“Kyiv Vocation College of Communication”
Cyclical Commission of Computer Engineering

REPORT ON EXECUTION LABORATORY WORK №3

on the discipline: "Operating Systems"

Topic: "Introduction to basic commands in CLI mode in Linux."

Performed by student
group RPZ-13a
Kateryna Hranat
Checked by teacher
V.S. Sushanova

Performed by student group RPZ-13a Kateryna Hranat

Kyiv 2024

Work objectives:

1. Introduction to basic commands in CLI mode in Linux.
2. Introduction to basic text commands in terminal mode operation in various operating systems.

Material Support for Classes:

1. IBM PC type computer.
2. Windows family operating system (Windows 7).
3. Virtual machine - Virtual Box (Oracle).
4. GNU/Linux operating system - CentOS.

Performed by student group RPZ-13a Kateryna Hranat

Tasks for Preliminary Preparation:

1. Read the brief theoretical information for the laboratory work and create a small dictionary of basic English terms related to the purpose of commands and their parameters.

English terms	**Ukrainian terms**
Inline editing	Інлайнове редагування
Scripting	Сценарії
Aliases	Псевдоніми
Variables	Змінні
Current Directory	Поточний каталог
The Shell	Оболонка (Shell)
Quoting	Цитування
Control statements	Контрольні оператори
Semicolon	Крапка з комою
Double ampersand	Подвійний амперсанд
Double pipe	Подвійна вертикальна риска

2. Study the materials of the online course "NDG Linux Essentials" from Cisco:
   - Chapter 5 - Command Line Skills
   - Chapter 6 - Getting Help

Complete✔

3. Complete testing in the NDG Linux Essentials course on the following topics:

- Chapter 05 Exam
- Chapter 06 Exam

Complete✔

4. Provide definitions for the following concepts:

- Command Interpreter: It is software that interprets and executes commands entered by the user via the command line or other means of interaction with the operating system. It can also execute scripts containing sequences of commands.

- Shell: It is the interface between the user and the operating system, allowing the user to interact with the system using text commands. The shell typically provides access to operating system functions such as launching programs, managing files, and input/output operations.

- Command: It is an instruction or program that a user enters into the command interpreter to perform a specific action in the operating system. Commands can be built into the shells themselves or represent programs that can be executed in the operating system.

5. Answer the following questions:

- What basic information does the prompt provide?

The command prompt provides basic information about the system's status and awaits input commands from the user. Typically, this includes the username, computer name, current directory, and possibly some additional information such as the current time or shell type.

- Why does a command need parameters and arguments?

Parameters and arguments help commands execute specific actions accurately and as needed. Parameters are flags or settings that can modify the behavior of a command. Arguments are data passed to the command for processing.

- What is the purpose of the ls command, what parameters and arguments can it have? Provide 3 examples.

The 'ls' command is intended to list files and directories in the current directory. Some parameters and arguments of the 'ls' command include:

- ❖ Parameter -l: displays detailed information about files and directories.
- ❖ Parameter -a: shows all files, including hidden ones.
- ❖ Argument directory_name: specifies that 'ls' should display the contents of a specific directory, not the current one.

- How can you utilize the history of commands, what advantages does it provide?

Command history can be used to view previous commands entered by the user. This is convenient for repeating previous actions, editing, or executing previous commands.

- What is the purpose of the echo command?

The 'echo' command is used to display text on the screen. It is often used to display variable values or messages to the user.

- Describe the concept of variables in the Bash shell, what types of variables does it support?

A variable in the Bash shell is a symbolic name that contains a certain value. The Bash shell supports various types of variables, such as local, global, system, etc.

- What is the purpose of the env, export, and unset commands?

The commands 'env', 'export', and 'unset' are used to work with the shell environment:

- ❖ env: displays the current environment, which is a set of environment variables and their values.
- ❖ export: is used to create global environment variables or export variables from the shell into the environment for use in subprocesses.
- ❖ unset: removes an environment variable.

- What commands for getting help on commands in the terminal do you know?

In the terminal, there are several commands and methods to get help on commands:

- ❖ man: Short for "manual." The man command is used to display manual pages for other commands and programs. For example, man ls will display the manual page for the ls command.
- ❖ --help or -h option: Many commands support the --help or -h options, which display a brief help on how to use the command. For example, ls --help.
- ❖ info: Some programs provide more detailed documentation through the info system, which can be accessed using the info command.
- ❖ apropos: The apropos command is used to search for commands by keywords. For example, apropos search will output a list of commands related to searching.
- ❖ help: In some shells, such as Bash, there is a built-in help command that provides brief help on built-in shell commands.

6. Prepare an initial version of the report in electronic form:

- Title page, topic, and purpose of the work
- Glossary of terms
- Answers to points 5 and 6 from the tasks for preliminary preparation

Complete✔

Performed by student group RPZ-13a Kateryna Hranat

Progress of Work:

1. Work through all the examples of commands presented in the laboratory work of the NDG Linux Essentials course - Lab 5: Command Line Skills and Lab 6: Getting Help. Create a table to describe these commands***

Command	**Description**
ls	List files and directories in the current directory.
cd	Change directory.
pwd	Print working directory - display the current directory path.
mkdir	Create a new directory.
touch	Create a new empty file.
rm	Remove files or directories.
rmdir	Remove directories.
cp	Copy files or directories.
mv	Move or rename files or directories.
cat	Concatenate and display file content.
less	View file content one page at a time.
head	Display the first part of a file.
tail	Display the last part of a file.
man	Display the manual page of a command.

--help	Display usage information and options for a command.
whatis	Display a brief description of a command.
apropos	Search the manual page names and descriptions for a specific keyword.
info	Display detailed information about a command.
which	Display the location of an executable command.
type	Display information about command type (builtin, alias, or executable file).
alias	Create an alias for a command.
history	Display the list of previously executed commands.
Ctrl + R	Search command history backward.
Ctrl + C	Interrupt and cancel the current process.
Ctrl + Z	Suspend current process
fg	Bring a suspended process to the foreground.
bg	Resume a suspended process in the background.
jobs	Display a list of currently running jobs.

Performed by student group RPZ-13a Kateryna Hranat

Control questions

1. What types of commands exist in the Bash shell?

Types of commands in the Bash shell include built-in commands, external commands, and scripts.

- Built-in commands: These are commands that are built directly into the shell and are executed without launching a separate process.
- External commands: These are executable programs or scripts located in separate executable files.
- Scripts: These are sequences of commands that can be saved in a text file and executed as a program.

2. What are environment variables? What types do they come in? How can they be viewed in the terminal?

Environment variables are variables that store information about the environment in which the shell is operating. Some of the most common environment variables include $HOME, $PATH, $USER, $LANG, $PWD, and others. To view the values of environment variables in the terminal, you can use the echo command, for example: echo $PATH.

3. Describe the $PS1 variable. How can its content be viewed in the terminal?

The $PS1 variable defines the prompt string that is displayed before each command input in the Bash shell. By default, it contains a value that indicates the current working directory path. To view the contents of the $PS1 variable in the terminal, use the command echo $PS1.

4. How can you change the value of the $PS1 variable? What happens to the command prompt in bash (the prompt string before each command)? How to change the value of this variable not for the current session, but as a default?

To change the value of the variable $PS1, you can assign it a new value. For example:

PS1="new value"

After this, the prompt string in the Bash shell will change to the new value. If you want to persist the changes, you should add this assignment to the shell configuration file, such as .bashrc or .bash_profile.

5. What are quotes used for in the Bash shell?

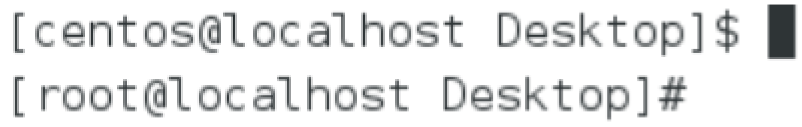
Quotes in the Bash shell are used to declare strings with special characters or spaces that need to be interpreted as a single argument. This helps to avoid breaking strings into parts when passing them to commands or performing other operations. Single quotes (') and double quotes (") are used.

6. What is the purpose of control instructions, and what types do you know?

Instructions for managing in the Bash shell are used to control the flow of command execution and scripts. Some types of control instructions include:

- <u>Conditional statements:</u> if, else, elif, case.
- <u>Loops:</u> for, while, until.
- <u>Exit instructions:</u> exit, return.
- <u>Error handling instructions:</u> trap, set.

7. What is the difference if the bash prompt ends with the symbol $ or #? For example, on the screen, we see the following entries:



```
[centos@localhost Desktop]$ █
[root@localhost Desktop]#
```

In the Bash shell, the $ symbol at the end of the prompt string indicates the regular user mode, while the # symbol indicates the superuser (root) mode. Therefore, when you see a prompt string ending with $, it means you are logged into the system as a regular user. And when you see #, it means you are logged into the system as a user with superuser (root) privileges.

8. What is the purpose of the commands whereis and locate? What is the difference between them?

The whereis command is used to search for executable files, documentation, and source code of programs in system directories. It returns the paths to these files. The locate command is used for quick file searching in a database.

It searches for files and directories based on the pattern you specify. The main difference between them is that locate is faster because it uses an indexed database, while whereis searches directly in system directories. However, whereis can only find those files listed in the system's database file.

Conclusions:
During the laboratory work, I introduced basic commands in CLI mode in Linux. Introduced to basic text commands in terminal mode operation in various operating systems. Practiced my English language skills in report writing.