“Київський фаховий коледж зв’язку”

Циклова комісія комп’ютерної та програмної інженерії

**ЗВІТ ПО ВИКОНАННЮ**

**ЛАБОРАТОРНОЇ РОБОТИ №3**

з дисципліни: «Операційні системи»

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

Виконав(ла/ли) студент(ка/и)

групи: КСМ-23а

The Awkward Turtles:

Когут Б.М., Михайленко О.О.,

Трощинський Я.П.

Перевірила викладач

Сушанова В.С.

Київ 2024

**Мета роботи:**

1. Знайомство з базовими командами CLI-режиму в Linux.
2. Знайомство з базовими текстовими командами в термінальному режимі роботи в різних ОС.

**Матеріальне забезпечення занять:**

1. ЕОМ типу IBM PC.

2. ОС сімейства Windows та віртуальна машина Virtual Box (Oracle).

3. ОС GNU/Linux (будь-який дистрибутив).

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

***Виконав Михайленко Олексій***

1.  **Command interpreter** – a program that receives text commands from the user, interprets them, and performs the corresponding actions in the operating system. An example of a command interpreter is the command line interface (CLI) or terminal. It translates user commands into system calls to manage the computer’s resources.

 **Shell** – an interface between the user and the operating system. It allows the user to input commands that the operating system will execute. The shell can be textual (command line) or graphical (graphical user interface). Software shells may also include additional features like script handling and automation.

 **Command** – an individual instruction or request that a user or program sends to the operating system via a command interpreter or shell to perform a specific task. Commands can carry out various operations such as launching programs, working with files, managing processes, etc.

2.  **What basic information does the prompt line provide?**  
The prompt line in a command shell provides basic information about the system's current state. Typically, this includes the username, hostname, the current directory, and a symbol indicating readiness to accept a command (e.g., $ for a regular user or # for the superuser). For example:

user@hostname:~$

 **Why do commands need parameters and arguments?**  
Parameters and arguments modify a command’s behavior and define what the command should operate on. Parameters (also called "options" or "flags") specify how the command should perform the task, while arguments indicate the objects (such as files or directories) on which the command will operate. For example, in the command ls -l /home:

* -l is a parameter that changes the output format to detailed;
* /home is an argument specifying the directory to list.

 **What is the purpose of the ls command, and what parameters and arguments can it have? Give 3 examples.**  
The ls command is used to display a list of files and directories in the current or specified directory. Some parameters and examples:

* ls -l: detailed list of files with permissions, owner, size, and last modification time;
* ls -a: displays all files, including hidden ones (files that start with a dot);
* ls -lh: detailed file list with sizes shown in human-readable units (KB, MB, etc.).

 **How can the command history be used, and what advantages does it offer?**  
Command history allows users to view and reuse previously entered commands. This improves efficiency by eliminating the need to retype long or complex commands. The history command lists previous commands, or the arrow keys can be used to navigate through them. A specific command can be executed again by its number using !<number>. For example, !45 will execute command number 45 from the history.

 **What is the purpose of the echo command?**  
The echo command is used to display text or the values of variables in the terminal. It’s useful for outputting information or checking the values of environment variables. For example:

echo "Hello, World!"

or to display the value of a variable:

echo $HOME

 **Define the concept of a variable in the Bash shell. What types of variables does it support?**  
A variable in Bash is an object that stores data for use in scripts or commands. Bash supports two types of variables:

* **Local variables** – available only within the current session or script;
* **Environment variables** – available to all processes spawned from the current environment.

Variables can store various data types: strings, numbers, or the results of commands.

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

* env: displays a list of all environment variables or runs a program with specified environment variables;
* export: declares a variable as an environment variable, making it available to child processes;
* unset: deletes a variable, freeing up memory and making it inaccessible.

 **What commands for obtaining help on terminal commands do you know?**  
To get help on terminal commands, you can use:

* man <command> – displays a detailed manual for the command;
* --help – many commands support this option for a quick help summary (e.g., ls --help);
* info <command> – another way to get structured help documentation.