**"Kyiv Vocational College of Communication"**

**Cyclic Commission of Computer Engineering**

**EXECUTION REPORT**

**LABORATORY WORK No. 10**

from the discipline: "Operating systems"

**Topic: "Changing owners and access rights to files in Linux. Special directories and files in Linux”**

**Performed by students of the group:**

Барабаш Матвій  
Погребняк Ілля

Целуйко Станіслав

**Checked by the teacher**

Sushanova V.S

**Work of group students КСМ-13Б Team:** **PMC wolf group**

**The goal of the work:**

1. Getting practical skills for working with the Bash command shell.

2. Familiarity with basic file system navigation commands.

3. Familiarity with basic commands for managing files and directories.

**Material provision of classes**

1. IBM PC type computer.

2. OS family Windows (Windows 7).

3. Virtual machine - Virtual Box (Oracle).

4. GNU/Linux operating system - CentOS.

5. Cisco network academy site netacad.com and its online Linux courses

**Tasks for preliminary preparation:**

**Ilya Pogrebnyak was looking for material**

**4.1. What is the purpose of the id command?**

The id command is used to get information about the current user.

It displays the following data on the screen:

- User ID (UID)

- Group ID (GID)

- User name

- User authority

**4.2. How to see what access rights the owner of a file has?**

\* You can use the ls command to view file access rights. In this case, you should use the -l option, which displays extended information about the file.

\* For example, to view access rights to the /etc/passwd file, you can execute the following command:

**ls -l /etc/passwd**

\* The resulting result will look something like this:

**-rw-r--r-- 1 root root 1299 2023-11-30 14:06 /etc/passwd**

\* The first three characters in the line determine the access rights to the file.

- In this case, the access rights are as follows:

**- - the file is normal**

**rw- - the owner has read and write rights**

**r-- - the group has read access**

**r-- - other users have read access**

**4.3. How to change the group owner?**

\* To change the owner of a file group, you can use the chown command. At the same time, you should specify the new owner of the group and the file to which the access rights need to be changed.

\* For example, to change the owner of the /etc/passwd file group to the users group, you can execute the following command:

**chown users /etc/passwd**

**4.4. How can you view the current file type in the terminal?**

\* You can use the file command to view the file type. It displays the file type and its possible use.

**\* For example, to view the type of the /etc/passwd file, you can execute the following command:**

file /etc/passwd

**\* The resulting result will look something like this:**

/etc/passwd: ASCII text, with very long lines

**4.5. What are Setuid and Setgid permissions used for?**

\* Setuid and Setgid permissions allow you to execute a program with the access rights of the file's owner or file's group.

\* Setuid permissions are used to run root applications. This can be useful, for example, for running programs that need access to system files or devices.

\* Setgid permissions are used to run programs with file group permissions. This can be useful, for example, for running programs that use shared resources, such as files or directories.

**4.6. Why does the system need the so-called "sticky bit" (Sticky Bit). Give examples of when this permission should be used.**

\* Sticky Bit allows only the file owner and users who are members of the file owner's group to delete or move the file.

\* This permission should be used for files that should be accessible only to a certain group of users. For example, for files that contain data that must be protected from unauthorized access.

**\* Here are some examples of when it makes sense to use a “sticky bit”:**

- A file with the configuration of the program, which is available only to a certain group of users.

- An error log file that is accessible only to system administrators.

- A file with a backup copy of data that is only available to users who are responsible for its creation and storage.

**Progress  
1)** **The table was made by Barabash Matviy**

Work through all the command examples presented in the NDG Linux course labs Essentials:

- Lab 17: Ownership and Permissions

- Lab 18: Special Directories and Files

**Commands**

**Lab 17: Ownership and right of access**

**\* Example 1: View file access rights**

ls -l /etc/passwd

-rw-r--r-- 1 root root 1299 2023-11-30 14:06 /etc/passwd

**Example 2: Changing the owner of a file**

chown users /etc/passwd

**Example 3: Changing the file group**

chgrp users /etc/passwd

**Example 4: Granting write access to other users**

chmod o+w /etc/passwd

**Example 5: Granting the execute right to all users**

chmod a+x /etc/passwd

**Example 6: Granting execute permission to the file's owner and group**

chmod ug+x /etc/passwd

**Example 7: Granting execute rights to the file owner and other users**

chmod u+x,o+x /etc/passwd

**Example 8: Granting execute permission to all users except the owner of the file**

chmod o+x,u-x /etc/passwd

**Lab 18: Special directories and files**

**Example 1: Viewing a custom directory**

ls /dev/disk/by-id/

**Example 2: Viewing a custom file**

cat /proc/cpuinfo

**Example 3: Viewing system status**

top

**Example 4: Viewing the process**

ps -ef

**Example 5: Starting the program**

sudo /usr/bin/vim

**Example 6: Exiting the program**

:q!

**Example 7: Restarting the system**

sudo reboot

**Example 8: Shutting down the system**

sudo poweroff