“Київський фаховий коледж зв’язку”

Циклова комісія Комп’ютерної інженерії

**ЗВІТ ПО ВИКОНАННЮ**

**ЛАБОРАТОРНОЇ РОБОТИ №3**

з дисципліни: «Операційні системи»

**Тема: «Ознайомлення з робочим середовищем віртуальних машин та операційних систем різних сімейств»**

Виконав студент

групи БІКС-13

Когут Ігор Святославович  
Береза Артем

Перевірив викладач

Сушанова В.С.

Київ 2022

**The goal of the work(Когут)**

1. Acquiring practical skills in working with virtual machine environments and operating systems of various types and families - their graphical shell, logging in and out of the system, familiarization with the structure of the desktop, learning the basic actions and settings when working in the system.

**Material provision of classes**

1. IBM PC type computer.

2. OS family Windows (Windows 7).

3. Virtual machine - Virtual Box (Oracle).

4. GNU/Linux operating system - CentOS.

**Progress (Береза)**

**Answers to control questions (Когут)**

**1.Types of commands in the Bash shell can be classified as follows:**

**Built-in commands:** These are commands that are built directly into the Bash shell. Examples include cd, echo, alias, etc.

**External programs:** These are commands located in executable files in system directories or directories specified in the PATH variable.

Shell scripts: Files with .sh extension containing a sequence of Bash shell commands.

**2.Environment variables are variables that hold information about the current shell session.** They can be system-defined (e.g., PATH, HOME) or user-defined. You can view them in the terminal using the printenv or env command.

3.The $PS1 variable defines the prompt string for the Bash shell. It is what is displayed before each command. To view the contents of this variable in the terminal, you can use the echo $PS1 command.

**4.To change the value of the $PS1 variable, you simply assign it a new value.** For example, PS1="new\_prompt\_string". After this change, the new string will be displayed in the prompt. To change the $PS1 value permanently, you would modify it in the shell configuration file, such as ~/.bashrc or ~/.bash\_profile.

**5.Quotes in the Bash shell** are used to indicate strings that contain spaces or special characters and should be treated as a single argument. For example, if you want to pass a string with spaces as a single argument to a command, you use quotes, like command "string with spaces". Quotes help the shell properly parse arguments.

**6. Control statements are used in programming to alter the flow of execution of a program. They enable conditional branching, looping, and other operations that change the logic of program execution. The main types of control statements include:**

**Conditional statements:**

**if:** Executes a block of code if a specified condition is true.

**else:** Executes a block of code if the condition in the if statement is false.

**elif:** Executes if the preceding if or elif condition is false and the specified condition is true.

**switch:** Executes different actions based on the value of an expression.

Iterative statements:

**for:** Executes a block of code a specified number of times or for each item in a list.

**while:** Executes a block of code as long as the specified condition is true.

**do-while:** Executes a block of code once and then checks the condition for continuation.

**Flow control:**

**break:** Terminates the execution of a loop or switch statement.

**continue:** Skips the rest of the code in the current iteration of a loop and proceeds to the next iteration.

**return:** Returns a value from a function and terminates its execution.

**exit:** Terminates the execution of a program.

These statements allow programmers to control the flow of program execution and perform different actions based on conditions and the current state of the program.

**7. The difference between the two prompts is the user's privilege level.**

$ indicates a regular user.

# indicates the root user (superuser).

The root user has full access to the system, while regular users have limited access.

**8.The commands *whereis* and *locate* are both used for searching files in the file system, but they have some differences in functionality and usage:**

**The whereis Command:** whereis is designed for quickly locating executable files, binary files, and sources for a given command in system paths.

**The locate Command:** locate is designed for quickly finding files and directories based on an indexed database. It can locate any file or directory in the file system, including those located in user directories.

Thus, the main difference between the **whereis** and **locate** commands lies in the fact that whereis searches for specified types of files in standard system paths, whereas locate searches for any files and directories in a large indexed database.

**Conclusion**

In the course of the laboratory work, I studied the Linux system and its distributions, theoretically studied the issue of the system's operation in more detail. Acquired practical skills of working with the operating system.