

Gestione dei Permessi di Lettura, Scrittura ed Esecuzione in Linux.

Obiettivo: Configurare e gestire i permessi di lettura, scrittura ed esecuzione per file o directory in un sistema Linux.

Logica della creazione della Directory e del File: L'esercizio è stato eseguito creando una cartella col nome di **plansoftoday** che avrebbe l'intento di contenere al suo interno tutti i piani di lavoro giornalieri. L'idea è quella di simulare uno spazio comune dove più utenti possono consultare i piani (quindi leggere il file), ma solo il proprietario può apportare modifiche di scrittura. All'interno della directory ho creato un file chiamato **plans.txt**, destinato a contenere i piani della giornata in questione.

1) Creazione della Directory e modifica dei permessi.

Per prima cosa, è stata creata la cartella con l'utilizzo del comando **mkdir**.

```
(kalivm@vboxkalivm)-[~]  
$ mkdir plansoftoday
```

I permessi della directory sono stati visualizzati prima della modifica. In questo caso:

```
drwxrwxr-x  2 kalivm kalivm  4096  3 giu 17.58 plansoftoday
```

Dove;

D indica che è una directory,

Rwx dove **r** = read, **w** = write, **x** = execute

Ciò indica che il proprietario può leggere, scrivere e eseguire il file, quindi entrare nella cartella.

Stessa cosa per gli utenti del gruppo.

Tutti gli altri utenti possono solo leggere la directory.

Sono stati quindi cambiati i permessi, in modo da permettere le modifiche alla directory solo al proprietario, il resto degli utenti, inclusi quelli del gruppo, avrebbero potuto solo visualizzarne il contenuto.

```
(kalivm@vboxkalivm)-[~/plansoftoday]
$ cd ..

(kalivm@vboxkalivm)-[~]
$ chmod 755 plansoftoday
```

Il comando utilizzato è stato **chmod**, il quale permette di cambiare i privilegi di un file.

Il comando è seguito dal valore **755**, il quale indica la rappresentazione dei permessi in valore ottale, dove:

R ha come valore 4, write 2 e execute 1.

7 è la somma di tutti i valori, quindi il proprietario può leggere, scrivere e eseguire il file, mentre il resto degli utenti hanno come valore assegnato 5, la somma di read e execute, possono quindi leggerlo ed eseguirlo, senza poterlo modificare.

```
drwxr-xr-x 2 kalivm kalivm 60 3 giu 16.33 plansoftoday
```

A questo punto infatti, i permessi sono stati aggiornati correttamente.

2) Creazione del file plans.txt e modifica dei permessi.

Successivamente, sono stati effettuati gli stessi passaggi per il file di testo.

E' stato creato con l'editor **nano**, e successivamente sono stati modificati i permessi con l'utilizzo del comando **chmod**.

```

(kalivm@vboxkalivm)-[~]
$ cd plansoftoday

(kalivm@vboxkalivm)-[~/plansoftoday]
$ nano plans.txt

(kalivm@vboxkalivm)-[~/plansoftoday]
$ ls -l
totale 4
-rw-rw-r-- 1 kalivm kalivm 27  3 giu 16.07 plans.txt

```

Anche qui, prima ancora di modificare i permessi, si può vedere che il gruppo ha permessi di scrittura del file.

```

(kalivm@vboxkalivm)-[~]
$ cd plansoftoday

(kalivm@vboxkalivm)-[~/plansoftoday]
$ chmod 644 plans.txt

(kalivm@vboxkalivm)-[~/plansoftoday]
$ ls -l
totale 4
-rw-r--r-- 1 kalivm kalivm 27  3 giu 16.07 plans.txt

```

Questa volta, **chmod** è seguito dal valore **644**, in quanto il file è un file di testo, quindi non eseguibile. Il valore 6 indica che il proprietario può leggere e scrivere nel file, mentre il resto degli utenti possono solo leggerlo.

3) Test di verifica.

Per assicurarci che quanto detto prima sia perfettamente funzionante alla pratica, è stata spostata la directory con il comando **mv** in un'altro percorso di cartelle pubblico, al fine di essere raggiungibile dagli utenti stessi.

```

(kalivm@vboxkalivm)-[~]
$ mv ~/plansoftoday /tmp/

```

Verificandone di nuovo quindi, i permessi applicati sopra.

```
drwxr-xr-x 2 kalivm kalivm 60  3 giu 16.33 plansoftoday
```

```
(kalivm@vboxkalivm)-[/tmp/plansoftoday]
$ ls -l
totale 4
-rw-r--r-- 1 kalivm kalivm 27  3 giu 16.33 plans.txt
```

E, successivamente, è stato creato un nuovo utente con l'utilizzo del comando **adduser**, in questo caso rinominato impiegato1 per praticità.

```
(kalivm@vboxkalivm)-[~]
$ sudo adduser impiegato1
```

A questo punto, non ci resta che verificare.

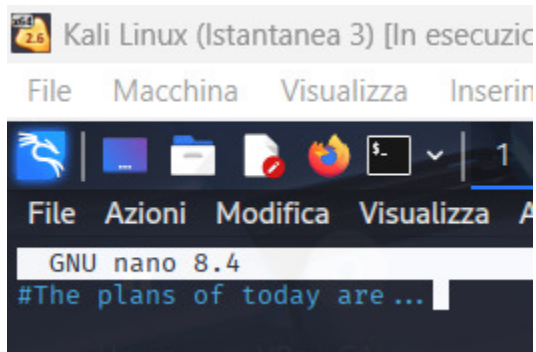
Con l'utilizzo di **su**, cambiamo utente:

```
(kalivm@vboxkalivm)-[/tmp/plansoftoday]
$ su impiegato1
Password:
(impiegato1@vboxkalivm)-[/tmp/plansoftoday]
$ ls -l
totale 4
-rw-r--r-- 1 kalivm kalivm 27  3 giu 16.33 plans.txt
```

Con il comando **ls -l** possiamo confermare quanto detto prima, abbiamo avuto accesso alla cartella.

Passiamo al file:

E' stato possibile visualizzarne il contenuto,



Ma...al momento della scrittura:



Tutto ciò conferma quanto detto prima.

Solo il proprietario può scrivere nella cartella e nel file, mentre tutti gli altri utenti, possono visualizzarne i contenuti come previsto.