Module Code & Module Title

CT5052NP Network Operating System

Assessment Type

Logbook 7

Semester

2023/24 Spring/Autumn

Student Name: Pujan Jung Thapa

London Met ID: 23057035

Assignment Due Date: 21 December 2024

Assignment Submission Date: 21 December 2024

Submitted To: Mr. Prasant Adhikari

Word Count (Where Required):

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.

Contents

Table of figures

1.Introduction

UNIX provides a variety of operating system commands for working with files, administering the system, and interacting with the user. Basic commands such as ls (listing a directory), cd (changing between directories), and cp (copying) are quite common. Other related operations include changing file permissions (chmod), process (kill), and inter-system communications. Basic principles needed to use the UNIX Commands effectively include the UNIX filesystem, shell environments, and regular expressions. Although traditional UNIX commands cannot support multiple processors, several parallel versions have been produced for high-performance computing environment. Examples of such utilities include the parallel versions of ls, ps, and find. These Parallel Unix Commands use MPI (Message Passing Interface) for good scalable performance on multi-node clusters. Detailed information about the various commands is available through the command manual pages, which a user can view using the 'man' command. (Mann, 2001)

1.2 Access Permissions:

Access permissions for files are an important part of data security and user management in multi-user environments. They ensure that data is accessible only to those who need it, and modifiable by only those who should modify it. The specification of permissions by a programmer is cumbersome. To alleviate this, developed a static analysis method using abstract interpretation to automatically infer access permissions. Their solution computes symbolic bounds of permissions to heap locations and outputs a constraint system solved by a linear program solver. The approach explicitly supports different permission systems, for which also promising case studies were reported. To alleviate the effort for non-experts to gain a comprehension and better handle file system permissions, provided a treemap visualization for. Their solution, though illustrated for NTFS, but applicable to other hierarchical file systems, provides an intuitive method of visualizing and maintaining access control permissions. (Heitzmann, 2008 Septembe 15)

1.3 Directories

The basic components of the Linux file system include directories, which hold and manage files and other directories hierarchically. It makes a system tidy and orderly and is quite effective since many similar kinds of files will be together. A directory thus acts like a path toward the root to reach or manipulate the data on a computer without confusion while traveling across the file system. Understanding how to create, modify, and remove directories, as well as manage their permissions, is essential for effectively working within a Linux environment. Directories play a crucial role in ensuring data organization and access control. (Nemeth, 2017)

1.4 Aim

This session of the workshop aims to familiarize us with more Linux commands. The session aims to enhance the commands syntax and understanding of the Linux commands for us which we need to learn for future projects.

1.5 Objectives

The main objective of this log is To gain proficiency in using Linux by working with shell commands, understanding file structures, creating, modifying, and removing files and directories, and managing access permissions to ensure secure and efficient file handling.

2 Questions for log 7

2.1 Create the directory structure presented in the figure below. Use mkdir command and relative pathnames from your home directory. Try both: no option and –p option, for the command.

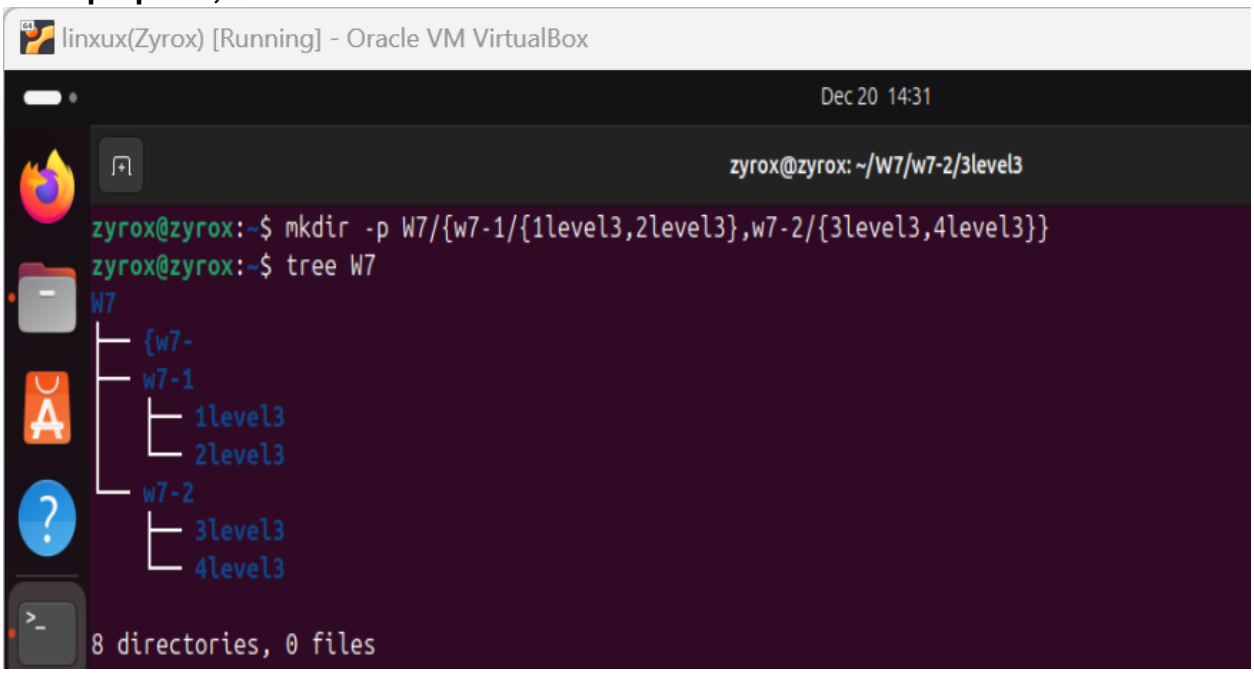

Figure 1: Question 1

2.2 Change to the 1level3 directory by one step using a relative pathname.

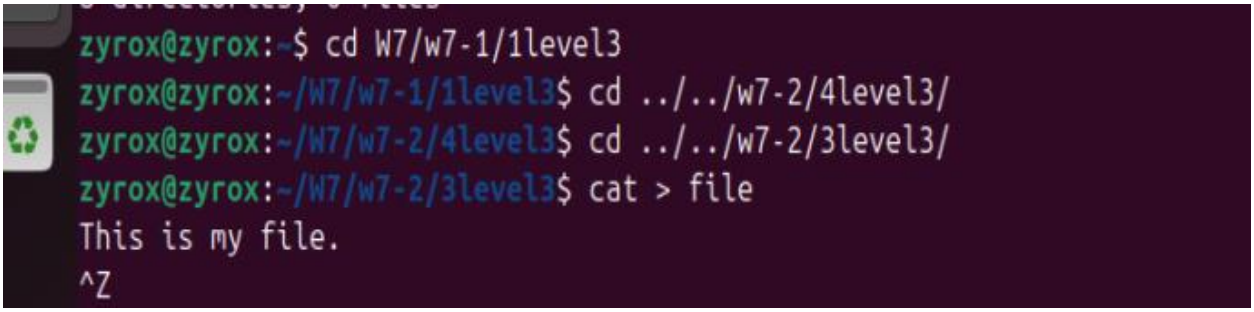

Figure 2:Question 2

2.3 Practice in changing directories in your directory structure by one command using relative pathnames, e.g., from 1level3 to 2level3, from 2level3 to 4level3, from 4level3 to W7, etc. Use names of parent and child directories ('.' and '..') as well.

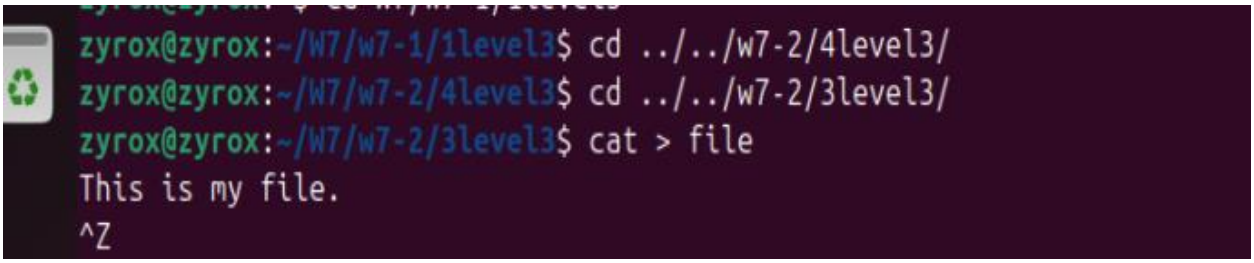

Figure 3: Question 3

2.4 Change to 1level3 and create a text file by any tool (e.g., by cat or cal like last tutorial).

Figure 4: Question 4

2.5 Copy this text file from 1level3 to 1level3 (with the name file1), 2level3, and to 3level3 changing its name. Show that there are these files in corresponding directories.

Figure 5: Question 5

2.6 Move this file to 4level3. Show that there is this file in 4level3 and there is not in 1level3.

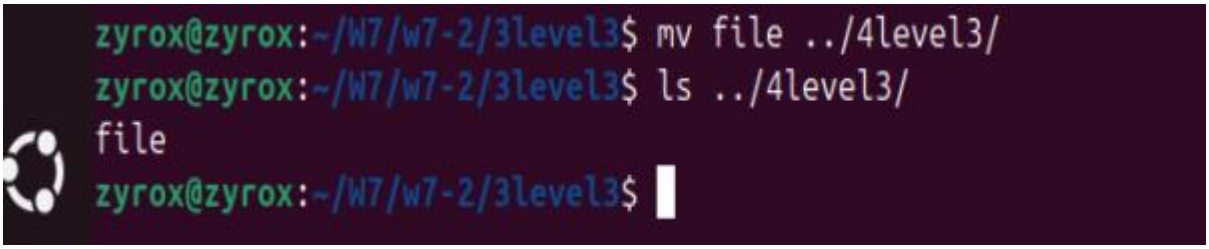

Figure 6: Question 6

2.7 Print the following texts each in one echo or printf command:

• Hello! I can do it

• 5 > (20: 8) < (30 * 2)

• Line 1 Line 2 a-b, A-B, –, +,

• , #, $, %, &.

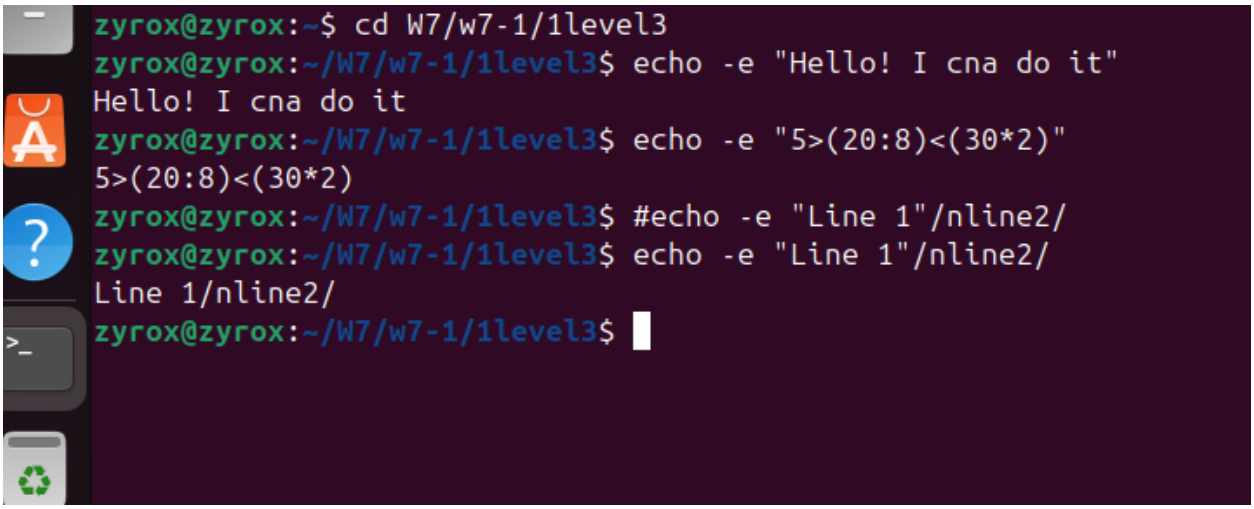

Figure 7: Question 7

2.8 Give the ls command (without options and with a, d, g, l, R options) in home directory, w7, w7-1, and 1level3 directories. Explain for yourself the results received.

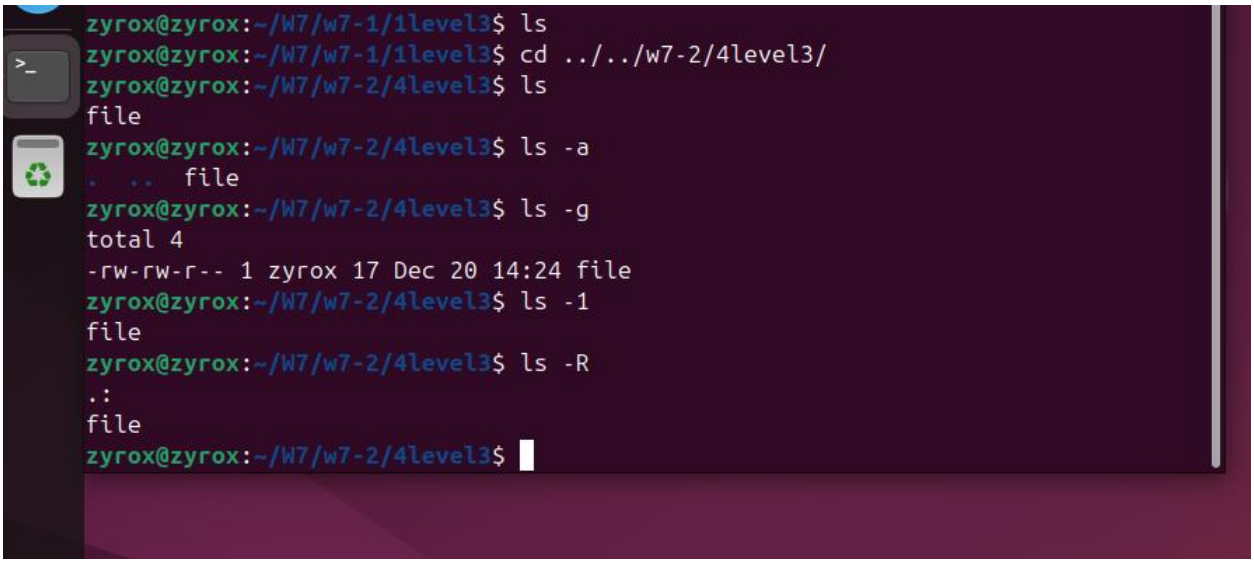

Figure 8: Question 8.1

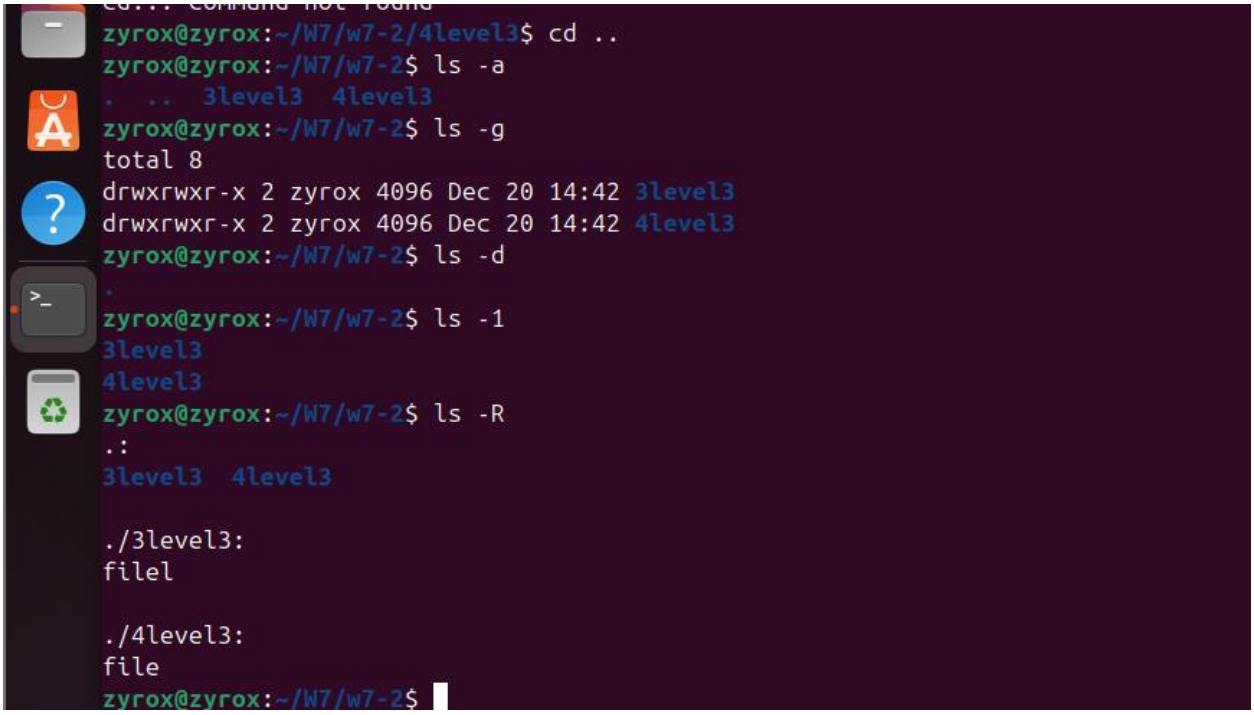

Figure 9:Question 8.2

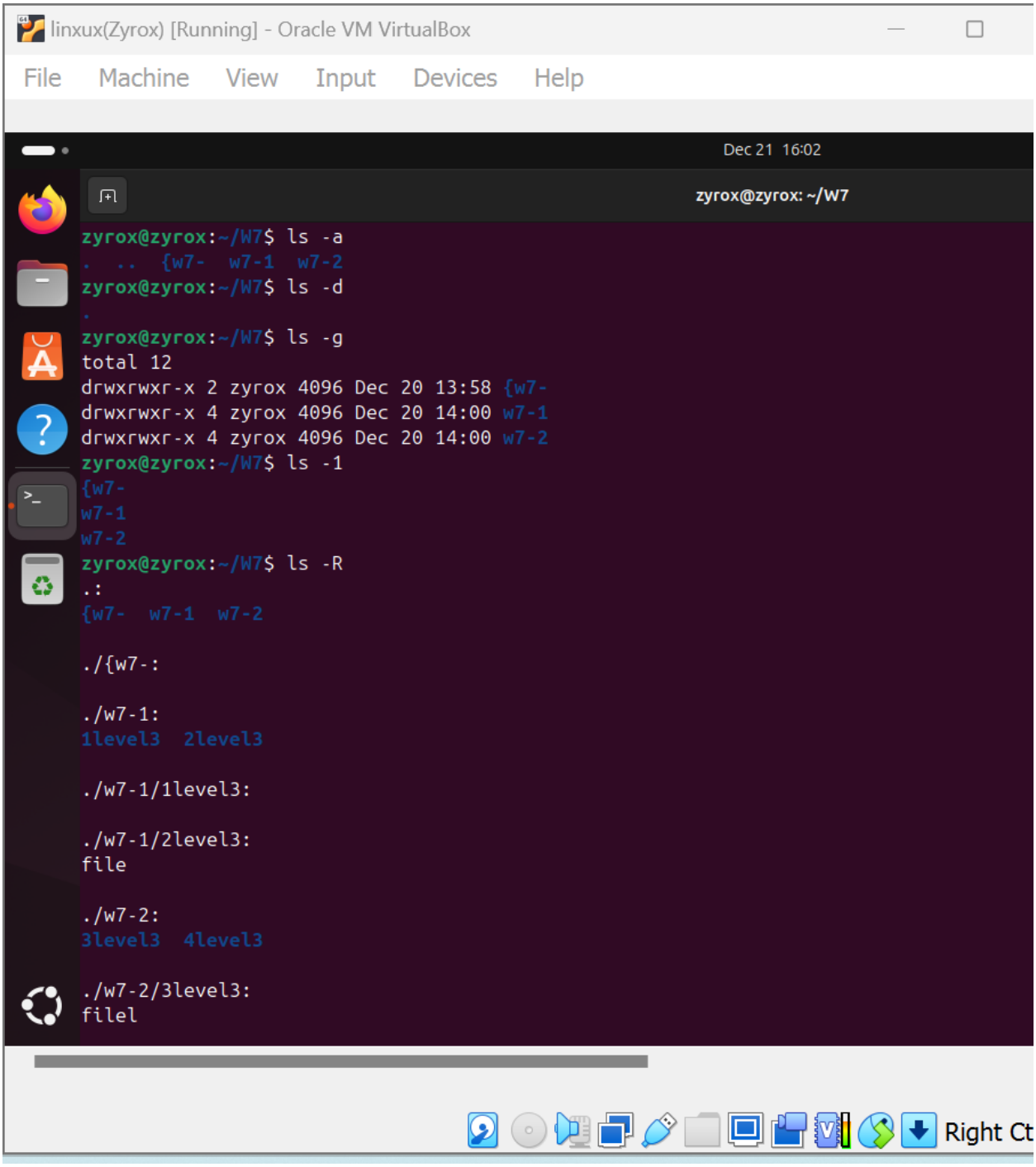

Figure 10: Question 8.3

2.9 Change to the W7 directory. Remove the directory files w7-2, 3level-3, 4level3 and all ordinary files in them. Use the option –i of the rm and rmdir commands. Show that there are not these ordinary and directory files in your file structure.

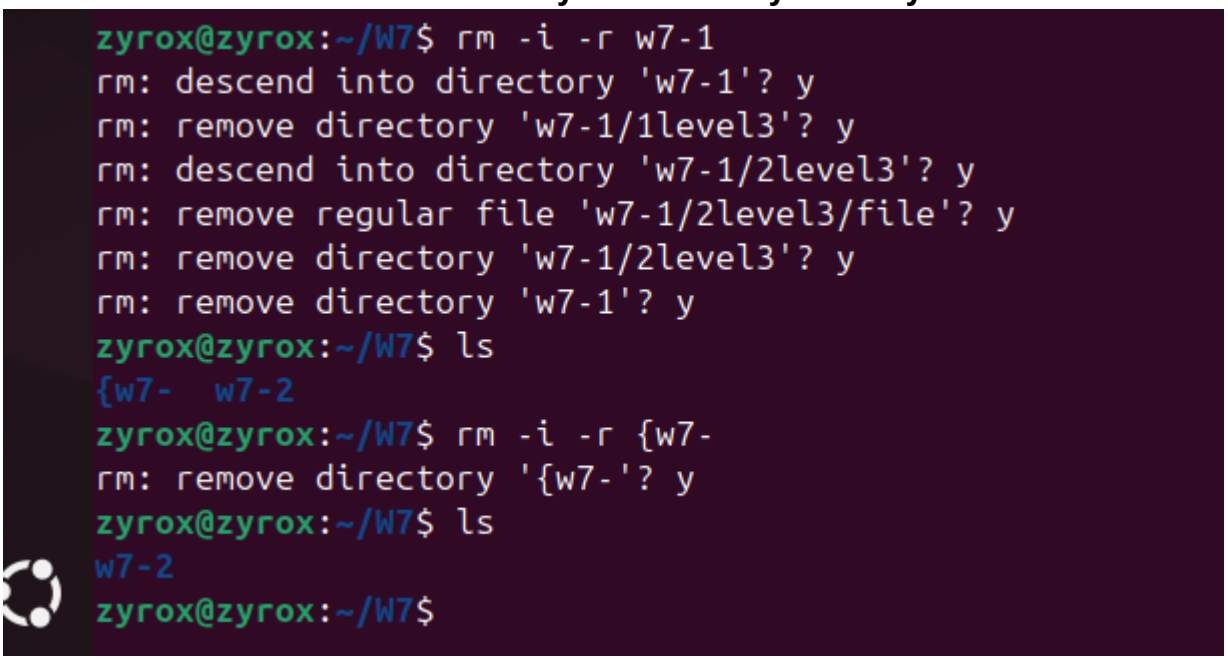

Figure 11: Question 9

2.10 Change to w7-1.

• Display access permissions for the file file1 in 1level3.

• Remove all access permissions for this file.

• Display access permissions for this file.

• Try to read this file using any utility (e.g., cat).

• Try to write into this file using any utility (e.g., cat with the sign >> – append).

• Add read and write access permissions for yourself for this file.

• Display access permissions for this file.

• Try to read this file using any utility.

• Try to write into this file using any utility.

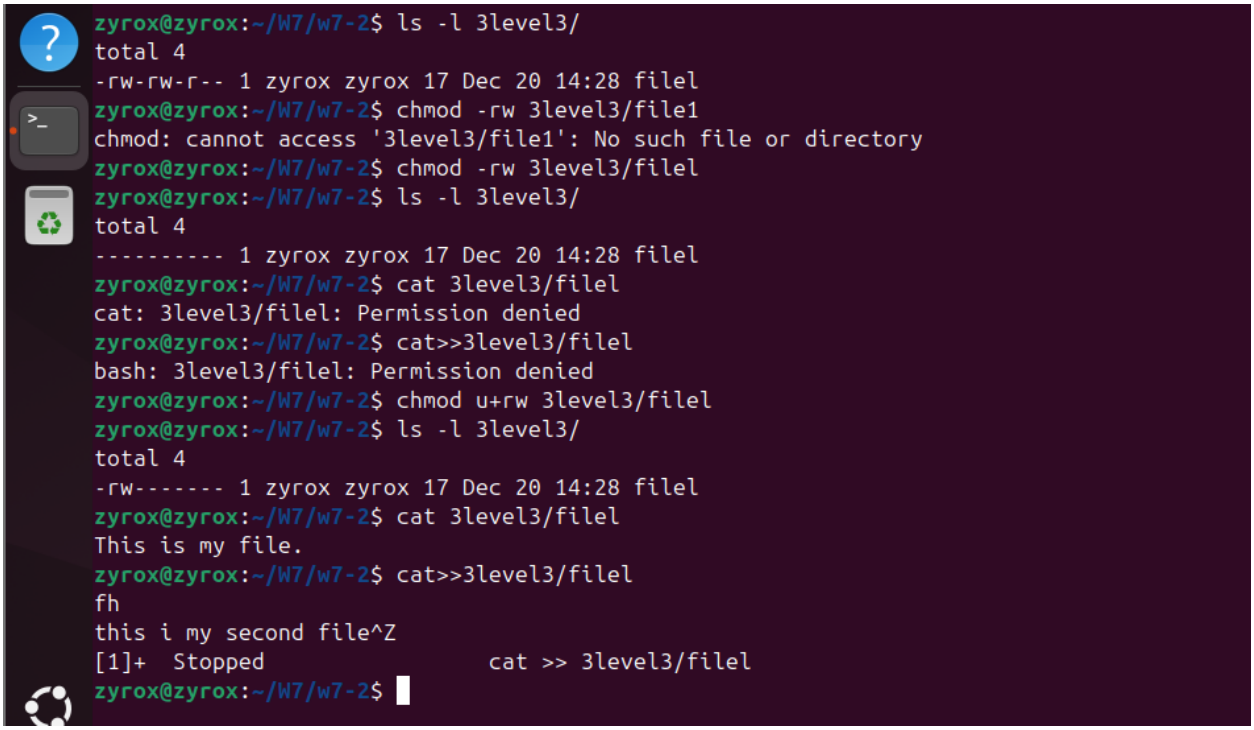

Figure 12: Question 10

2.11 (Now,)

• Display access permissions for 1level3.

• Remove all access permissions for the 1level3 directory.

• Display access permissions for 1level3.

• Try to read a file from 1level3 using any utility.

• Try to put a file into 1level3 using any utility.

• Try to search in 1level3 using any command (e.g., the ls command).

• Add read, write, and execute access permissions for yourself for the 1level3 directory.

• Display access permissions for 1level3.

• Try to read a file from 1level3 using any utility.

• Try to put a file into 1level3 using any utility.

• Try to search in 1level3 using any command (e.g., the ls command).

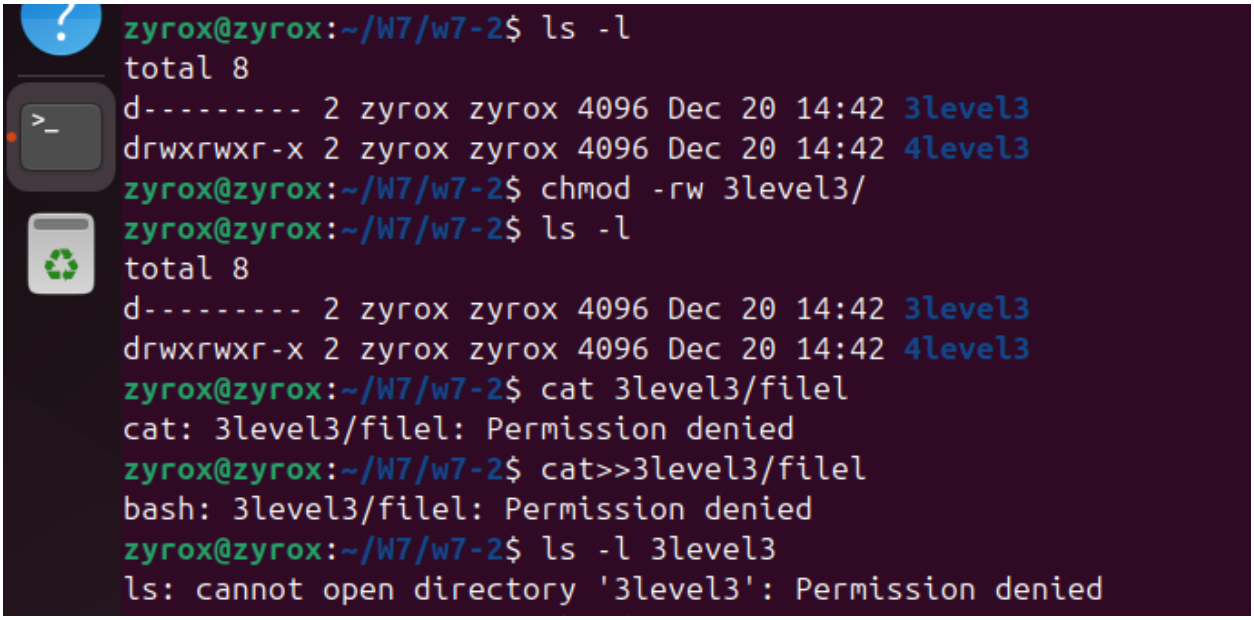

Figure 13:Question 11

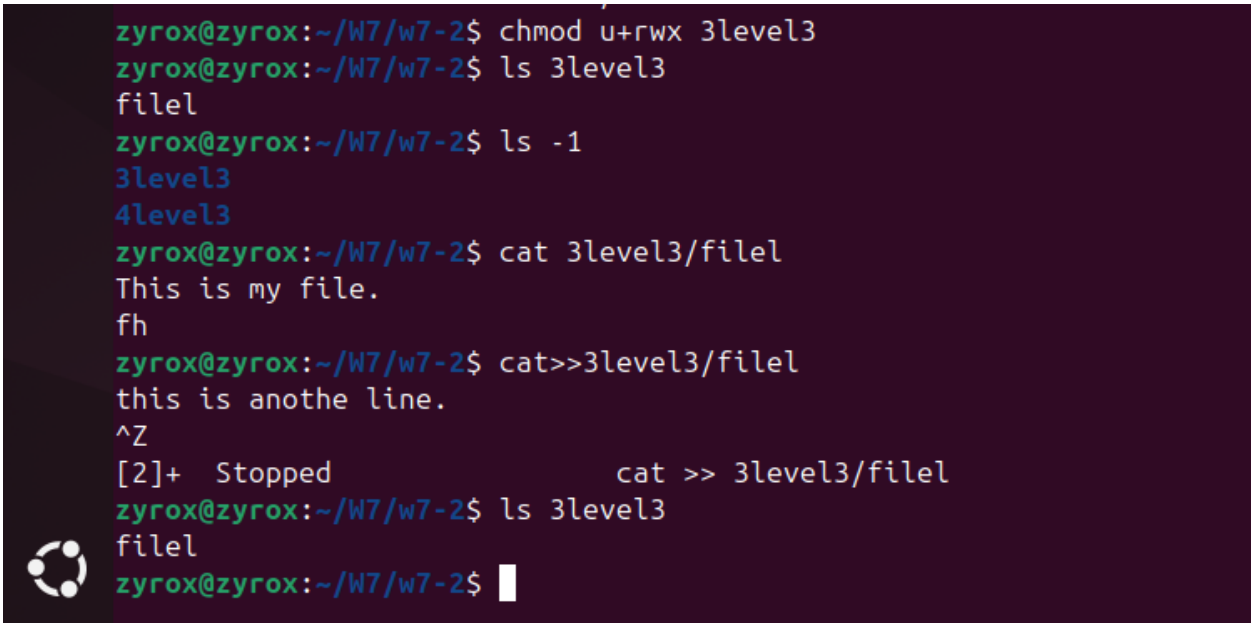

Figure 14:Question 11.1

3 Conclusion

This workshop session was very important to gain knowledge with regard to simple file and directories creation, modification and deletion. The file structure system and the process of creating a tree was also understood. Methods to modify access permissions of users were also administered. The session was successfully completed with no to minimal complications. The workings and findings of this session will surely provide a certain impact in the overall learning outcomes of this module.

4 References (wwww, n.d.)

Heitzmann, A. B. P. C. P. a. R. T., 2008 Septembe 15. In: *Effective Visualization of File System Access-Control.* s.l.:Visualization for Computer Security, pp. 18-25[Accessed 2024 12 21].

Mann, J., 2001. In: *Using UNIX,.* s.l.:Biomedical instrumentation & technology, pp. 52-349. [Accessed 2024 12 21].

Nemeth, E. S. G. H. T. W. B. a. M. D., 2017. In: *UNIX and Linux System Administration Handbook. .* Boston: Addison-Wesley Professional. [Accessed 2024 12 21].