“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 БІКС-13

Лапчик Софія

Коломієць

Олександра

Перевірила

викладачка

Сушанова В.С.

Київ 2024

Робота студентів групи БІКС-13: Лапчик С., Коломієць О.

**Мета роботи:**1. Знайомство з базовими командами CLI-режиму в Linux.2. Знайомство з базовими текстовими командами в термінальному режимі роботи в різних ОС.  
  
**Матеріал підготувала Лапчик Софія**

**Завдання попередньої підготовки:** 1. \*Прочитайте короткі теоретичні відомості до лабораторної роботи та зробіть невеликий словник базових англійських термінів з питань призначення команд та їх параметрів..

|  |  |
| --- | --- |
| Термін Англійською | Термін Українською |
| A command the terminal | Команда терміналу |
| Man Pages | Довідка |
| Aliases | Псевдоніми |
| Variables | Змінні |
| Quoting | Цитування |
| Double quotes, single quotes, back quotes | Подвійні лапки, одинарні лапки, зворотні лапки |
| Control statements | Інструкції керування |

**Матеріал підготувала Коломієць О.**

4. Definitions:

- Command-line interpreter: Software that allows users to interact with the operating system by entering commands.

- Shell: A set of programs for organizing user interaction with the computer.

- Command: An expression that defines a single operation and its operands.

5. \*\*Answers to the questions:\*\*

- \*\*What basic information does the prompt provide?\*\* The prompt provides users with basic information to interact with the command line. This information includes the username or identifier, possibly information about the current machine or subsystem, the current working directory (path), a greeting message or instructions, and special symbols or indicators to display the system's state. The prompt facilitates navigation and user interaction with the command line by providing context and orientation in its usage.

- \*\*What are parameters and arguments needed for commands?\*\* Parameters and arguments in commands of various programs and operating systems are used to pass additional information to the program during its invocation.

- \*\*What is the purpose of the `ls` command, and what parameters and arguments can it have? Provide 3 examples.\*\* The `ls` command in UNIX-like operating systems is used to view the contents of the current directory. It displays a list of files and subdirectories in the current working directory.

Basic usage: `ls` - This command lists all files and subdirectories in the current directory.

Displaying additional information: `ls –l` - In this case, the `-l` parameter instructs the `ls` command to display additional information about files, such as permissions, owner, size, creation date, etc.

Displaying hidden files: `ls –a` - Using the `-a` parameter, the `ls` command will list all files in the current directory, including those that begin with a dot, which typically denote hidden files.

- \*\*How can you use command history, and what advantages does it provide?\*\* Using command history is an important feature of command-line shells that allows users to efficiently manage and repeat previously entered commands. This capability enables viewing and using previous commands, saving time, and assisting in performing routine tasks. Utilizing command history allows not only quickly finding and repeating previous actions but also editing commands before execution, enhancing productivity, and reducing the likelihood of errors.

- \*\*What is the purpose of the `echo` command?\*\* The `echo` command in UNIX-like operating systems is used to output a text message or the value of a variable to standard output (usually the terminal). The primary purpose of the `echo` command is to display textual information on the screen. It is useful for executing shell scripts, debugging, displaying variables, and more.

- Characterize the concept of variables in the Bash shell, and what types of variables does it support?A variable in the Bash shell is a character string that contains information that can change during the execution of a script or command. It is used to store data such as texts, numbers, or file names that can be used in a script for various calculations or operations.

The types of variables supported by the Bash shell include:

1. String variables: They store textual information and can contain any characters, including letters, digits, and special characters.

2. Numeric variables: They are used to store numerical values, such as integers or floating-point numbers. In Bash, numeric variables typically do not require defining data types and can contain digits or mathematical expressions.

3. Arrays: Arrays in Bash allow storing multiple values as a sequence of elements, which can be addressed using indices.

4. Associative arrays: This is a special type of array where array elements are addressed using keys rather than numeric indices.

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

- `env` command: It allows executing commands in a specified environment, consisting of a set of environment variables.

- `export` command: Used to create new environment variables or to extend the scope of existing variables, making them available to child processes.

- `unset` command: Used to remove environment variables.

Хід роботи:  
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

|  |  |
| --- | --- |
| Command name | Its purpose and functionality |
| ls | Displays information about directories and files. By default, with no arguments, displays information for the current directory |
| ls -l | Using the **-l** option in the **ls** command allows you to display information about files located in the current working directory in a long format that provides more extensive additional information |
| ls -l /tmp | Using the **/tmp** argument in combination with the **-l** option in the **ls** command allows you to display detailed information about the files in the /tmp directory. |
| date | Display today's date. |
| uname | Displays information about the current system. |
| man 5 passwd | To display a man page for a different section, provide the section number as the first argument to the man command. |

Kolomiiets

2.1# Створення змінних для імен та прізвищ студентів

var\_name1="Lapchik"

var\_name2="Kolomiiets"

# Виведення імен студентів за допомогою команди echo

echo "Ім'я студента 1: $var\_name1"

echo "Ім'я студента 2: $var\_name2"

# Створення псевдонімів для команди cal для автоматичного виведення календарю року народження

alias mycal1="cal 2006"

alias mycal2="cal 2006"

2.2students\_report() {

echo "Ім'я студента 1: $var\_name1"

echo "Ім'я студента 2: $var\_name2"

echo "Рік народження студента 1: 2006"

echo "Рік народження студента 2: 2006"

}

2.3

echo "We create such variables as \$var\_name1, \$var\_name2, which stored our names Sasha1, Sofia2"

echo "We create such Aliases as mycal1, mycal2, which can show our calendars: Calendar1, Calendar2, Calendar3"

**Відповіді на контрольні запитання:  
  
Матеріал підготувала Лапчик Софія**

1. What types of commands exist in the Bash Shell?  
Built-in and external.

2. What are environment variables? What are they? How can they be viewed in the terminal?  
Environment variables are essentially name-value pairs that store information about the environment in which programs run.  
Viewing in Bash:  
env: This command displays a list of all environment variables.  
echo $VARNAME: This command shows the value of a specific named variable.  
export: This command lists the exported environment variables, which are accessible by child processes started from your current shell.

3. \* Describe the $PS1 variable. How to view its contents in the terminal?  
The $PS1 variable in various shells, most notably Bash, is responsible for defining the primary prompt string.

Viewing:  
echo $PS1  
set | grep PS1

Матеріал підготувала Коломієць О.

4. To change the value of the `$PS1` variable in the current Bash shell session, use the command:

```bash

export PS1="new value"

```

To persist this change as the default, add the appropriate command to your shell configuration file (typically `.bashrc`). For example:

```bash

echo 'export PS1="new value"' >> ~/.bashrc

```

This will change the prompt in the Bash shell to "new value" for subsequent logins.

5. In the Bash shell, quotes (single or double) are used for several purposes:

1. Specifying string values: Quotes are used to enclose string values to combine them into a single object. For example:

- Single quotes: `'hello'`

- Double quotes: `"world"`

2. Escaping special characters: Quotes allow entering a string that contains spaces or special characters that are interpreted by the shell. For example:

- `"Hello World"`

- `'This is a test'`

3. Variable substitution: When using double quotes, variable values are evaluated (usually), while single quotes are treated literally. For example:

- Double quotes: `"Hello, $USER"`

- Single quotes: `'Hello, $USER'`

6. \*\*Control Instructions Purpose and Types:\*\*

Control instructions are used to change the sequence of command execution in programs or scripts. They allow conditional branching, looping, and other actions that control the flow of program execution. Some types of control instructions include:

- Conditional statements (if-else): Used to execute code based on certain conditions.

- Iterative statements (for, while): Used to repeat the execution of a certain block of code until a condition is met.

- Jump statements (break, continue): Used to alter the flow of program execution, such as exiting a loop or jumping to the next iteration of a loop.

- Selection statements (switch-case): Used to select the execution of a specific block of code based on the value of a variable.

7. \*\*Difference if the bash prompt ends with $ or #?\*\*

In bash, the symbol $ or # at the end of the prompt indicates the user's status or access level:

- The $ symbol indicates that the current user is a regular user (not an administrator), and thus, a regular user shell is being used.

- The # symbol indicates that the current user is an administrator or has elevated privileges (such as root), and thus, an administrative shell is being used.

8. \*\*Purpose of `whereis` and `locate` commands? What is the difference between them?\*\*

- The `whereis` command is used to search for the location of executable files, program sources, and their manual pages in system directories and typically searches in standard search directories such as `/bin`, `/sbin`, `/usr/bin`, etc.

- The `locate` command is used to search for files in a database created using the `updatedb` program. It quickly searches by utilizing an indexed database.

The difference between them lies in the fact that `whereis` searches for a file in standard system directories, whereas `locate` searches in a previously created database. `locate` can be faster but may not show the latest changes in the file system as it needs to update the database before use.

In total

Шляхом виконання роботи ми ознайомились з базовими командами CLI-режиму в Linux та з базовими текстовими командами в термінальному режимі роботи в різних ОС.