“Київський фаховий коледж зв’язку”

Циклова комісія Комп’ютерної та програмної інженерії

ЗВІТ ПО ВИКОНАННЮ

ЛАБОРАТОРНОЇ РОБОТИ №5

з дисципліни: «Операційні системи»

Тема: “Знайомство з базовими командами CLI-режиму в Linux”

Виконали

студенти

групи РПЗ-03

Команда 6: Sichkar Maxim,

Brytyuk Bohdan

Kyiv 2023

**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 support to occupy:**

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.**

1. Read the short theoretical information for the laboratory work and make a small dictionary

basic English terms for command assignments and their parameters.

2. On the basis of the considered material, answer the following questions:

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

Windows and Linux operating systems have different approaches to organizing files and directories.

In a Windows-like system, files and folders are organized in a hierarchical structure starting from the root directory, which is represented by a drive letter (such as C:\ or D:). The root directory contains various system folders, such as Program Files, Windows, and Users, which contain the applications and user data. Windows also uses the concept of file extensions to identify file types, such as .doc for Microsoft Word documents or .exe for executable files.

In a Linux-like system, the file structure is based on a single directory tree, which starts at the root directory (represented by a forward slash /). Everything in Linux is considered a file, including devices, directories, and programs. Linux uses a system of permissions to control access to files and directories, with three different levels of access: read, write, and execute. Linux also uses a different naming convention for files, with no concept of file extensions. Instead, the file type is determined by the file's contents and attributes.

Overall, while both Windows and Linux use a hierarchical file structure, Linux's file system is more unified and flexible, with a focus on security and permissions. Windows, on the other hand, places more emphasis on file extensions and organizing files by program and user.

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

FHS stands for Filesystem Hierarchy Standard. It is a set of guidelines and recommendations for organizing file systems in Unix-like operating systems, including Linux. The goal of FHS is to promote consistency and interoperability between different Unix-like systems, by defining a common file system structure that can be used by all Unix-like systems.

FHS specifies the location and purpose of directories and files in a Unix-like file system. For example, it defines the use of the /bin directory for storing essential user commands, the /etc directory for system configuration files, the /usr directory for user applications, and the /var directory for variable data such as logs, mail, and spool files.

By following the FHS standard, developers can ensure that their applications will work on different Unix-like systems without requiring significant modifications. System administrators can also use the standard to manage file systems consistently across different machines.

The FHS standard is used in the context of file systems by providing a consistent and predictable layout for directories and files, which simplifies system administration, software development, and software deployment. It also ensures that Unix-like systems are interoperable and can work with each other seamlessly. Overall, FHS is an important standard that helps maintain the stability and reliability of Unix-like operating systems.

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

**move, copy, delete.**

Here are some basic commands for working with files and directories in Linux:

Creating a new directory: mkdir directory\_name

Moving between directories: cd directory\_name

Listing the contents of a directory: ls

Creating a new file: touch file\_name

Opening a file for editing: nano file\_name or vi file\_name

Copying a file to another location: cp file\_name destination

Moving or renaming a file: mv old\_file\_name new\_file\_name or mv file\_name directory\_name

Deleting a file: rm file\_name

Deleting a directory and all its contents: rm -r directory\_name

It's important to note that many of these commands have additional options and arguments that can be used to customize their behavior. For example, the ls command can take arguments such as -l to display a long format listing or -a to show hidden files. It's a good idea to consult the documentation or use the man command to learn more about the options available for each command.

3. Study the materials of the online course of the Cisco Academy “NDG Linux Essentials”:

- Chapter 7 - Navigating the Filesystem

- Chapter 8 - Managing Files and Directories

4. Take the test in the NDG Linux Essentials course on the following topics:

- Chapter 07 Exam

- Chapter 08 Exam

5. Prepare the initial version of the report electronically:

- Title page, topic and purpose of the work

- Glossary of terms

- Answers to items 2.1-2.3 from tasks for preliminary preparation

**Progress.**

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

1.1. Start the VirtualBox virtual machine, select CentOS and run it. Log in

under user: CentOS, password for login: reverse (if you run LR in 401 aud.) and run

terminal.

1.2. Start the Ubuntu\_PC virtual machine (if you are doing the LR tasks through the netacad academy)

1.3. Start your Linux family operating system (if you are working on your own PC and its

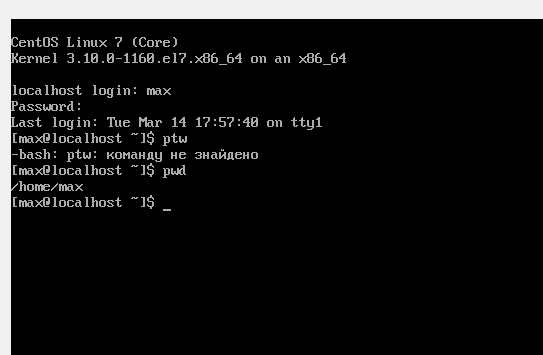
installed) and launch the terminal.

2. Work through all the command examples presented in the NDG Linux Essentials course labs - Lab 7: Navigating the Filesystem and Lab 8: Managing Files and Directories. Create a table for description of these commands\*\*\*

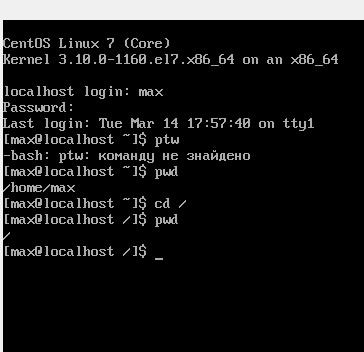
|  |  |
| --- | --- |
| pwd | The pwd command in Linux stands for "print working directory." When you type pwd into a terminal, it displays the absolute path of the current working directory.  The purpose of the pwd command is to allow you to determine the exact location of the directory you are currently in. This can be helpful when navigating through the file system, especially if you are working with a complex directory structure or if you need to reference the current directory in a script or command.  The pwd command has a simple functionality - it outputs the absolute path of the current directory to the terminal. For example, if you are in the directory /home/user/documents, typing pwd will output /home/user/documents.  In summary, the pwd command is a simple but essential tool for working with directories in Linux. It allows you to easily determine your current directory and reference it as needed when running commands or scripts. |
| cd Documents | The command cd Documents is used to change the current working directory in a Linux or Unix-like operating system to the "Documents" directory, assuming it exists in the current directory. The purpose of this command is to allow the user to navigate to the "Documents" directory and access or modify files within that directory or any subdirectories under it.  The cd command stands for "change directory", and it is a fundamental command in Linux and Unix-like systems. By default, when a user logs in to a shell or terminal, the current working directory is the user's home directory. The cd command is used to navigate to different directories on the file system. For example, if a user is in their home directory and they want to navigate to the "Documents" directory, they can enter cd Documents and press enter. The shell prompt will then change to reflect the new working directory, and any subsequent commands will be executed in the context of that directory. |
| echo $HOME | The command echo $HOME is used to display the path of the current user's home directory in a Linux or Unix-like system. The $HOME is a special shell variable that holds the absolute path to the user's home directory.  When the command echo $HOME is executed, the shell expands the $HOME variable and displays the result to the standard output. For example, if the current user's home directory is /home/username, the command echo $HOME will display /home/username.  This command is useful for various purposes, such as scripting and automation, as it allows scripts to refer to the user's home directory without hardcoding the path. Additionally, it can be used to verify that the shell has correctly set the $HOME variable, which can be important for some applications and scripts that rely on this variable. |
| cd / pwd | The cd (change directory) and pwd (print working directory) commands are fundamental commands for navigating and working with directories in a Linux or Unix-like operating system.  The cd command is used to change the current working directory. It takes one argument, which is the name of the directory to change to. For example, cd Documents changes the current directory to the Documents directory within the current directory. Using cd without any arguments takes you to your home directory.  The pwd command is used to print the current working directory. It displays the full path of the directory you are currently in. For example, pwd might display /home/user/Documents if you are currently in the Documents directory of the user home directory. |
| cd /home  pwd | The command "cd /home" is used to change the current working directory to the "/home" directory in the file system. This means that any subsequent commands or actions performed in the terminal will take place within the "/home" directory.  The purpose of changing the working directory is to make it easier to navigate the file system and access files or directories located in a specific location. For example, if a user wants to access files located in their home directory, they can use the "cd /home/user" command to navigate to that directory and then perform any necessary actions.  The command "pwd" is used to print the current working directory. When this command is executed, the terminal will display the absolute path of the current working directory. This is useful for confirming the current location in the file system and ensuring that you are working in the correct directory. |
| echo ~ ~sysadmin ~root ~mail ~nobody | The "echo" command is a basic command in Unix and Unix-like operating systems that prints (echoes) a string of text or variable value to the standard output (usually the terminal). In this case, the command:  echo ~ ~sysadmin ~root ~mail ~nobody  will simply print the following text to the terminal:  ~ ~sysadmin ~root ~mail ~nobody  This command does not have any particular purpose or functionality, as the text string does not have any special meaning or significance in the context of the operating system or any specific application. The text string consists of a series of tilde (~) characters followed by various usernames (sysadmin, root, mail, nobody), which may be interpreted as Unix user accounts or simply as arbitrary strings of text. |
| cd ~root | The command cd ~root is used to change the current working directory to the home directory of the root user. Here, the tilde symbol (~) is used as a shortcut for the home directory. The root user is the superuser in Unix-like operating systems and has the highest level of privileges.  The purpose of this command is to navigate to the home directory of the root user, which can be useful in a number of scenarios. For example, if you need to access files or directories that are only accessible by the root user, you can use this command to navigate to the root user's home directory and then access the required files or directories.  Additionally, if you need to perform administrative tasks on your system, such as installing software or configuring system settings, you may need to be logged in as the root user or use the sudo command with root privileges. In these cases, using the cd ~root command can help you quickly navigate to the root user's home directory before performing these tasks. |
| cd .. pwd | The command cd .. is used to navigate up one level in a directory hierarchy. It is used to move to the parent directory of the current working directory. For example, if the current working directory is /home/user/Documents/, then cd .. would take the user to the directory /home/user/.  The command pwd (print working directory) is used to display the current working directory. It shows the full path of the current working directory from the root directory. For example, if the current working directory is /home/user/Documents/, then pwd would display /home/user/Documents/.  Together, cd .. and pwd can be used to navigate through the file system and display the current location within it. |
| ls -a | The ls -a command is used to list all files and directories in a directory, including hidden files and directories that start with a period (.). The ls command itself lists only the visible files and directories.  Here's an example output of ls -a command:  $ ls -a  . .. .bashrc Documents Downloads .git .gitignore .ssh .vimrc  In the output, you can see that in addition to the regular files and directories, there are also hidden files and directories listed, such as .bashrc, .git, .ssh, etc.  The -a option is particularly useful when you need to view hidden files and directories, which are often configuration files or files that are not meant to be edited directly. It can also be used in conjunction with other ls options to provide a more detailed view of the directory's contents. |
| ls -d /etc/s\* | The command "ls -d /etc/s\*" has the following purpose and functionality:  Purpose: To list only the directories in the "/etc" directory that start with the letter "s".  Functionality:  "ls" is the command used to list the contents of a directory.  "-d" is an option that tells "ls" to list only the directories themselves, rather than the contents of those directories.  "/etc/s\*" specifies the directory path and the pattern to match for the directories to list. Here, the pattern is "s\*", which matches any directory name in "/etc" that starts with the letter "s".  Overall, the command will only list the names of the directories in the "/etc" directory that begin with the letter "s". It will not list the contents of those directories or any other files in the "/etc" directory. |
| ls –d /etc/[abcd]\* | The command "ls -d /etc/[abcd]\*" is a Unix/Linux command that performs the following:  "ls" is the command used to list the files and directories in a specified location.  "-d" option tells ls to list only the directories specified and not their contents.  "/etc" is the directory path that the command will be executed in.  "[abcd]\*" is a file globbing pattern that matches any file or directory that begins with a, b, c, or d in the /etc directory.  So, the purpose and functionality of the command "ls -d /etc/[abcd]\*" is to list only the directories in the /etc directory that begin with a, b, c, or d. It will not list the contents of those directories. This command can be useful when you want to quickly identify directories in the /etc directory that match a certain pattern or criteria. |

3. Working in the terminal (consolidation of practical skills), you must present your screenshots:

- Define your current working directory;

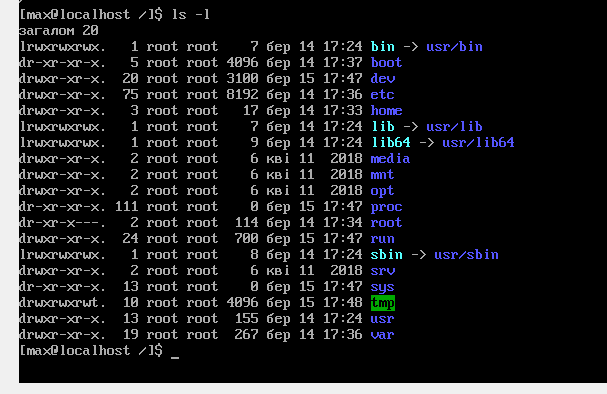


- Go to the root directory and define your current working directory (two commands);

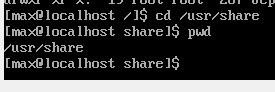


- View the contents of the current directory in long format (use the appropriate key

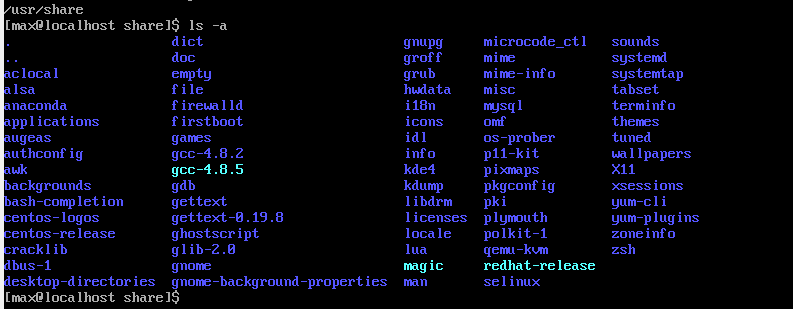
ls commands);



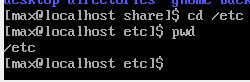
- Go to the /usr/share directory and define your current working directory (two commands)



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

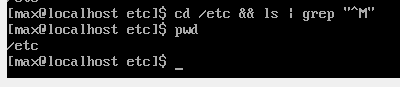


- Go to the /etc directory;



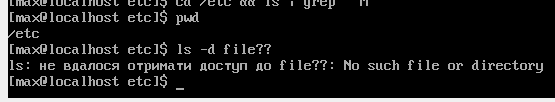
- Browse the contents of a given directory, but only output filenames starting with

the letters of your name;

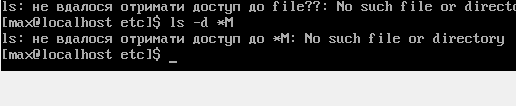


- Browse the contents of a given directory, but only output files whose names consist of 6

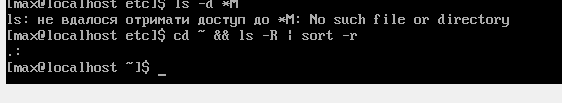
letters;



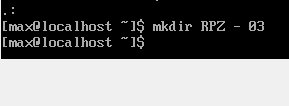
- Browse the contents of a given directory, but only display files whose names end with the letters of your names, for example, if your names are Ivan, Anna, Maks, then I will make a selection for the names files ended with the letters [i,a,m];



- Go to the current user's home directory and view its contents in recursive (back to alphabetical) format (perform this action through the command pipeline);

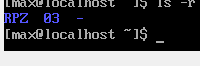


- In the current directory, create a directory with the name of your group;



- View the updated contents of the current user's home directory. Take advantage

with the -r switch of the ls command, what information do you get?



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

lab5



- Create in this directory 3 directories with the surnames of students of your team surname1,

surname2, surname3\* (multi argument mkdir command, so all three directories can be created one team);



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



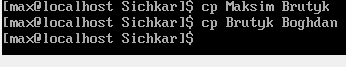
- Using the command echo "Hello, my name is Name1" &gt; name1 enter data about student (the symbol &gt; allows the output of the echo command to be redirected immediately to the file name1;



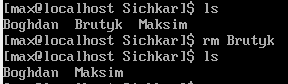
- View the contents of the file name1 using the command cat name1 (should contain the just typed you information)



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



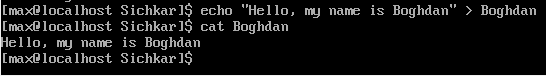
- Browse the contents of the directory, both files should appear;



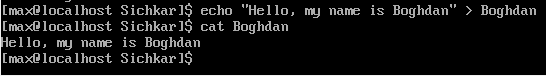
- Review the contents of the second file cat name2 (it should by now contain a full copy of the contents of the file name1)



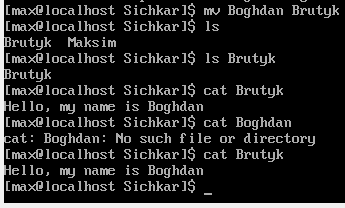
- Replace the contents of the file name2 to contain the corresponding name of the second student using commands echo "Hello, my name is Name2"; &gt; name2



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

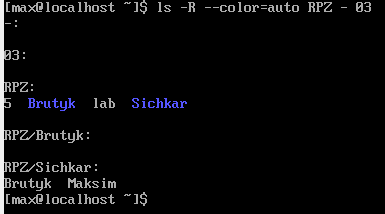


- Move the name2 file to the surname2 directory;



- Return to the user's home directory;

cd

- View the contents of this directory, but to display only your subdirectory with the name of the group and all its contents (subdirectories surname1, surname2, surname3 and files name1, name2, name3) to the same files and directories were separated by colors (use the appropriate key -R ls command and don't forget to use the custom glob template [directory name]\*)  


**Control questions:**

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

Yes, there are at least two ways to view the path to the user's home directory using the echo command. Here are the examples:

Using the tilde (~) symbol which represents the current user's home directory:

echo ~

This will output the path to the current user's home directory.

Using the environment variable $HOME which also points to the user's home directory:

echo $HOME

This will output the path to the current user's home directory as well.

Both commands will output the path to the user's home directory, and which one you use may depend on personal preference or the specific context in which you're using the 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.**

No, it is not possible to view the contents of the root directory while in the user's home directory without going to the root directory or specifying its absolute path.

The root directory (/) is the highest-level directory in the file system, and it contains all other directories and files in the system. To view its contents, you must navigate to it using its absolute path or relative path.

Here's an example to illustrate this. Let's say you are currently in the user's home directory and you want to view the contents of the root directory without changing your working directory. If you try to list the contents of the root directory without specifying its path, you will get an error:

ls /

This will list the contents of the root directory because we're specifying its absolute path. However, if you try to list its contents without specifying its path, you'll get an error message like this:

ls: cannot open directory '.': Permission denied

This error occurs because you are attempting to list the contents of the root directory without actually navigating to it or specifying its path, and you do not have permission to list the contents of directories above your current working directory.

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

To add information to an empty file in the terminal, you can use any text editor of your choice such as nano, vim or emacs. Here are the steps to create and add information to an empty file using the nano text editor:

Open a terminal window.

Navigate to the directory where you want to create the file using the cd command.

Create a new empty file using the touch command. For example:

touch newfile.txt

Open the file in a text editor. For example, to open the file using the nano text editor, type:

nano newfile.txt

Once the editor opens, start typing the information that you want to add to the file. You can also paste text from another source if needed.

When you are done adding the information, save the file by pressing Ctrl+O, then exit the editor by pressing Ctrl+X.

After following these steps, the file will be created with the added information. You can view the contents of the file using the cat command or by opening it again in a text editor.

1. **How to 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 and delete an existing directory, you can use the cp and rm commands in the terminal.

To copy an existing directory, you can use the cp command with the -r option, which stands for recursive. This option ensures that all the contents of the directory are copied along with the directory itself. For example, to copy a directory named myfolder to a new directory named backup, you can use the following command:

cp -r myfolder backup

To delete an existing directory, you can use the rm command with the -r option to delete the directory and its contents recursively. For example, to delete the directory named myfolder, you can use the following command:

rm -r myfolder

If the directory is not empty, and you try to delete it using the rm command without the -r option, you will get an error message saying that the directory is not empty. In this case, you will need to use the -r option to delete the directory and its contents recursively. For example:

rm -r myfolder

Note that when using the rm command with the -r option, the operation is irreversible and all contents of the directory will be permanently deleted. Therefore, you should use this command with caution and ensure that you have backed up any important data before proceeding.

**5. In which of the following examples does a file move occur? renaming it?**

**both actions 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**

The first example moves the file "comp.png" from "/work/tech/" directory to "/Desktop" directory. Therefore, a file move occurs in the first example.

The second example renames the file "comp.png" to "my\_car.png" within the same directory "/work/tech/". Therefore, a file rename occurs in the second example.

The third example moves the file "comp.png" from "/work/tech/" directory to "/Desktop" directory, and also renames it to "computer.png". Therefore, both actions, file move and file rename, occur in the third example.

**Conclusion:** I got hands-on with the Bash shell, learned basic file system navigation commands, and learned basic commands for managing files and directories.