Module Code & Module Title

Level 5 – Network Operating Systems

Assessment Type

Logbook 8

Semester

2023/24 Autumn

Student Name: Aayush Wanem Limbu

London Met ID: 22072043

College ID: np05cp4a220010

Assignment Due Date: Wednesday, December 20, 2023

Assignment Submission Date: Wednesday, December 20, 2023

Submitted To: Enjina Ghimire

Word Count (Where Required):2417

I confirm that I understand my coursework needs to be submitted online via Google Classroom under the relevant module page before the deadline in order for my assignment to be accepted and marked. I am fully aware that late submissions will be treated as non-submission and a mark of zero will be awarded.

Table of Contents

Table of Figures

1. UNIX Workshop Log: Directories, Files, and Alias

1.1 Aims and Objectives

1.1.1 Aims

The aim of this workshop is to improve the core UNIX skills and to streamline the command line tasks and to further improve the knowledge of directories and files.

1.1.2 Objectives

Directory Skills:

Create a clear directory structure using UNIX commands.

File Handling with cat and grep:

Practice making and managing files with cat.

Explore effective text searching using grep.

Alias Proficiency:

Learn to use, create, and remove aliases for common commands.

Search Expertise:

Master text searches with grep, trying out different options.

System Understanding via Aliases:

Connect aliases to system information with the nwho alias.

Command History Tricks:

Explore command history and redo actions using history and fc.

Alias Persistence:

Understand how aliase can be made permanent by editing the .bashrc

1.2 Required Tools and Concepts

Linux virtual machine or WSL machine

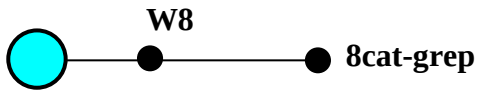
Linux basics

Text editors like nano, geany, vim etc

Linux file structures knowledge

2. Questions with their answers

1. Create the directory structure presented in the figure below.



Your home directory

Ans,

Figure 1 Creating Directory Structure

2. Change to the 8cat-grep directory by one step using a relative pathname.

Ans,

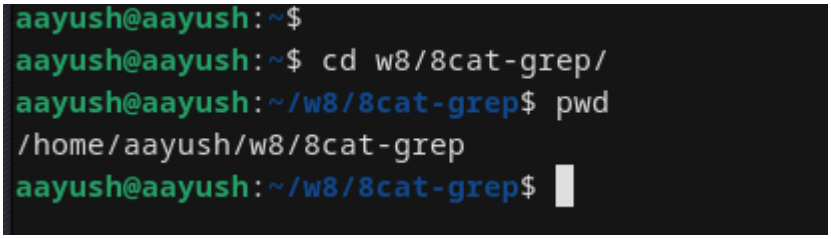


Figure 2 Changing Directory Using Relative Path

3. **Using the cat utility, create two files**

File testa	File testb
Kkkll	KKKKK
lllmm	LLLLL
oo–oo	MMMMM
mmmdd	DDDDD
dddkk	

Ans, We are using the write functionality of the cat utility to create and write into a file.

Figure 3 Using cat command to create and write into a filea.

Figure 4 Using cat command to create and write into a fileb.

4. **Give the following commands and explain the results for yourself**

- grep ll testa
- grep -v ll testa
- grep -n ll testa
- grep -l ll *
- grep -i ll *
- grep -i LL *
- grep -c ll *
- grep '^K' testa testb
- grep -n '^' testa

Ans,

The grep is one of the most used command line tool for UNIX based operating systems used mainly for filtering certain patterns of texts or characters within certain files.

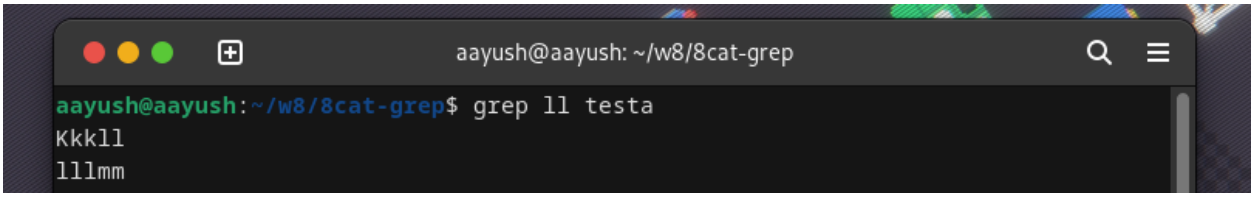


Figure 5 grep command without options.

Using just the grep command without any option will just search for the pattern 'll' in the supplied file i.e., filea, In the picture above we can see that it displays the lines of text containing the pattern ll in them, in some terminals or other Linux distributions the specified pattern of text is highlighted.

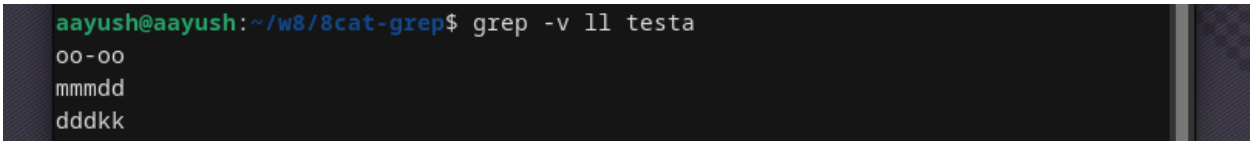


Figure 6 Using grep with -v option.

Using the grep with -v option or –invert-match option will select the non-matching lines from the file specified. In the figure above we have used the option -v with grep with the

'll' pattern and specified the file named testa and as the output we can see that it returned the lines which do not have the pattern 'll' in the specified file.

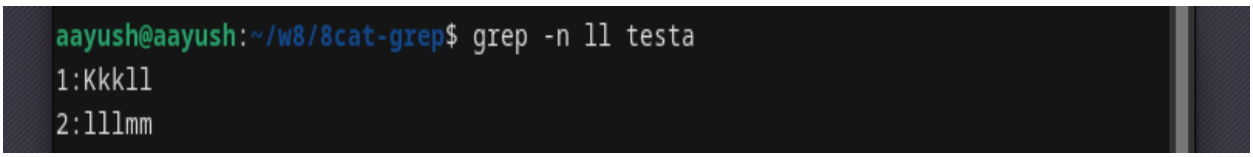


Figure 7 Using grep with -n option.

Using grep command with the -n option or –line-number or –line-buffered will do either one of two things:

--line-number: this option will print the line number with output lines.

--line-buffered: this option will flush the output on every line.

In the figure above we can see that when we are using the -n option of grep to search for 'll' pattern in test we can see it shows the line from the file which was supplied in the command i.e., testa that contains the pattern 'll' with its line number within that file.

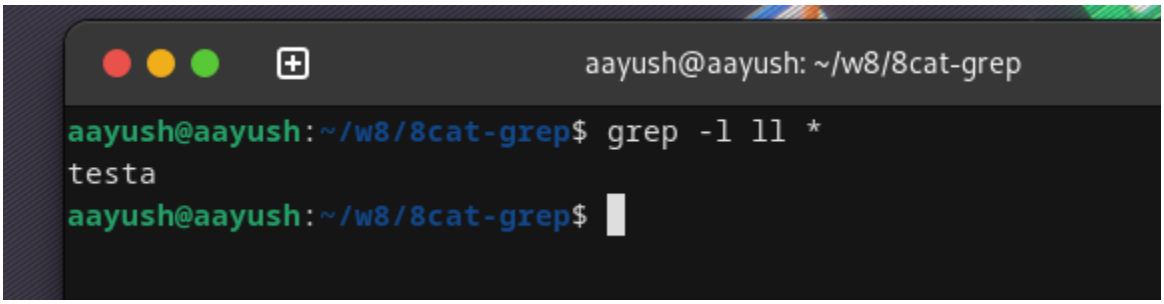


Figure 8 Using grep with -l option with * to check for all files.

Using the grep command with lower caps -l or –files-with-matches option will print only the names of the file with selected lines, in the above figure we used asterisk symbol "*****" in place of any file names, using '*****' will tell our system to look for all the multiple files in our current working directory or any directory that we specify explicitly. This is helpful for us when we have a large number of files, and we have to check it all at once.

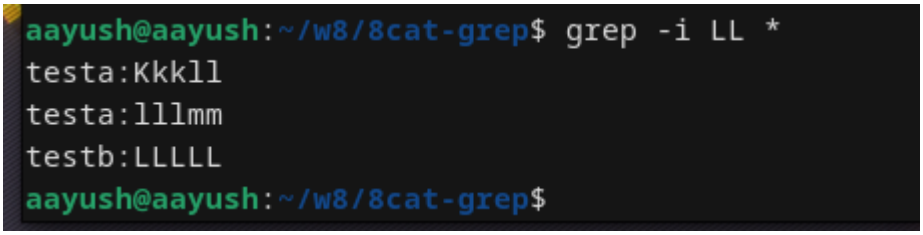


Figure 9 Using grep with -i option to search for all files.

Using the grep command with -i or –no-ignore-lower-case option to search will performs a case-insensitive search for the pattern "ll" in all files in the current directory. In the figure above we can see that we have used the option -i of grep to search for ‘ll’ lowercase ‘ll’ on all of the files of our current working directory and we were expected to see the names of files having “ll” in them, but we can see that it showed the name of the file and the lines containing both lowercase and uppercase “ll” as the output.

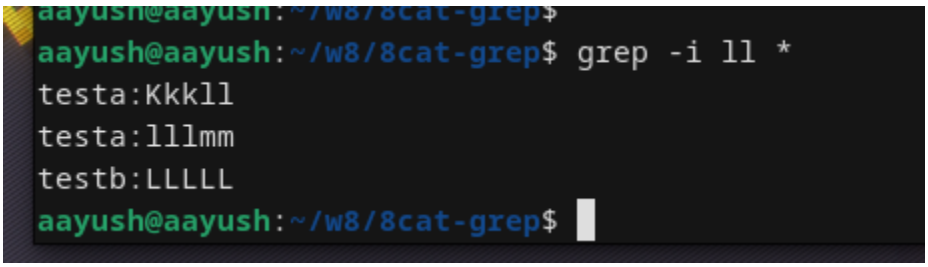


Figure 10 Again using grep with -i option to search for uppercase text patterns.

Again, using the -i option of grep to search for uppercase text patterns will display the same output as the previous step as it will only search for text patterns and ignore all the uppercase and lowercase texts.

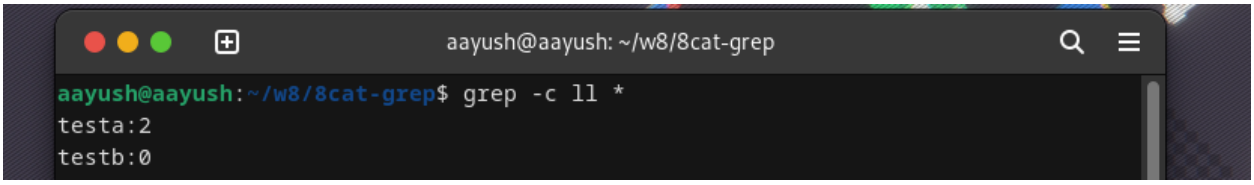


Figure 11 Using grep with -c option.

Using grep with -c option or –count will print only the count of selected lines per file. In the above figure we are trying to count how many lines we have “ll” pattern in each file of our current working directory. We can see that at our output it displays the filenames present in our current directory and the number of times “ll” pattern has occurred within that files.

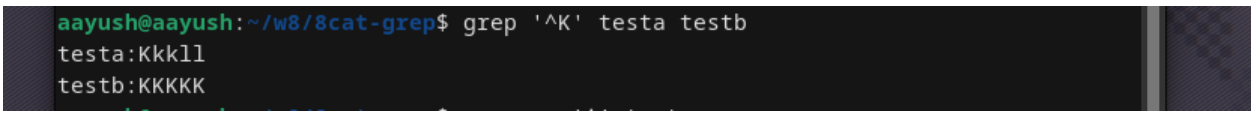


Figure 12 Using grep to search for capital K

This command searches for lines in files **testa** and **testb** that start with the character 'K'. It will display lines from both files where the line begins with 'K'.

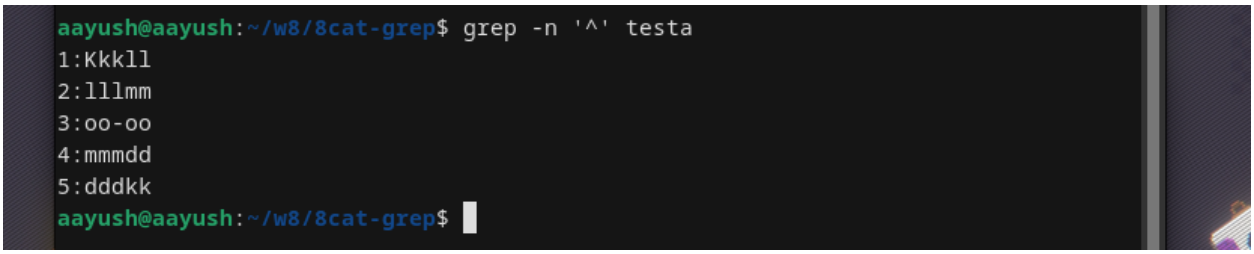


Figure 13 Using grep with -n option.

Using the -n option of grep we can display line numbers along with lines that match the specified pattern. In this case, it searches for lines that start with the beginning of the line in the file **testa**. It will display lines from **testa** where each line is preceded by its line number.

5. **(6) Define the lsal alias for ls -al command**

Show that your system stores it giving the alias command (without arguments).

Use it in your home directory.

Ans,

An alias is a user-defined shorthand or substitute for a command or series of commands. It allows users to create custom shortcuts for complex or frequently used commands, making it easier and quicker to execute them. Aliases are commonly used in command-line interfaces (CLIs) in operating systems like UNIX, Linux, and macOS.

Figure 14 Defining alias.

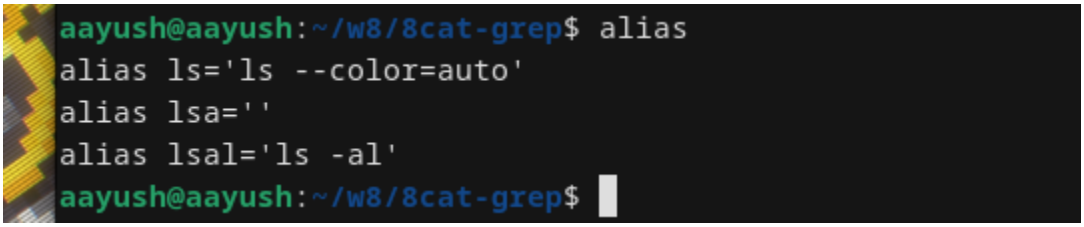


Figure 15 Verifying for created alias.

Using the alias command without any arguments will show us all the available aliases which are present in our system. We can also see that there is our newly created alias named “**lsal”**

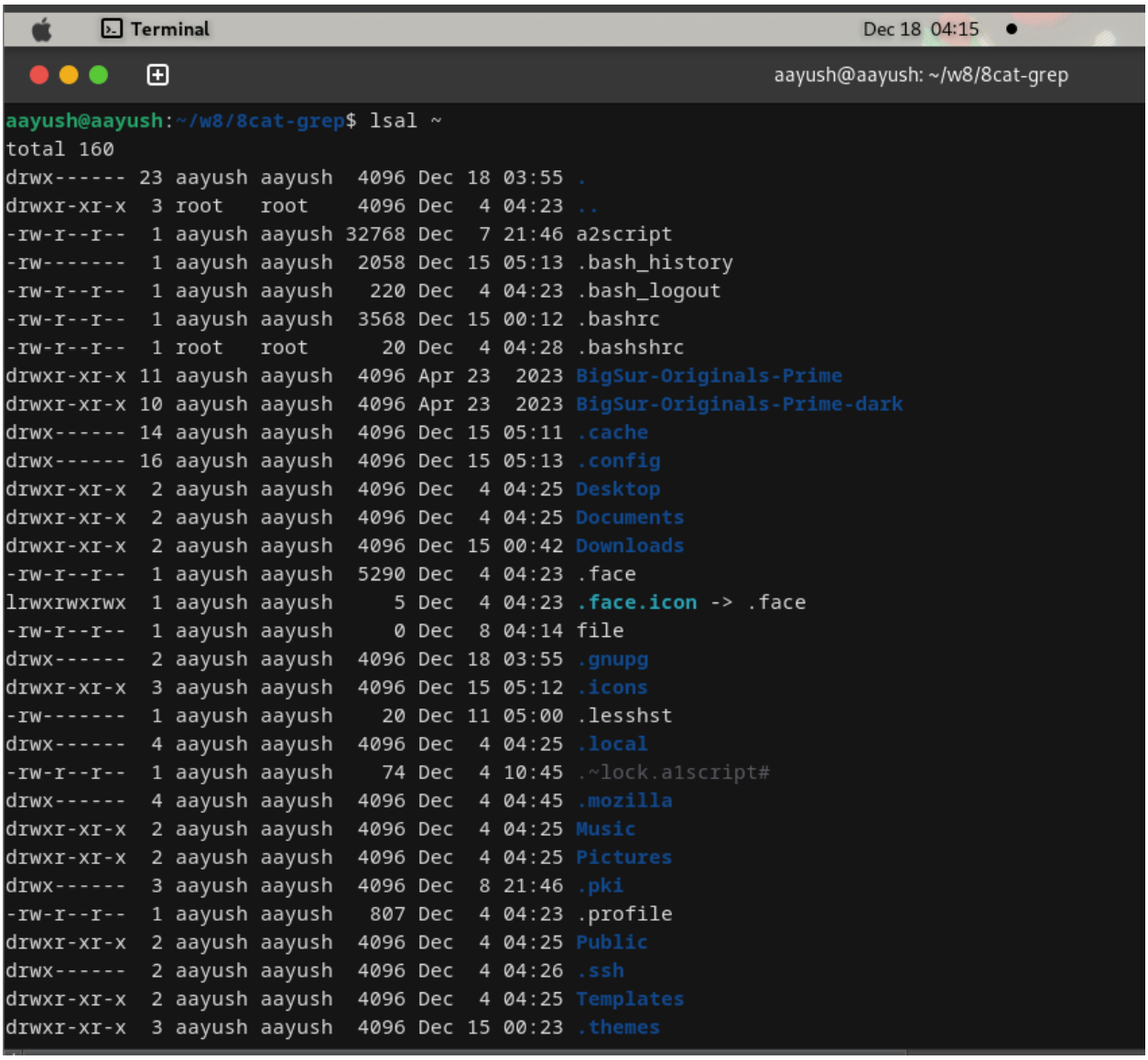


Figure 16 Using our alias in our home directory.

To use the alias that we recently created we need to use the name of that alias in our case it is the **“lsal”** to use it on our home directory without changing directories to the home directory we can use **“lsal ~”** and it will be equivalent of using this **“ls -al ~”** command.

6. **Remove the alias.**

Show that your system does not store it.

Ans, To remove any alias that we made we just need to use the unalias command with the name of the alias. In our case it will be **"unalias lsal"** it will just remove the alias having name lsal from our system.

```
aayush@aayush:~/w8/8cat-grep$
aayush@aayush:~/w8/8cat-grep$ unalias lsal
```

Figure 17 Removing alias named lsal.

```
aayush@aayush:~/w8/8cat-grep$ alias
alias ls='ls --color=auto'
alias lsa=''
aayush@aayush:~/w8/8cat-grep$
```

Figure 18 Verifying aliases.

After using the alias command, we can see that it will show us the alias command and we can see that the alias that we created named **lsal** is removed.

7. **Define this alias again preserving it for the next session**

Show that the system still keeps this your alias.

Ans, To make an alias permanent we need to edit the .bashrc or .zshrc located in the home directory via a text editor such as nano, vim, geany etc. The .bashrc or .zshrc stores the configuration files for the shell that the device is using for example if a device is using bash shell as the terminal emulator it will have the .bashrc file in the user's home directory.

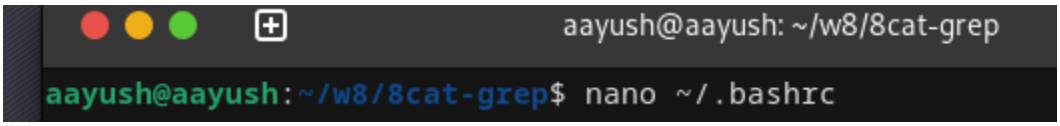


Figure 19 using nano to open .bashrc

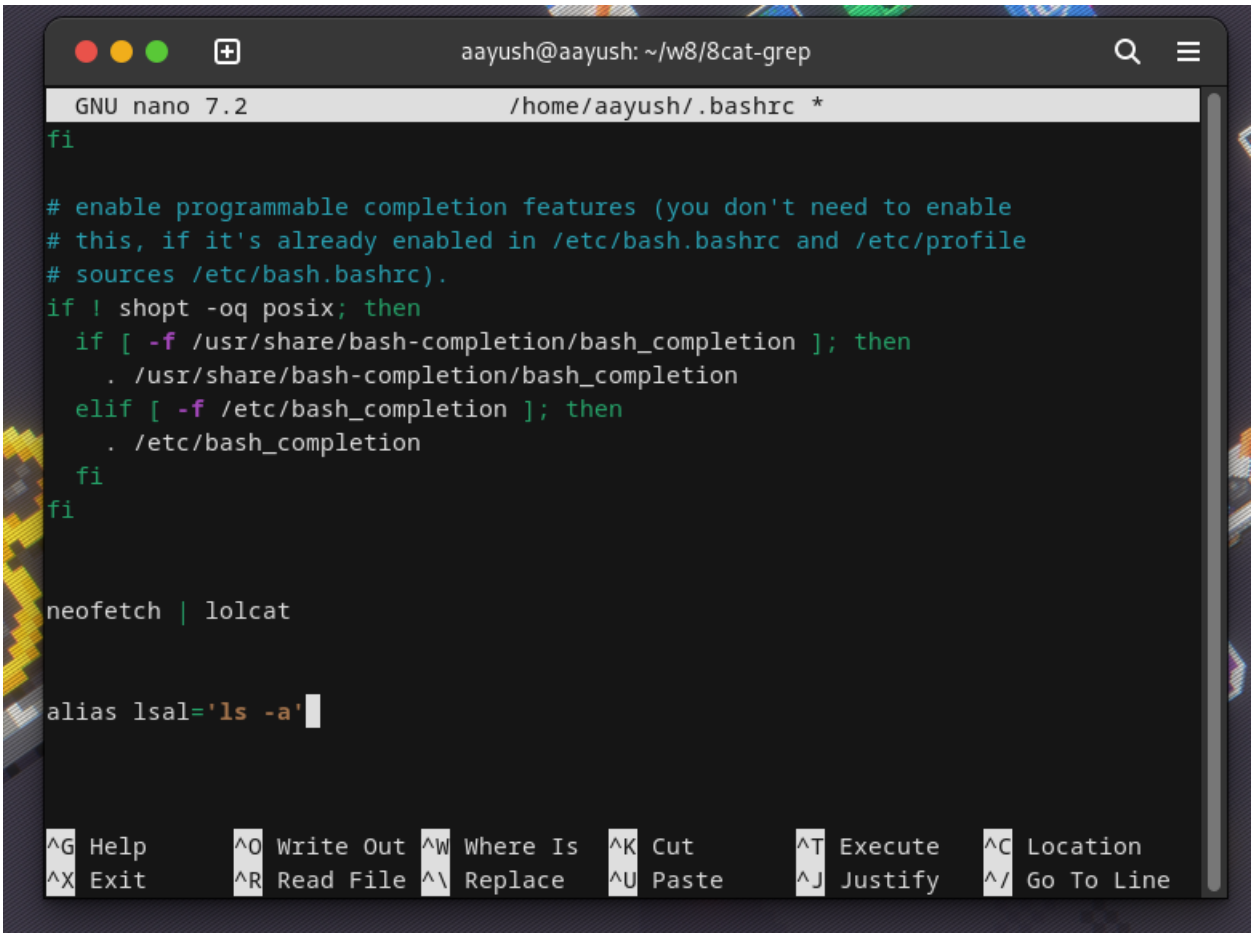


Figure 20 Adding alias in .bashrc

To open nano as a text editor, we need to type nano and the name of the file so it would be nano [file name]. since my Linux distribution has bash as its default terminal emulator

i have the .bashrc file so the command will be **nano ~/.bashrc**, it will open a new window which shows the .bashrc configurations we need to scroll down using the down arrow key and type **alias lsal='ls -al'** and exit nano text editor.

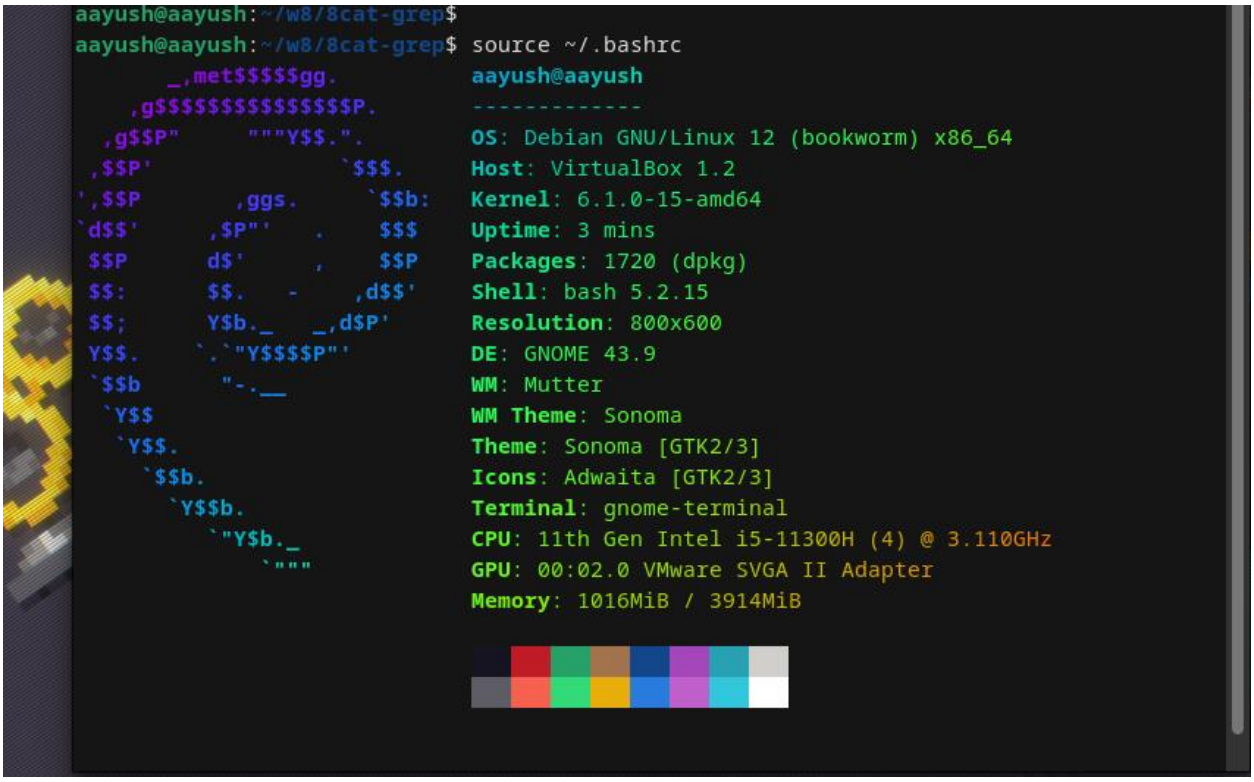


Figure 21 Reading again from .bashrc

After exiting we need to read from the .bashrc file again for that we can restart our device, or we can type source ~/.bashrc to read from the file again. If we type source .~/.bashrc it will restart our terminal and the new alias which we set can be used.

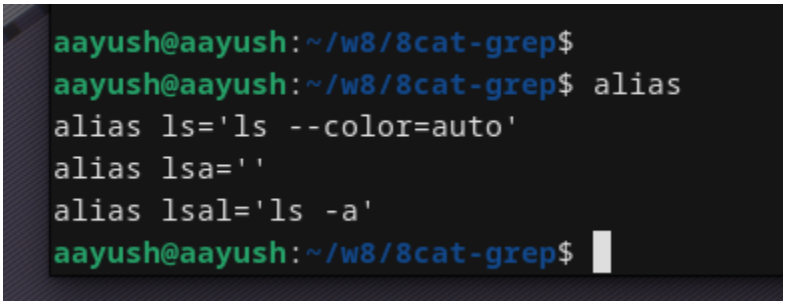


Figure 22 Verifying again

Verifying again to see all the aliases using the alias command.

8. **Define the nwho alias for the number of system file at UNIX computers.**

alias nwho='getent passwd|wc -l'

Ans,

```
aayush@aayush:~/w8/8cat-grep$
aayush@aayush:~/w8/8cat-grep$ alias nwho='getent passwd | wc -l'
aayush@aayush:~/w8/8cat-grep$ alias
alias ls='ls --color=auto'
alias lsa=''
alias lsal='ls -a'
alias nwho='getent passwd | wc -l'
aayush@aayush:~/w8/8cat-grep$
```

Figure 23 Adding nwho alias and verifying.

9. **Give the command nwho. Compare the figure displayed with ones got by your UNIX-mates.**

Ans,

This nwho alias will count the number of user accounts on the system using the **getent passwd|wc -l** command.

```
aayush@aayush:~/w8/8cat-grep$ nwho
36
aayush@aayush:~/w8/8cat-grep$
```

Figure 24 Running nwho alias command.

The output of the **nwho** alias, counting user accounts with **getent passwd | wc -l**, may vary across different Linux or UNIX operating systems. Various factors contribute to these differences, including the distinct user management approaches employed by different OS distributions. System configurations, default user accounts, and the presence of networked users can also influence the count. Therefore, when executing **nwho** on different UNIX systems, one can expect divergent values due to the unique characteristics and user environments of each operating system.

10. List your last commands executed giving the history command.

Ans,

The **history** command in UNIX-like operating systems is used to display a list of previously executed commands. When you append a number (e.g., **history 10**), it shows the specified number of most recent commands from your command history.

In the case of **history 10**, the command will display the last 10 commands that you have executed in the current terminal session. The output typically includes the command numbers and the corresponding commands.

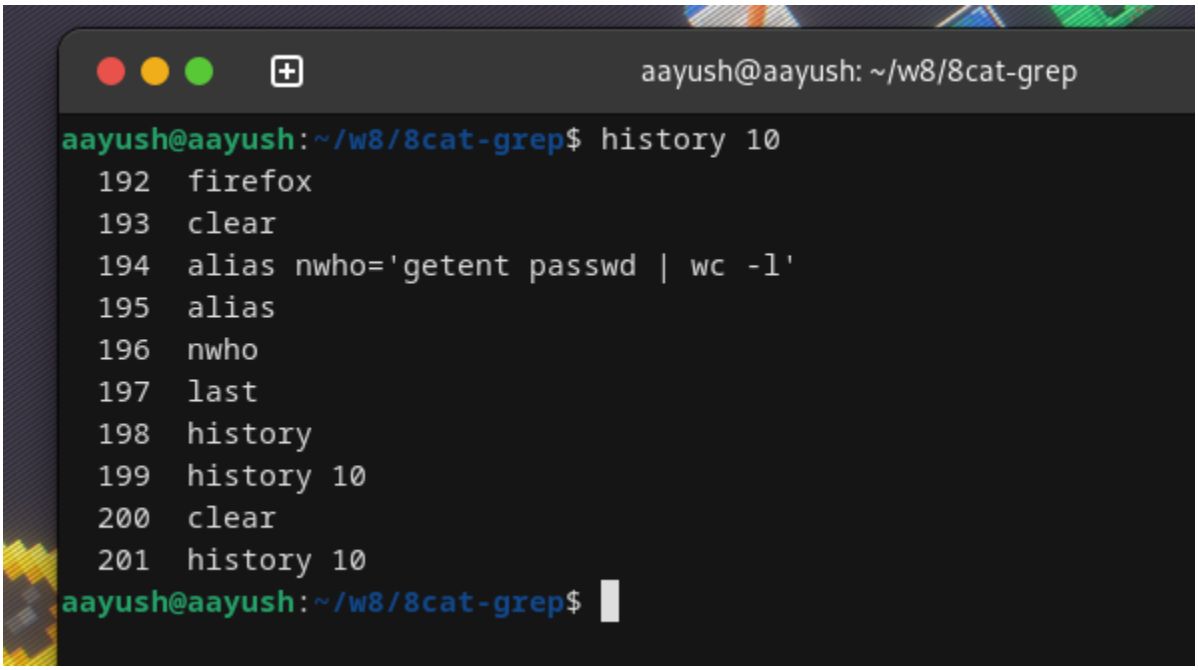


Figure 25 Executing history command

11. **Re-execute the *last but one* command using the redo (r) command and the number of the event.**

fc -r

Ans,

The **fc -r** command is a shell command used to redo (re-execute) the last command that was previously executed. It simplifies the process of rerunning the most recent command without retyping it.

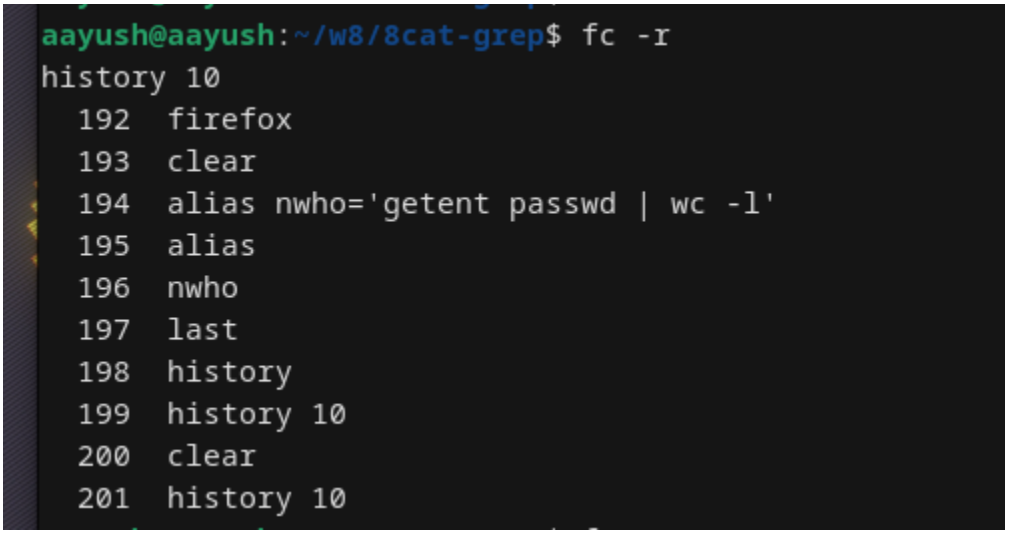


Figure 26 function redo command

12. **Re-execute the command given *three commands ago* using the negative integer.**

!-3

Ans, The command **!-3** is a history expansion in UNIX-like shells. It retrieves and re-executes the command that was executed three commands ago from the current command history. This provides a quick way to repeat a specific command without typing it out again.

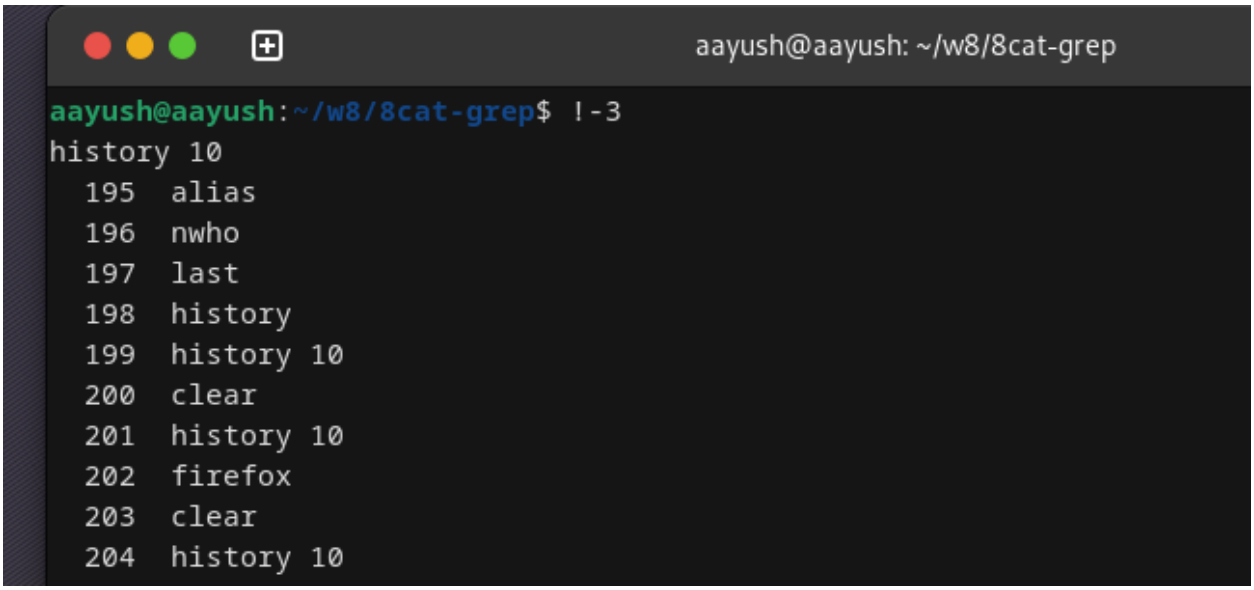


Figure 27 executing the third last command

13. **Re-execute the last command which name begins with ‘l’.**

fc -e- l

Ans,

The **fc -e- l** command is a history expansion in UNIX-like shells. It re-executes the most recent command from the command history that starts with the letter 'l'. This provides a quick way to repeat a specific command with a particular prefix.

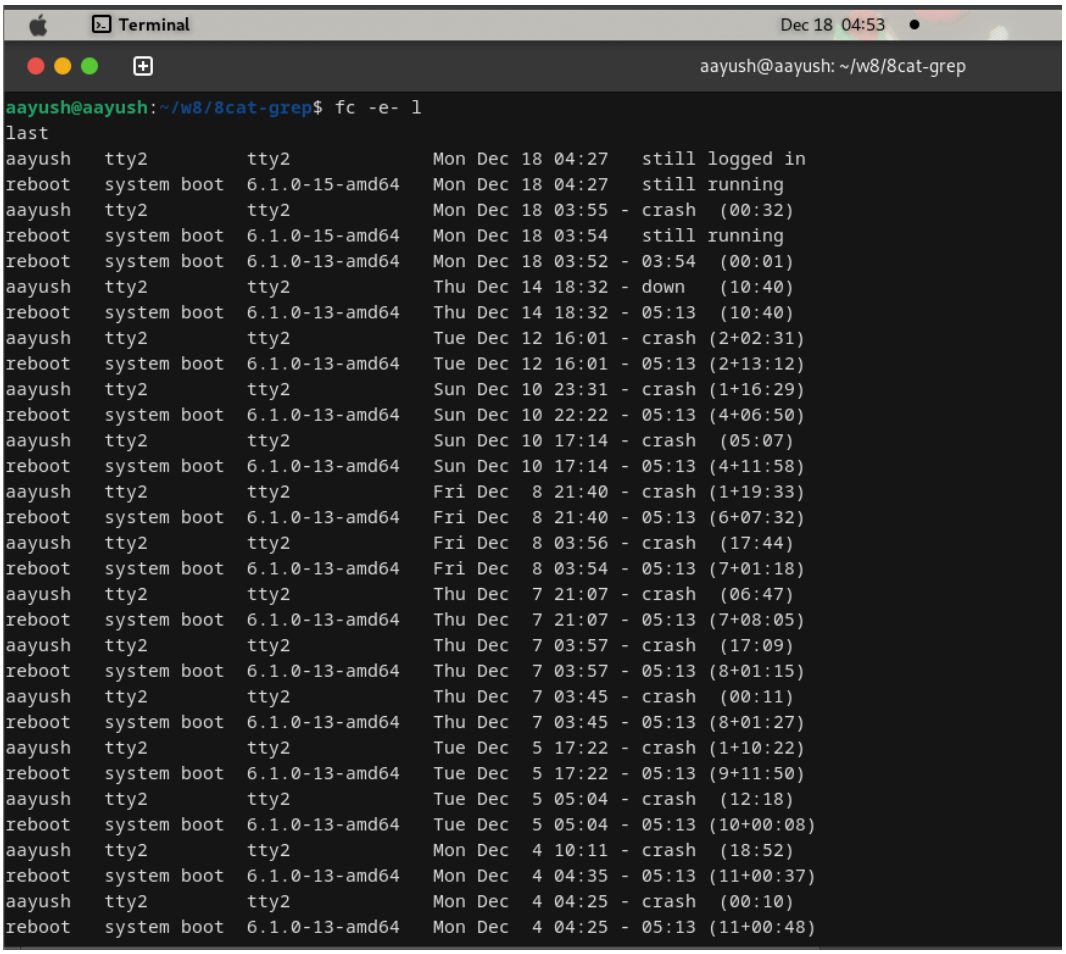


Figure 28 Executing last command that starts with l.

3. Conclusion:

During this week's workshop tasks, I practiced using UNIX utilities, focusing on directory manipulation and text processing commands. Successfully creating directories, generating files, and exploring various **grep** commands enhanced my proficiency. Alias management, including definition, verification, removal, and redefinition, demonstrated practical skills. The workshop also covered history navigation, command redo, and history expansion, contributing to a comprehensive understanding of essential UNIX concepts. Overall, the exercises provided valuable hands-on experience, reinforcing key skills in UNIX system navigation and manipulation.