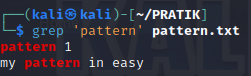
More on Grep

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



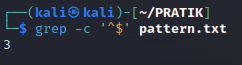
*  This searches for the exact word "pattern" in the file.

1. Pick out the blank lines in the file



*  ^$ matches lines that start and end with nothing, which are blank lines.

1. Count total number of empty lines in the file.



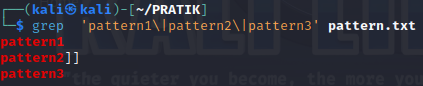
*  -c counts the number of matching lines.

1. Print the line which have both “Sir and Madam”.



*  This uses a pipe (|) to combine two grep commands, ensuring both words are present in a line.

1. pick out lines with “pattern1” “pattern2” or “pattern3”. (use the alternator |)



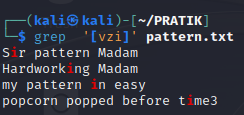
 The | symbol acts as an OR operator between the patterns.

1. pick out lines that have at least two p's followed by any number of letters followed by 'ore'. The p's do not have to be next to each other.



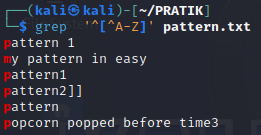
.\* allows any characters between the p and ore.

1. pick out all the lines with v, z or I in them



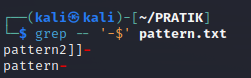
*  This matches any line containing one of the specified characters.

1. pick out all the lines that do not start with an uppercase letter.



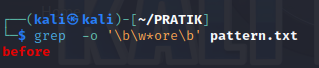
*  [^A-Z] matches anything that is not an uppercase letter.

1. pick out all the lines that end with a dash –

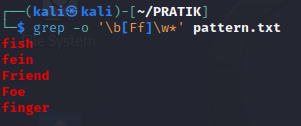


*  -$ matches lines ending with a dash.

10. pick out all the words that end with ore



*  \b marks word boundaries, and \w\*ore matches any word ending with "ore".
  1. pick out all the words that start with f or F



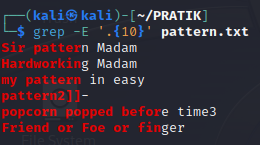
*  [Ff] matches f or F.
  1. pick out lines that uses first letter alliteration - starting two words with the same letter.



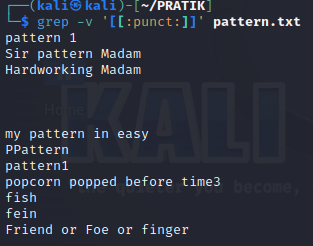
*  This uses a backreference \1 to match two words starting with the same letter.
  1. determine how many times contains the word "pattern".



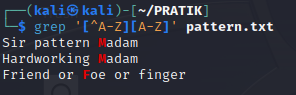
*  This counts occurrences of "pattern" in the file.
  1. to pick out lines with at least 10 characters:



*  .{10,} matches any line with 40 or more characters.
  1. to pick out lines with no punctuation



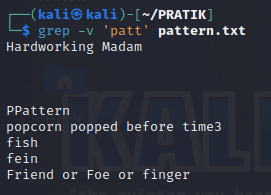
*  -v inverts the match, selecting lines without punctuation.
  1. to pick out lines with an uppercase letter other than the first character. (The first character on the line does not count.)



*  This matches lines with an uppercase letter not at the beginning.

17. To pick out lines without rav

Quotes:



* + -v selects lines that do not contain "patt".

18. Write a shell script to generate a report with the following details.

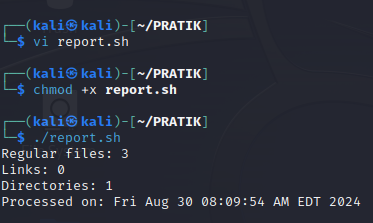
- Number of regular files

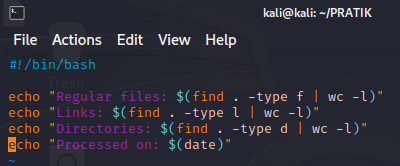
- Number of links

- Number of directories

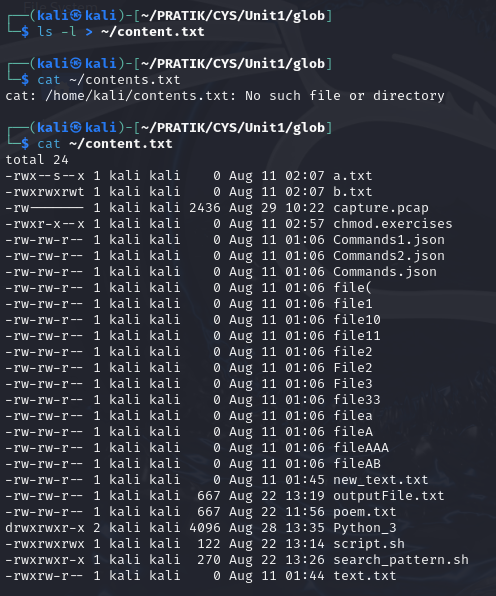
- Print the date when it was processed!

Redirection

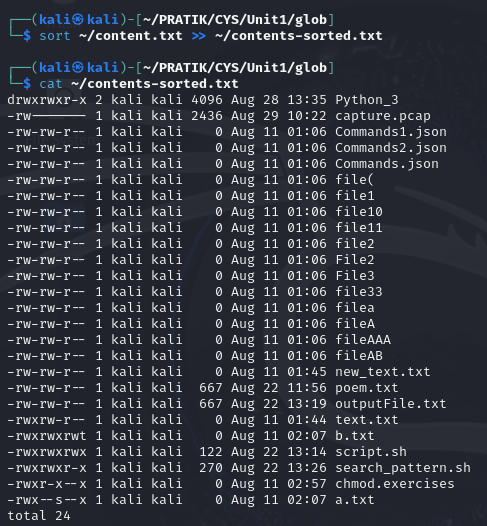




19. List the contents of your current directory, including the ownership and permissions, and store the output to a file called contents.txt within your home directory.



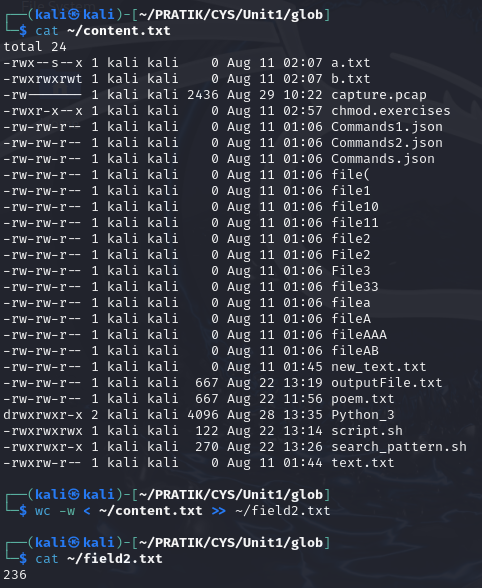
20. Sort the contents of the contents.txt file from your current directory and append it to the end of a new file named contents-sorted.txt.



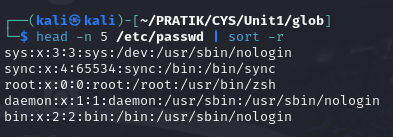
21. Display the last 10 lines of the /etc/passwd file and redirect it to a new file in the your user’s Documents directory.



22. Count the number of words within the contents.txt file and append the output to the end of a file field2.txt in your home directory. You will need to use both input and output redirection.



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



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

Debug



25. Debug the script 1\_debug.sh

1. #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

