

ESERCITAZIONE DI VENERDÌ 26/03/2020

CAT

- 1) Usando la ridirezione dello standard output del filtro cat, creare un file di nome prova (inserire almeno 5-6 linee con più parole per linea).
- 2) Usando la ridirezione dello standard output del comando pwd e del comando ls -l, aggiungere tale output al file prova.
- 3) Usando il comando cat con le opportune ridirezioni, creare una copia del file prova e dargli nome p.txt.
- 4) Usando più volte la ridirezione dello standard output in append aggiungere il contenuto del file prova al file p.txt almeno altre 5 volte in modo da avere un contenuto di molte linee.

MORE

- 5) Usando la ridirezione dello standard input e il filtro more, visualizzare il contenuto del file p.txt.
- 6) Verificare il comportamento del comando more, usando il comando more p*.

SORT

- 7) Usando la ridirezione dello standard input e il filtro sort, verificare se il file di nome prova è ordinato alfabeticamente.
- 8) Usando la ridirezione dello standard input e il filtro sort, mostrare il contenuto del file prova ordinato secondo il normale ordinamento alfabetico.
- 9) Stessa cosa di 8 ma invertendo l'ordinamento.
- 10) Stessa cosa di 8, ma ignorando la differenza fra maiuscole e minuscole.
- 11) Stessa cosa di 8, ma ridirigendo lo standard output su un file di nome ordinato.
- 12) Stessa cosa di 7, ma sul file di nome ordinato.
- 13) Editare il file prova andando a duplicare e/o triplicare alcune linee.
- 14) Stessa cosa di 8, ma attuando l'ordinamento alfabetico senza replicazioni e scrivendo lo standard output su un file di nome ordinato-senza-doppi.

GREP

- 15) Usando la ridirezione dello standard input e il filtro grep, cercare le linee che contengono una certa stringa (o anche un semplice carattere) nel file ordinato-senza-doppi.
- 16) Stessa cosa di 15, ma mostrando i numeri di linea.
- 17) Assicurandosi di avere nel file ordinato-senza-doppi la stessa stringa scritta in maiuscola e in maiuscolo almeno 2-3 volte, stessa cosa di 15, ma cercando la stringa ignorando maiuscole/minuscole.
- 18) Stessa cosa di 15, ma cercando le linee che NON contengono una certa stringa (o anche un semplice carattere).
- 19) Stessa cosa di 15, ma cercando solo le linee che INIZIANO per una certa stringa (o anche un semplice carattere).
- 20) Stessa cosa di 15, ma cercando solo le linee che TERMINANO per una certa stringa (o anche un semplice carattere).
- 21) Stessa cosa di 15, ma cercando solo le linee che TERMINANO per il carattere '.' (PUNTO), assicurandosi di averne almeno UNA
- 22) Utilizzando la soluzione di uno degli esercizi fra 15 e 21, ridirigere lo standard output in un file di nome prova-grep.

WC

- 23) Usando la ridirezione dello standard input e il filtro wc, contare le linee del file p.txt (vedi esercizio 4).
- 24) Stessa cosa di 23, ma contando i caratteri.
- 25) Stessa cosa di 23, ma contando le parole.
- 26) Usando il comando wc su un file di nome pippo (che non esista) e ridirigendo lo standard error su /dev/null, verificare il valore di ritorno del comando.
- 27) Utilizzando il comando ps in piping con tee temp e in piping con wc -l verificare: a) il numero visualizzato; b) il contenuto del file di nome temp creato dal filtro tee.

HEAD e TAIL

- 28) Usando la ridirezione dello standard input e il filtro head, selezionare le prime 10 linee del file p.txt (vedi esercizio 4).
- 29) Stessa cosa di 28, ma selezionando la prima linea.

- 30) Stessa cosa di 28, ma selezionando le prime 3 linee.
- 31) Usando la ridirezione dello standard input e il filtro tail, selezionare le ultime 10 linee del file p.txt.
- 32) Stessa cosa di 31, ma selezionando l'ultima linea.
- 33) Stessa cosa di 31, ma selezionando le ultime 3 linee.
- 34) Utilizzando la soluzione di uno degli esercizi fra 28 e 33, ridirigere lo standard output in un file di nome prova-head o prova-tail a seconda dei casi.
- 35) Utilizzando il piping dei comandi, isolare in un file di nome p.txt.terza la terza linea a partire dall'inizio del file p.txt.
- 36) Utilizzando il piping dei comandi, isolare in un file di nome p.txt.terzultima la terza linea a partire dalla fine del file p.txt.

ESECUZIONE IN BACKGROUND

- 37) Mandare in esecuzione con le opportune ridirezioni di standard output e di standard error il comando ls -lR / usando l'esecuzione in background.
- 38) Verificare i processi attivi nella sessione prima che finisca il comando lanciato in background.

DATE

- 39) Verificare la data (e l'ora corrente).

DIFF

- 40) Creare con l'editor vi (oppure con la ridirezione dello standard output) un file che contenga almeno 5 o 6 linee; crearne una copia e quindi modificare con l'editor vi alcune parti; quindi, verificare il funzionamento del comando diff utilizzando i due file prodotti.

VARIABILI

- 41) Memorizzare in una variabile di nome x, il numero di linee del file p.txt.
- 42) Memorizzare in una variabile di nome y, il numero di linee del file prova.
- 43) Memorizzare in una variabile di nome z, la somma della variabile x e della variabile y e visualizzarne il valore.
- 44) Con l'editor vi, scrivere un file comandi prova-var.sh che visualizzi il valore delle variabili x, y e z inserendo anche dei commenti significativi. Rendere eseguibile tale file comandi (verificare che sia eseguibile con ls -l prova-var.sh) e mandarlo in esecuzione. Quale è il risultato?
- 45) Rendere la variabile z una variabile di ambiente e riprovare ad eseguire prov-var.sh. Quale è il risultato?
- 46) Verificare con il comando env, la presenza di z nell'ambiente.
- 47) Copiare il file comandi prova-var.sh nel file comandi di nome prova-var-bis.sh, quindi aggiungere in fondo un comando che modifica il valore della variabile z e lo visualizza nuovamente, sempre inserendo commenti significativi. Provare ad eseguire prova-var-bis.sh. Quale è il risultato? Quale valore ha la variabile z nella shell interattiva di partenza?

METACARATTERI

- 48) Utilizzando la ridirezione a vuoto, creare diversi file con nomi che iniziano e terminano con varie lettere dell'alfabeto maiuscole e minuscole e con numeri; creare anche un paio di directory. Verificare quindi il comportamento dei comandi:
 - a) echo [a-z]*
 - b) echo [A-Z]*
 - c) echo [0-9]*
 - d) echo *[a-z]
 - e) echo *[A-Z]
 - f) echo *[0-9]

Rifare i comandi precedenti usando la negazione: ad esempio per a) echo [!a-z]*

Utilizzare nuovamente i pattern precedenti utilizzando però il comando ls -l: che cosa cambia?