"Kyiv Professional College of Communication"

Cycle Commission of Computer Engineering

REPORT ON THE IMPLEMENTATION

LABORATORY WORK №10

in the discipline: "Operating systems"

Topic: "Changing file owners and permissions in Linux. Special directories and files in Linux"

Performed by

student

of BICS-13 group

Zlata Volodymyrivna Los

Checked by the teacher

Sushanova V.S.

Kyiv 2024

**Objectives:**

1. Gaining practical skills in working with the Bash shell.

2. Familiarization with basic actions when changing file owners, file permissions.

3. Familiarity with special directories and files in Linux.

**Material support of classes:**

1. Computer such as IBM PC.

2. Windows operating system and Virtual Box (Oracle) virtual machine.

3. GNU/Linux OS (any distribution).

4. The Cisco Network Academy website netacad.com and its online Linux courses.

**Tasks for preliminary preparation:**

1. \*Read the brief theoretical information for the laboratory work and make a small dictionary of basic English terms on the purpose of commands and their parameters.

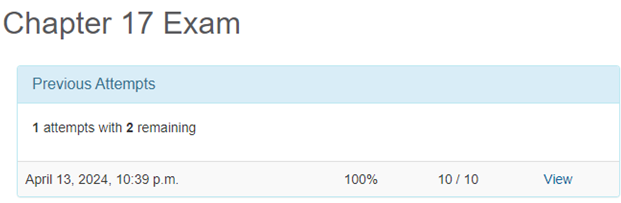
2. Study the materials in the NDG Linux Essentials online course (netacad.com):

- Chapter 17 - Ownership and Permissions

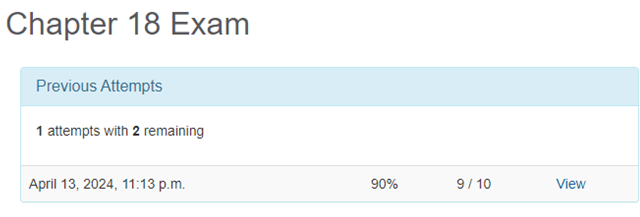
- Chapter 18 - Special Directories and Files

3. Take the NDG Linux Essentials quizzes on the following topics:

- Chapter 17 Exam



- Chapter 18 Exam



4. Based on the material reviewed, answer the following questions:

4.1 What is the purpose of the id command?

The id command is designed to display information about the user's ID and the group to which he belongs.

4.2 How do I view what access rights the file owner has?

To view the access rights of the file owner, you can use the ls -l command, which displays a line starting with access permissions. The owner's permissions are displayed on the first line of this output.

4.3 \*How do I change the group owner?

To change the owner of a group, you can use the chown command, for example: 'sudo chown :newgroup file'.

4.4. \*How can I see in the terminal what type of file is currently being used? Give examples for different file types.

To view the file type in the terminal, you can use the file command. For example:

- file text.txt for a text file

- file image.jpg for an image

- file program.exe for an executable file

4.5 \*\*What are the Setuid and Setgid permissions used for?

The Setuid and Setgid permissions are used to ensure that a program is executed with the rights of the owner or group of a file, respectively. This allows users to run programs with elevated privileges without having to enter their password.

4.6 \*\*What is the purpose of the so-called "Sticky Bit" in the system. Give examples of when this permission should be used.

"Sticky Bit is used to control access to folders in shared environments. It ensures that only the owner of a file can delete or rename it, even if other users have write permissions to that folder. Examples of using the sticky bit include the /tmp folder, where this permission allows all users to create files, but they can only delete their own files.

5. Prepare the initial version of the report in electronic form:

- Cover sheet, topic and purpose of the work

- Glossary of terms

- Answers to p.4.1-4.6 from the preliminary preparation tasks

**Procedure:**

1. Initial work in CLI mode in the Linux OS of the Linux family:

1.1. Start the VirtualBox virtual machine, select CentOS, and start it. Log in to the system as a user: CentOS, login password: reverse (if you are performing the LP in 401) and close the terminal.

1.2. Start the Ubuntu\_PC virtual machine (if you are performing the PL task through the netacad academy)

1.3. Start your Linux operating system (if you are working on your own PC and have installed it) and launch the terminal.

2. Work through all the sample commands presented in the labs of the NDG Linux Essentials course: Lab 17: Ownership and Permissions and Lab 18: Special Directories and Files. Create a table of commands learned in step 2 of the workflow in the following form:

|  |  |
| --- | --- |
| **Name of the commands** | **Its purpose and functionality** |
| chmod | A command for changing access rights to files and directories. For example, it allows you to set or change the read (r), write (w), and execute (x) permissions for files for a user, group, and other users. |
| chown | Command to change the owner and group of a file or directory. |
| chgrp | Command to change the owner group of a file or directory. |
| mkdir | Command to create new directories (folders) in the file system. |
| ls | Command to display a list of files and directories in the current working directory. |
| ls -la | An extended version of the ls command that displays a list of files and directories with detailed information about access rights, owner, group, etc. |
| ls -ld | Displays information about the directory itself, not its contents. |
| date | Command to display or set the current date and time. |
| stat | Command to display detailed information about a file or directory, such as size, indexing, owner, group, file type, etc. |
| passwd | Command to change the user's password. |
| wall | A command to send a message to all users on the system. |
| ln | Command to create links to files or directories. |
| rm | Command to delete files or directories. |

**3. Perform the following practical tasks in the terminal as follows (show screenshots):**

- create three new users;

- create a new user group, add two of the three created users to it;

- create a new file that will be available for reading, editing, and execution by the file owner, for example, a simple script;

- for users of the owner's group, grant permissions to view and execute (without permission to edit) this file;

- for other users, deny access to this file;

- \*do the same for directories - create a directory that will be available to all three users, create a directory that will be available only to the owner, create a directory that users of the owner's group will be able to view but not edit;

- create an empty file called emptyfile using the touch emptyfile command. Now "zero out" the permissions for the file with chmod 000 emptyfile. What happens if we change the permissions for emptyfile by passing only one value for chmod in numeric mode, such as chmod 4 emptyfile? What happens if we use two numbers, such as chmod 44 emptyfile? What can we learn about how chmod reads a numeric value?

- Create a directory named where all files automatically belong to your user group and can only be deleted by the user who created them?

- Create one new file under each user, and a hard and symbolic link to it;

- Have other users try to view these files;

- have other users try to delete these files and draw conclusions.

**Control questions:**

**1. What are some examples of changing access rights using the Symbolic Method?**

- chmod u+x file: Adds the right to the owner to execute the file.

- chmod g-w file: Removes the group's right to write to a file.

- chmod o=rw file: Sets read and write permissions for other users.

**2. What are some examples of changing access rights using the numeric method (numeric method, octal method)?**

- chmod 755 file: Sets permissions to rwxr-xr-x for the owner, and r-xr-xr-x for the group and other users.

- chmod 644 file: Sets the permissions to rw-r--r-- for the owner, and r--r--r-- for the group and other users.

**3. What is the purpose of the umask command?**

The umask command sets or displays the default file mask for creating new files. It subtracts the specified bits from the default permissions.

**4. Compare hard links and symbolic links?**

- Hard links: A file reference that points to the same location of data on the file system. Deleting the original file does not affect a hard link.

- Symbolic links: A link that contains the path to the original file. Deleting the original file will invalidate the symbolic link.

**5. \*Can I execute a file that has execute privileges but not read privileges (--x)? Explanation.**

No, a file that has runtime but not read permissions cannot be executed because the operating system will not be able to read its contents to execute it.

**6. \*If we change access rights and permissions in the current session, will they be saved in the next session?**

Changes to access rights made in the current session are usually not saved after the session is closed. They are only applied in the current environment.

**7. \*Is there a template that the system uses for permissions and accesses when creating new files? How can I change the default permissions?**

Usually, when creating new files on UNIX-like systems, the permissions defined by the umask mask are applied. The umask can be changed to set the default permissions for all new files.

**8. \*How can I create a hard link? In what situations is it advisable to use them?**

A hard link can be created using the ln command. They are used when you want to have multiple names for the same file on the same file system.

**9. \*How can I create a symbolic link? In what situations is it advisable to use them?**

Symbolic links are also created with the ln command, but with the -s option. They are used to create links to files or directories that can switch between different file systems.

**10. \*\*Imagine that a program needs to create a one-time temporary file that will never be needed again after the program is closed. What is the correct directory to create this file?**

The correct directory to create a one-time temporary file is /tmp, because this directory is for temporary files that are deleted after the system is shut down or rebooted.

**11. \*\*I have an original file and two links have been created for it - a symbolic link and a hard link. What will happen to the other files if you delete them?**

- original file: deleting the original file will not affect the hard link because they point to the same data

- symbolic link: deleting a symbolic link will not affect the original or other links

- hard link: deleting the original file will invalidate the symbolic link

**Conclusion:** I gained practical skills in working with the Bash command shell, learned the basic actions for changing file owners and file access rights, and got acquainted with special directories and files in Linux.