

SUMMER SCHOOL 2021

Search and Edit Text Files

Dr Krishna K. Govender

Summary of commands used in this lecture

Command	Meaning
nano	Command line text editor in Linux
Ctrl+X > Y > Enter	Sequence used when saving changes to a file created/edited in nano
cat	Concatenate a file/s
ls	List items in current directory
grep	Used to find text in a file
sed	Used to find and replace text in a file

- As in the previous lectures all commands provided in **BOLD** and center aligned from here onwards are commands that can be typed/executed directly in a terminal.
- We will create a file called **randominfo.txt** using **nano**:

```
~$ nano randominfo.txt
```

```
GNU nano 4.8 randominfo.txt Modified
this is a fil with sme random Information
we is going to make sme chnges to the file

^G Get Help    ^O Write Out   ^W Where Is    ^K Cut Text    ^J Justify
^X Exit        ^R Read File   ^_ Replace     ^U Paste Text  ^T To Spell
```

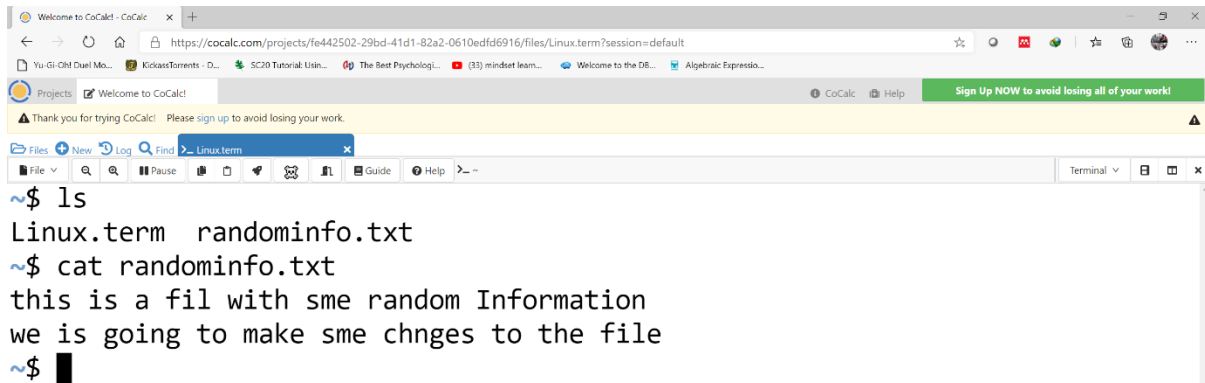
- Please type the information as it appears in the image above (with the errors included).
- Use **Ctrl+X > Y > Enter** to exit **nano**, save the changes to the file and end up back on the terminal:

```
~$ nano randominfo.txt
~$
```

- Check that the file does exist and contains the relevant information:

ls

cat randominfo.txt



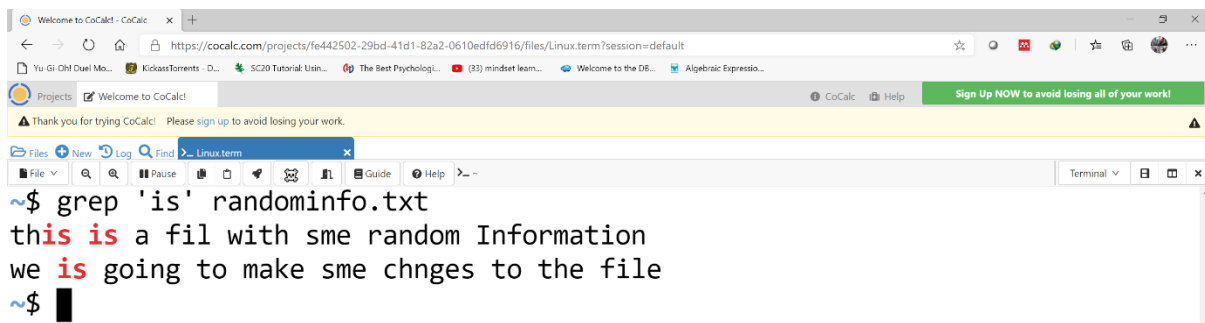
```

Welcome to CoCalc - CoCalc
https://cocalc.com/projects/fe442502-29bd-41d1-82a2-0610edfd6916/files/Linux.term?session=default
Projects Welcome to CoCalc!
Thank you for trying CoCalc! Please sign up to avoid losing your work.
Files New Log Find Linux.term
File Find Pause Copy Paste Undo Redo Guide Help ~
~$ ls
Linux.term randominfo.txt
~$ cat randominfo.txt
this is a fil with sme random Information
we is going to make sme chnges to the file
~$ █

```

- To search for a word in a file we use:

grep 'is' randominfo.txt



```

Welcome to CoCalc - CoCalc
https://cocalc.com/projects/fe442502-29bd-41d1-82a2-0610edfd6916/files/Linux.term?session=default
Projects Welcome to CoCalc!
Thank you for trying CoCalc! Please sign up to avoid losing your work.
Files New Log Find Linux.term
File Find Pause Copy Paste Undo Redo Guide Help ~
~$ grep 'is' randominfo.txt
this is a fil with sme random Information
we is going to make sme chnges to the file
~$ █

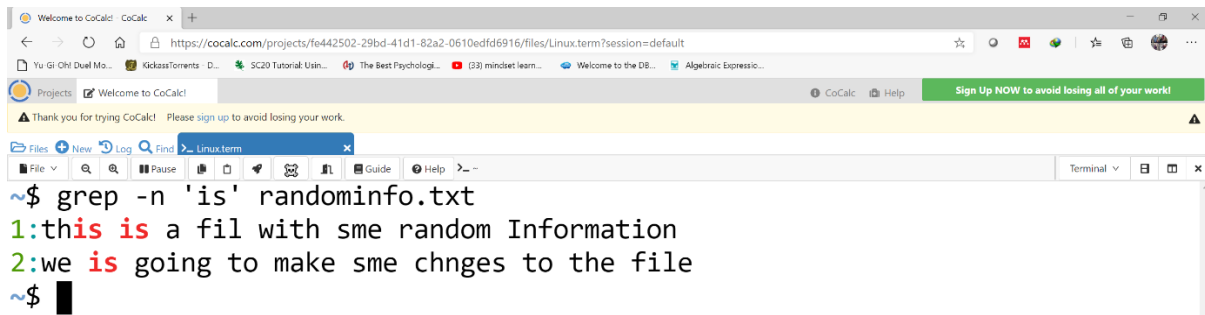
```

- **grep** will search the file **randominfo.txt** and look for the words/characters that are a match to what is provided in the quotes.

We see that all instances where the word **is** has been found are highlighted in red.

- In the above example we do only have two lines of text in **randominfo.txt**, but what if you have a file that has multiple lines of text and you wish to know the line number where your word/character appears:

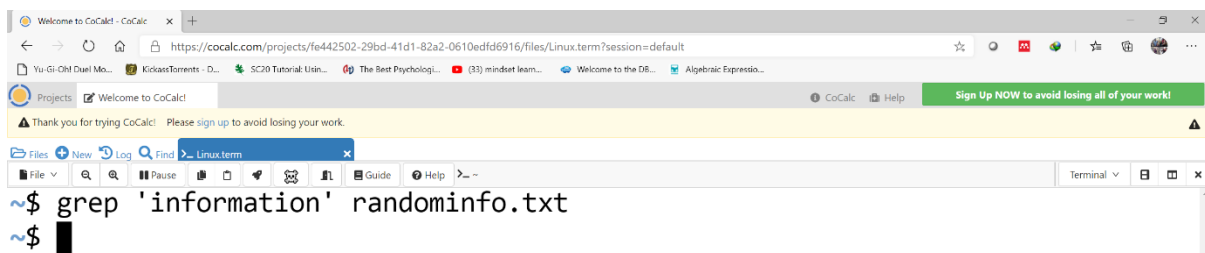
grep -n 'is' randominfo.txt



```
~$ grep -n 'is' randominfo.txt
1:this is a fil with sme random Information
2:we is going to make sme chnges to the file
~$
```

- Now you find the line numbers provided on the left corresponding to the line where the word/character appears inside the **randominfo.txt** file.
- Search the file **randominfo.txt** for the word information:

grep 'information' randominfo.txt



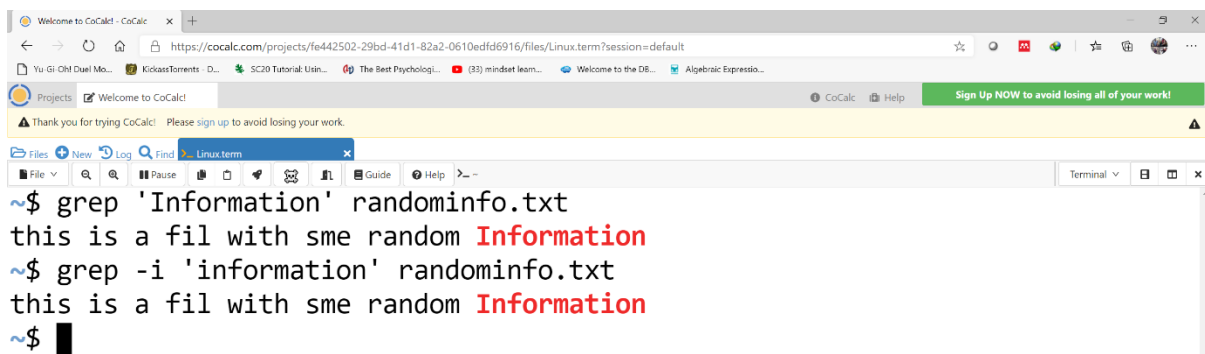
```
~$ grep 'information' randominfo.txt
~$
```

- This does not seem to produce anything, despite the file containing the word. As with most Linux commands **grep** is case sensitive and the word in the file contains a capital I.

There are two approaches that could be used to get the result we desire:

grep 'Information' randominfo.txt

grep -i 'information' randominfo.txt



```
~$ grep 'Information' randominfo.txt
this is a fil with sme random Information
~$ grep -i 'information' randominfo.txt
this is a fil with sme random Information
~$
```

- In the first case we use **grep** to find the word with the same pattern by ensuring the capital letter is present.

In the second case we switch off the case sensitivity with the **-i** flag so that **grep** will find all text matching the word regardless of whether it has uppercase or lowercase characters.

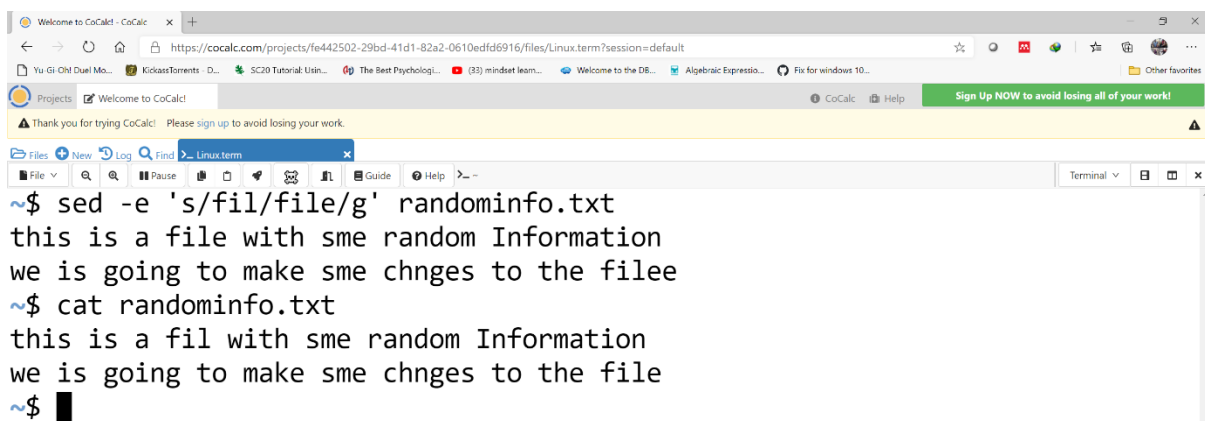
Both cases yield the same result.

- Now we need to correct the errors contained in the file.

We will start by changing the word **fil**:

sed -e 's/fil/file/g' randominfo.txt

cat randominfo.txt



```

~$ sed -e 's/fil/file/g' randominfo.txt
this is a file with sme random Information
we is going to make sme chnges to the filee
~$ cat randominfo.txt
this is a fil with sme random Information
we is going to make sme chnges to the file
~$

```

- **sed** is the command used when you wish to find and replace text.

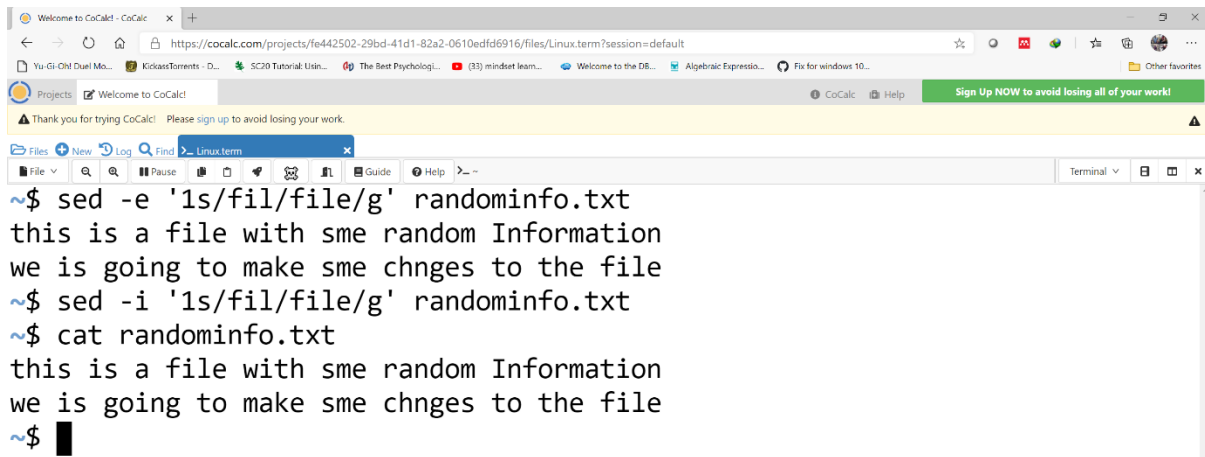
The **-e** tells **sed** to make the changes and display the result on the screen. As you can see from the image above the result of what the new text will look like is displayed on the screen, however no change takes place on the file **randominfo.txt**. Displaying the contents of the file (**cat randominfo.txt**) shows that the original information remains. The **'s/fil/file/g'** tells **sed** to look through every line of text (**s**) to find the word **fil** and replace it with **file** and do this for every instance that the word appears (**g**).

- The above **sed** command is not providing the desired result though as we end up with the word **filee** on the last line of our file. We need to make some changes to the command:

sed -e '1s/fil/file/g' randominfo.txt

sed -i '1s/fil/file/g' randominfo.txt

cat randominfo.txt



```
~$ sed -e '1s/fil/file/g' randominfo.txt
this is a file with sme random Information
we is going to make sme chnges to the file
~$ sed -i '1s/fil/file/g' randominfo.txt
~$ cat randominfo.txt
this is a file with sme random Information
we is going to make sme chnges to the file
~$
```

- By using **1s** instead of **s** we are telling **sed** to only make the changes to the first line of the file.
sed -e shows us the result of the change on the screen.
Now that we have the expected result, we execute the same command and just change **-e** to **-i**. This will make the changes directly to the **randominfo.txt** file.
We confirm that the changes did take place by displaying the contents of the file (**cat randominfo.txt**)
- This brings us to the end of **Search and Edit Text Files**.
- At this point you should give **Exercise 4** a try and see if you can make the file above grammatical correct.