dimostrando la corretta applicazione del comando e la riuscita dell'operazione di pulizia.

## CREAZIONE DEL FILE PIPPO

```
(kali㉿kali)-[~]  
$ cd ~/studenti/nicola/lavoro  
  
(kali㉿kali)-[~/studenti/nicola/lavoro]  
$ touch pippo.txt  
  
(kali㉿kali)-[~/studenti/nicola/lavoro]  
$ ls -l  
total 0  
-rw-rw-r-- 1 kali kali 0 Oct 21 09:10 compito.doc  
-rw-rw-r-- 1 kali kali 0 Oct 21 09:35 pippo.txt  
-rw-rw-r-- 1 kali kali 0 Oct 21 09:18 relazione.doc
```

In questo passaggio è stato creato un nuovo file di testo denominato *pippo.txt* all'interno della sottocartella *lavoro*, appartenente alla directory dell'utente *nicola*. Dopo essersi posizionati nella directory di destinazione, è stato utilizzato il comando appropriato per la creazione del file vuoto. Successivamente, mediante il comando di visualizzazione dettagliata dei contenuti, è stata confermata la presenza del nuovo file *pippo.txt* accanto ai documenti *compito.doc* e *relazione.doc*. Dall'output risulta che tutti i file condividono la stessa configurazione di permessi, indicata da -rw-rw-r--, che assegna all'utente proprietario (*kali*) e al gruppo i diritti di lettura e scrittura, mentre concede agli altri utenti il solo permesso di lettura.

## MODIFICA DEI PERMESSI DEL FILE PIPPO

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

(kali㉿kali)-[~/studenti/nicola/lavoro]
$ chmod 644 pippo.txt

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

-rw-r--r-- 1 kali kali 0 Oct 21 09:35 pippo.txt
```

In questa fase è stato modificato il livello di accesso del file *pippo.txt* all'interno della directory *lavoro*, con l'obiettivo di limitare i diritti di utilizzo agli altri utenti mantenendo la piena operatività per il proprietario.

Dopo aver verificato il percorso assoluto mediante il comando di controllo della posizione corrente, è stato applicato il comando di modifica dei permessi.

La nuova configurazione, assegna al proprietario del file (*kali*) i diritti di lettura e scrittura (*rw-*), mentre al gruppo e agli altri utenti solo il permesso di lettura (*r--*).

La successiva verifica tramite visualizzazione dettagliata dei file ha confermato l'avvenuto aggiornamento: l'output mostra infatti il set di permessi *-rw-r--r--*, corrispondente alla nuova politica di accesso.

Questa operazione garantisce un controllo più restrittivo sull'uso del file, assicurando che soltanto il proprietario possa modificarne il contenuto.

## RINOMINA DIRECTORY

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

(kali㉿kali)-[~/studenti/anna]
$ ls -l

total 4
drwxrwxr-x 2 kali kali 4096 Oct 21 09:26 casa

(kali㉿kali)-[~/studenti/anna]
$ mv casa .casa

(kali㉿kali)-[~/studenti/anna]
$ ls -l

total 0

(kali㉿kali)-[~/studenti/anna]
$ ls -la
total 12
drwxrwxr-x 3 kali kali 4096 Oct 21 09:44 .
drwxrwxr-x 5 kali kali 4096 Oct 21 08:58 ..
drwxrwxr-x 2 kali kali 4096 Oct 21 09:26 .casa
```

In questa fase è stata effettuata la rinomina della directory *casa* in *.casa* all'interno della sottocartella *anna*.

L'obiettivo dell'operazione era quello di trasformare una cartella visibile in una directory nascosta, sfruttando la convenzione di Linux secondo cui qualsiasi file o directory il cui nome inizia con un punto (.) viene considerato nascosto dal sistema. Dopo essersi spostati nella directory *anna* e aver verificato la presenza iniziale della cartella *casa*, è stato applicato il comando di rinomina.

Una successiva verifica con il comando *ls -l* non ha mostrato la cartella, poiché il suo nome, preceduto dal punto, la rende invisibile nella visualizzazione standard.

Per confermarne l'effettiva esistenza, è stato quindi utilizzato il comando *ls -la*, che elenca anche i file e le directory nascoste: da questo output si osserva la presenza di *'.casa'*, confermando il corretto completamento dell'operazione.

## VISUALIZZAZIONE DEL CONTENUTO DEL FILE PIPPO

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

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

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

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

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

In questa fase è stata verificata la possibilità di leggere il contenuto del file *pippo.txt* precedentemente creato nella directory *lavoro* dell'utente *nicola*.

Dopo essersi posizionati nel percorso corretto, è stato eseguito il comando di lettura del contenuto, senza tuttavia ottenere alcuna uscita a video.

L'assenza di testo nel risultato conferma che il file *pippo.txt* è effettivamente vuoto, poiché era stato generato in precedenza con il comando di creazione semplice, senza l'aggiunta di contenuti.

Questa verifica dimostra il corretto funzionamento dei permessi impostati nel passaggio precedente i quali consentono al proprietario di accedere e leggere il file, ma impediscono modifiche da parte di altri utenti.



## RIMOZIONE DELLA DIRECTORY "AMICI"

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

(kali㉿kali)-[~/studenti/matteo]
$ ls -l
total 4
drwxrwxr-x 2 kali kali 4096 Oct 21 08:59 amici

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

(kali㉿kali)-[~/studenti/matteo]
$ ls

(kali㉿kali)-[~/studenti/matteo]
$ ls -l
total 0
```

In questa parte dell'esercitazione è stata rimossa la directory *amici*, presente all'interno della cartella *matteo*.

Dopo essersi posizionati nella directory *matteo* e aver verificato la presenza della sottocartella *amici* tramite la visualizzazione dettagliata dei contenuti, è stato eseguito il comando di eliminazione

Poiché la directory non conteneva file al suo interno, la rimozione è avvenuta senza errori; la conferma è stata ottenuta attraverso un nuovo elenco dei contenuti, che ha restituito un risultato vuoto, attestando la corretta eliminazione della directory *amici*.

Questo passaggio dimostra la differenza operativa tra i comandi di rimozione semplice e quelli ricorsivi.

## RIMOZIONE DIRECTORY PRINCIPALI

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

(kali㉿kali)-[~]
$ ls -l

total 44
drwxr-xr-x 2 kali kali 4096 Oct 20 15:26 Desktop
drwxr-xr-x 2 kali kali 4096 Oct 18 11:01 Documents
drwxrwxr-x 2 kali kali 4096 Oct 21 08:53 dos
drwxr-xr-x 2 kali kali 4096 Oct 20 15:22 Downloads
drwxr-xr-x 2 kali kali 4096 Oct 18 11:01 Music
drwxr-xr-x 2 kali kali 4096 Oct 18 11:01 Pictures
drwxr-xr-x 2 kali kali 4096 Oct 18 11:01 Public
drwxrwxr-x 5 kali kali 4096 Oct 21 08:58 studenti
drwxr-xr-x 2 kali kali 4096 Oct 18 11:01 Templates
drwxr-xr-x 2 kali kali 4096 Oct 18 11:01 Videos
drwxrwxr-x 2 kali kali 4096 Oct 21 08:53 windows

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

(kali㉿kali)-[~]
$ ls -l

total 32
drwxr-xr-x 2 kali kali 4096 Oct 20 15:26 Desktop
drwxr-xr-x 2 kali kali 4096 Oct 18 11:01 Documents
drwxr-xr-x 2 kali kali 4096 Oct 20 15:22 Downloads
drwxr-xr-x 2 kali kali 4096 Oct 18 11:01 Music
drwxr-xr-x 2 kali kali 4096 Oct 18 11:01 Pictures
drwxr-xr-x 2 kali kali 4096 Oct 18 11:01 Public
drwxr-xr-x 2 kali kali 4096 Oct 18 11:01 Templates
drwxr-xr-x 2 kali kali 4096 Oct 18 11:01 Videos
```

Nel passaggio conclusivo dell'attività è stata eseguita la rimozione delle directory principali create all'inizio dell'esercizio, ovvero *dos*, *studenti* e *windows*.

Lo scopo era ripristinare lo stato iniziale dell'ambiente di lavoro, verificando al contempo l'applicazione del comando di eliminazione ricorsiva su più cartelle in un'unica operazione.

Dopo essersi spostati nella directory *home* e aver elencato i contenuti per individuare le directory da eliminare, è stato utilizzato un comando di rimozione con opzione ricorsiva, che consente di cancellare sia le directory sia i loro eventuali contenuti.

L'output successivo alla nuova visualizzazione del contenuto della Home ha confermato la scomparsa delle directory *dos*, *studenti* e *windows*, a testimonianza della corretta esecuzione del comando.

Questa fase conclude l'esercitazione, garantendo che tutte le operazioni di creazione, modifica e cancellazione di file e directory siano state comprese e applicate in modo efficace.

## CONCLUSIONE

L'esercitazione ha consentito di approfondire in modo pratico la gestione di file e directory in ambiente Linux attraverso l'uso dei principali comandi della shell.

Ogni fase ha illustrato un aspetto fondamentale dell'amministrazione del file system, dalla creazione e organizzazione gerarchica delle directory, fino alla gestione dei permessi e alla rimozione controllata di cartelle e documenti.

Le attività hanno permesso di consolidare la comprensione delle operazioni di base fornendo una visione chiara delle logiche di sicurezza e struttura che caratterizzano i sistemi Linux.

Nel complesso, l'esercitazione ha mostrato come un corretto utilizzo dei comandi della shell permetta di mantenere un ambiente di lavoro ordinato, efficiente e sicuro, ponendo le basi per una gestione più avanzata dei processi e delle risorse del sistema operativo.