

Esercizio 2

2023-04-28

1) Visualizzare il numero dei processi attivi sulla vostra macchina.

(Suggerimento: Il comando `ps` opportunamente lanciato mostra a video tutti i processi presenti sul sistema, elencandoli ciascuno su una linea.)

```
pierpaolo@Patricio:~$ ps -e | wc -l
```

2) Creare un file contenente il numero di utenti che utilizzano la shell `/bin/bash`.

(Suggerimento: Il file `/etc/passwd` memorizza le informazioni relative agli account utente ad esempio user ID, group ID, home directory, shell utilizzata, etc.)

```
pierpaolo@Patricio:~$ cat /etc/passwd | grep "/bin/bash" | wc -l > file.txt
```

3) Visualizzare username e home directory degli utenti che hanno associata la shell `/bin/bash`.

(Suggerimento: Il file `/etc/passwd` memorizza le informazioni relative agli account utente. Il comando `cut` consente di estrarre da file o standard input delle sezioni di testo. Può essere usato per estrarre campi di testo definiti mediante un separatore.)

```
pierpaolo@Patricio:~$ cat /etc/passwd | grep "/bin/bash" | cut -d ":" -f 1,6
```

4) Creare un file contenente un estratto (dalla 20-esima alla 30-esima riga) del file /etc/passwd.

```
pierpaolo@Patricio:~$ cat /etc/passwd | head -30 | tail -10 > file.txt
```

5) Creare un file elenco.txt contenente l'elenco (in ordine inverso) di tutte le entries del file /etc/passwd in cui compare la parola home utilizzando sort

```
pierpaolo@Patricio:~$ cat /etc/passwd | grep "home" | sort -r > elenco.txt
```

oppure

```
pierpaolo@Patricio:~$ cat /etc/passwd | grep "home" | tac > elenco.txt
```

6) Creare tre file chiamati file-1.txt, file-2.txt, file-3.txt usando il seguente comando shell (da impartire tutto su un rigo):

```
pierpaolo@Patricio:~$ for F in {1..3}; do for N in {1..100}; do od -An -N1 -i /dev/urandom | tr
```

7) A partire dai file generati (file-1.txt, file-2.txt, file-3.txt) stampare su un nuovo file (file.txt) l'elenco ordinato di ciascuna linea contenuta nei tre file, avendo cura di eliminare le righe duplicate. Si usi l'ordinamento numerico (sort -n)

```
pierpaolo@Patricio:~$ cat file-1.txt file-2.txt file-3.txt | sort -n | uniq > file.txt
```

8) Creare un file contenente l'elenco degli ultimi 4 comandi eseguiti, dove questi 4 comandi devono comparire ordinati in base al nome del comando e non all'ordine di esecuzione

(Suggerimento: il comando sort consente di definire una chiave di ordinamento su una porzione della linea.)

```
pierpaolo@Patricio:~$ history | tail -4 | sort -k 2 > file.txt
```

9) Creare un file contenente il numero di file presenti nella /home di dimensione superiore a 2M.

(Suggerimento: utilizzare opportunamente il comando find.)

```
pierpaolo@Patricio:~$ find /home -size +2M | wc -l > file.txt
```

10) Cercare tutti i file di backup presenti nella home più vecchi di 4 giorni e rimuoverli.

Il comando find ricerca file e directory che soddisfano i criteri specificati tramite direttive. Tra le varie direttive, ad esempio, è possibile specificare il nome o il numero di giorni trascorsi dalla data di creazione rispetto a quella odierna. Infine, l'opzione -exec consente di eseguire il comando specificato subito dopo con eventuali parametri. Ad esempio, "exec COMANDO ;" esegue COMANDO su ogni file verificato da find. La sintassi del comando termina con ';'.

```
pierpaolo@Patricio:~$ find /home -name "*~" -mtime +4 -exec rm {} \;
```

10) Come funziona il comando usato per creare i tre file file-1.txt, file-2.txt, file-3.txt? A che serve il comando od? Che cos'è /dev/urandom? A cosa serve il comando "tr"?

```
for F in {1..3}; do
    for N in {1..100}; do
        od -An -N1 -i /dev/urandom | tr -d " \t" >> file-$F.txt;
    done
done
```

- Il ciclo esterno for F in {1..3} si esegue tre volte, una per ogni valore numerico (1, 2 e 3) contenuto nell'elenco racchiuso tra le parentesi graffe. In ogni iterazione, la variabile F assume uno dei tre valori numerici.
- Il ciclo interno for N in {1..100} si esegue 100 volte per ogni iterazione del ciclo esterno, generando quindi un totale di 300 numeri casuali (100 per ogni file). In ogni iterazione, la variabile N assume uno dei 100 valori numerici.
- Il comando od -An -N1 -i /dev/urandom viene utilizzato per generare un numero casuale tra 0 e 255 in formato decimale, e lo scrive nella riga corrente del file di output.
- /dev/urandom è un file speciale di dispositivo di input/output che fornisce un flusso di byte pseudo-casuali, ovvero dati casuali generati da un generatore di numeri pseudo-casuali nel kernel di Linux.

Al contrario di /dev/random, /dev/urandom fornisce dati casuali di "bassa qualità" ma non si blocca se il pool di entropia del kernel è esaurito.

/dev/urandom è preferibile a /dev/random per la maggior parte degli scopi.

- Il comando tr -d " \t" viene utilizzato per eliminare eventuali spazi o tabulazioni presenti nella riga appena generata dal comando od. (ogni carattere inserito viene considerato come una "singola entry", e.g. verranno eliminati tutti gli spazi regardless della loro posizione nella stringa o dei caratteri loro vicini)