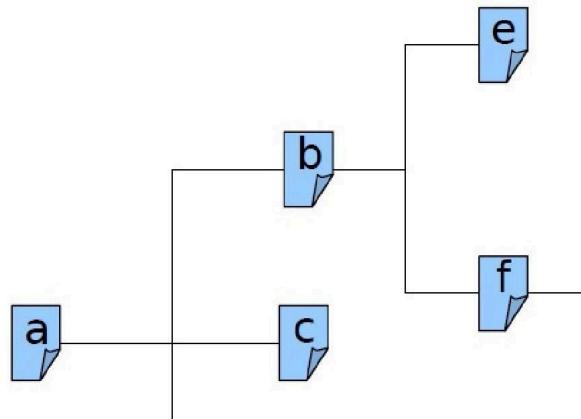


Esercitazione Shell Linux

N.B. Se usi il sottosistema WSL, assicurati di eseguire l'esercizio in una cartella nativa Linux e non NTFS (Windows). Per esempio puoi partire dalla tua cartella home con il comando `cd ~`

Esercizio 1

Costruire il seguente albero di cartelle:



- Creare nella cartella 'a' i file: (utilizzare il comando 'touch' dopo averne letto la documentazione)

- file1.txt
- file2.txt
- file3.tmt
- xfile.txt
- doc1.doc
- doc2.doc
- doc3.doc
- xdoc.doc

- Scegliere 'a' come cartella corrente;
- Creare una copia di 'file1.txt' con nome 'copiafile1.txt' all'interno di 'a';
- Copiare 'file1.txt' e 'copiafile1.txt' nella cartella 'b/e' utilizzando il path assoluto di 'e';
- Rinominare la copia di file1.txt presente in 'e' usando il path relativo di 'file1.txt' partendo da 'a'. Il nuovo file dovrà chiamarsi 'rinominato.txt';
- Spostare tutti i file in 'a' con estensione 'txt' in 'b';
- Copiare tutti i file nella cartella 'a' che contengono la sottostringa 'doc' a partire dal secondo carattere, nella cartella 'c'. ad esempio, devono essere copiati file con nome **p**documento.gif, **x**documento.txt, **_doc**, ecc.
- Spostare tutti i file nella cartella 'a' che iniziano per 'd' nella cartella 'f'
- Scaricare nella cartella 'a' la home page di 'google': (utilizzare il comando `wget` dopo averne letto la documentazione)

- Modificare i diritti di accesso alle cartelle in modo che:

- chiunque possa leggere e scrivere nella cartella a;

- others non possa vedere il contenuto di c;
- others possa entrare in f ma non in e;
- others abbia il diritto di lettura su b, mentre group abbia tutti i diritti;
- Rimuovere la cartella 'f' con tutto il suo contenuto

Esercizio 2

- 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.

- 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.) .

- 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*.

- Creare un file contenente un estratto (dalla 20-esima alla 30-esima riga) del file `/etc/passwd`.
- 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*.
- Creare tre file chiamati `file-1.txt`, `file-2.txt`, `file-3.txt` usando il seguente comando shell (da impartire tutto su un rigo):

```
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
```

- 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`)
- 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.
- Creare un file contenente il numero di file presenti nella `/home` di dimensione superiore a 2M. *Suggerimento:* utilizzare opportunamente il comando `find`.
- 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 *';*’.

- 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*”?