“Kyiv specialized College of Communications”

Commission of computer engineering

**REPORT ON THE IMPLEMENTATION**

**LABORATORY WORK №7**

From the discipline: "Operating systems"

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

The students

performed Groups RPZ-03

Team 3:

Kanavets K.S.,

Kryvenko A.I.,

Kulikovska M.V.

Checked by the teacher

Sushanova V.S.

Кyiv 2023

Tasks for preliminary preparation.

**The material was prepared by student Kulikovska Maria (@Smith5004)**

Read the brief theoretical information before the lab and make a small vocabulary of basic English terms related to the purpose of commands and their parameters.

automate repetitive parts - автоматизація повторюваних частин

passing it as an argument - передаючи як аргумент

variables - змінні

conditionals - умови

loops - цикли

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

A shell script is a file containing a series of commands. The shell reads this file and carries out the commands as though they have been entered directly on the command line. The shell is somewhat unique, in that it is both a powerful command line interface to the system and a scripting language interpreter.

*4.2 How are scripts created and edited, and what should be done to run a script?*

**Shell Scripts** are written using text editors. On your Linux system, open a text editor program, open a new file to begin typing a shell script or shell programming, then give the shell permission to execute your shell script and put your script at the location from where the shell can find it.

Create a command line script:

1. **Create a file** **using** a **vi** editor(or any other editor). Name script file with **extension .sh**
2. **Start** the script with **#! /bin/sh**
3. Write some code.
4. Save the script file as filename.sh
5. For **executing** the script type **bash filename.sh**

Linux has several different tools that you can use to edit files. The two most popular are Vi (or Vim) and Nano.

To open a file in Nano, you have to type the command nano , followed by the path to the file you are trying to open. You can also first navigate to the desired directory with the command cd , and then open the file for editing by simply using nano , followed by the file name. For example: nano SampleText.txt. If the specified file already exists, it will be opened for editing.

The default mode in which you enter Vi is command mode , used for navigating and entering commands. Like Nano, Vi uses the arrow keys to navigate. However, unlike Nano, any text entered in Vi will not be treated as a line of text added to the document, but rather as a command sent directly to Vi. To add text to a document, you must first enter insert mode.

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

Computer motherboard parts:

1. CPU socket

2. RAM slots

3. CMOS battery

4. BIOS (Basic Input Output System)

5. Chipset

6. Data connector

7. Power connector

8. Expansion slots

9. Ports /connectors

10. Jumper pins

11. Switches

12. Cache memory

13. CPU clock

14. Heat sink

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

**The master boot record (MBR) and GUID partition table (GPT) are the two partitioning schemes for hard disk drives (HDD) and solid state drives (SSD) everywhere, with GPT being the newer standard.**

The main difference between MBR and GPT is that MBR has some limitations for modern usage. Namely, **MBR can only handle four primary partitions and 2TB of drive space**. **GPT has no partition limit**, so you can have ten partitions if you want and **up to 9400000000 TB of SSD/HDD space**. However, **Windows versions earlier than Vista can’t boot off GPT drives**. Windows 7 and Vista can boot in GPT but with exceptions, such as only on 64-bit systems and by following a specific process. Windows 8 and above work with GPT by default as long as you use a 64-bit system. Another difference is MBR stores all information in one place, which could get corrupted and fail. **GPT writes information in several drive areas and includes a secondary backup GPT Table for recovery** if the first one gets corrupted. Other than the differences between MBR and GPT mentioned above, **GPT can use newer device technologies, and it’s compatible with BIOS/MBR functions** for backward compatibility of older, non-UEFI devices.

*4.5. What is the essence of the mount operation and why is it needed?*

Mounting in Linux allows you to access the contents of a disk and organize the file system structure. Mounting can also be used to open a disk image (such as one created with the dd program) for work, and to access and edit a wide variety of file systems and disk images (such as virtual machine disk images); even remote network directories can be mounted, making them available as if they were files on any other local storage.

**The material was prepared by student Andrew Kryvenko (@AndrewKryvenko)**

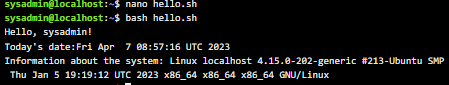
*2. Work through all the sample commands presented in the labs of the NDG Linux Essentials course - Lab 11: Basic Scripting and Lab 12: Understanding Computer Hardware. Create a table to describe these commands\*\*\*.*

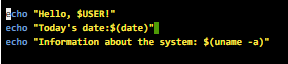
|  |  |
| --- | --- |
| Command name | Its purpose and functionality |
| vi myfile | The command "vi myfile" calls the "vi" text editor to edit the file "myfile". After the "vi" editor starts, you can enter text. |
| :x | Will save and close the file. |
| :wq | Will write to file and quit |
| :wq! | Will write to a read-only file, if possible, and quit. |
| ZZ | Will save and close. Notice that no colon : is used in this case. |
| :q! | Exit without saving changes |
| :e! | Discard changes and reload file |
| :w! | Write to read-only, if possible. |
| vi sample.sh | The command "vi sample.sh" calls the "vi" text editor to edit the file named "sample.sh". After starting the "vi" editor, you can enter or edit the script in the shell command line language. |
| ls -l sample.sh | is used to check the properties of the "sample.sh" file. The output of this command includes information about file permissions, owner, group, and other file parameters. |
| chmod a+x sample.sh | sets the execution permission for all users (owner, group, and others) for the file "sample.sh". This will allow both the owner and other users to execute the file. |
| ls -l sample.sh | repeatedly checks the properties of the "sample.sh" file. The output of this command should show that execution permission has been added to the file. |
| ./sample.sh | is used to execute the "sample.sh" file. The "./" command indicates the current directory where the file is located. If the file "sample.sh" has the correct execution permission, then the file execution should start. If the file contains a script in a command line language, then this script will be executed in accordance with its purpose. |
| lscpu | is used to display information about the processor (CPU) in the system. |
| head -n 20 /proc/cpuinfo | is used to display the first 20 lines of the "/proc/cpuinfo" file, which contains information about the processors in the system that is readable by the Linux kernel. |

**The material was prepared by student Kanavets Kateryna (@kanavetsk)**

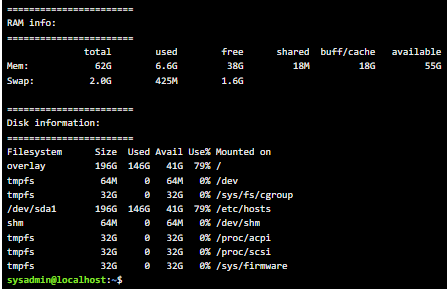
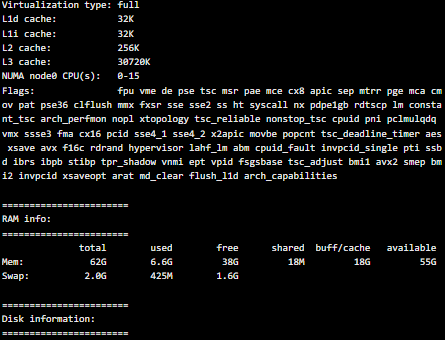
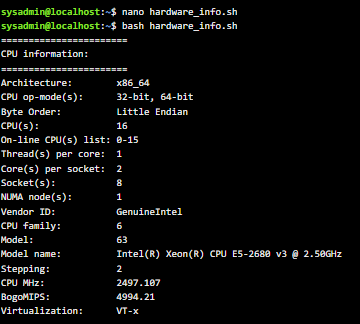
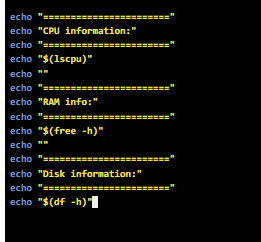
1. *Create scripts with text message output for the user (show screenshots):*

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





- the script should output information about the hardware configuration of the current system (use the commands discussed in Lab 12).



Control questions

**The material was prepared by student Kulikovska Maria (@Smith5004)**

1.How can variables be handled in scripts and how can you create branched and looping scenarios?

Variables store data in the form of characters and numbers. Similarly, shell variables are used to store information, and they can only be used by the shell.

For example, the following creates a shell variable and then prints it:

variable ="Hello"

echo $variable

Below is a small script which will use a variable.

#!/bin/sh

echo "what is your name?"

read name

echo "How do you do, $name?"

read remark

echo "I am $remark too!"

In Linux, you can create branching scripts using conditional statements such as if, elif, and else.

The syntax of the conditional if statement is as follows:

if condition

then

# code that is executed if the condition is true

elif other\_condition

then

# code that is executed if the previous condition was false and this condition is true

else

# code that is executed if none of the previous conditions is true

fi

In Linux, you can create cyclic scripts using for and while loops. The syntax of the for loop is as follows:

for variable in list

do

# code that is executed for each item in the list

Done

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

The arch and lscpu commands are used to get information about the computer's CPU architecture, but they have different purposes and provide different types of information.

lscpu is to display information about the CPU architecture. This command gives you a better idea of the hardware capabilities of your system and may be useful for troubleshooting performance problems or optimizing software for specific hardware configurations.

The arch command in Linux is used to get the current architecture of your computer. This command is often used to determine if a program or binary is compatible with a particular architecture. For example, running the arch command on an x86 architecture machine will display x86\_64 if it has a 64-bit version of the operating system, or i686 if it has a 32-bit version.

In general, the arch command displays the architecture of the machine, while the lscpu command provides more detailed information about the processor architecture and hardware specifications.

3. What command can I use to get information about the current system's RAM usage?

The cat command is used to display information about Linux memory.

Entering cat /proc/meminfo into your terminal opens the file /proc/meminfo .

This is a virtual file that reports the amount of memory available and in use. It contains real-time information about the use of system memory, as well as the buffers and shared memory used by the kernel. The output may vary slightly depending on the architecture and operating system in question.

4. What commands can be used to view the connection status of peripherals in the terminal?

Commands:

lsusb can be used to list all connected USB devices on Linux.

dmesg to get more information about connected USB devices. The easiest way to find the last connected USB device is to use the dmesg command.

usb-devices command to get a list of all the attached USB devices on your system.

lsblk or fdisk to list all block USB drives.

5. What are the possibilities of gparted?

GParted is a free partition editor that is designed to effectively and securely manage hard disk partitions (create, delete, move, copy, resize, etc.) without losing data. It also allows you to create partition tables (MS-DOS or GPT), enable, disable and change attributes, shred partitions, recover data from damaged partitions and much more.

GParted supports various storage devices including SATA, IDE and SCSI hard drives, Flash and SSD drives, RAID systems and more. It also handles all sector sizes and the most popular file systems ext2, ext3, ext4, FAT16, FAT32, HFS, HFS+, NTFS, UFS and others.

Conclusion

In this lab, we learned how to create scripted scenarios and created scripted scenarios that display text messages to the user.