“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»**

Виконали студенти

групи КСМ-23А

Команда КГК:

Корольов Є.Ю.,

Горохов Д .С. та

Коваленко С.О.

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.

**Завдання для попередньої підготовки**

***Готував матеріал студентка Корольов Є.***

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

|  |  |
| --- | --- |
| Термін англійською | Термін українською |
| **Operating System** | Операційна система |
| **Command Line** | Командний рядок |
| **Shell** | Оболонка (командний інтерпретатор) |
| **Terminal** | Термінал |
| **Command** | Команда |
| **Scripting** | Скриптування |
| **Aliases** | Псевдоніми |
| **Variables** | Змінні |
| **Current Directory** | Поточний каталог |
| **Options** | Параметри |
| **Inline Editing** | Вбудоване редагування |

1. Прочитавши матеріал з коротких теоретичних відомостей дайте відповіді на наступні питання:

***Готував матеріал студент Горохов Д.***

1. Дайте визначення наступним поняттям:

* Командний інтерпретатор
* Оболонка
* Команда
* **Command Interpreter**: A program that processes commands entered by the user, typically via a command-line interface (CLI), and sends them to the operating system for execution.
* **Shell**: A user interface for accessing the operating system's services, typically a command-line interface (e.g., Bash, Zsh) that interprets and executes commands.
* **Command**: An instruction given by the user to perform a specific task, like copying files, listing directories, or running a program.

1. Яку базову інформацію надає рядок запрошення prompt?

The **prompt** in a command-line interface provides basic information such as:

1. **Username**: The current user logged into the system.
2. **Hostname**: The name of the computer or server.
3. **Current Directory**: The directory in which the user is currently working.
4. **Privileges**: It may also indicate user privileges, for example, showing # for root (administrator) or $ for regular users.
5. Для чого команді потрібні параметри та аргументи?

Parameters and arguments are used by the command to specify its actions and indicate how to perform the task.

* + Parameters (or options) change the behaviour of a command. They usually start with one or two dashes (- or --), such as -l for a detailed list in the ls -l command.
  + Arguments specify the objects on which the command performs an action, such as file or directory names: in the cp file1 file2 command, file1 and file2 are arguments.

This makes the command more flexible and specific.

1. Яке призначення команд ls, які параметри та аргументи вона може мати? Наведіть 3 приклади.

The **ls command** is used to list the contents of a directory in Linux. It shows files and directories by default, but can be modified with options (parameters) and arguments.

**Common options and arguments:**

* **-l**: Displays detailed information about files (permissions, size, etc.).
* **-a**: Shows hidden files.
* **-h**: Shows file sizes in human-readable format (e.g., KB, MB).

**Examples:**

1. ls -l: Lists files with detailed information.
2. ls -a /home: Lists all files, including hidden ones, in the /home directory.
3. ls -lh: Lists files with detailed, human-readable file sizes.
4. Яким чином можна використати історію команд, які переваги це надає?

The **command history** in Linux allows users to view and reuse previously executed commands. You can access it by typing history or using the **up/down arrow keys** to scroll through past commands.

**Advantages:**

1. **Saves time**: Quickly repeat or modify previous commands without retyping.
2. **Error correction**: Easily correct or rerun a command that failed.
3. **Efficiency**: Speeds up repetitive tasks.

Example: Use !23 to rerun the 23rd command in the history list.

1. Яке призначення команди echo?

The **echo command** is used to display text or variables in the terminal. It outputs the provided string or content to the screen or a file.

**Example:**

* echo "Hello, World!": Prints "Hello, World!" to the terminal.
* echo $USER: Displays the current username.

It is commonly used in scripts for displaying messages or outputting variables.

1. Охарактеризуйте поняття змінної в оболонці Bash, які типи змінних вона підтримує?

In **Bash**, a **variable** is used to store data that can be referenced and manipulated within scripts or the command line.

**Types of variables in Bash:**

1. **Local variables**: Defined and accessible only within the current shell or script.
2. **Environment variables**: Global variables accessible to the shell and any child processes (e.g., PATH, USER).
3. **Positional parameters**: Special variables representing arguments passed to a script (e.g., $1, $2).
4. **Special variables**: Predefined variables for specific functions (e.g., $? for the exit status).
5. Яке призначення команд env, export та unset?

The propose of these command:

1. **env**: Displays the current environment variables or runs a command in a modified environment.
2. **export**: Sets a variable as an environment variable, making it available to child processes.
3. **unset**: Removes a variable, either local or environment, from the shell.

These commands manage variables in the Bash environment.

1. Які команди для отримання довідки по командам в терміналі ви знаєте?
   1. help.
   2. help pwd.
   3. help -d pwd.
   4. help -s pwd.
   5. help -m pwd.

**Хід роботи**

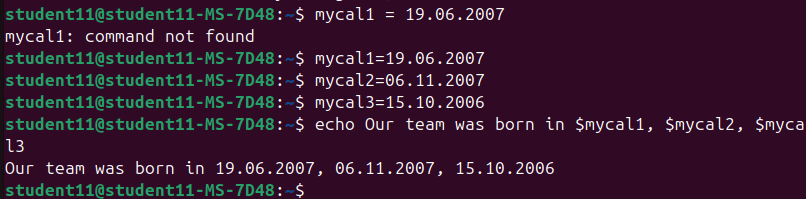
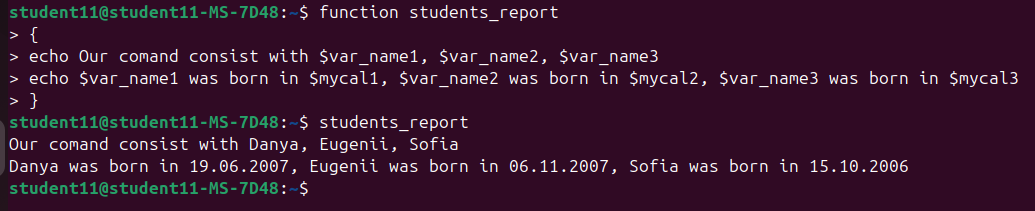
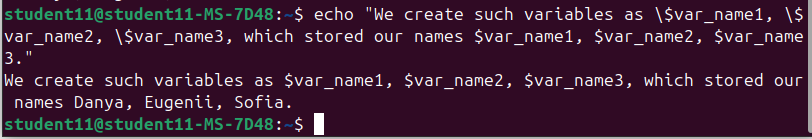
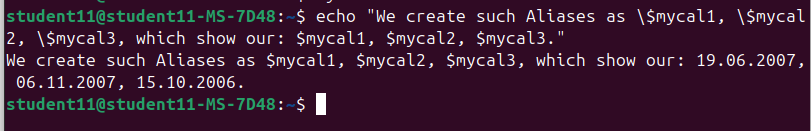
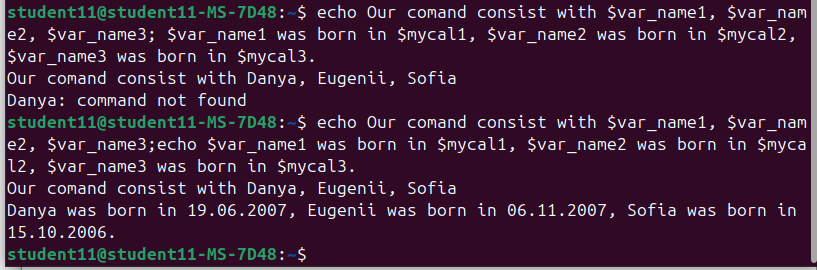
***Готували матеріали студенти Горохов Д., Корольов Є., Коваленко С.***

1. Робота в графічному режимі в ОС сімейства Linux:.

|  |  |
| --- | --- |
| 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. |
| ls -l /home | Використовуючи опцію -l і аргумент /home, ми можемо побачити, що в каталозі /home є каталог з назвою sysadmin, розташований всередині нього |
| whoami | У результаті виконання команди whoami, sysadmin, буде показано ім'я поточного користувача. Хоча у цьому випадку ваше ім'я користувача відображається у запиті, цю команду можна використати для отримання цієї інформації у ситуації, коли запит не містить цієї інформації. |
| uname | Ця команда покаже інформацію про поточну систему. |
| pwd | Команда pwd використовується для відображення вашого поточного «місцезнаходження» або поточного «робочого» каталогу. |
| history 5 | Щоб переглянути обмежену кількість команд, команда history може приймати число як параметр для відображення саме такої кількості останніх записів. |

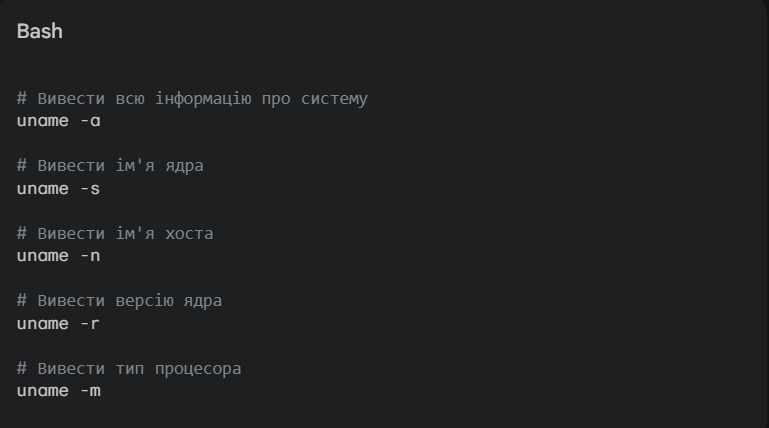
* 1. Опрацюйте всі приклади команд, що представлені у лабораторній роботі курсу ***NDG Linux Essentials - Lab 5: Command Line Skills*** та ***Lab 6: Getting Help.*** Створіть таблицю для опису цих команд.

1. Робота в терміналі (закріплення практичних навичок) **обов'язково представити свої скріншоти:**
   1. Робота зі змінними (Variables) та псевдонімами (Aliases) в терміналі:

* Створіть змінні, що будуть містити Ваші імена та прізвища $var\_name1, $var\_name2, $var\_name3
* За допомогою команди echo виведіть імена студентів вашої команди
* Створіть псевдоніми mycal1, mycal2, mycal3 для команди cal для автоматичного виведення календарю вашого року народження
  1. Робота з функціями (Functions) в терміналі:
* Створіть функцію students\_report, що порядково буде виводити спочатку імена студентів Вашої команди, а потім роки їх народження
  1. Робота з лапками (Quoting) в терміналі. Виведіть в командному рядку наступні речення:
* “We create such variables as $var\_name1, $var\_name2, $var\_name3, which stored our names Name1, Name2, Name3” (у реченні спочатку виводимо назви змінних, а потім їх вміст)
* “We create such Aliases as mycal1, mycal2, mycal3, which can show our calendars: Calendar1, Calendar2, Calendar3”  (у реченні спочатку виводимо назву команди-псевдонімів, потім вивід цих команд).
  1. Робота з інструкціями керування (Control Statements) в терміналі:.
* Чи можна завдання 2.1 та 2.2 ходу роботи виконати через інструкції керування без написання окремої функції, як це буде виглядати?
  1. Робота з командами довідки (Man Pages) в терміналі:
* На прикладі команди uname продемонструйте як отримати довідку. На основі отриманої додаткової інформації наведіть 5 різних варіантів виводу результату інформації по даній команді з використанням 5 різних параметрів (Options)

The uname command displays information about the system on which it is executed. This information can be useful for diagnosing and solving problems, as well as for writing scripts.

5 different options for displaying the output of the uname command using 5 different parameters:

* + uname -a: Displays all available information about the system, including the kernel name, host name, processor type, operating system version, and other detailed information.
  + uname -s: Displays the name of the operating system kernel (e.g. Linux, Darwin).
  + uname -n: Displays the host name of the system.
  + uname -r: Prints the kernel version.
  + uname -m: Displays the processor type.

**Відповіді на контрольні запитання**

***Готував матеріал студент Коваленко С.***

1. There are three main types of commands in the Bash shell:

Built-in commands: These are commands that are integrated directly into the Bash shell itself, such as cd, echo, pwd, exit. They don't require calling external programs and execute quickly.

External commands: these are separate programs or scripts that are located on the disk, for example, ls, grep, find. They are called from the command line.

Aliases: These are short names that refer to one or more commands. For example, alias ll='ls -la' creates the command ll, which executes ls -la.

2. Environment Variables are variables that store information about the shell and system session settings. They are used to store data such as directory paths, username, settings, and other parameters. There are the following basic environment variables:

System variables: configured by the operating system or the user, for example, PATH, HOME, SHELL.

User variables: created by the user to store their own settings.

View environment variables in the terminal:

Use the printenv or env command to print all variables.

Or use the echo $VAR\_NAME command to view a specific variable (for example, echo $HOME).

3. $PS1 is an environment variable that determines the appearance of the main prompt in Bash, that is, the line that is displayed before each command is entered. It usually includes the username, system name, current directory, etc.

You can view the contents of the $PS1 variable with the command:echo $PS1

4. You can change the value of the $PS1 variable by assigning a new value:

PS1="New prompt >”

After that, the changes are instantly displayed in the invitation line before the next command is entered.

To change the value of a variable on a permanent basis (not for the current session), you need to edit the ~/.bashrc file and add a line:

export PS1="New invitation > ”

After saving the changes in the file, you need to execute

source ~/.bashrc

to apply the new settings.

5. There are three main types of quotes in Bash, which are used for different purposes:

Single quotes (' '): keep the text as it is, without processing special characters and variables. For example, 'Hello $USER' will print Hello $USER.

Double quotes (” ”): allow you to interpret variables and special characters inside the quotes. For example, 'Hello $USER' will print Hello username.

\*\*Back quotes ( ):\*\* are used to execute commands inside an expression. For example, echo date will print the current date.

6. Control instructions in Bash are used to control the execution of commands and programs, create conditional statements, loops, and handle errors. Types of control instructions:

Conditional statements:

if, else, elif

case - an alternative for many if conditions.

Loops:

for - a loop to iterate through a list of values.

while - a loop that runs while the condition is true.

until - a loop that runs until the condition is true.

Jump instructions:

break - interrupts the execution of the loop.

continue - goes to the next iteration of the loop.

7. The $ symbol:

[centos@localhost Desktop] $ means that you are running as a regular user (non-root).

A normal user has limited access to system resources and cannot run commands that require elevated privileges.

Symbol #:

[root@localhost Desktop] # means that you are working as the root user (superuser).

Root has full system privileges: it can change system settings, edit all files, and execute high-level commands.

Explanation of the lines:

[centos@localhost Desktop] $

centos - the name of a regular user.

localhost - the name of the host (computer).

Desktop - the current working directory (Desktop folder).

$ - indicates that this is a regular user.

[root@localhost Desktop] #

root - the name of the superuser.

localhost - the name of the host.

Desktop - the current working directory.

# - indicates that this is the root user.

8. The whereis command:

Purpose: Searches for the location of executables (binaries), source code (source), and documentation (man pages) for a given command.

Usage: Often used to quickly find the location of system utilities or programs.

Command locate:

Purpose: Quickly searches for files by name throughout the file system.

Usage: Suitable for global search of files and folders by a specified mask or part of the name.Main differences:

Purpose:

whereis - for searching executable files, documentation, and source code.

locate - for global search of files and directories by name.

Search method:

whereis searches in specific system directories.

locate uses a database that indexes the contents of the file system, so searches are fast.

Scope of application:

whereis is typically used to find system programs.

locate is suitable for searching for ordinary user files and directories.

**Висновки**

In the course of this work, we have familiarized ourselves with the basic commands of the CLI mode (Command Line Interface) in the Linux operating system, which allow you to effectively work with the file system, manage processes, and configure system parameters. We also reviewed the basic text commands that provide terminal mode operation in various operating systems, such as Linux and Windows, which allows the user to automate routine tasks and increase overall productivity in command-line environments.