“Київський фаховий коледж зв’язку”

Циклова комісія Комп’ютерної інженерії

**ЗВІТ ПО ВИКОНАННЮ**

**ЛАБОРАТОРНОЇ РОБОТИ №5**

з дисципліни: «Операційні системи»

**Тема: «Знайомство з командами навігації по файловій системі та керування файлами та каталогами»**

Виконавли студенти

групи РПЗ-03

Команда: Губенко Є.О.,

Заїка С.В. та Кресан Р.А.

Перевірив викладач

Сушанова В.С.

Київ 2022

**Мета роботи:**

1. Отримання практичних навиків роботи з командною оболонкою Bash.
2. Знайомство з базовими командами навігації по файловій системі.
3. Знайомство з базовими командами для керування файлами та каталогами.

**Матеріальне забезпечення занять**

1. ЕОМ типу IBM PC.

2. ОС сімейства Windows (Windows 7).

3. Віртуальна машина – Virtual Box (Oracle).

4. Операційна система GNU/Linux – CentOS.

5. Сайт мережевої академії Cisco netacad.com та його онлайн курси по Linux

**Завдання для попередньої підготовки**

***Готував матеріал студент Губенко Є.О***

1. Прочитавши матеріал з коротких теоретичних відомостей дайте відповіді на наступні питання:
   1. Порівняйте файлові структури Windows-подібної та Linux-подібної системи.

*The file structures of Windows-based and Linux-based systems differ in several ways:*

1. *File hierarchy:*

*Windows uses a file hierarchy with a single directory tree. The most well-known directories include Program Files, Windows, Users, and Documents and Settings.*

*In Linux, the file hierarchy has many levels and starts with the root directory "/". Typical directories include "/bin", "/etc", "/home", "/usr", and "/var".*

1. *File extensions:*

*On Windows, file extensions are appended to the file name after a period separating the name and extension (e.g., "file.txt").*

*On Linux, file extensions are optional, and a file name can consist of just a name without a period (for example, "file").*

1. *File systems:*

*On Windows, NTFS and FAT32 are the most common file systems.*

*On Linux, EXT4 and Btrfs are the most popular file systems, and there is support for others such as XFS, ZFS, and NTFS.*

1. *Manage file permissions:*

*On Windows, file and folder permissions are managed using ACLs (Access Control Lists).*

*On Linux, file and folder permissions are controlled by file access levels, which are set for owner, group, and all users.*

1. *Command line:*

*On Windows, the command line is known as the Windows Command Prompt (CMD), and uses commands that can be executed at the CMD.*

*On Linux, the command line is known as a shell, and the most common shell is bash, which uses special commands that can be executed from the command line.*

1. *Symbolic links:*

*In Windows, symbolic links are known as shortcuts, which are links to a file or folder and can be placed anywhere.*

*In Linux, symbolic links are files that contain a link to another file or folder and can be created anywhere.*

1. *Partitioning a disk:*

*In Windows, you can partition a disk into several partitions that appear as separate disks with different drive letters.*

*In Linux, you can partition a disk into several partitions that appear as separate partitions in the directory system.*

1. *Setting up the system:*

*On Windows, system configuration is usually done through the control panel or other graphical interfaces.*

*On Linux, system configuration is usually done through configuration files located in /etc or via the command line.*

1. *System size:*

*On Windows, the system typically takes up more disk space because of the large number of installed applications and drivers.*

*On Linux, the system usually takes up less disk space because it has fewer applications and drivers installed by default.*

1. *Availability of applications:*

*On Windows, many programs and games are developed specifically for that operating system, which can make them more accessible to users.*

*On Linux, some programs may be limited in terms of accessibility, but open source allows developers to create and distribute their software for free, which can make them more accessible to users.*

1. *System requirements:*

*On Windows, the system requirements can be high, depending on the version of the operating system and the applications that need to be installed.*

*On Linux, the system requirements are usually lower because the operating system can run on less powerful computers.*

1. *Console:*

*In Windows, the console (command prompt) is available to users, but it does not have the same power as the console in Linux.*

*In Linux, the console is a powerful tool for interacting with the operating system and performing various tasks.*

1. *Access to files:*

*On Windows, access to files can be restricted by a security system that uses ACLs (Access Control Lists).*

*In Linux, access to files can be controlled by using permissions that are set for the owner, group, and other users.*

1. *Antivirus protection:*

*On Windows, antivirus protection is an important aspect of security because this operating system is more susceptible to viruses and malware.*

*On Linux, antivirus protection may be less important as this operating system has more security by default.*

1. *Driver support:*

*In Windows, driver support is usually provided by hardware manufacturers who develop drivers for their devices.*

*On Linux, drivers are usually found in the operating system kernel and are updated with it.*

*In general, Windows and Linux have different ways of working with the operating system and different advantages and disadvantages. Windows usually has a simpler and more intuitive user interface, as well as a wider selection of programs and applications, but it can be less stable and secure. Linux usually has greater stability and security, as well as more controllability and customization, but it can be less accessible to newcomers and have a limited selection of programs and applications.*

* 1. Розкрийте поняття FHS. Як даний стандарт використовується в контексті файлових систем?

*FHS is an acronym for Filesystem Hierarchy Standard, which translates to File System Hierarchy Standard. It is a standard that describes how files and directories should be organized in the file system in Unix-like operating systems.*

*The FHS is an important standard in Unix-like operating systems because it provides a standard file system hierarchy that allows program developers and system administrators to work with files and directories more easily and efficiently.*

*According to the FHS, the file system of a Unix-like OS should have the following hierarchy:*

* *- /: root directory*
* *- /bin: system binaries*
* *- /sbin: system binaries*
* *- /boot: files needed to boot the system*
* *- /dev: device files*
* *- /etc: system configuration files*
* *- /home: user home directories*
* *- /lib: libraries for programs*
* *- /media: mounted devices*
* *- /mnt: temporarily mounted file systems*
* *- /opt: additional programs*
* *- /proc: information about processes*
* *- /root: home directory of the root user*
* *- /run: runtime files*
* *- /sbin: system binaries*
* *- /srv: data for services*
* *- /sys: device and kernel information*
* *- /tmp: temporary files*
* *- /usr: additional programs and files*
* *- /var: system variable files*

*Compliance with the FHS standard allows application developers and system administrators to more easily locate and manage files and directories on a system. In addition, compliance with the standard provides greater portability of programs between different Unix-like operating systems.*

* 1. Перерахуйте основні команди для роботи з файлами та каталогами в Linux: створення, переміщення, копіювання, видалення.

*The main commands for working with files and directories in Linux include the following:*

* *Create files and directories:*

*touch file.txt: create an empty file named file.txt*

*mkdir directory: create a directory named directoryMove files and directories:*

*mv source\_file.txt destination/: move the file source\_file.txt to the directory destination/*

*mv source\_directory/ destination/: move the source\_directory directory to the destination/ directory*

*mv old\_file.txt new\_file.txt: rename the file from old\_file.txt to new\_file.txtКопіювання файлів та каталогів:*

*cp source\_file.txt destination/: скопіювати файл source\_file.txt до каталогу destination/*

*cp source\_directory/ destination/: скопіювати каталог source\_directory до каталогу destination/*

* *Видалення файлів та каталогів:*

*rm file.txt: delete the file file.txt*

*rm -r directory: delete the directory and all files and directories in it*

*rmdir directory: delete the empty directory directory*

*It's important to remember that the rm and mv commands can be dangerous because they delete or move files without confirmation. It is recommended that you always check before running these commands to make sure they are executed correctly.*

**Хід роботи**

1. Початкова робота в CLI-режимі в Linux ОС сімейства Linux:
   1. Start the VirtualBox virtual machine, select CentOS, and start it. Log in to the system as user: CentOS, login password: reverse (if you are performing the LP in 401) and launch the terminal.
   2. Start the Ubuntu\_PC virtual machine (if you are performing the PL task through the netacad academy)
   3. Start your operating system of the Linux family (if you are working on your own PC and have installed it) and launch the terminal.
2. Опрацюйте всі приклади команд, що представлені у лабораторних роботах курсу NDG Linux Essentials - Lab 7: Navigating the Filesystem та Lab 8: Managing Files and Directories. Створіть таблицю для опису цих команд\*\*\*?

|  |  |
| --- | --- |
| Назва команди | Її призначення та функціональність |
| pwd | Determines the user's location in the file system, shows the current working directory (print working directory) |
| cd Documents | The cd command navigates to the directory specified as its  as an argument. In this case, it is the Documents directory |
| cd / | cd is a terminal command that is used to change the current working directory. When you execute the cd command, you change to the other directory specified as the command argument. For example, if you run the cd /home/user/Documents command, you change to the /home/user/Documents directory. |
| cd /home | cd /home is a command in the terminal that is designed to change to the /home directory. If you execute this command, you will be taken to the /home directory, which is where users' home directories are usually stored. Running the cd /home command is equivalent to running the cd command without arguments, since /home is the home directory on most Linux systems. |
| cd ~ | cd ~ is a command in the terminal that is designed to switch to the current user's home directory. The ~ sign in the cd command indicates the home directory. This command helps you quickly navigate to your main directory so you can continue working with files and documents. |
| echo ~ ~sysadmin ~root ~mail ~nobody | ~sysadmin, ~root, ~mail, and ~nobody are special abbreviations that denote the home directories for the sysadmin, root, mail, and nobody users, respectively. By using these abbreviations in the echo command, you can print the path to the home directory for these users. For example, if you run the echo ~sysadmin command, you will see the path to the home directory of the sysadmin user. |
| cd ~root | cd ~root is a command in the terminal that is designed to change to the home directory of the root user. The ~ sign indicates the home directory of the current user, so ~root points to the home directory of the root user |
| cd /usr/bin | cd /usr/bin is a command in the terminal that is designed to change to the /usr/bin directory. The /usr/bin path is one of the standard paths in most Linux distributions and contains the system's executable files. |
| cd /usr | cd /usr is a terminal command that is used to change to the /usr directory. The /usr directory contains files that are used to provide features that are not required for the system to work when it starts up, but are required to provide additional features after it starts up. |
| cd /usr/share/doc | cd /usr/share/doc is a terminal command that navigates to the /usr/share/doc directory. The /usr/share/doc directory contains documentation that can be used with programs installed on the system. |
| cd bash | cd bash is a command in the terminal that is designed to change to a subdirectory named "bash" in the current working directory.  However, in most systems, directories named "bash" are not contained in the user's current working directory. Most often, "bash" is the name of a command line interpreter used in most Linux distributions. |
| cd .. | cd.. is a command in the terminal that is designed to switch to the parent directory relative to the current working directory. |
| cd ../dict | The cd ../dict command is used to change to a directory named "dict" in the parent directory relative to the current working directory. |
|  |  |
| ls | The ls command in the terminal is designed to display a list of files and folders that are in the current working directory.  When you call the ls command without any arguments, it displays a list of files and folders in the current directory in plain text. At the same time, files and folders are displayed in the form of their names and some information about them, such as the date and time of the last modification, file size, and access rights. |
| -a | The -a option of the ls command is used to display all files and folders, including those beginning with a dot (hidden files).  By default, the ls command does not display hidden files. However, if you issue the ls command with the -a option, all files and folders that are in the current working directory will be displayed, including files with names beginning with a dot. |
| -l /etc/hosts | The -l option of the ls command is used to display detailed information about files and folders, including permissions, owner, group, file size, and the date and time of the last modification.  If you run the ls command with the -l option and specify the /etc/hosts argument, you will see detailed information about the /etc/hosts file. This file contains a list of IP addresses and their corresponding domain names. |
| -R /etc/udev | The -R option of the ls command is used to display directories and subdirectories in recursive order. This means that all files and folders in the specified directory and its subdirectories will be displayed.  If you run the ls command with the -R option and specify the /etc/udev argument, all files and folders in the /etc/udev directory and its subdirectories will be displayed in recursive order. |
| -d /etc/s\* | The -d option of the ls command is intended to display only the directories specified on the command line. It does not allow you to display detailed information about files and folders, but only directory names.  If you run the ls command with the -d option and specify the /etc/s\* argument, the names of all directories starting with the letter "s" in the /etc directory will be displayed. |
| –d /etc/[abcd]\* | The -d option of the ls command is intended to display only the directories specified on the command line. It does not allow you to display detailed information about files and folders, but only directory names.  If you run the ls command with the -d option and specify the /etc/[abcd]\* argument, the names of all directories beginning with the letters a, b, c, or d in the /etc directory will be displayed. |
|  |  |
| echo \* | The echo command displays the text data that is passed to it as arguments. When the echo command is prefixed with the joker character \*, it displays a list of all files and folders in the current directory (except for those that start with a ".", i.e., hidden files). |
| echo D\*  echo P\* | The echo command displays the text that is passed to it as arguments. If the echo command is prefixed with the joker character \*, it displays a list of all files and folders whose names begin with the specified sequence of characters.  In the case of echo D\*, the command displays the names of all files and folders whose names begin with the letter "D".  In the case of echo P\*, the command displays the names of all files and folders whose names begin with the letter "P". |
| echo \*s | echo \*s, the command displays the names of all files and folders whose names contain the letter "s". |
| echo D\*n\*s | echo D\*n\*s, the command displays the names of all files and folders whose names begin with the letter "D", contain a certain number of characters between the first and last letter "n", and end with the letter "s". |
| echo ?????? | echo ??????, the command displays the names of all files and folders whose names consist of exactly 6 characters |
| echo D???????? | echo D????????, the command displays the names of all files and folders that begin with the letter D and have a length of 9 characters |
| echo ?????\*s | ?????\*s, the command displays the names of all files and folders that are at least 6 characters long, start with any 5 characters, and end with s. |
| echo [DP]\*  echo [!DP]\* | [DP]\*, the command displays the names of all files and folders that begin with the letter "D" or "P".  [!DP]\*, the command displays the names of all files and folders that do not begin with the letters "D" or "P".  The ! sign indicates that the pattern is inverted |
| echo [D-P]\*  echo [!D-P]\* | [D-P]\*, the command displays the names of all files and folders that begin with letters between "D" and "P" inclusive.  [!D-P]\*, the command displays the names of all files and folders that do not begin with letters between "D" and "P" inclusive. |
| rm hosts  cp –v /etc/hosts hosts | rm hosts - this command deletes the file named "hosts" in the current directory.  cp -v /etc/hosts hosts - this command copies the file "/etc/hosts" to the current directory with a new name "hosts". The -v option provides information about the copying process on the screen. |
| -l hosts  –p hosts /home/sysadmin | ls -l hosts - this command displays detailed information about the file named "hosts" in the current directory, such as permissions, owner, size, etc.  cp -p hosts /home/sysadmin - this command copies the file named "hosts" in the current directory to the "/home/sysadmin" directory, preserving all file attributes such as creation date, modification date, etc. The "-p" option ensures that all file attributes are preserved during copying. |
| mkdir Myetc  cp –R /etc/udev Myetc  ls –l Myetc  ls –lR Myetc | mkdir Myetc - this command creates a new directory named "Myetc" in the current directory.  cp -R /etc/udev Myetc - this command copies the "/etc/udev" directory with its contents to the "Myetc" directory. The "-R" option allows you to copy the directory along with its subdirectories and files.  ls -l Myetc - this command displays detailed information about the directory named "Myetc" in the current directory, such as permissions, owner, size, etc.  ls -lR Myetc - this command displays detailed information about all files and directories contained in the directory "Myetc" in the current directory and all its subdirectories. The -R option allows you to display information about all subdirectories and files recursively. |
| rm -r Myetc | the rm -r command is potentially dangerous because it recursively deletes files and directories without confirmation |

\*\*\* You don't need to provide screenshots of commands in the terminal, just briefly describe the commands in the table.

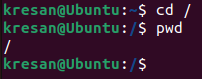
***Готував матеріал студент Кресан Руслан***

1. Робота в терміналі (закріплення практичних навичок) обов’язково представити свої скріншоти:

* Identify your current working directory;



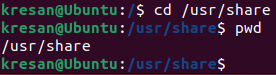
* Go to the root directory and identify your current working directory (two commands);



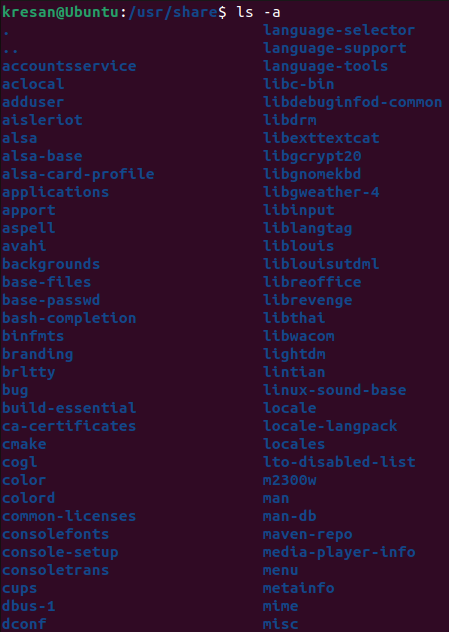
* View the contents of the current directory in long format (use the appropriate key of the ls command);



* Change to the /usr/share directory and define your current working directory (two commands)



* View the contents of the current directory, including hidden files (use the appropriate key of the ls command);



* Change to the /etc directory;



* Browse through the contents of this directory, but display only file names that begin with the letter of your name;



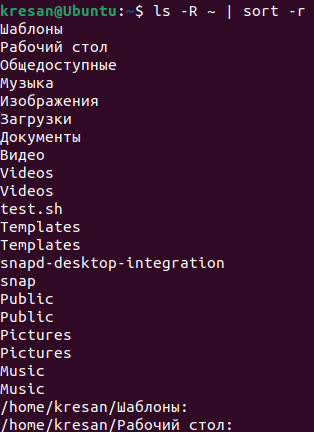
* Browse through the contents of this directory, but display only files with names consisting of 6 letters;



* Look through the contents of this directory, but make sure that it displays only files whose names end with the letters of your names, for example, if your names are Ivan, Anna, Maks, then I will select files whose names end with the letters [i,a,m];

|  |  |
| --- | --- |
|  | *r – Руслан*  *s – Сергій*  *e – Єгор* |

* Navigate to the current user's home directory and view its contents in a recursive (reverse alphabetical) format (perform this action through the command pipeline);



* In the current directory, create a directory with the name of your group;

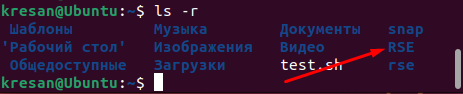
I decided that the name of our group will consist of the first letters of our names.

The name of the group is "RSE*”*



* View the updated content of the current user's home directory.

Use the -r switch of the ls command to see what information you get?



* Go to the directory you created with the name of your group and create an empty lab5 file in it



* Create 3 directories in this directory with the names of the students of your team surname1, surname2, surname3\* (the mkdir command has multiple arguments, so all three directories can be created with one command);



* Go to the first subdirectory surname1 and create an empty file with the name of the first student name1;





* Use the echo command "Hello, my name is Name1" > name1 to enter the student's data into this file (the > symbol allows the output of the echo command to be redirected directly to the name1 file;



* View the contents of the file name1 with the cat command name1 (should contain the information you just entered)



* Make a copy of the first file name1 and rename it to the file with the second name of the student of your team name2;



* Browse the contents of the directory, both files should appear;



* View the contents of the second cat file name2 (it should still contain a complete copy of the contents of the file name1)



* Replace the contents of the file name2 with the corresponding name of the second student using the command echo "Hello, my name is Name2" > name2



* Review the contents of the second file cat name2 (it should already contain the updated information)



* Move the file name2 to the surname2 directory;



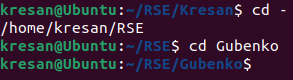
* Make a copy of the first file name1 and rename it to the file with the third name of the student of your team name3;



* Move the file name3 to the surname3 directory;



* Change to the surname3 directory;



* View the contents of the third file with the cat name3 command (it should contain information about the second student)



* *It contains the data of the first student, not the second, because we copied the first student's file in the previous paragraph (Make a copy of the first file name1 and rename it to the file with the third student's name of your team name3;)*
* Replace the contents of the file name3 with the corresponding name of the third student using the command echo "Hello, my name is Name3" > name 3



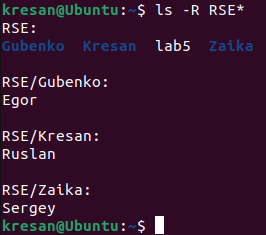
* Check the contents of the file with cat name3 (it should already contain the updated information)



* Return to the user's home directory;



* Browse through the contents of this directory, but only your subdirectory with the group name and all its contents (subdirectories surname1, surname2, surname3 and files name1, name2, name3), and color-coded files and directories (use the appropriate -R switch of the ls command and don't forget to use the special glob-template [directory name]\*)



1. Опишіть дії, які виконують команди для переміщення по системі каталогів:

* Command cd /

Command *cd / moves the user to the root directory of the system. That is, if the user is currently in any other directory, this command moves him to the "/" directory (root directory).*

* Command cd /home

Command *cd /home moves the user to the "/home" subdirectory. This directory often contains users' home directories, so this command is often used to quickly access home directories.*

* Command cd ~

Command *cd ~ moves the user to their home directory. This command performs the same action as the cd command without arguments.*

* Command cd (without argument)

Command *cd (without argument) also moves the user to their home directory. This command performs the same action as the cd ~.*

* Command cd ..

Command *cd .. moves the user one directory back from the current one. For example, if the user is in the directory "/home/user/Documents", the command "cd .." will move him to the directory "/home/user".*

* Command cd ../..

Command *cd ../.. moves the user two directories back from the current one. For example, if the user is in the directory "/home/user/Documents", the command "cd ../..." will move him to the directory "/home".*

* Command cd –

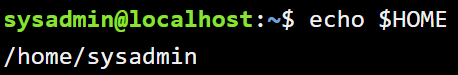
The cd - command moves the user to the last directory they were in before. For example, if the user is in the "/home/user/Documents" directory, and before that he was in the "/home/user" directory, the "cd -" command will move him back to the "/home/user" directory.

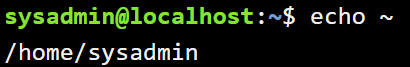
**Відповіді на контрольні запитання**

1. How can I view the path to a user's home directory using the echo command? There are 2 ways, please see both examples in the terminal (the answer is available in the cisco academy materials on netacad.com)

You can get the path to the user's home directory using the echo command and the special environment variable $HOME. Here are two ways to get the path to the user's home directory in the terminal:

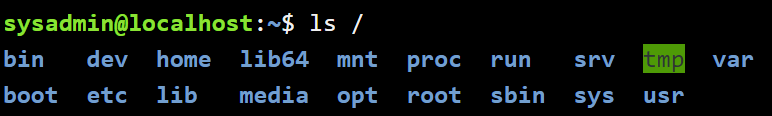
Using the $HOME environment variable with the echo command*:*

 Using the tilde ~ character in combination with the echo command*:*



1. Is it possible to view the contents of the root directory while in the user's home directory without going to the root directory? Demonstrate this on the command line.

*Yes, you can view the contents of the root directory without navigating to it by using the absolute path to this directory. The absolute path to the root directory on most Linux systems is usually /. So, to view the contents of the root directory without changing the current directory, you can use the following command:*

*This command displays a list of files and folders in the root directory. In this case, the user will be in their home directory.*

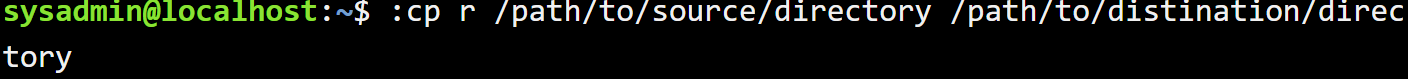
1. How can you add information to an empty file in the terminal?

*Using the echo command and redirecting output to a file. For example, the command below will add the string "Hello World" to an empty myfile.txt file:*



1. How do I copy and delete an existing directory? Will there be a difference in commands if the directory is not empty at the same time.

*You can copy an empty directory using the cp command with the -r option:*

 You can delete an empty directory using the rm command with the -r option *:*

*If the directory contains files and/or subdirectories, you should use the -r option along with the cp and rm commands to copy and delete them, as in the above examples. It is important to be careful and make sure that you use the correct command and the correct directory path to avoid data loss.*

1. In which of the following examples is a file moved? renamed? both actions performed at the same time?

- mv /work/tech/comp.png. /Desktop

- mv /work/tech/comp.png. /work/tech/my\_car.png

- mv /work/tech/comp.png. /Desktop/computer.png

mv /work/tech/comp.png. /Desktop: in this example, the comp.png file is moved to the /Desktop directory.

mv /work/tech/comp.png. /work/tech/my\_car.png: In this example, the comp.png file is renamed to my\_car.png in the same /work/tech directory.

mv /work/tech/comp.png. /Desktop/computer.png: In this example, the comp.png file is moved to the /Desktop directory and renamed to computer.png at the same time.

Thus, the file is moved in the first and third examples, renamed in the second and third, and both actions are performed simultaneously only in the third example.

**Conclusion.**

During the lab, we gained practical skills in working with the Bash shell, and learned basic file system navigation commands and basic commands for managing files and directories.