"Kyiv Professional College of Communication"

Cycle Commission of Computer Engineering

REPORT ON THE IMPLEMENTATION

LABORATORY WORK №7

in the discipline: "Operating systems"

Topic: " Create scripted scenarios and define the hardware configuration of the system "

Performed by

student

of BICS-13 group

Zlata Volodymyrivna Los

Checked by the teacher

Sushanova V.S.

Kyiv 2024

**Objectives:**

1. Gaining practical skills in working with the Bash shell.

2. Familiarization with basic actions when working with scripting.

**Material support of classes:**

1. Computer such as IBM PC.

2. Windows operating system and Virtual Box (Oracle) virtual machine.

3. GNU/Linux OS (any distribution).

4. The Cisco Network Academy website netacad.com and its online Linux courses.

**Tasks for preliminary preparation:**

1. \*Read the brief theoretical information for the lab and make a small glossary of basic English terms on the purpose of commands and their parameters.

Shell Script is a text file containing a series of commands that can be executed by the shell or the underlying operating system.

Shebang - a sequence of two characters: "lattice" and exclamation mark (#!), which is placed at the beginning of a file (text) written in a scripting language.

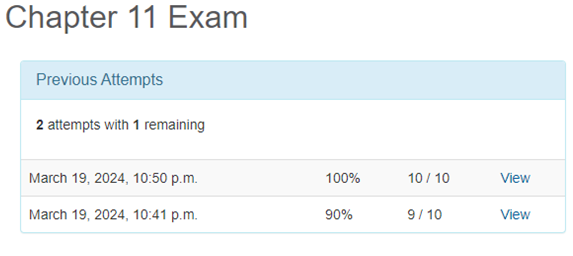
Nano - a text editor for the console.

2. Study the materials in the NDG Linux Essentials online course at Cisco Academy:

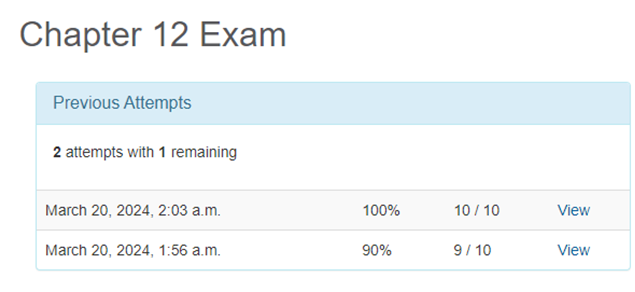
* Chapter 11 - Basic Scripting
* Chapter 12 - Understanding Computer Hardware

3. Take the NDG Linux Essentials course exams on the following topics:

* Chapter 11 Exam



* Chapter 12 Exam



4. Based on the material reviewed, answer the following questions:

**4.1. \* Describe the concept of a scripting script in a command shell.**

A shell script is a file that contains a sequence of commands that are executed at the command line of an operating system. These scripts allow you to automate repetitive tasks by executing several commands sequentially or depending on conditions.

**The main characteristics of shell scripts are:**

1. Programming language: Scripting scripts can be written in a variety of programming languages such as Bash, Python, Perl, or others.

2. Command execution: They contain a sequence of commands that are executed automatically.

3. Variables and conditions: Scripts can contain variables, conditions, loops, and other programming constructs to control the flow of execution.

4. Parameterization: Scripts can take arguments from the command line, allowing them to be used with different inputs.

5. Task automation: They allow you to automate repetitive tasks, which increases efficiency and reduces the likelihood of errors.

6. Support for functions: Scripts can contain functions, which simplifies the structure of the code and makes it easier to understand and modify.

7. Information output: Scripts can display information on the screen, which allows you to control and track the execution process.

**4.2 \*How are scripts created and edited, and what do I need to do to run a script?**

To create and edit scripts, you will usually need a text editor such as Notepad (for Windows), Nano, Vim, or Emacs (for Unix-like systems). Here are the steps to create and edit scripts:

1. Start a text editor on your computer.

2. Write the script code using the syntax of the programming language you plan to use.

3. Save the file with a name of your choice and an extension that indicates the programming language. For example, if you are writing a script in Bash, the file name could be myscript.sh.

To run the script:

1. Open a terminal or command prompt on your computer.

2. Use the cd command to change to the directory where your script is stored.

3. Run your script, type its name at the command prompt, and press Enter. For example, for the myscript.sh script, you can run the command ./myscript.sh.

4. On Windows, you may need to run the script through the command line or by double-clicking on the script file (if it is configured to do so).

**4.3 \*\*What are the main components of a motherboard that you know?**

* Processor socket: The socket in which the processor is installed. It allows the processor to communicate with other system components.
* Chipset: This is the set of chips on the motherboard that control the operation of other components, such as the speed of data transfer between the CPU, RAM, and other devices.
* Memory slots (DIMM/RAM): The slots that hold the random access memory (RAM). They provide the processor with access to memory to store temporary information while the system is running.
* Expansion slots (PCIe/PCI/AGP): Slots that allow you to install expansion cards such as graphics cards, sound cards, network adapters, and other devices.
* Connectors (SATA/IDE): Connectors for attaching hard drives, optical drives, and other storage devices to the motherboard.
* BIOS/UEFI chip: This is the software that initializes the system and controls various hardware functions.
* Power connector (ATX/EPS): The connector for connecting power from the power supply to the motherboard.
* External ports (USB/Ethernet/Audio): Connectors for external devices such as a keyboard, mouse, monitor, network cable, audio devices, and more.
* Buttons and LEDs: For example, the power button, BIOS reset buttons, power status LEDs, disk activity LEDs, and others.

**4.4 \*\*Briefly describe for which devices the concepts of MBR and GPT are used?**

**1. MBR (Master Boot Record):**

* Used for BIOS-compatible computers and operating systems.
* Supports volumes up to 2 terabytes.
* Allows up to four primary partitions, or three primary partitions and one extended partition, which can contain multiple logical partitions.
* Widely supported by operating systems such as Windows, Linux, and many others.

**2. GPT (GUID Partition Table):**

* Used for UEFI-compliant computers and operating systems.
* Supports large amounts of data (more than 2 terabytes).
* Allows up to 128 primary partitions.
* Contains more detailed information about partitions and has improved data backup and recovery capabilities.
* More commonly used for modern versions of operating systems such as Windows 10, macOS, Linux, and many others.

**4.5 \*\*What is the essence of the mount operation, why is it needed?**

The essence of the mount operation is that it allows the operating system to access data that is stored on external media or other devices. Since the operating system is organized according to the hierarchical structure of the file system, mounting allows you to connect external devices to this hierarchy, providing them with a path and access point to the data.

5. Prepare an initial version of the report in electronic form:

- Cover sheet, topic and purpose of the work

- Glossary of terms

- Answers to p.4.1 and p.4.5 from the preliminary preparation tasks

**Progress:**

1. Initial work in CLI mode in the Linux OS of the Linux family:

1.1. Start the VirtualBox virtual machine, select CentOS, and start it. Log in to the system as a user: CentOS, login password: reverse (if you are performing the LP in 401) and launch the terminal.

1.2. Start the Ubuntu\_PC virtual machine (if you are performing the PL task through the netacad academy)

1.3. Start your Linux operating system (if you are working on your own PC and have installed it) and launch the terminal.

2. Work through all the sample commands presented in the labs of the NDG Linux Essentials course - Lab 11: Basic Scripting and Lab 12: Understanding Computer Hardware. Create a table to describe these commands.

|  |  |
| --- | --- |
| **Name of the commands** | **Its purpose and functionality** |
| lscpu | is used to display information about the processor (CPU) of the system |
| lspci | is used to display information about all PCI (Peripheral Component Interconnect) devices connected to the system via the PCI bus |
| lsusb | is used to display information about all USB devices connected to the system |
| lsmod | used to display the list of loaded kernel modules |
| fdisk | used to work with hard disk partitions |

3. Create scripts to display text messages to the user (show screenshots):

- \* the script should display a greeting to the current user indicating the current date and information about the current system;

- \*the script should display information about the hardware configuration of the current system (use the commands discussed in Lab 12: Understanding Computer Hardware);

- Provide an example of your script.

**Control questions:**

**1. What is the difference between the arch and lscpu commands?**

The difference is that the arch command provides general information about the operating system architecture, while the lscpu command specifically points to processor details that can be useful for analyzing the system hardware.

**2. What command can I use to get information about the current system's RAM utilization?**

You can get information about the current system's RAM usage status by using the `free` command.

The `free` command displays statistics about the memory usage in the system, including the total amount of available memory, used memory, free memory, swap usage, and other important indicators.

**3. \*How can I use scripts to manipulate variables and create branching and looping scenarios?**

1. Variables:

* Setting the value of a variable: variable=value.
* Using the value of a variable: echo $variable.
* Command line parameters: $1, $2, and so on to get the value of the passed arguments.

2. Branched scripts:

* Conditional statements: if, elif, else.

3. Cyclic scenarios:

* For, while loops.

**4. \*What commands can be used in the terminal to view the connection status of peripheral devices?**

* **lsusb:** This command displays a list of all USB devices connected to your system. It provides information about the manufacturer, model, and other details about each connected USB device.
* **lspci:** This command displays a list of all PCI devices connected to your system. It provides detailed information about graphics cards, network adapters, audio and video adapters, and other PCI devices.
* **lsblk:** This command displays a list of all block devices, such as hard disks and flash drives, that are connected to your system. It also shows information about their partitions.
* **lscpu:** This command displays information about your processor, including the number of cores, model, frequency, and other characteristics.

**5. \*\*What are the features of gparted?**

* **Create, delete, and edit partitions:** GParted allows you to create new partitions on your hard disk, delete existing partitions, and modify their parameters such as size, file system, and type.
* **Formatting partitions:** You can format partitions with different file systems such as ext4, NTFS, FAT32 and others using GParted.
* **Rename partitions**: With GParted, you can rename partitions on your hard disk for convenience.
* **Clone partitions:** GParted allows you to create copies of partitions and clone them, including copying operations between different disks or partitions.
* **Move partitions:** You can move partitions on your hard disk without losing data, expand or shrink them.
* **Check and repair file systems:** GParted has the ability to scan and repair various types of file systems, which can help you identify and fix file system problems.
* **Work with RAID and LVM:** GParted supports partition creation, deletion and editing operations in RAID and LVM configurations.

**Conclusion:** I gained practical skills in working with the Bash shell. I got acquainted with the basic actions when working with scripted scenarios.