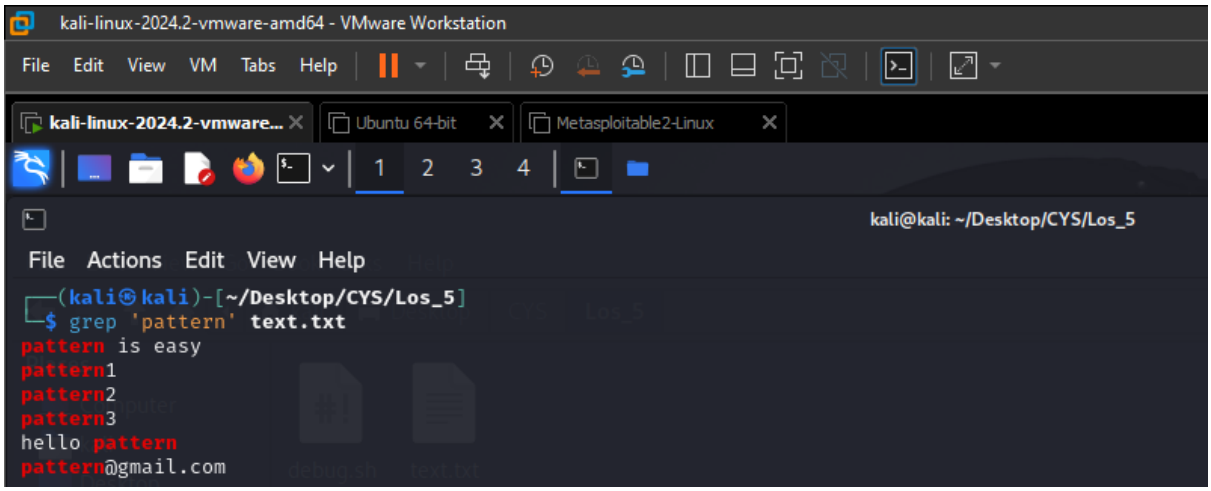
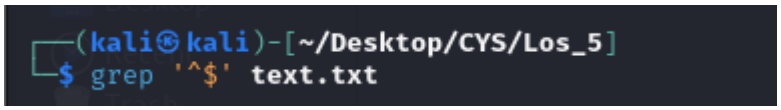
More on Grep

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



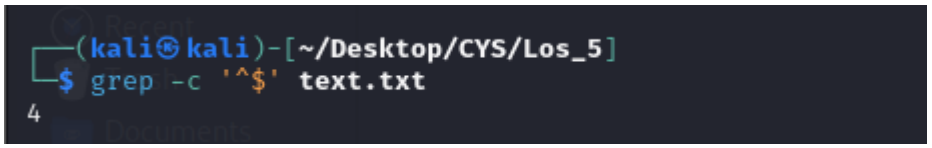
- This searches for the exact word "pattern" in the file.

2. Pick out the blank lines in the file



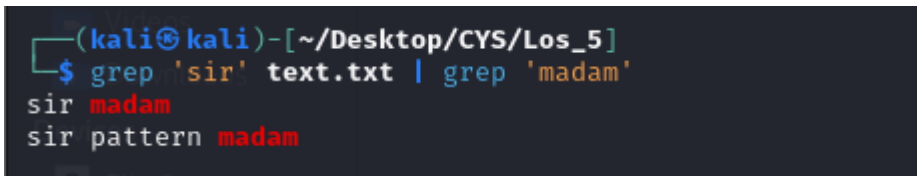
- ^$ matches lines that start and end with nothing, which are blank lines.

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



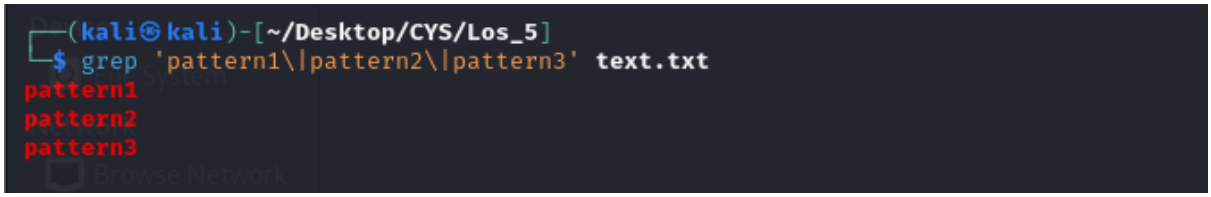
- · -c counts the number of matching lines.

4. 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.

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



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

6. 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.

```
┌──(kali㉿kali)-[~/Desktop/CYS/Los_5]
└─$ grep 'p.*p.*ore' text.txt
Popcorn popped before

┌──(kali㉿kali)-[~/Desktop/CYS/Los_5]
└─$ grep 'P.*p.*ore' text.txt
Popcorn popped before
```

.* allows any characters between the p and ore.

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

```
┌──(kali㉿kali)-[~]
└─$ cd Desktop/CYS/Los_5

┌──(kali㉿kali)-[~/Desktop/CYS/Los_5]
└─$ grep '[vzi]' text.txt
pattern is easy
good Morning!
geood Evening
pattern@gmail.com
sir madam
sir
sir pattern madam

┌──(kali㉿kali)-[~/Desktop/CYS/Los_5]
└─$
```

- · This matches any line containing one of the specified characters.

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

```
┌──(kali㉿kali)-[~/Desktop/CYS/Los_5]
└─$ grep '^[^A-Z]' text.txt
pattern is easy
pattern1
pattern2
pattern3
hello pattern
good Morning!
geood Evening
pattern@gmail.com
sir madam
sir
madam
sir pattern madam
```

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

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

```
┌──(kali㉿kali)-[~/Desktop/CYS/Los_5]
└─$ grep -- '-$' text.txt
--
hello pattern-
sir madam-
```

- · -$ matches lines ending with a dash.

10. pick out all the words that end with ore

```
┌──(kali㉿kali)-[~/Desktop/CYS/Los_5]
└─$ grep -o '\b\w*ore\b' text.txt
before
```

- · \b marks word boundaries, and \w*ore matches any word ending with "ore".

11. pick out all the words that start with f or F

```
┌──(kali㉿kali)-[~/Desktop/CYS/Los_5]
└─$ grep -o '\b[fF]\w*' text.txt
fun
fish
for

┌──(kali㉿kali)-[~/Desktop/CYS/Los_5]
└─$
```

- · [Ff] matches f or F.

12. pick out lines that uses first letter alliteration - starting two words with the same letter.

```
┌──(kali㉿kali)-[~/Desktop/CYS/Los_5]
└─$ grep -E '\b(\w)\w*\s+\1\w*' text.txt
hello hello hello
```

- · This uses a backreference \1 to match two words starting with the same letter.

13. determine how many times contains the word "pattern".

```
┌──(kali㉿kali)-[~/Desktop/CYS/Los_5]
└─$ grep -o 'pattern' text.txt | wc -l
7

┌──(kali㉿kali)-[~/Desktop/CYS/Los_5]
└─$
```

- · This counts occurrences of "pattern" in the file.

14. to pick out lines with at least 10 characters:

```
┌──(kali㉿kali)-[~/Desktop/CYS/Los_5]
└─$ grep -E '.{10}' text.txt
pattern is easy
hello pattern--
good Morning!
geood Evening
pattern@gmail.com
sir madam-
for sir pattern madam
Popcorn popped before

┌──(kali㉿kali)-[~/Desktop/CYS/Los_5]
└─$
```

- · .{10,} matches any line with 40 or more characters.

15. to pick out lines with no punctuation

```
┌──(kali㉿kali)-[~/Desktop/CYS/Los_5]
└─$ grep -v '[[:punct:]]' text.txt
pattern is easy
pattern1
pattern2
pattern3

fun
geood Evening
fish

sir
madam
for sir pattern madam

Popcorn popped before

┌──(kali㉿kali)-[~/Desktop/CYS/Los_5]
└─$
```

- · -v inverts the match, selecting lines without punctuation.

16. to pick out lines with an uppercase letter other than the first character. (The first character on the line does not count.)

```
┌──(kali㉿kali)-[~/Desktop/CYS/Los_5]
└─$ grep '[^A-Z][A-Z]' text.txt
good Morning!
geood Evening

┌──(kali㉿kali)-[~/Desktop/CYS/Los_5]
└─$
```

- · This matches lines with an uppercase letter not at the beginning.

17. To pick out lines without rav

Quotes:

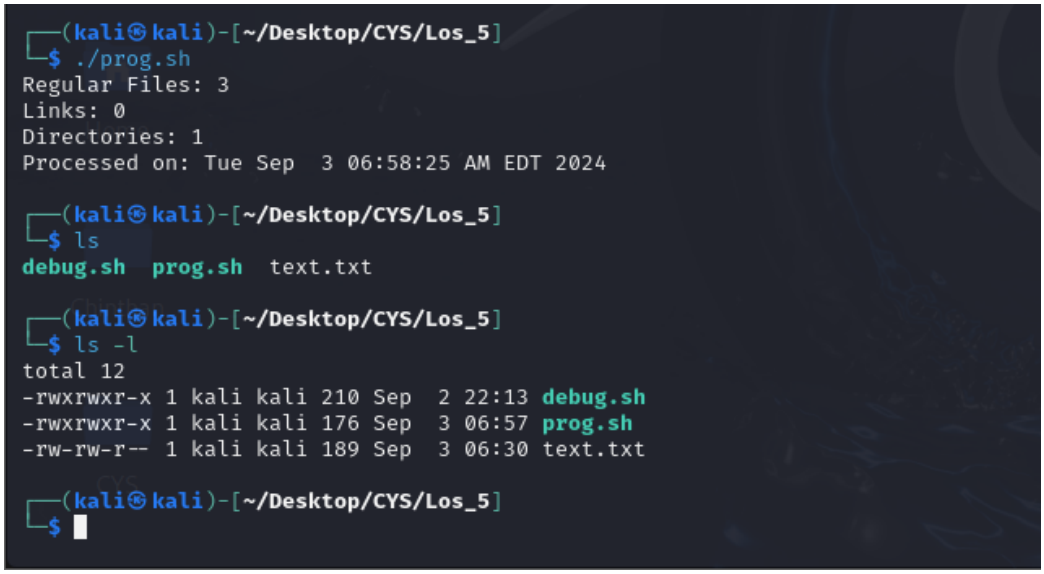
```
┌──(kali㉿kali)-[~/Desktop/CYS/Los_5]
└─$ grep -v 'patt' text.txt

--
good Morning!
fun
geood Evening
fish

sir madam-
sir
madam

Popcorn popped before
```

- -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.

```
┌──(kali㉿kali)-[~/Desktop/CYS/Los_5]
└─$ ls -l > ~/contents.txt

┌──(kali㉿kali)-[~/Desktop/CYS/Los_5]
└─$ cat ~/contents.txt

total 12
-rwxrwxr-x 1 kali kali 210 Sep  2 22:13 debug.sh
-rwxrwxr-x 1 kali kali 176 Sep  3 06:57 prog.sh
-rw-rw-r-- 1 kali kali 189 Sep  3 06:30 text.txt

┌──(kali㉿kali)-[~/Desktop/CYS/Los_5]
└─$
```

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.

```
┌──(kali㉿kali)-[~/Desktop/CYS/Los_5]
└─$ sort ~/contents.txt >> ~/contents_sorted.txt

┌──(kali㉿kali)-[~/Desktop/CYS/Los_5]
└─$ cat ~/contents_sorted.txt

-rw-rw-r-- 1 kali kali 189 Sep  3 06:30 text.txt
-rwxrwxr-x 1 kali kali 176 Sep  3 06:57 prog.sh
-rwxrwxr-x 1 kali kali 210 Sep  2 22:13 debug.sh
total 12

┌──(kali㉿kali)-[~/Desktop/CYS/Los_5]
└─$
```

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.

```
┌──(kali㉿kali)-[~/Desktop/CYS/Los_5]
└─$ tail -n 10 /etc/passwd > ~/content1.txt

┌──(kali㉿kali)-[~/Desktop/CYS/Los_5]
└─$ cat ~/content1.txt

_gophish:x:124:130::/var/lib/gophish:/usr/sbin/nologin
iodine:x:125:65534::/run/iodine:/usr/sbin/nologin
miredo:x:126:65534::/var/run/miredo:/usr/sbin/nologin
statd:x:127:65534::/var/lib/nfs:/usr/sbin/nologin
redis:x:128:131::/var/lib/redis:/usr/sbin/nologin
postgres:x:129:132:PostgreSQL administrator,,,:/var/lib/postgresql:/bin/bash
mosquitto:x:130:133::/var/lib/mosquitto:/usr/sbin/nologin
inetsim:x:131:134::/var/lib/inetsim:/usr/sbin/nologin
_gvm:x:132:135::/var/lib/openvas:/usr/sbin/nologin
kali:x:1000:1000:,,,:/home/kali:/usr/bin/zsh

┌──(kali㉿kali)-[~/Desktop/CYS/Los_5]
└─$
```

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.

```
┌──(kali㉿kali)-[~/Desktop/CYS/Los_5]
└─$ cat ~/contents.txt
total 12
-rwxrwxr-x 1 kali kali 210 Sep  2 22:13 debug.sh
-rwxrwxr-x 1 kali kali 176 Sep  3 06:57 prog.sh
-rw-rw-r-- 1 kali kali 189 Sep  3 06:30 text.txt

┌──(kali㉿kali)-[~/Desktop/CYS/Los_5]
└─$
```

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

```
┌──(kali㉿kali)-[~/Desktop/CYS/Los_5]
└─$ head -n 5 /etc/passwd | sort -r
sys:x:3:3:sys:/dev:/usr/sbin/nologin
sync:x:4:65534:sync:/bin:/bin/sync
root:x:0:0:root:/root:/usr/bin/zsh
daemon:x:1:1:daemon:/usr/sbin:/usr/sbin/nologin
bin:x:2:2:bin:/bin:/usr/sbin/nologin

┌──(kali㉿kali)-[~/Desktop/CYS/Los_5]
└─$
```

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

Debug

```
┌──(kali㉿kali)-[~/Desktop/CYS/Los_5]
└─$ tail -n 9 ~/contents.txt | wc -m
155

┌──(kali㉿kali)-[~/Desktop/CYS/Los_5]
└─$
```

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