“Kyiv specialized College of Communications”

Commission of computer engineering

**REPORT ON THE IMPLEMENTATION**

**LABORATORY WORK №5**

From the discipline: "Operating systems"

**Topic: "Learn how to navigate the file system and manage files and directories"**

The students

performed Groups RPZ-03

Team 3:

Kanavets K.S.,

Kryvenko A.I.,

Kulikovska M.V.

Checked by the teacher

Sushanova V.S.

Кyiv 2023

***The material was prepared by student Kulikovska Maria (@Smith5004)***

Tasks for preliminary preparation.

1. **Read the brief theoretical information before the lab and make a small glossary of basic English terms on the purpose of commands and their parameters.**

boot(boot directory) - каталог завантаження

sbin(system binary directory) - системний бінарний каталог

mnt(mount directory) - каталог підключення

tmp(temporary directory) - тимчасовий каталог

recursive option -рекурсивна опція

**2. Based on the material reviewed, answer the following questions:**

*2.1. Compare the file structures of Windows-like and Linux-like systems.*

|  |  |
| --- | --- |
| Window | Linux |
| Windows uses different data drives, such as C:D:E, to store files and folders. | Linux uses the tree as a hierarchical file system. |
| In Windows, "My Documents" is the default home directory. | For each user, a directory /home/username is created, which is called his home directory. |
| The Windows file system is not case-sensitive, so it treats their names as the same file. | In Linux the file system is case-sensitive. This means that you can have files named file, File and FILE respectively in the same folder. |

*2.2. Explain the concept of FHS. How is this standard used in the context of file systems?*

Linux uses the file system structure of the File System Hierarchy Standard ( FHS ), which defines the names, locations, and permissions for many types of files and directories.

This standard consists of a set of requirements and guidelines for placing files and directories in UNIX-like operating systems. It is intended to support interoperability between applications, system administration tools, development tools, and scripting tools, and to provide more uniform documentation for these systems.

*2.3. List the basic commands for working with files and directories in Linux: create, move, copy, delete.*

Files are created with the command: touch filename.

Creating directories is done with a command like: mkdir directory\_name.

To delete files, use the command rm: rm filename.

To delete a directory with all subdirectories and files, use the -r option of the command rm: rm -r directory\_name.

To move and rename files and directories, use the mv command.

To copy the file filename1 and name it filename2, use the command: cp file1 file2.

To copy directory\_name1 to directory\_name2, use the command: cp -a directory\_name1/ directory\_name2/.

Work in progress

1. **Work through all of the command examples presented in the NDG Linux Essentials labs - Lab 7: Navigating the Filesystem and Lab 8: Managing Files and Directories. Create a table to describe these commands.**

|  |  |
| --- | --- |
| Command name | Its purpose and functionality |
| pwd | Determines the user's location in the file system, shows the current working directory. |
| echo $HOME | Shows the path to your home directory, using the values of the HOME variable for viewing. |
| cd / | Used to change the current directory. |
| cd | Used to return to your home directory. |
| cd /home | Used to return to your home directory in the absolute path, indicated by the slash. |
| cd ~ | Used to return to your home directory. With the tilde character ~, the terminal will expand this character to the user's home directory with an account on the system. |
| echo ~ ~sysadmin ~root ~mail ~nobody | Displaying the use of the squiggle as part of the path. |
| cd ~ root | Navigating to the home directory of the user root. In our case, an error is displayed, through denial of permission. |
| cd /usr/bin | Navigate to the /usr/bin directory using the absolute path. |
| cd /usr/ | Navigate to the /usr directory using the absolute path. |
| cd /usr/share/doc | Navigate to the /usr/share/doc directory using the absolute path. |
| cd bash  pwd | Using the relative path, browse to the /usr/share/doc/bash directory and display the working directory. |
| cd .. | Used to navigate to the directory above the current directory by relative path. |
| cd ../dict | Used to navigate to a directory below the current directory in /dict. |
| cd  ls | Used to display the current directory. |
| ls -a | Used to display all files, including hidden files. |
| ls -l /etc/hosts | The -l option provides additional information about the /etc/hosts file. |
| ls -R /etc/udev | The -R option provides not only the contents of the directory, but also the contents of the /etc/udev subdirectories. |
| ls -d /etc/s\* | The -d option prevents displaying files from subdirectories, and s\* displays only those /etc files where they begin with the letter s. |
| ls -d /etc/???? | The /etc files are displayed with exactly four characters. |
| ls -d /etc/[abcd]\* | All /etc files whose names begin with the letters a, b, c, or d are displayed. |
| echo \* | All file names in the current directory are displayed, using the \*. |
| echo D\* | Displays all file names in the current directory that begin with the letter D. |
| echo P\* | Displays all file names in the current directory that begin with the letter P. |
| echo \*s | Displays all file names in the current directory that end with the letter s. |
| echo D\*n\*s | All file names in the current directory that begin with D, have n in the middle and end with s are displayed. |
| echo ?????? | All file names in the current directory that are exactly 6 characters long are displayed. |
| echo D????????? | Displays all file names in the current directory that begin with the letter D and consist of exactly 9 characters. |
| echo ?????\*s | All file names in the current directory that end with s and are at least 5 characters long are displayed. |
| echo [DP]\* | Displays all file names in the current directory that begin with D or P. |
| echo [!DP]\* | Displays all file names in the current directory that do not begin with D or P. |
| echo [D-P]\* | Displays all file names in the current directory that begin with the characters between D and P. |
| echo [!D-P]\* | Displays all file names in the current directory that do not begin with the characters between D and P. |
| cp /etc/hosts hosts | The cp command makes a copy of the /etc/hosts file and places it in the current directory. |
| rm hosts | The rm command is used to delete the hosts file. |
| cp -v /etc/hosts | The -v option tells you what is done with /etc/hosts and cp makes a copy of the file. |
| cp -v /etc/hosts . | The -v option tells you what is done with /etc/hosts and cp makes a copy of the file, . - is used to say "current file". |
| cp -p hosts /home/sysadmin | The -p option is used to copy from the source directory and save the file attributes. |
| rm hosts newname | The rm command can remove more than one file at a time, in our case the hosts and newname files are removed. |
| mkdir Myetc | Creating the Myetc directory using the mkdir command. |
| cp -R /etc/udev Myetc | Copying all files into Myetc. |
| rm -r Myetc | Deleting a directory with the -r parameter of the rm command. |
| touch premove | Create an empty file named premove. |
| mv premove postmove | The mv command "cuts" the premove file and "inserts" it into the file named postmove. |
| rm postmove | Deleting the postmove file. |

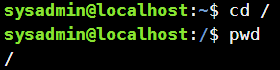
***The material was prepared by student Kryvenko Andrew (@Andrewkryvenko)***

**3. Work in the terminal (consolidation of practical skills)**

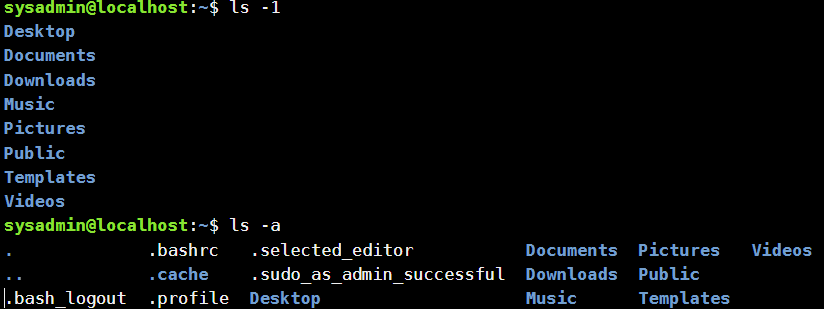
* *Enter your current work directory;*



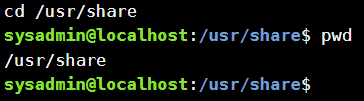
* *Go to the root directory and identify your current working directory (two commands);*



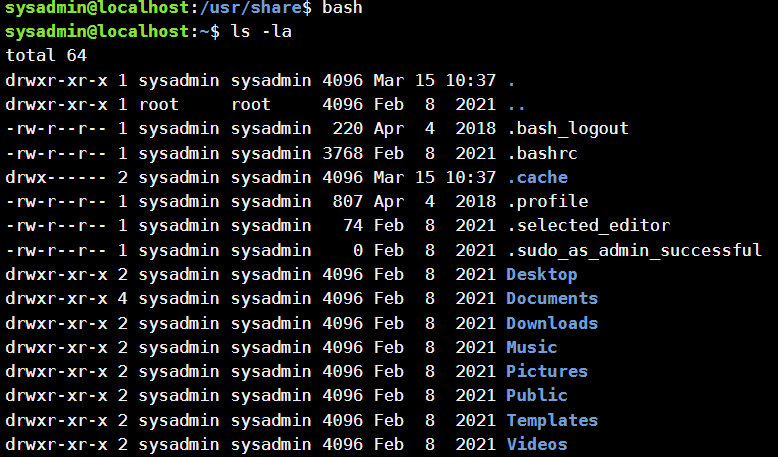
* *View the contents of the current directory in long format (use the appropriate ls command key);*



* *Navigate to the /usr/share directory and determine your current working directory (two commands)*



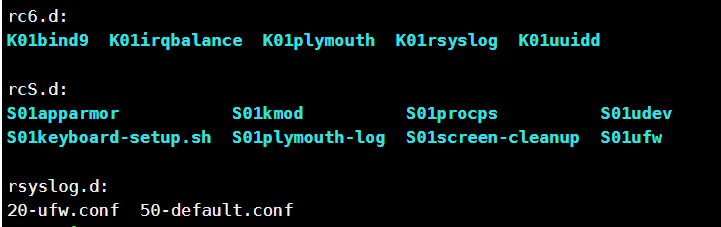
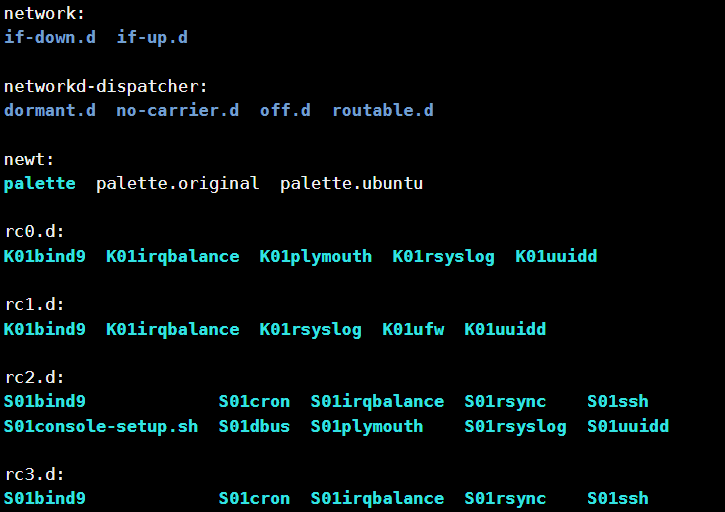
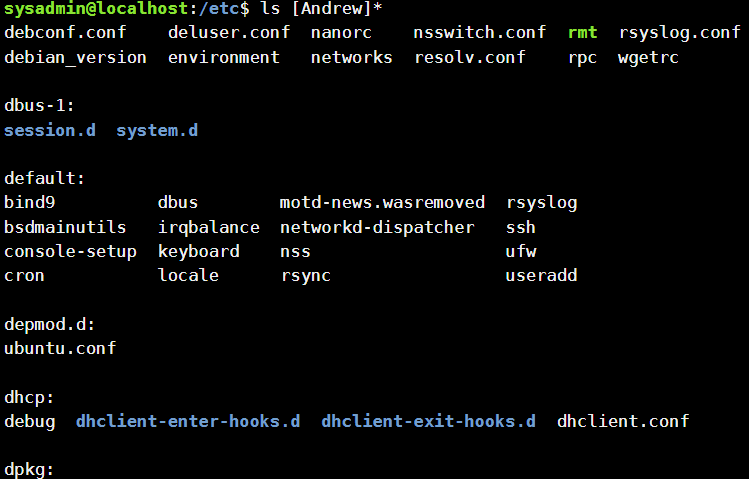
* *View the contents of the current directory, including hidden files (use the appropriate ls command key);*



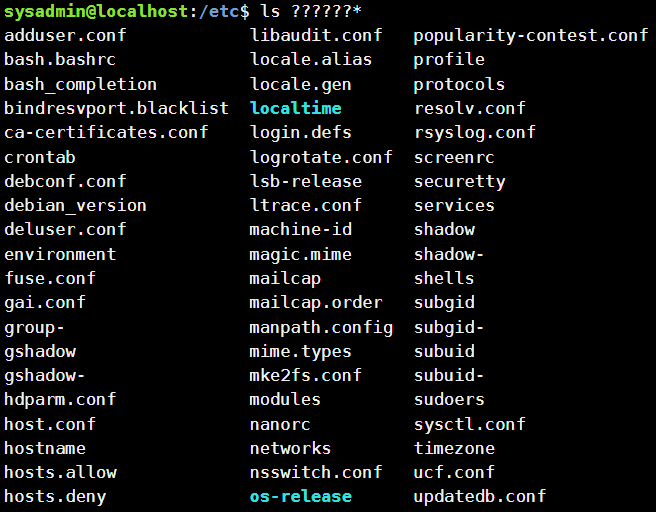
* *Navigate to the /etc directory;*



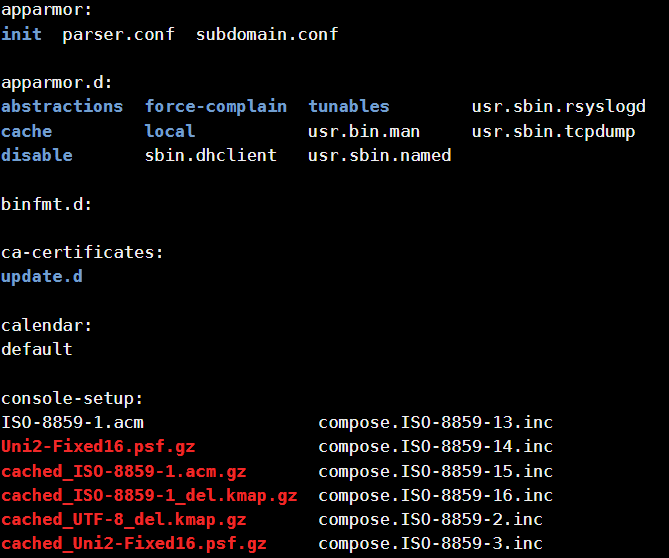
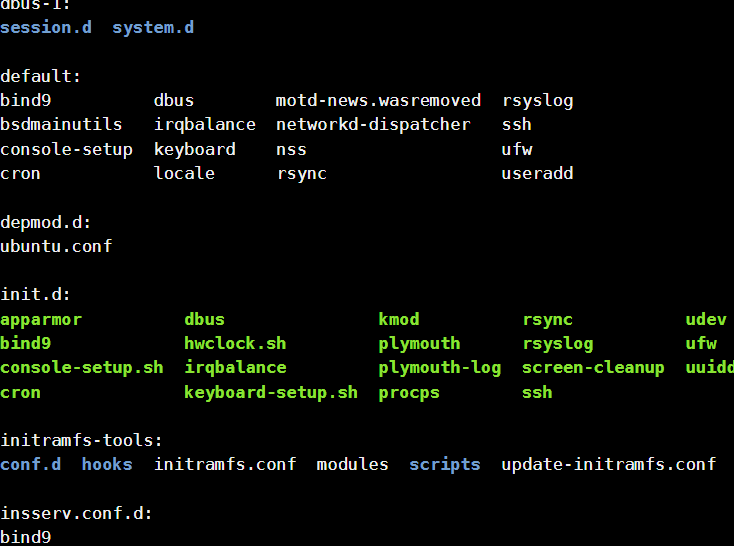
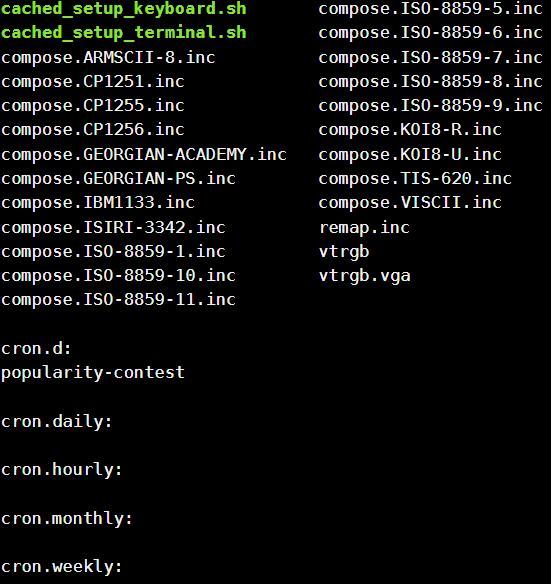
* *View the contents of this directory, but make sure that it displays only file names that begin with the letter of your name;*

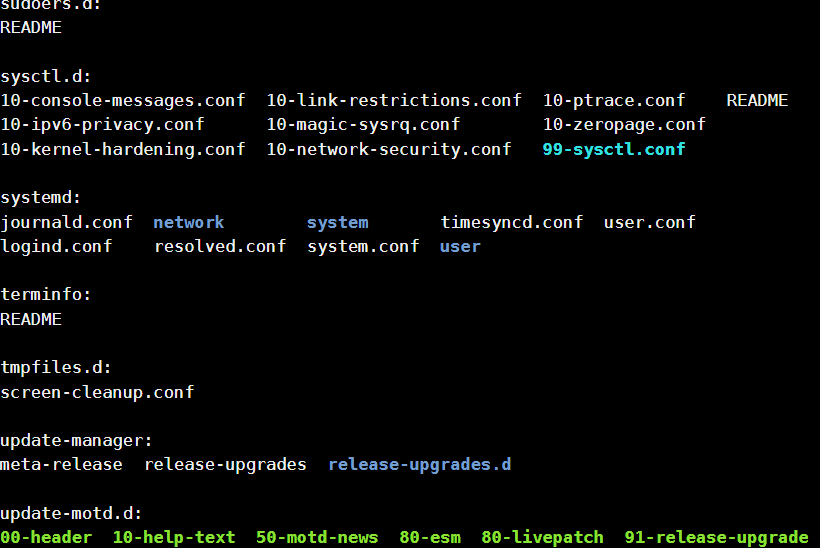


* *View the contents of this directory, but make sure that it displays only files with names consisting of 6 letters;*

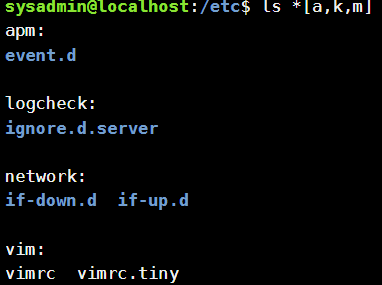








* *Browse the contents of this directory, but make sure that it displays only files whose names end with the letters of your names, for example, if your names are Ivan, Anna, Maks, then I will select files whose names end with the letters [i,a,m];*



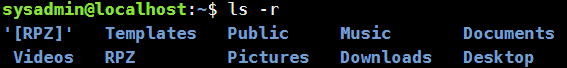
* *Navigate to the current user's home directory and view its contents in a recursive (reverse alphabetical) format (perform this action through the command pipeline);*



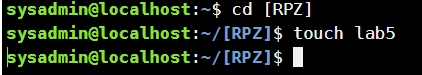
* *In the current directory, create a directory with the name of your group;*



* *View the updated contents of the current user's home directory. Use the -r switch of the ls command, what information will you get?*



* *Navigate 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);*



* *Navigate 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 in your team name2;*



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



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

**

* *View 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 a file with the third name of the student in your team, name3;*

**

* *Move the file name3 to the surname3 directory;*

**

* *Navigate to the surname3 directory;*

**

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

**

* *Replace the contents of the file name3 with the corresponding name of the third student using the command echo "Hello, my name is Name3" > name3*

**

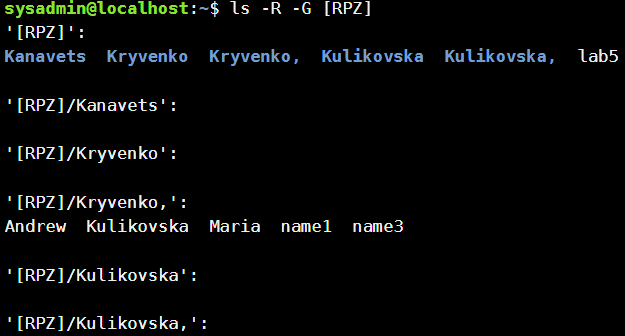
* *Browse the contents of the file with cat name3 (it should already contain the updated information)*

**

* *Return to the user's home directory;*

**

* *Browse 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]\*)*



***The material was prepared by student Kanavets Kateryna (@kanavetsk)***

**5. Describe the actions that the commands perform to navigate the directory system:**

*- The cd / command*

The cd / command moves the user to the root directory of the system. This means that after executing this command, the user will be in the directory that is the base for the entire file system.

*- cd /home command*

The cd /home command moves the user to the /home directory. This directory usually contains subdirectories corresponding to individual users on the system. After executing this command, the user will be in the /home directory, where he can view and work with files and directories belonging to him or other users on the system.

- *cd ~ command*

The cd ~ command moves the user to their home directory. The home directory usually contains the user's personal files and directories, including their documents, music, videos, and other personal data. After executing this command, the user will be in their home directory, where they can view and work with their personal files and directories.

*- cd command (no argument)*

The cd command without an argument moves the user to their home directory. This command is similar to the cd ~ command. After executing this command, the user will be in his home directory, where he can view and work with his personal files and directories.

*- The cd .. command.*

The cd .. command moves the user one level higher in the file system hierarchy. This means that after executing this command, the user will be in the directory containing the previous directory in which he or she was before executing the command. For example, if a user is in the /home/user/Documents directory and executes the cd .. command, he or she will be moved to the /home/user directory.

*- The cd ../... command.*

The cd ../... command moves the user two levels higher in the file system hierarchy. This means that after executing this command, the user will be in a directory that contains a directory that contains the previous directory in which he or she was before executing the command. For example, if a user is in the /home/user/Documents directory and executes the cd ../... command, he or she will be moved to the /home directory.

*- The cd - command*

The cd - command moves the user to the previous directory in which he was before executing the current cd command. This command allows you to quickly navigate between two directories, keeping the history of movements. For example, if a user first navigated to /home/user/Documents, then cd /var/log, and now executes the cd - command, the user is returned to /home/user/Documents, which was the previous directory before the cd - command.

Checklist questions

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

To display the path to the user's home directory, use the special environment variable $HOME. There are two ways to display the value of this variable using the echo command:

Using the dollar sign $ with the $HOME variable:

echo $HOME

Using the ~ tilde with the username character:

echo ~

Both methods will return the path to the current user's home directory.

**2. 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, it is possible to view the contents of the root directory without navigating to it from the home directory using the absolute path to the root directory. To do this, you can use the ls command together with the absolute path to the root directory /.

An example of using the ls command to view the contents of the root directory from the user's home directory:

$ ls /

bin dev home lib64 mnt proc run snap sys usr

boot etc init media opt root sbin srv tmp var

In this example, ls / displays a list of files and directories in the root directory of the system, you do not need to change the current directory to the root directory.

**3. How can I add information to an empty file in the terminal?**

To add information to an empty file in the terminal, you can use the nano text editor or the echo command.

To use the nano editor, you need to execute the nano command along with the name of the file you want to open or create. For example:

$ nano myfile.txt

This command will open the myfile.txt file in the nano editor, where you can enter the desired information. To save the entered information, press Ctrl + O, confirm the file by pressing Enter, and exit the nano editor by pressing Ctrl + X.

To use the echo command, you need to execute the echo command along with the text you want to add to the file and redirect its output to the file using the > symbol (or >> if you want to add information to an existing file). For example:

$ echo "Hello, world!" > myfile.txt

This command will add the line "Hello, world!" to the file myfile.txt. If the file does not exist, it will be automatically created. If the file exists, the contents of the file will be replaced with the information you entered.

**4. How do I copy and delete an existing directory? 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.**

To copy an existing directory, you can use the cp command. The syntax of the command to copy a directory and its contents using the example of the source\_dir directory to the target\_dir directory:

$ cp -r source\_dir target\_dir

To delete an existing directory, you can use the rm command. If the directory is empty, you can use the -r command to recursively delete the directory. The syntax of the command to delete an empty directory using the example of the mydir directory:

$ rm -r mydir

If the directory is not empty, the rm command will generate an error, since deleting a non-empty directory is unsafe. In such a case, you can use the rm command with the -r switch to recursively delete the directory and its contents. The syntax of the command to delete a non-empty directory using the example of the mydir directory:

$ rm -r mydir

Make sure that you specify the name of the directory correctly to avoid unintentional data loss.

**5. Which of the following examples moves a file? renames it?**

both actions at the same time?

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

The following example moves the file comp.png from the /work/tech/ path to the /Desktop path and renames it to comp.png.

If you use the mv command correctly, you can move and rename a file at the same time. In this case, the file is moved and renamed in separate actions, because the dot at the end of the new file name may be a typo, so you should check the correctness of the new name before executing the command. Also, an extra space before /Desktop can cause the command to fail.

- mv /work/tech/comp.png.

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

The mv command is used to move or rename a file. In this case, we are renaming the file that is already in the /work/tech/ path to the new name my\_car.png. The dot at the end of the original file name may be a typo, so you should check that the new name is correct before executing the command.

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

The example above moves the file comp.png from the /work/tech/ directory to the /Desktop/ directory and renames it to computer.png.

The mv command is used to move or rename a file. In this case, we are moving the file from the /work/tech/ path to the /Desktop/ directory and renaming it to computer.png. There was a typo at the end of the original filename, which has been corrected in the new filename.

Conclusions: in this lab, we learned how to work with file system navigation commands, manage files and directories, and examined commands for navigating the directory system in Linux.