“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 РПЗ-03
Кошіль Владислав та
Фещенко Эвгеній.
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 Фещенко Эвгеній.

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

Термін англійською	**Термін англійською**
prompt	**підказка**
shell	**оболонка**
environment	**середовище**
variables	**змінні**

Дайте відповіді на наступні питання:

- **Яку базову інформацію надає рядок запрошення prompt?**

The prompt line provides the user with basic information about the current state of the system or process in the terminal that allows you to enter commands. Typically, the prompt contains the user name, the system name, the current directory (or other information about the current context), and a prompt character (such as $ or >) to indicate that the system is ready to accept the next command. The user can enter commands after this prompt character.

- **Для чого команді потрібні параметри та аргументи?**

Parameters and arguments are important elements of commands in most operating systems. They allow you to pass additional information to commands, allowing you to perform different actions depending on the values passed.

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

The ls command is used to display a list of files and folders in the current working directory. Some of the options and arguments that can be used with the ls command include:
-l: displays detailed information about files and folders such as permissions, owner, group, size, creation date, etc.
-a: displays all files and folders, including hidden ones.
<directory path>: used to display a list of files and folders in a specific directory, rather than the current working directory.

- **Яким чином можна використати історію команд, які переваги це надає?**

Command history is a list of previously executed commands in the terminal that is stored in the computer's memory.
You can use the history commands:
history: display a lis.t of all commands that have been executed in the terminal
!n: execute the nth command from the history list
!!: execute the last command entered
Using the command history allows you to work more efficiently with the terminal, reduces the number of characters you type, and helps you avoid repeated errors. It also allows you to quickly repeat previous actions or execute the same commands with different parameters

- **Яке призначення команди echo?**

The echo command is used to display a text string on the screen or add it to a file. It is often used in scripts to display messages to the user or to save data to a file.

- **Охарактеризуйте поняття змінної в оболонці Bash, які типи змінних вона підтримує?**

A variable in the Bash shell is a name that is assigned to a specific value. It can contain any data, such as strings of text, integers, or Boolean values.
Bash supports two types of variables: local and global. Local variables are defined within the current shell and are not available to other processes. Global variables are available to all processes running in the current Bash shell session.

- **Яке призначення команд env, export та unset?**

The commands env, export, and unset are used to manage environment variables in a Unix/Linux shell environment.

The env command is used to display the current environment variables or run a program in a modified environment.
The export command is used to create or modify environment variables that can be accessed by child processes. It is used to make a variable global or visible to all subprocesses of the current shell.
The unset command is used to remove a variable or function from the environment.
Environment variables are used to store information that can be accessed by different processes running on the system. They are also used by various programs to configure their behavior based on the value of these 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***

Готував матеріал студент Кошіль Владислав

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 дозволяє відобразити детальну інформацію про файли в каталозі /tmp.
whoami	The output of the whoami command, sysadmin, displays the user name of the current user. Although in this case your username is displayed in the prompt, this command could be used to obtain this information in a situation when the prompt did not contain this information.
uname	Execute the uname command again twice in the terminal, once with the option -n and again with the option --nodename.
pwd	This is referred to as your home directory, a special place where you have control of files and other users normally have no access.
history	To view a limited number of commands, the history command can take a number as a parameter to display exactly that many recent entries.
!9	To execute a command again, type the exclamation point and the history list number. For example, to execute the 9th command in your history list
echo Hello Student	The echo command can be used to print text and the value of a variable, and to show how the shell environment expands metacharacters (more on metacharacters later in this lab)
echo $PATH	The PATH variable is displayed by placing a $ character in front of the name of the variable.
type command	The type command identifies the cd command as an internal command.
which ls	External commands are binary executables stored in directories that are searched by the

	shell. Use the which command to display the full path to the ls command.
Aliases	Aliases can be used to map longer commands to shorter key sequences. When the shell sees an alias being executed, it substitutes the longer sequence before proceeding to interpret commands.
/file	Searches are not case sensitive and do not "wrap" around from the bottom to top, or vice versa. Start a forward search for the word "file" by typing:
man -k password	In some cases you may not remember the exact name of the command. In these cases you can use the -k option to the man command and provide a keyword argument.
man -f passwd	There are often multiple man pages with the same name. For example, the following command shows three pages for passwd. Execute the following command to view the man pages for the word passwd:
apropos password	Note that the apropos command is another way of viewing man page summaries with a keyword. Type the following command:
whatis passwd	Instead of using man -f to display all man page sections for a name, you can also use the whatis command:
info date	Almost all system features (commands, system files, etc.) have man pages. Some of these features also have a more advanced feature called info pages.
date --help	Another way of getting help is by using the --help option to a command. Most commands allow you to pass an argument of --help to view basic command usage:
date	Execute commands in the bash shell by typing the command and then pressing the Enter key.
man date	To learn more about commands, access the manual page for the command with the man command.
date --help	Another way of getting help is by using the --help option to a command. Most commands allow you to pass an argument of --help to view basic command usage:
which date	The output of the which command tells you that when you execute the date command, the system will run the command /bin/date.

2.Робота в в терміналі (закріплення практичних навичок):

1. Робота зі змінними (Variables) та псевдонімами (Aliases) в терміналі:

Готував матеріал студент Фещенко Эвгеній

```
[root@localhost ~]# var_name1="Koshil Vlad"
[root@localhost ~]# var_name2="Feshenko Jenya"
[root@localhost ~]# echo $var_name1 $var_name2
Koshil Vlad Feshenko Jenya
```

```
[root@localhost ~]# alias mycal1="cal 2 2005"
[root@localhost ~]# alias mycal2="cal 4 2005"
```

2. Робота з функціями (Functions) в терміналі:

Готував матеріал студент Кошіль Владислав

```
[root@localhost ~]# students_report () { echo $var_name1, $var_name2; mycal1; mycal2; }
[root@localhost ~]# students_report
Koshil Vlad, Feshenko Jenya
    Февраль 2005
Пн Вт Ср Чт Пт Су Вс
    1  2  3  4  5  6
 7  8  9 10 11 12 13
14 15 16 17 18 19 20
21 22 23 24 25 26 27
28

     Апрель 2005
Пн Вт Ср Чт Пт Су Вс
             1  2  3
 4  5  6  7  8  9 10
11 12 13 14 15 16 17
18 19 20 21 22 23 24
25 26 27 28 29 30
```

3. Робота з лапками (Quoting) в терміналі. Виведіть в командному рядку наступні речення:

Готував матеріал студент Фещенко Эвгеній

```
[root@localhost ~]# echo "We create such variables as \$var_name1, \$var_name2, wich stored our name
s $var_name1, $var_name2"
We create such variables as $var_name1, $var_name2, wich stored our names Koshil Vlad, Feshenko Jeny
a
```

```
[root@localhost ~]# echo We create such Aliases as \mycal1, \mycal2 wich can show our calendars:;myc
al1;mycal2;
We create such Aliases as mycal1, mycal2 wich can show our calendars:
    Февраль 2005
Пн Вт Ср Чт Пт Су Вс
    1  2  3  4  5  6
 7  8  9 10 11 12 13
14 15 16 17 18 19 20
21 22 23 24 25 26 27
28

     Апрель 2005
Пн Вт Ср Чт Пт Су Вс
             1  2  3
 4  5  6  7  8  9 10
11 12 13 14 15 16 17
18 19 20 21 22 23 24
25 26 27 28 29 30
```

4. Робота з інструкціями керування (Control Statements) в терміналі:

Готував матеріал студент Кошіль Владислав

```
[root@localhost ~]# echo $var_name1, $var_name2 ;mycal1;mycal2;
Koshil Vlad, Feshenko Jenya
    Февраль 2005
Пн Вт Ср Чт Пт Су Вс
    1  2  3  4  5  6
 7  8  9 10 11 12 13
14 15 16 17 18 19 20
21 22 23 24 25 26 27
28

     Апрель 2005
Пн Вт Ср Чт Пт Су Вс
            1  2  3
 4  5  6  7  8  9 10
11 12 13 14 15 16 17
18 19 20 21 22 23 24
25 26 27 28 29 30
```

5. Робота з командами довідки (Man Pages) в терміналі:

Готував матеріал студент Фещенко Эвгеній

```
[root@localhost ~]# uname --kernel-name
Linux
[root@localhost ~]# uname --nodename
localhost.localdomain
[root@localhost ~]# uname --kernel-release
3.10.0-1160.el7.x86_64
[root@localhost ~]# uname --machine
x86_64
[root@localhost ~]# uname --operating-system
GNU/Linux
```


Відповіді на контрольні запитання

Готував матеріал студент Фещенко Эвгеній та Кошіль Владислав

1. Які типи команд існують в оболонці Bash?

There are several types of commands that exist in the Bash shell:

1. Shell built-in commands: These are commands that are built into the shell itself, and are executed directly by the shell. Examples of built-in commands include cd, alias, and echo.
2. Shell functions: These are user-defined functions that are defined within the shell environment. They are similar to shell scripts, but are stored in memory instead of as separate files. Functions are executed by the shell just like any other command.
3. Aliases: These are user-defined shortcuts for longer commands or command options. They are defined using the alias command and are expanded by the shell before execution.
4. Executable programs: These are external commands that exist as separate executable files in the file system. They are executed by the shell by searching the directories listed in the PATH environment variable for the command name.

5. Shell scripts: These are files containing a series of shell commands and instructions that are executed as a single command. They can be executed directly from the command line or called from other scripts or programs.

2. Що таке змінні оточення? Які вони бувають. Як їх можна переглянути в терміналі?

Environment variables are variables used by the operating system to store values that can be accessed by processes or programs running on the system. They are typically used to define system-wide settings, such as default paths for executables or settings for specific applications.

Environment variables are stored in a special area of memory that is accessible to all processes on the system. They are created and modified using specific commands or tools provided by the operating system or shell.

In Bash, environment variables can be viewed using the printenv command or by simply typing the name of the variable preceded by a dollar sign ($). For example, to view the value of the PATH environment variable, you can type echo $PATH.

3. Опишіть змінну $PS1. Як в терміналі переглянути її вміст?

The $PS1 variable in Bash stands for "Prompt String 1". It is a shell variable that specifies the primary prompt string that is displayed in the terminal. The prompt string defines the format of the command line prompt, which typically includes information such as the current working directory, username, hostname, and other customized text.

The format of the $PS1 variable can be customized by setting various escape sequences, which begin with a backslash (\) followed by a special character that represents a particular piece of information. For example, the escape sequence \u represents the current username, and \W represents the current working directory.

To view the contents of the $PS1 variable in the terminal, you can simply enter the following command:

bash
Copy code
echo $PS1
This will display the value of the $PS1 variable, which can be modified using the export command.

4. Як можна змінити значення змінної $PS1? Що при цьому відбудеться в рядку запрошенні в bash (рядок запрошення перед початком кожної команди). Як змінити значення цієї змінної не на поточний сеанс, а за замовчуванням?

To change the value of the $PS1 variable, you can simply assign a new value to it using the = operator. For example, to change the prompt to display the current user and working directory, you can use the following command:

python
Copy code
PS1='\u:\w\$ '
This will set the value of $PS1 to the string '\u:\w\$ ', where \u is replaced with the current user's username and \w is replaced with the current working directory. The $ character indicates the end of the prompt.

Once you have changed the value of $PS1, the new prompt will be displayed before each command in the terminal.

If you want to change the value of $PS1 permanently (i.e., not just for the current session), you can add the export command to your .bashrc file. This file is executed each time you start a new Bash session, so any variables that you set or change in this file will be available each time you open a new terminal window.

5. Для чого використовують лапки в оболонці Bash?

Quotation marks in the Bash shell are used to control how the shell interprets special characters and spaces in strings. There are three types of quotes that can be used:

Single quotes (' '): When a string is enclosed in single quotes, all special characters within the quotes lose their special meaning. The string is treated literally, so variables and command substitution are not evaluated within single quotes. For example, the string '$PATH' will be printed as is, instead of the actual value of the PATH environment variable.

Double quotes (" "): When a string is enclosed in double quotes, some special characters lose their special meaning, but variables and command substitution are still evaluated within the quotes. For example, the string "$PATH" will be replaced with the value of the PATH environment variable.

Backticks ( ): Backticks are used for command substitution. When a command is enclosed in backticks, it is executed and the output is substituted into the command line. For example, the command date will be replaced with the current date and time.

Висновок: During this lab, we learned about the basic commands of the CLI mode in Linux and also learned how to work with basic text commands in terminal mode in different operating systems.