“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)

1.

command	purpose and functionality
ls	Type the ls command at the prompt, which will list the files and directories contained in your current working directory. In this example, no options or arguments will be used.
ls -l	Use the -l option to display this information in the long format, which gives additional information about files located in the current working directory
ls -l /home	Use the argument /home to display detailed information about files in the /home directory
whoami	The output of the whoami command, sysadmin, displays the user name of the current user
uname	command displays information about the current system
uname -n	Traditionally in UNIX, options were expressed by a hyphen followed by another character
uname --nodename	In Linux, options can sometimes also be given by two hyphen characters followed by a word, or hyphenated word
pwd	command is used to display your current "location" or current "working" directory
history	command to be able to view a numbered history list
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)
which	command to determine if there is an executable file, in this case named date, that is located within a directory listed in the PATH value
type	command can be used to determine information about command type
alias	To determine what aliases are set on the current shell
date	display today's date
man	To learn more about commands, access the manual page for the command with the man command
apropos password	command is another way of viewing man page summaries with a keyword
--help	Another way of getting help is by using the --help option to a command
locate	An easy way to search for a file is to use the locate command
whereis	You may just want to find where a command (or its man pages) is located.

2.

```
viczk@viczk-VirtualBox:~$ NEW_VAR="Bereza"
viczk@viczk-VirtualBox:~$ NEW_VAR2="Kogut"
viczk@viczk-VirtualBox:~$ echo $NEW_VAR2
Kogut
viczk@viczk-VirtualBox:~$ echo $NEW_VAR
Bereza
viczk@viczk-VirtualBox:~$
```

2.1.

```
viczk@viczk-VirtualBox:~$ alias mycal1='cal 2005'
viczk@viczk-VirtualBox:~$ alias mycal2='cal 2005'
viczk@viczk-VirtualBox:~$ $mycal1
viczk@viczk-VirtualBox:~$ mycal1
                                2005
       Січень                  Лютий                  Березень
нд пн вт ср чт пт сб  нд пн вт ср чт пт сб  нд пн вт ср чт пт сб
                   1           1  2  3  4  5           1  2  3  4  5
 2  3  4  5  6  7  8   6  7  8  9 10 11 12   6  7  8  9 10 11 12
 9 10 11 12 13 14 15  13 14 15 16 17 18 19  13 14 15 16 17 18 19
16 17 18 19 20 21 22  20 21 22 23 24 25 26  20 21 22 23 24 25 26
23 24 25 26 27 28 29  27 28                 27 28 29 30 31
30 31

       Квітень                 Травень                 Червень
нд пн вт ср чт пт сб  нд пн вт ср чт пт сб  нд пн вт ср чт пт сб
                1  2   1  2  3  4  5  6  7              1  2  3  4
 3  4  5  6  7  8  9   8  9 10 11 12 13 14   5  6  7  8  9 10 11
10 11 12 13 14 15 16  15 16 17 18 19 20 21  12 13 14 15 16 17 18
17 18 19 20 21 22 23  22 23 24 25 26 27 28  19 20 21 22 23 24 25
24 25 26 27 28 29 30  29 30 31              26 27 28 29 30
```

2.2.

```
viczk@viczk-VirtualBox:~$ function students_report {
> echo "Імена:"
> echo $NEW_VAR
> echo $NEW_VAR2
> echo "календарі:"
> mycal1
> mycal2
> }
viczk@viczk-VirtualBox:~$ studets_report
studets_report: команду не знайдено
viczk@viczk-VirtualBox:~$ ^C
viczk@viczk-VirtualBox:~$ students_report
Імена:
Bereza
Kogut
календарі:
                        2005
       Січень                  Лютий                  Березень
нд пн вт ср чт пт сб  нд пн вт ср чт пт сб  нд пн вт ср чт пт сб
                   1           1  2  3  4  5           1  2  3  4  5
 2  3  4  5  6  7  8   6  7  8  9 10 11 12   6  7  8  9 10 11 12
 9 10 11 12 13 14 15  13 14 15 16 17 18 19  13 14 15 16 17 18 19
16 17 18 19 20 21 22  20 21 22 23 24 25 26  20 21 22 23 24 25 26
```

2.3.

```
viczk@viczk-VirtualBox:~$ echo "We create such variables as \$NEW_VAR, \$NEW_VAR
2, which stored our names $NEW_VAR, $NEW_VAR1"
We create such variables as $NEW_VAR, $NEW_VAR2, which stored our names Bereza,
viczk@viczk-VirtualBox:~$
```

2.4.

```
viczk@viczk-VirtualBox:~$ echo "We create such Aliases as \mycal1, \mycal2, whic
h can show our calendars: mycal1, mycal2"
We create such Aliases as \mycal1, \mycal2, which can show our calendars: mycal1
, mycal2
```

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:
1. 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.
2. 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` (in some programming languages): Executes a block of code once and then checks the condition for continuation.
3. 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.
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:
1. The `whereis` Command:Purpose: `whereis` is designed for quickly locating executable files, binary files, and sources for a given command in system paths.
2. 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.