Module Code & Module Title

Level 5 – CT5052, Network Operating Systems

Assessment Type

Logbook 8.

Semester

2023/24 Spring/Autumn

Student Name: Asim Sapkota

London Met ID: 23048768

College ID: NP04CP4A230225

Assignment Due Date: 28th Dec 2024

Assignment Submission Date: 28th Dec 2024

Submitted to: Mr Prasant Adhikari

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

Introduction

UNIX is a powerful, multi-tasking, and multi-user operating system. From the very beginning, UNIX was designed to be simple, efficient, and flexible. Its underlying philosophy is one of modularity—creating small, focused programs that each do one thing well. This approach allows users to combine these small programs to accomplish more complex tasks. The UNIX programming environment encourages programmers to write simple tools and then use them in combination to solve more complicated problems. These tools are often referred to as UNIX utilities and form the backbone of the operating system's power. (Brian W. Kernighen, 1984)

The power of UNIX utilities lies in their ability to manage files, directories, and system permissions with remarkable efficiency. Unlike graphical user interfaces (GUIs), which prioritize user-friendliness, UNIX commands provide fine-grained control and are highly customizable. They are indispensable for automation, scripting, and managing large-scale systems. For instance, a task that might require multiple steps in a GUI can often be accomplished with a single, well-crafted UNIX command. This efficiency underscores the enduring relevance of UNIX utilities, even in today's GUI-dominated landscape.

This workshop focuses on practical exercises with fundamental UNIX commands such as mkdir, cd, cat, grep, alias, and history. Participants will explore the creation of directory structures, manipulation of files, and management of aliases, all of which are foundational skills for navigating and utilizing the UNIX operating system. By practicing these commands, learners will gain a deeper understanding of the UNIX philosophy of doing one thing well, and they will develop the technical skills necessary to perform tasks efficiently.

In addition to these basic utilities, learners will also explore advanced features such as the use of aliases for command simplification, the grep command for text searching, and the history command for managing command-line history. These exercises will not only enhance their command-line proficiency but also provide insights into the powerful scripting and automation capabilities of UNIX.

Mastery of UNIX commands and utilities equips learners with the tools to handle complex tasks, from file manipulation to system-level operations. It also lays a strong foundation for understanding advanced topics in operating systems and network management, making it a valuable skill set for IT professionals. (Evi Nemeth, 2017)

Objectives

The Objectives of this workshop is to apply a hands on exercise with some basic UNIX utilities and commands that deal with file manipulation, text searching, managing of aliases, and the use of command history. This workshop aims to teach the way of how to work with files using the cat command as well as to navigate in the directories using the relative and the absolute pathnames and to search for the text in the files using different grep options. The workshop also teaches how to define and manage aliases for commands you frequently run to streamline workflow, and how to do so such that subsequent sessions will dutifully run the aliases this time and not next time. Additionally, students will be introduced to the use of command history to dramatically increase efficiency in re-executing previous commands. The workshop also concludes with practical exercises that gather system information, like users with gerent. And these skills are very important for working in a UNIX environment (used for system administration, automation, and scripts) since they won't only help you but all other users around you.

Required Tools and Concepts

Software:

A UNIX-based operating system (Linux or macOS) that provides access to the command line interface (CLI). We will use any terminal or shell, such as Bash (Bourne Again Shell), which is commonly available on UNIX systems.

Basic UNIX Commands:

- cat: This utility allows us to create and display the contents of files.
- grep: A powerful command-line tool for searching text patterns within files. It supports various options such as -n for line numbers, -i for case-insensitivity, and -v for inverted matching.
- ls: Used for listing directory contents. We will also work with aliases to customize this command.
- alias: A command that lets us define shortcuts for frequently used commands.
- history: Displays the command history, allowing us to reuse previous commands.
- fc: A command for editing and re-executing commands from the history.
- getent: A command to retrieve system information, such as the number of users, by accessing the system's databases.

Concepts:

- File Manipulation: We will create, display, and edit files in the UNIX environment using cat and other text editors.
- Search Techniques: We will use grep with various options to search for specific patterns in files, including case-sensitive, case-insensitive, and regular expressions.
- Aliases: We will create custom shortcuts for commands to improve efficiency in our tasks. Understanding how to define, use, and remove aliases both temporarily and permanently is crucial.
- Command History: We will leverage the history and fc commands to quickly re-execute past commands and streamline our workflow.
- UNIX File System Navigation: We will use relative and absolute pathnames to navigate and manipulate files and directories effectively.

Steps of Replicate:

1.

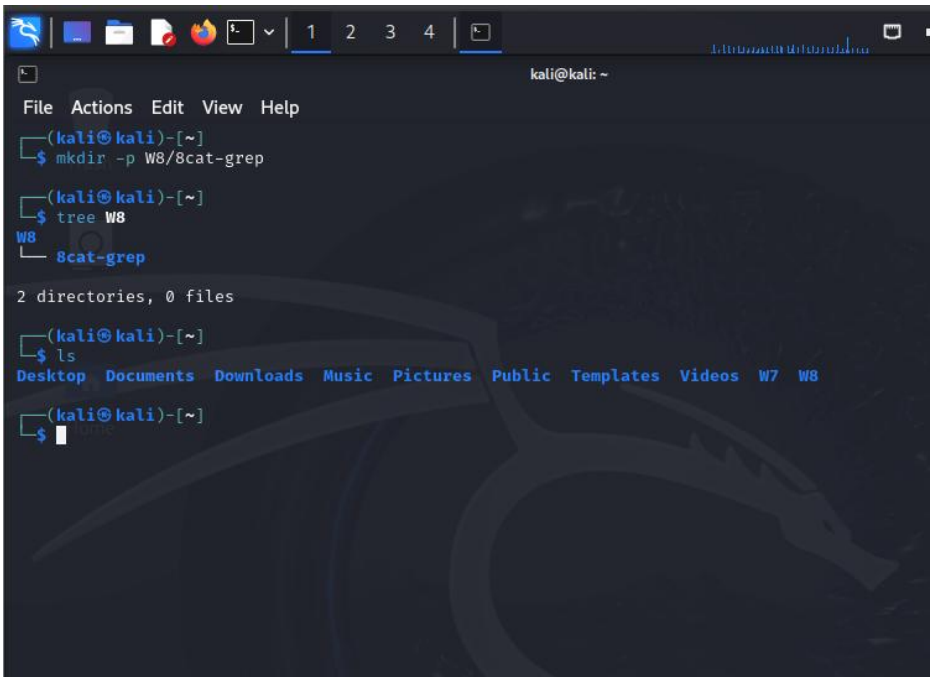

Figure 1: Creating new directory structure W8/8cat-grep

2.

Figure 2: Changing to the 8cat-grep directory

3.

Figure 3: Creating two files using cat command

Figure 4: Reading the two files

4.

Figure 5Searching for lines containing "ll" in the testa file.:

Figure 6: Searching for lines not containing "ll" in the testa file.

Figure 7: Displaying line numbers for lines containing "ll" in the testa file.

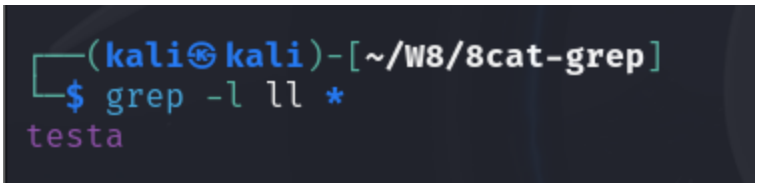

Figure 8: Listing files containing "ll" in their content.

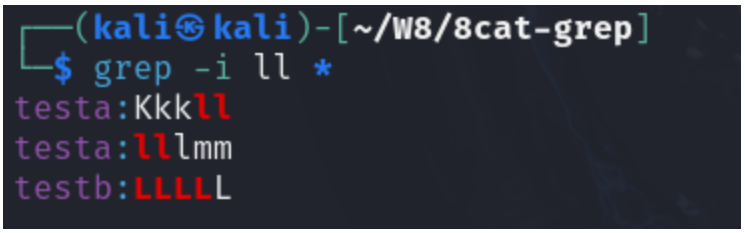

Figure 9: Performing a case-insensitive search for "ll" in all files.

Figure 10: Performing a case-insensitive search for "LL" in all files.

Figure 11: Counting the number of lines containing "ll" in each file.

Figure 12: Searching for lines starting with "K" in the testa and testb files.

Figure 13: Displaying all lines with their line numbers from the testa file.

6.

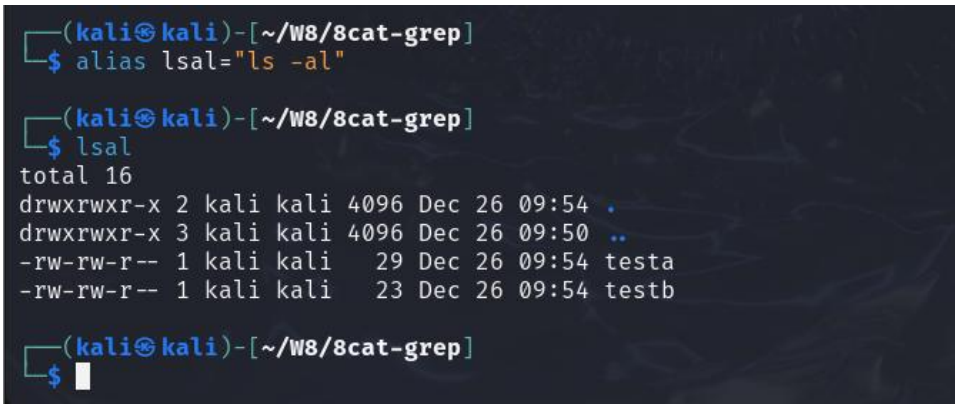

Figure 14: Defining the lsal alias for the ls -al command and checking it it is stored using alias

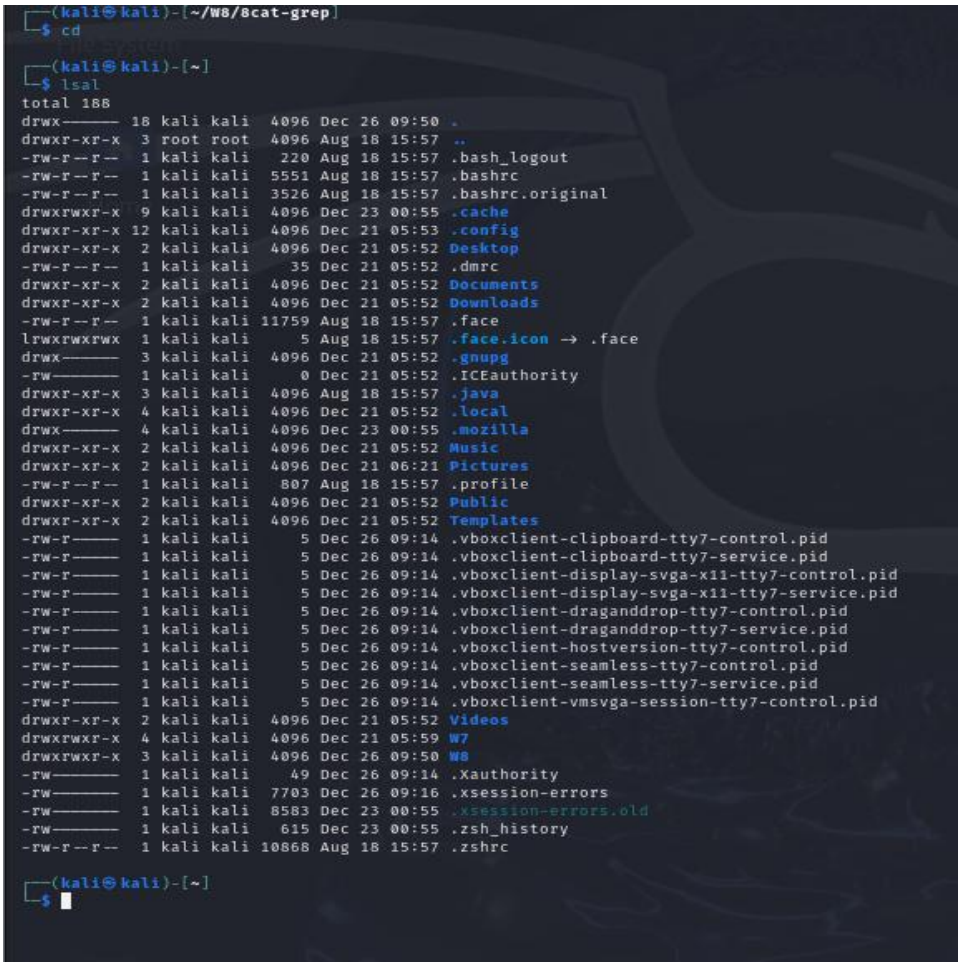

Figure 15: Using the alias in our home directory

7.

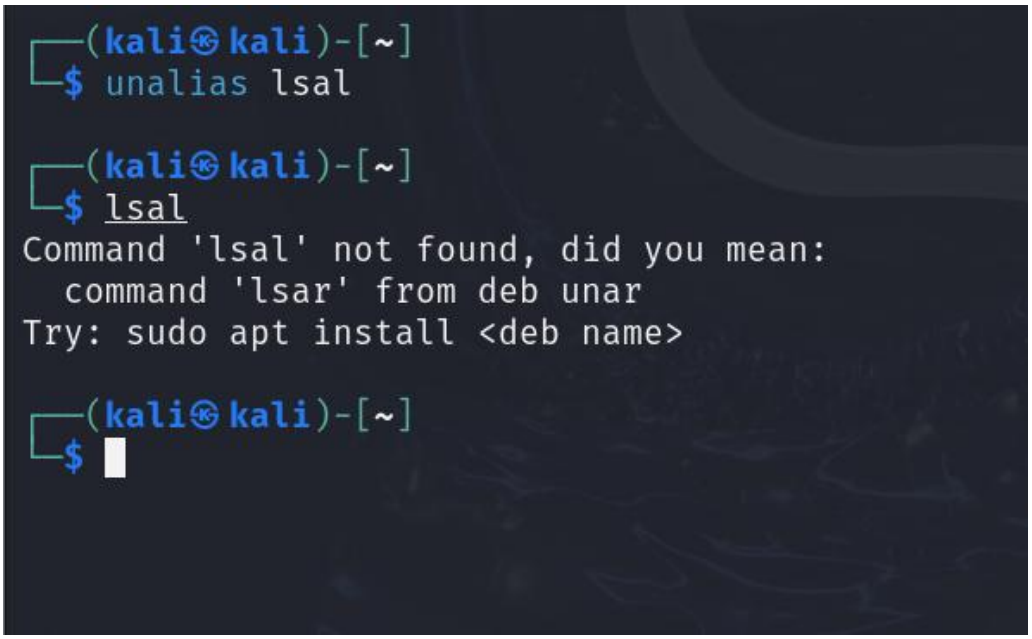

Figure 16: Removing the alias and verifying it is not stored

8.

Figure 17: Defining the alias again and making it permanent

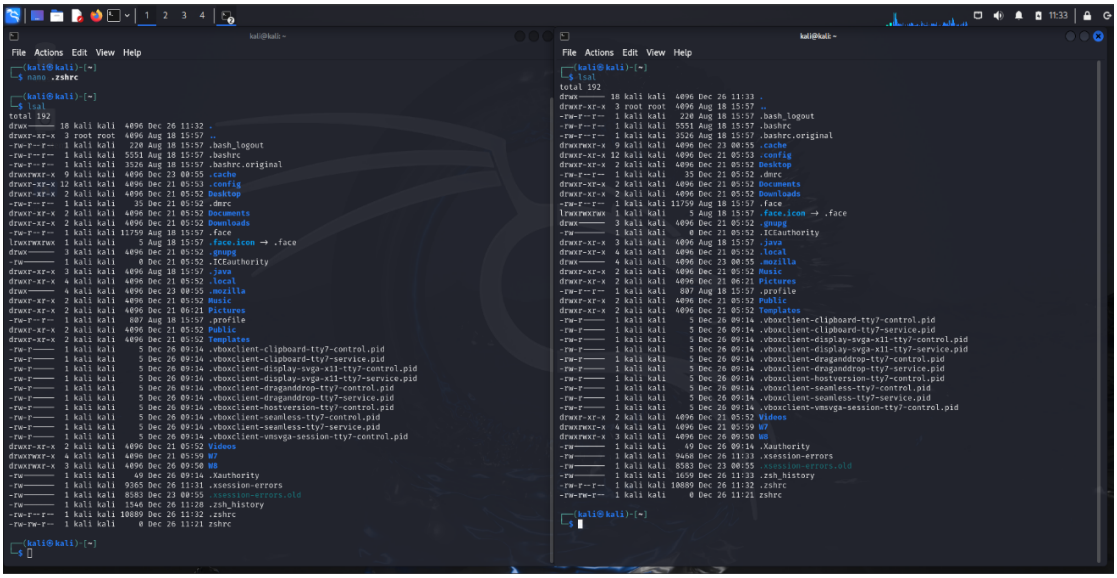

Figure 18: Checking the define alias work in another terminal or not

9 and 10

Figure 19: Defining the nwho alias and running nwho alias command

11.

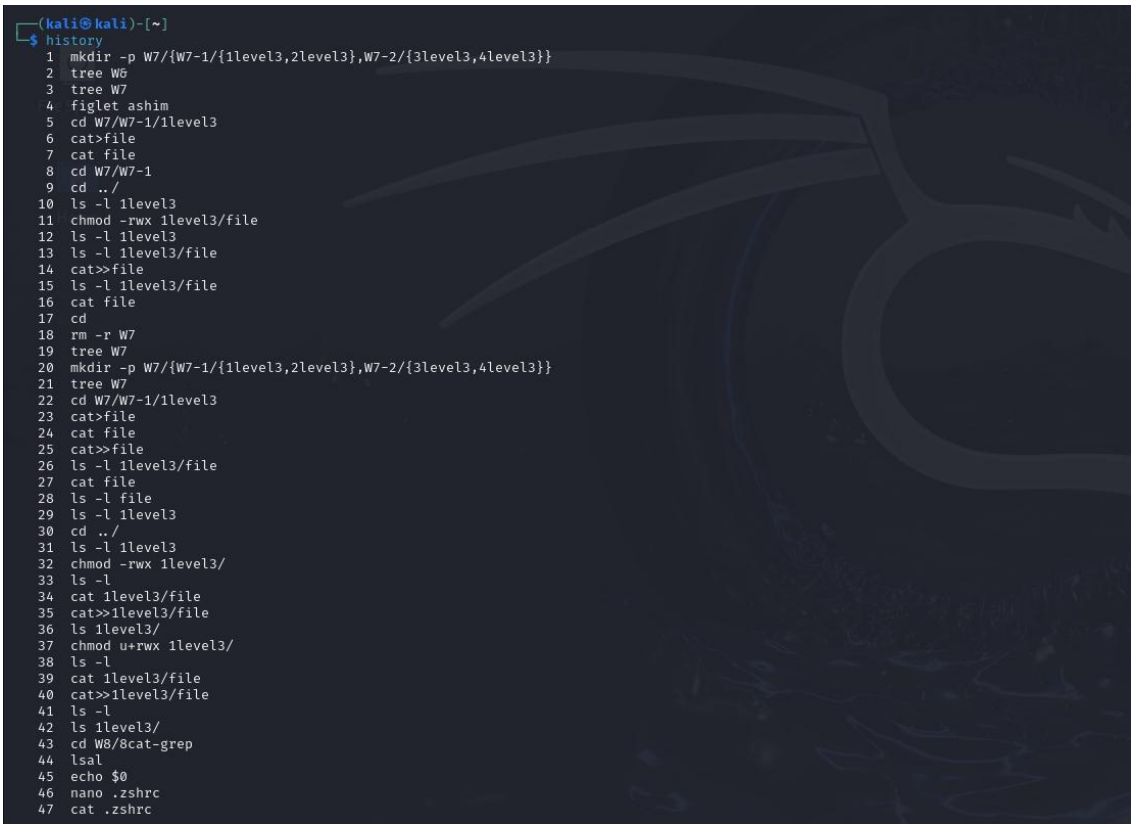

Figure 20: List of commands executed giving the history command

12.

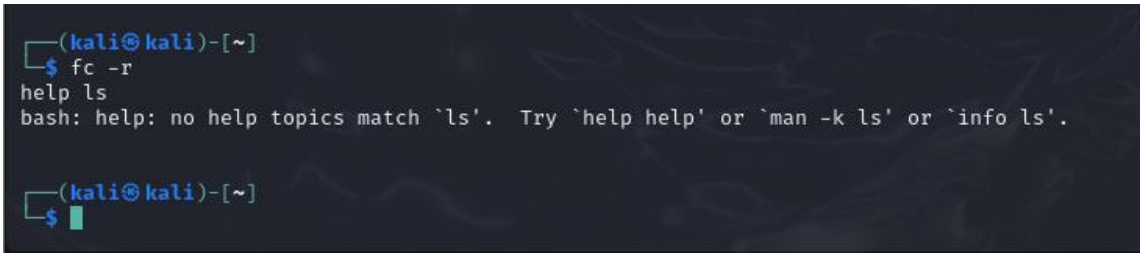

Figure 21: Re-executing the last but one command using the redo(r) command and the number of the event

13.

Figure 22: Re-executing the command !-3 three commands ago using the negative integer

14.

Figure 23: Re-executing the last command which name begins with "l"

Conclusion

In this workshop, we explored a range of essential UNIX utilities that are fundamental for efficient system management and file handling. We gained practical experience in creating and navigating directories using relative and absolute pathnames. Additionally, we learned how to use the cat command to create files, which formed the foundation for later tasks involving text manipulation. By practicing with the grep command, we explored its various options for searching text patterns within files, such as case-sensitive, case-insensitive, and line-numbered searches. These exercises provided valuable insights into how powerful and versatile the grep tool is, especially for data analysis and system administration.

We also learned the importance of managing aliases in UNIX to simplify and optimize our workflow. Through this workshop, we defined custom aliases for commonly used commands, reducing the time spent on repetitive tasks. We also explored how to preserve aliases across sessions, ensuring that our environment remains efficient even after a reboot. This aspect of the workshop helped us understand the flexibility of UNIX in adapting to user needs by providing options for customization, which is crucial for both system administrators and developers working in a UNIX-based environment.

Moreover, the hands-on experience with UNIX's command history feature allowed us to re-execute previous commands quickly and efficiently. This not only saved time but also highlighted how UNIX enables users to work more effectively by offering commands like fc and the history list for easy command retrieval. The combination of these tools—file management, text searching, alias creation, and command history—enhanced our overall understanding of the UNIX system. The skills we have developed through this workshop will undoubtedly be valuable in future tasks involving system administration, scripting, and automation, where efficiency and precision are essential.

References

Brian W. Kernighen, R. P. (1984). Introduction to UNIX and its Philosophy. In R. P. Brian W. Kernighen, *The Unix Programming Environment* (p. 368). Prentice Hall.

Evi Nemeth, G. S. (2017). Exploring UNIX and LInux Utilities. In G. S. Evi Nemeth, *Unix and Linux System Administration Handbook* (p. 134). Pearson Education.

Shotts, W. (2019). Exploring UNIX Command Line Tools for Workflow Optimization. In W. Shotts, *The Linux Command Line: A Complete Introduction* (p. 251). No Starch Press.