“Київський фаховий коледж зв’язку”

Циклова комісія Комп’ютерної інженерії

**ЗВІТ ПО ВИКОНАННЮ**

**ЛАБОРАТОРНОЇ РОБОТИ №3**

з дисципліни: «Операційні системи»

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

Виконали студенти

Групи КСМ – 13а

Команда Better Call Chekh:

Бродзінський Є.В.

Кравченко Т.І.

Тунда Р.О.

Перевірив викладач

Сушанова В.С.

Київ 2022

**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 (Windows 7).

3. Віртуальна машина – Virtual Box (Oracle).

4. Операційна система GNU/Linux – CentOS.

**Завдання для попередньої підготовки**

***Готував матеріал студент Бродзінський Є.В.***

1. Прочитайте короткі теоретичні відомості до лабораторної роботи та зробіть невеличкий словник базових англійських термінів з питань класифікації ОС та класифікації віртуальних середовищ.

| **Термін англійською** | **Термін українською** |
| --- | --- |
| Prompt | Запит |
| Bash Shell | Оболонка Bash |
| Inline Editing | Редагування на місці |
| Internal Commands | Внутрішні команди |
| External Commands | Зовнішні команди |
| Control Statements | Керуючі вирази |
| Quoting | Цитування |
| Double Quotes | Подвійні лапки |
| Single Quotes | Одинарні лапки |
| Back Quotes | Зворотні лапки |

1. Вивчіть матеріали онлайн-курсу академії Cisco “NDG Linux Essentials”:

- Chapter 5 - Command Line Skills

- Chapter 6 - Getting Help

1. Пройдіть тестування у курсі NDG Linux Essentials за такими темами:

- Chapter 05 Exam

- Chapter 06 Exam

1. Дайте визначення наступним поняттям:

- Command Interpreter

A command interpreter is software that runs in a text-based command-line interface, allowing users to input commands that are then executed by the operating system or other programs. The interpreter processes commands and passes them for execution, and it can also provide various useful features such as command history, inline editing, variables, and scripting. The most common command interpreter for many Linux systems is Bash (Bourne-Again Shell).

- Shell is an interface between the user and the operating system that allows users to interact with the system by entering text commands. The shell translates these commands into actions that the operating system or other programs execute.

- Command is a program or instruction entered by a user in a text-based command-line interface that tells the operating system or other program to perform an action. Commands can include options and arguments to change their behavior or provide additional information.

1. Дайте відповіді на наступні питання:

- What basic information does the prompt provide?

1)Username: The name of the user account that is using the shell.

2)Hostname: The name of the computer or server where the shell is running.

3)Current Directory: The path to the current working directory where the user is executing commands. Typically, this path is represented as a tilde (~) to abbreviate the long path to the user's home directory.

-What are the purposes of parameters and arguments for a command?

parameters and arguments are required for:

1)Data transmission: Parameters and arguments allow for the transfer of data from one part of the program to another. For example, you can pass a number or a string from one function to another for processing.

2)Passing information to functions: Functions in a program use parameters to receive data from the calling code. This enables functions to perform operations on this data and return results.

3)Configuration and settings: Parameters can be used to configure the behavior of functions and methods. For instance, you can pass a parameter to specify which options to enable or disable in a function.

4)Making code more generic: Using parameters allows you to create more generic and reusable functions. Instead of hardcoding values, functions can work with parameters, making them more adaptable to different situations.

5)Ensuring readability: Using named parameters and arguments helps improve the readability of code. Other programmers can more easily understand how your code works when they see which values are passed to functions.

6)Validation and error handling: Parameters enable the validation of input data for correctness and the handling of errors. Functions can check whether the passed arguments conform to expected data types and values.

1. Підготувати в електронному вигляді початковий варіант звіту:

- Титульний аркуш, тема та мета роботи

- Словник термінів

- Відповіді на п.4 та п.5 з завдань для попередньої підготовки

**Хід роботи**

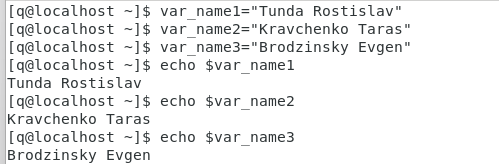
***Готував матеріал Бродзінський Є.В.***

1. Опрацюйте всі приклади команд, що представлені у лабораторній роботі курсу NDG Linux Essentials -

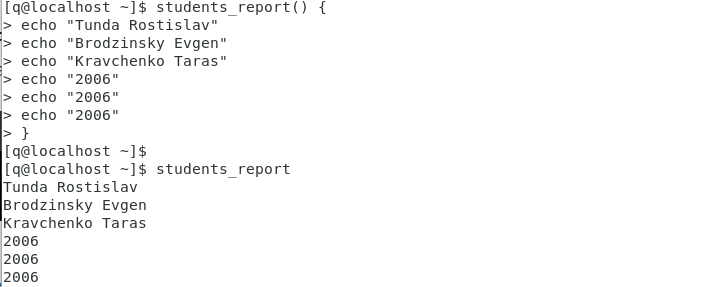
Lab 5: Command Line Skills та Lab 6: Getting Help. Створіть таблицю для опису цих команд\*\*\*

| Назва команди | Її призначення та функціональність |
| --- | --- |
| ls | Виводить інформації про каталоги та файли. За замовчуванням без аргументів відображає інформацію для поточного каталогу |
| ls -l | Використанні параметру **-l** в команді **ls** дозволяє відобразити інформацію про файли, розташовані в поточному робочому каталозі, у довгому форматі, який надає більш розширену додаткову інформацію |
| ls -l /tmp | Використання аргументу **/tmp** в поєднанні з параметром **-l** в команді **ls** дозволяєвідобразити детальну інформацію про файли в каталозі /tmp. |
|  |  |
| pwd | Виводить поточний каталог. |
| cd directory | Змінює поточний каталог на вказаний. |
| touch filename | Створює файл з вказаним ім'ям. |
| mkdir dirname | Створює новий каталог з вказаним ім'ям. |
| cp source destination | Копіює файл або каталог з одного місця в інше. |
|  |  |
| mv source destination | Переміщує файл або каталог з одного місця в інше. |
| rm file | Видаляє файл. |
|  |  |
| rmdir directory | Видаляє каталог. |
| cat file | Виводить вміст файлу на екран. |
| more file | Відображає вміст файлу посторінково. |
| less file | Відображає вміст файлу посторінково, дозволяючи прокручувати в обидві сторони. |
| head file | Виводить перші декілька рядків файлу. |
| tail file | Виводить останні декілька рядків файлу. |
| man command | Відображає довідкову інформацію для команди. |
| man -f keyword | Відображає список man-сторінок, що містять задане ключове слово. |
| whatis command | Показує короткий опис команди та пов'язаних з нею man-сторінок. |
| info command | Відображає докладну інформацію про команду за допомогою системи info. |
| date | Виводить поточну дату та час. |
| echo text | Виводить вказаний текст на екран. |
| echo Today is $(date) | Використовує дужки для виконання команди date та виводить поточну дату та час. |
| echo D\* | Виводить список файлів та каталогів, що починаються з літери "D". |
| echo This is the command \date`` | Використовує обернені лапки, щоб вивести текст з лапками, а не виконувати команду. |
| echo This is the command \date`` | Використовує обернені лапки, щоб вивести текст з лапками, а не виконувати команду. |
| echo "D\*" | Виводить текст "D\*". |
| man 5 passwd | Відображає докладну інформацію про файл "passwd" з розділу 5 (формати та конвенції файлів). |
| man -k keyword | Виводить короткий опис всіх man-сторінок, що містять задане ключове слово. |
| apropos keyword | Те ж саме, що і man -k keyword, виводить короткий опис всіх man-сторінок, що містять ключове слово. |
| whereis command | Показує місцезнаходження виконуваного файлу, man-сторінок та джерелного коду для команди. |

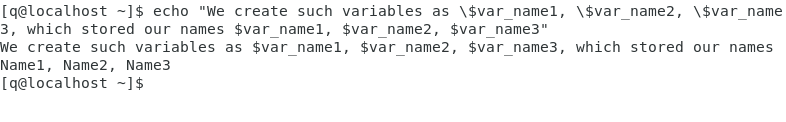
2. Робота в терміналі (закріплення практичних навичок) обов'язково представити свої скріншоти:



2.1.



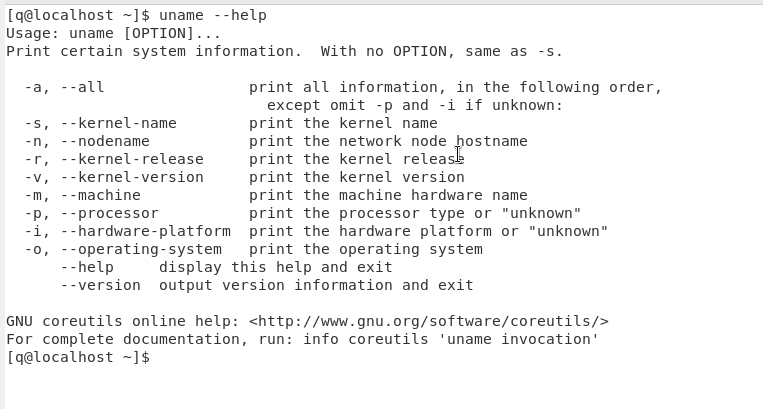
2.2.

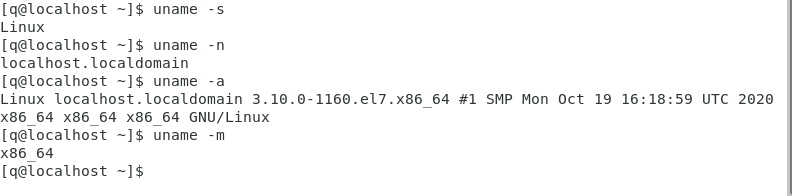
2.3.

—-------------------------------

2.4

2.5.





**Відповіді на контрольні запитання**

**Готував матеріал студент**

1. Types of commands in the Bash shell:

* Built-in commands: These are commands that are built directly into the Bash shell and execute faster as they do not require the creation of a new process. For example, cd, echo, alias are built-in commands.
* External commands: These are commands represented by separate executable files or programs and are executed as separate processes. For example, ls, grep, cat are external commands.
* Aliases: These are short names or alternatives for existing commands or combinations of commands. They can be created by users to simplify command usage.
* Functions: Users can create their own functions in the Bash shell to execute a sequence of commands.

2.Environment variables are variables that are defined for the entire system or for the current shell session and affect the behavior of processes and programs in the system. Environment variables can be global (available in all sessions) or local (limited to the context of the current session). They can be viewed in the terminal using the env or printenv command, and specific variables can be displayed, for example, echo $HOME will display the value of the HOME variable.

3.The $PS1 variable is a variable that defines the format of the prompt string in the Bash shell. The prompt string is displayed before each command. By default, it contains a value that looks something like this: \u@\h:\w\$, where \u represents the username, \h is the hostname, \w is the current directory, and $ is the $ symbol for regular users or # for superusers (root). The content of the $PS1 variable can be viewed using echo $PS1.

4.You can change the value of the $PS1 variable directly in the terminal. For example, to change the prompt string to MyPrompt>, enter the following command:

bash

Copy code

PS1="MyPrompt> "

The changes will immediately reflect in the prompt before each new command. To change the value of $PS1 permanently, not just for the current session, you can add this command to your .bashrc or .bash\_profile file, depending on your system.

5.Quotation marks in the Bash shell are used for:

Specifying strings that contain spaces or special characters, for example, 'this is a string with spaces'.

Indicating variables that need to be substituted into a string, for example, echo "Variable A = $A".

Defining commands that need to be executed and inserting their results, for example, `date` or $(date).

6.Control statements include conditional constructs, loops, and other structures that allow you to control the flow of command execution in the Bash shell. Some examples of control statements are if, for, while, case, and many others. They enable the creation of complex scripts and automation of tasks.

**Conclusion:**

In this work, we familiarized ourselves with the basic commands of the command-line interface (CLI) in the Linux operating system. Learning these commands is an essential part of working with Linux systems and enables us to perform various tasks, such as managing files and directories, navigating the file system, working with text files, and much more.