Assignment 05

linux 0s & scripting – b keerthana

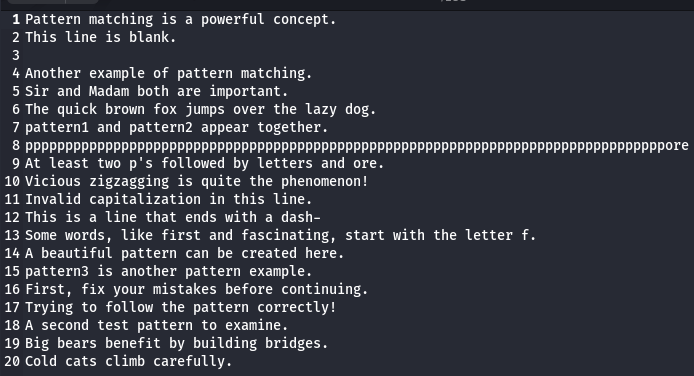
AMRUTHESH

241059041

M.E – Cyber Security

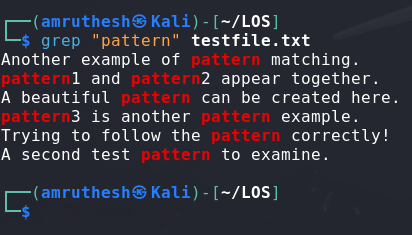
MSIS, MANIPAL

**Test file**

****

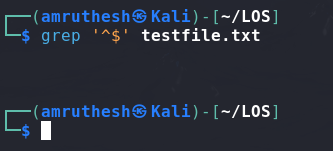
### 1. Print all the lines having the word "pattern":

grep "pattern" testfile.txt



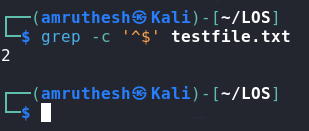
### 2. Pick out the blank lines in the file:

grep '^$' testfile.txt



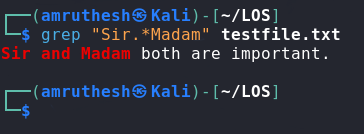
### 3. Count the total number of empty lines in the file:

grep -c '^$' testfile.txt



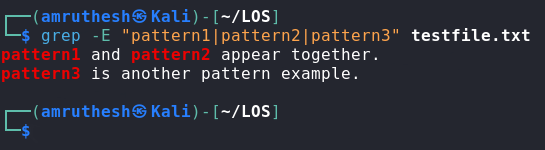
### 4. Print the line which has both "Sir and Madam":

grep "Sir.\*Madam" testfile.txt



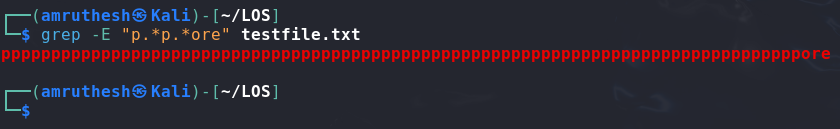
### 5. Pick out lines with "pattern1", "pattern2", or "pattern3" (use the alternator |):

grep -E "pattern1|pattern2|pattern3" testfile.txt



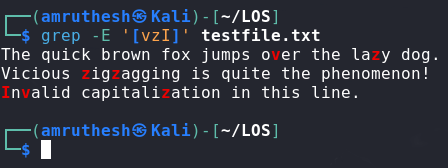
### 6. Pick out lines that have at least two p's followed by any number of letters followed by 'ore':

grep -E "p.\*p.\*ore" testfile.txt



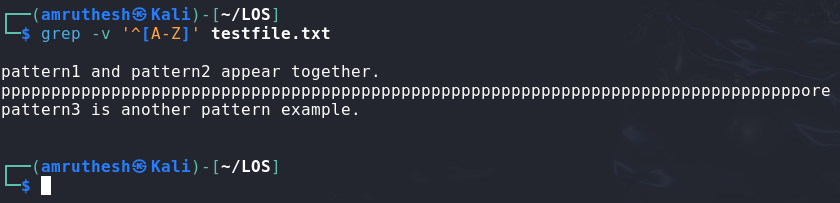
### 7. Pick out all the lines with v, z, or I in them:

grep -E '[vzI]' testfile.txt



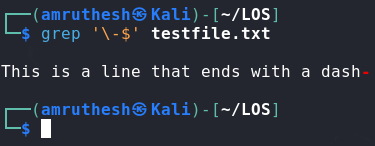
### 8. Pick out all the lines that do not start with an uppercase letter:

grep -v '^[A-Z]' testfile.txt



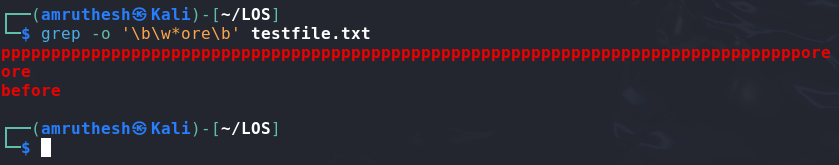
### 9. Pick out all the lines that end with a dash -:

grep '\-$' testfile.txt



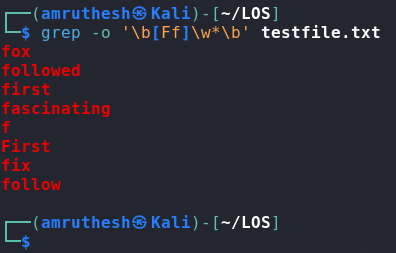
### 10. Pick out all the words that end with "ore":

grep -o '\b\w\*ore\b' testfile.txt



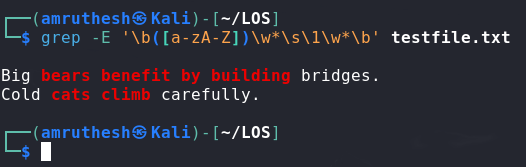
### 11. Pick out all the words that start with "f" or "F":

grep -o '\b[Ff]\w\*\b' testfile.txt



### 12. Pick out lines that use first letter alliteration (starting two words with the same letter):

grep -E '\b([a-zA-Z])\w\*\s\1\w\*\b' testfile.txt



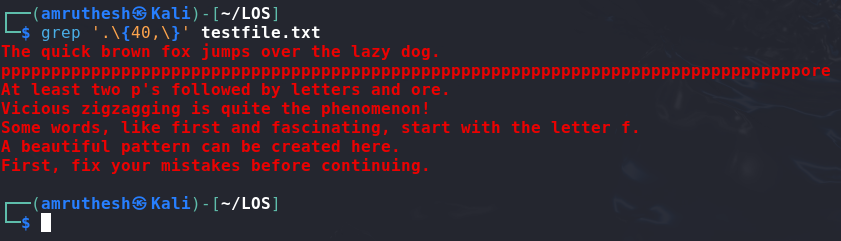
### 13. Determine how many times the word "pattern" occurs:

grep -o 'pattern' testfile.txt | wc -l



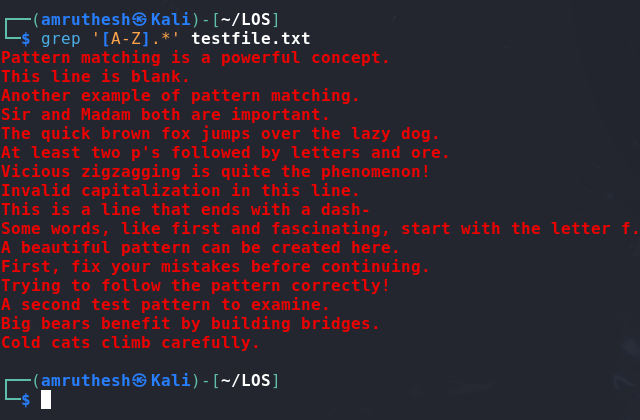
### 14. Pick out lines with at least 40 characters:

grep '.\{40,\}' testfile.txt



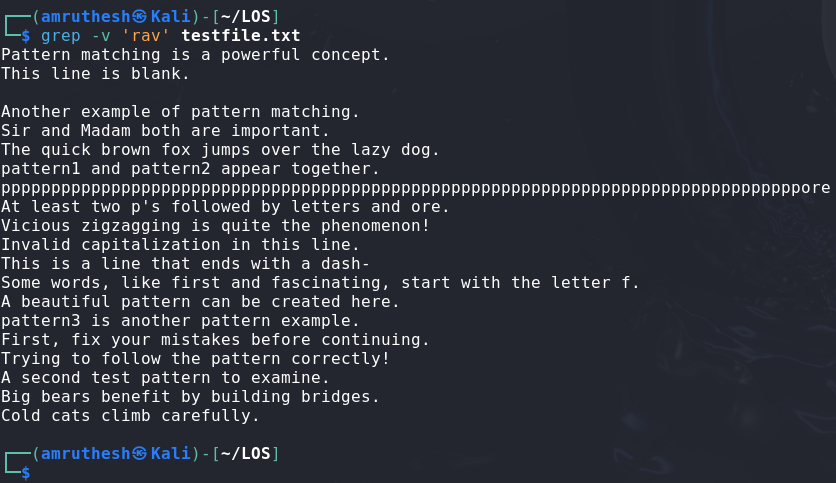
### 15. Pick out lines with an uppercase letter other than the first character:

grep '[A-Z].\*' testfile.txt



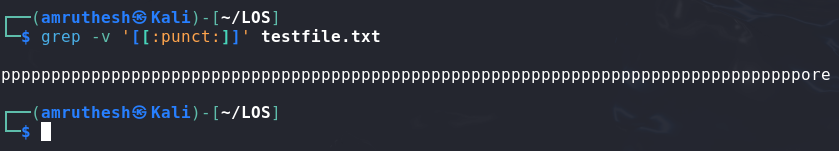
### 16. To pick out lines without "rav":

grep -v 'rav' testfile.txt



### 17. Pick out lines with no punctuation:

grep -v '[[:punct:]]' testfile.txt



### 18. Shell script to generate a report with the following details:

### - Number of regular files

### - Number of links

### - Number of directories

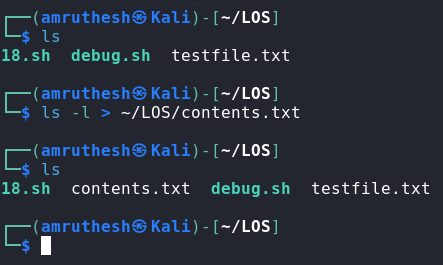
### - Print the date when it was processed!

### 

### 

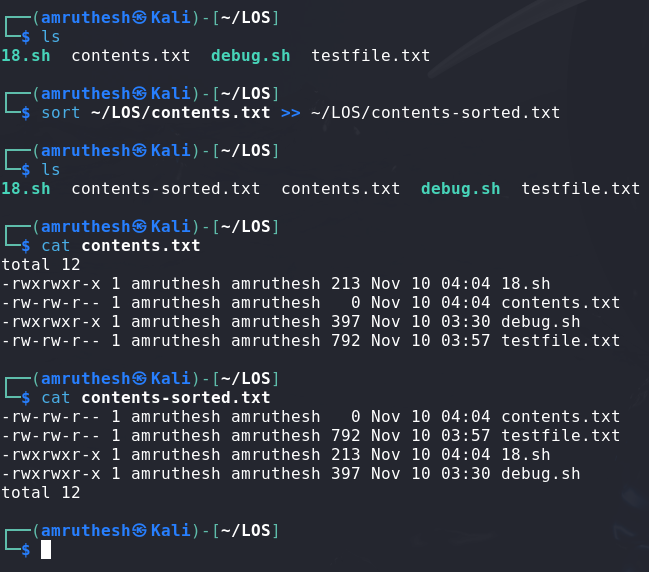
### 19. List the contents of your current directory, including ownership and permissions, and store the output to a file called contents.txt within your home directory:

ls -l > ~/contents.txt



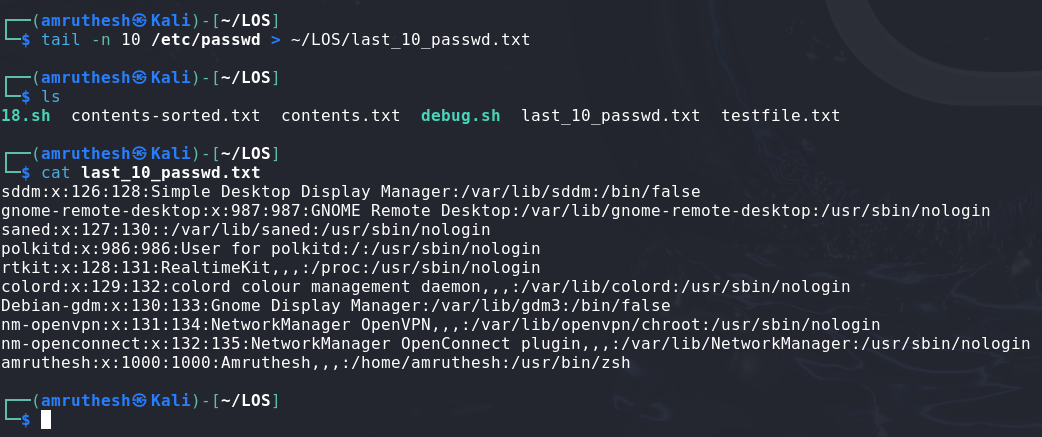
### 20. Sort the contents of contents.txt and append it to a new file contents-sorted.txt:

sort ~/contents.txt >> ~/contents-sorted.txt



### 21. Display the last 10 lines of the /etc/passwd file and redirect it to a new file in your user's Documents directory:

tail -n 10 /etc/passwd > ~/Documents/last\_10\_passwd.txt



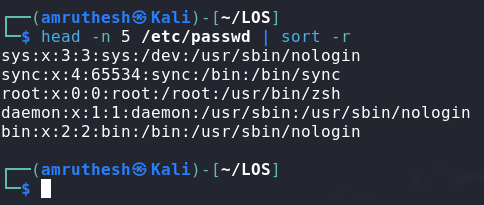
### 22. Count the number of words within contents.txt and append the output to the end of a file field2.txt in your home directory:

wc -w ~/contents.txt >> ~/field2.txt



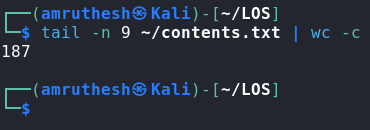
### 23. Display the first 5 lines of the /etc/passwd file and sort the output reverse alphabetically:

head -n 5 /etc/passwd | sort -r



### 24. Using the previously created contents.txt file, count the number of characters of the last 9 lines:

tail -n 9 ~/contents.txt | wc -c



### 25. Debug the script (fix the errors):

### #fix the error

### /!bin/bash

### fruit1 = Apples

### fruit2 = Oranges

### if [ $1 -lt $# ]

### then

### echo "This is like comparing $fruit1 and $fruit2!"

### elif test [$1 -gt $2 ]

### then

### echo '$fruit1 win!'

### else

### echo "Fruit2 win!"

### done

#!/bin/bash

fruit1="Apples"

fruit2="Oranges"

if [ $1 -lt $# ]; then

echo "This is like comparing $fruit1 and $fruit2!"

elif [ $1 -gt $2 ]; then

echo "$fruit1 win!"

else

echo "Fruit2 win!"

fi