“Київський фаховий коледж зв’язку”

Циклова комісія Комп’ютерної інженерії

**ЗВІТ ПО ВИКОНАННЮ**

**ЛАБОРАТОРНОЇ РОБОТИ №7**

з дисципліни: «Операційні системи»

**Тема: «** **Створення скриптових сценаріїв та визначення апаратної конфігурації системи»**

Виконала студентка

групи РПЗ-13б

Дімітрова С.П.

Перевірив викладач

Сушанова В.С.

Київ 2024

**Мета роботи:**

1. Отримання практичних навиків роботи з командною оболонкою Bash.
2. Знайомство знайомство з базовими діями при роботі зі скриптовими сценаріями.

**Матеріальне забезпечення занять:**

1. ЕОМ типу IBM PC.

2. ОС сімейства Windows та віртуальна машина Virtual Box (Oracle).

3. ОС GNU/Linux (будь-який дистрибутив).

4. Сайт мережевої академії Cisco netacad.com та його онлайн курси по Linux

**Завдання для попередньої підготовки:**

1. \*Прочитайте короткі теоретичні відомості до лабораторної роботи та зробіть невеликий словник базових англійських термінів з питань призначення команд та їх параметрів.

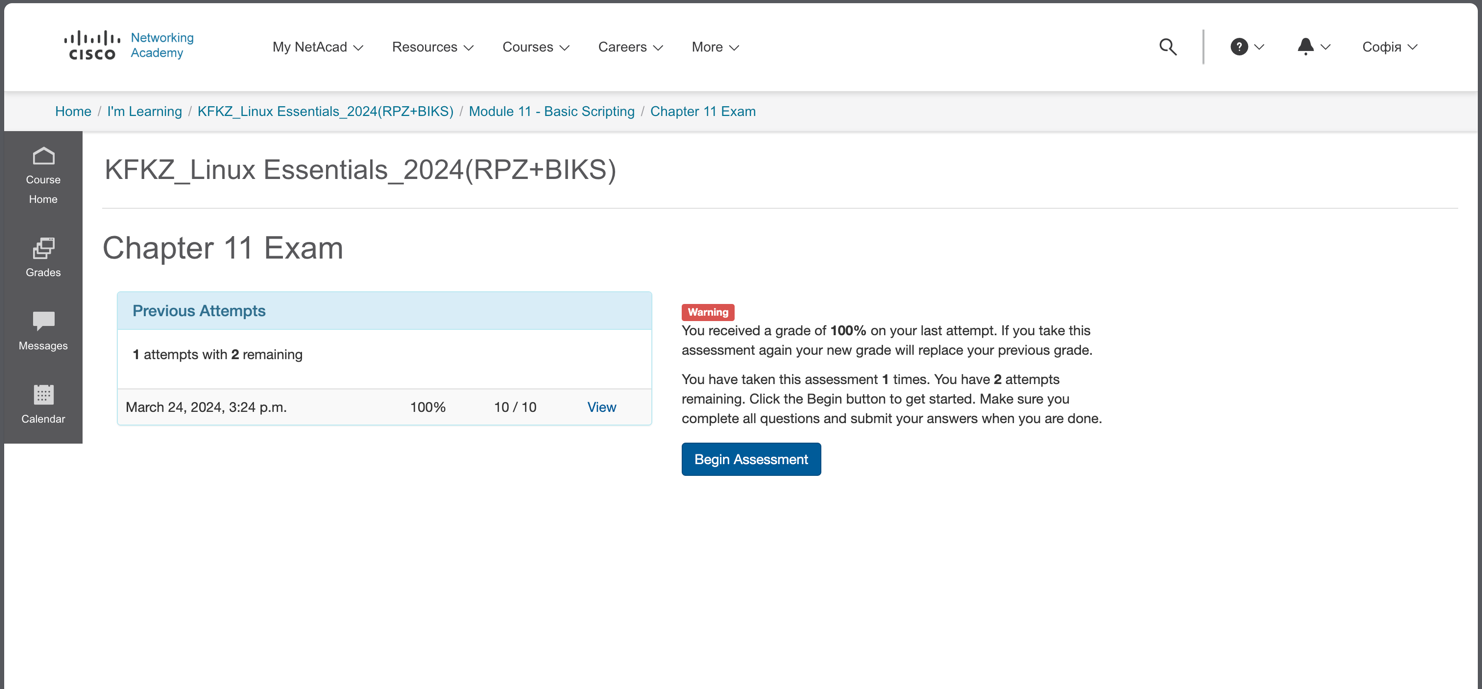
|  |  |
| --- | --- |
| Термін англійською | Термін українською |
| Shell script | Скрипт оболонки |
| Script editor | Редактор сценаріїв |
| Repetitive | Повторюваний |
| Consistency | Послідовність |
| Execute | Виконати |
| Loop | Цикл |
| Shebang | Шебанг (#!) |

1. Вивчіть матеріали онлайн-курсу академії Cisco “NDG Linux Essentials”:

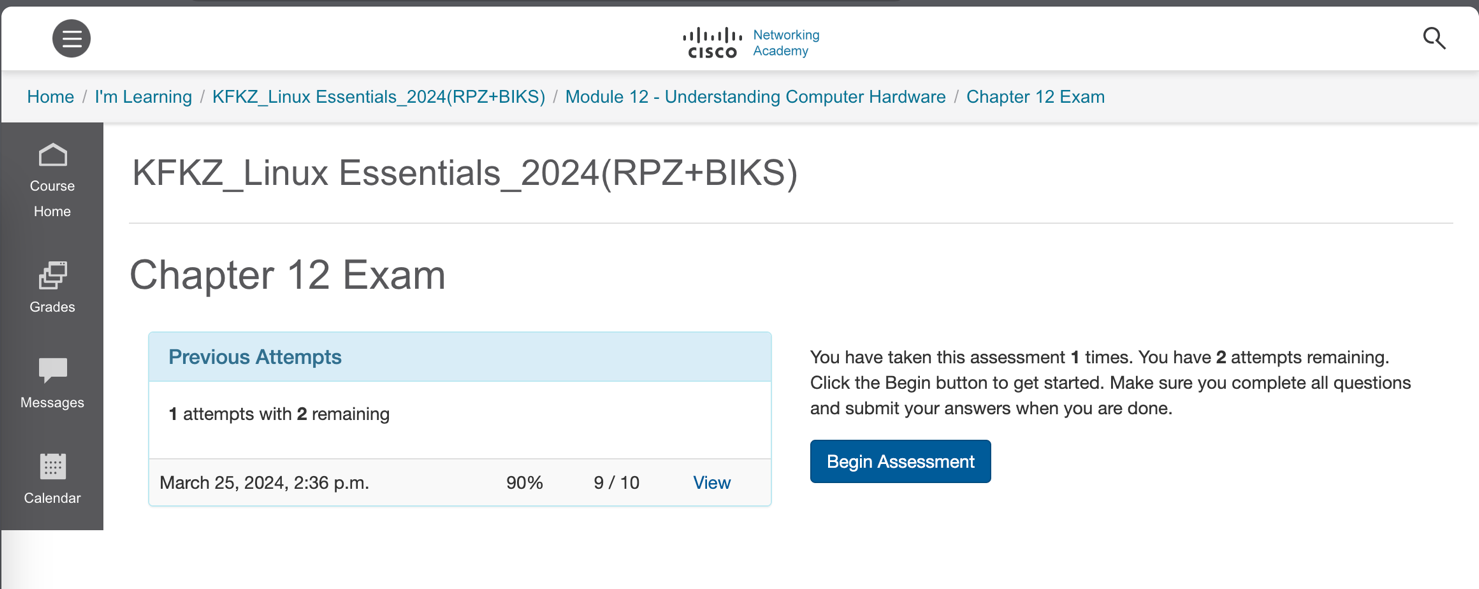
* Chapter 11 - Basic Scripting
* Chapter 12 - Understanding Computer Hardware

1. Пройдіть тестування у курсі NDG Linux Essentials за такими темами:

* Chapter 11 Exam



* Chapter 12 Exam



1. На базі розглянутого матеріалу дайте відповіді на наступні питання:
   1. \*Охарактеризуйте поняття скриптового сценарію у командній оболонці.

A shell script is a file that contains a set of commands that are executed one after the other when the script is run. Shell scripts have access to all shell commands, including logic. Thus, a script can check for a file or look for a certain output and change its behaviour accordingly. Scripts automate repetitive tasks, improve consistency, and can perform complex operations.

* 1. \*Яким чином створюються та редагуються скрипти, що треба зробити щоб запустити скрипт?

Scripts are created using a text editor like nano or vim. Save the script with a .sh extension. For example, to create a new script using the Nano editor, simply type *nano myscript.sh.* This will open the Nano text editor and create a new file called "myscript.sh". Next, enter the required code for your script in this file.

There are two ways to run a script:

* + Directly from the shell: ./myscript.sh. This requires the script to have execute permissions. You can set them using chmod +x myscript.sh.
  + As an argument to the shell: sh myscript.sh (replace sh with your shell if different, e.g., bash script.sh).
  1. \*\*Які основні компоненти материнської плати ви знаєте?

The main components of a motherboard are the central processing unit (CPU) socket, memory slots (RAM), expansion slots, chipset, input/output (I/O) ports, power connectors, and battery.

* 1. \*\*Коротко охарактеризуйте для яких пристроїв оперують поняттями MBR та GPT?

The concepts of MBR (Master Boot Record) and GPT (GUID Partition Table) are used for partitioning and managing disk space on storage devices such as hard disk drives (HDDs), solid-state drives (SSDs), USB flash drives, and other storage media. They determine how the disk is organized into partitions, allowing users to store data, install operating systems, and manage storage efficiently.

MBR is used for older systems. It has a limit of 4 partitions and does not support drives larger than 2 TB. GPT is a modern format that allows for more partitions, supports larger drives, and provides reliability.

* 1. \*\*В чому суть операції монтування, для чого вона потрібна?

The essence of the mounting operation in operating systems like Linux is to integrate a filesystem, stored on a storage device (such as a hard drive or USB flash drive), into the directory tree of the operating system. Once mounted, the filesystem becomes accessible through a designated path within the system. Mounting is necessary to access the data stored on the storage device. It allows the operating system to interact with the files and directories within the filesystem, enabling users to read, write, and manipulate data. Without mounting, the data on the storage device would remain inaccessible to the operating system and its users.

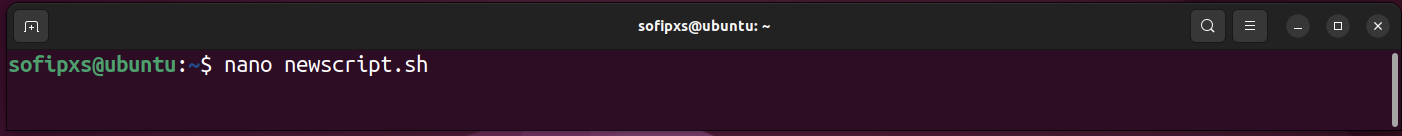
**Хід роботи:**

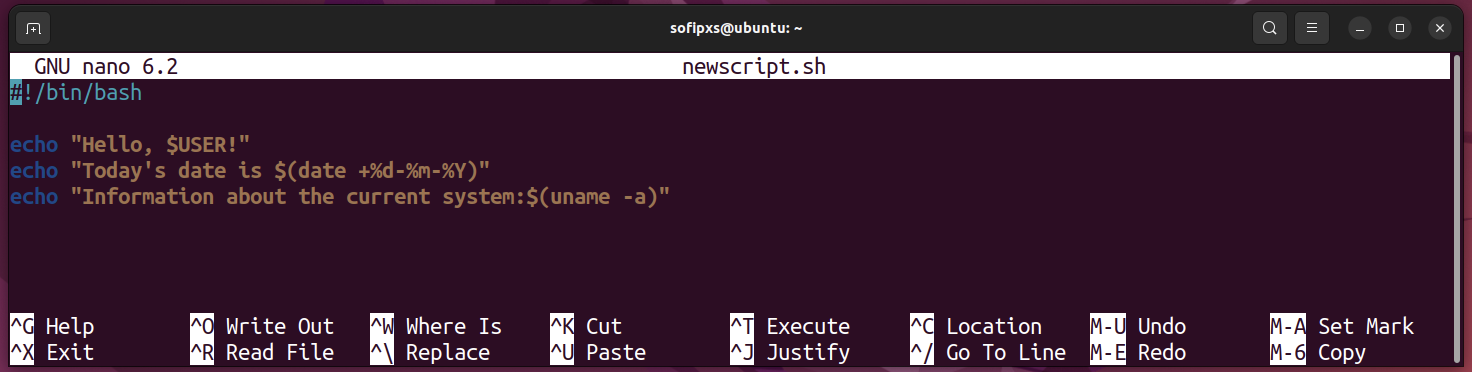
1. Початкова робота в CLI-режимі в Linux ОС сімейства Linux:
   1. Запустіть віртуальну машину VirtualBox, оберіть CentOS та запустіть її. Виконайте вхід в систему під користувачем: CentOS, пароль для входу: reverse ***(якщо виконуєте ЛР у 401 ауд.)*** та запустіть термінал.
   2. Запустіть віртуальну машину Ubuntu\_PC ***(якщо виконуєте завдання ЛР через академію netacad)***
   3. Запустіть свою операційну систему сімейства Linux ***(якщо працюєте на власному ПК та її встановили)*** та запустіть термінал.
2. Опрацюйте всі приклади команд, що представлені у лабораторних роботах курсу ***NDG Linux Essentials - Lab 11: Basic Scripting*** та ***Lab 12: Understanding Computer Hardware.*** Створіть таблицю для опису цих команд.

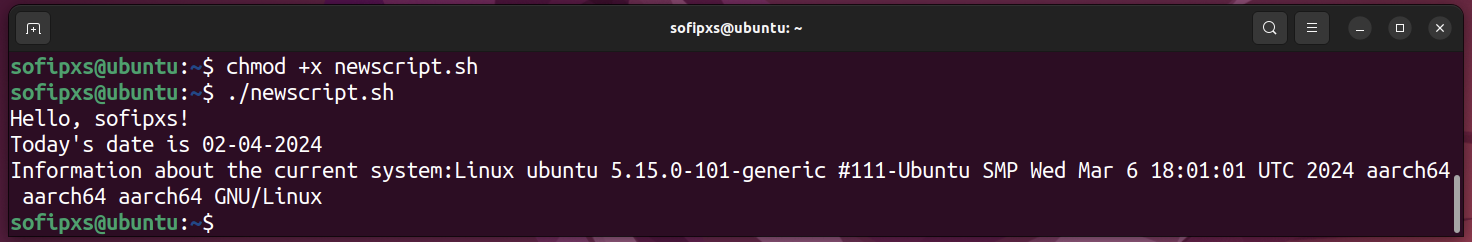
|  |  |
| --- | --- |
| Назва команди | Її призначення та функціональність |
| Lab 11  Basic Scripting | |
| vi myfile | Сreate a new file. |
| wq | Write the file to disk and quit. |
| j | Moves cursor down one line (same as down arrow) |
| k | Moves cursor up line (same as up arrow) |
| l | Moves cursor to the right one character (same as right arrow) |
| h | Moves cursor to the left one character (same as left arrow). |
| w | Moves cursor to beginning of next word. |
| e | Moves cursor to end of word. |
| b | Moves cursor to beginning of previous word. |
| $ | Moves cursor to end of current line (same as End key). |
| 0 (zero) | Moves cursor beginning of current line (same as Home key). |
| 3G | Jumps to third line (nG jumps to the nth line). |
| 1G | Jumps to first line. |
| Shift + G | Jumps to the last line. |
| dw | Delete the word |
| u | Undo the last operation. |
| 2dw | Delete two words. |
| xxxx | Delete four characters, one at a time. |
| 4u | Undo the last 4 operations and recover the deleted characters. |
| 14x | Delete 14 characters. |
| 5x | Delete the five characters to the left of the cursor (type 5 then Shift+x). |
| dd | Delete the current line. |
| p | Whatever was lasted deleted or yanked can be “pasted”. Paste the deleted lines below the current line. |
| 2u | Undo the last two operations |
| 2dd | Delete two lines, the current and the next. |
| 4w  D | Move to the fourth word then delete from the current position to the end of the line Shift+D. |
| J | Join two lines, the current and the next by typing a capital J (Shift+J). |
| yw | Copy (or “yank”) the current word. |
| P | Paste (or “put”) the copied word before the current cursor by typing Shift+p. |
| 1G  3J | Move to the first line, then join three lines. |
| :%s/text //g | Search for and delete the word text (add a space after the word text). |
| i | Enter insert mode |
| ~ | Shift+` changes letter to lower case |
| :w | Save the file. |
| 10l | 10 followed by the lowercase ‘L’ |
| a | Enter insert mode. |
| o | Open a blank line below the current line. |
| O | Open a blank line above the current line by pressing uppercase O. |
| :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. |
| /line | Search forward for the word line. The cursor moves to the beginning of the first instance of the word line. |
| n | Search for the next instance of the word line. |
| ?line | Search backward for the word line. Cursor moves to the beginning of the previous instance of the word line. |
| cw  entry | You will replace the word line with the word entry. When you press cw you will be in insert mode and you will be able to type over the word line. |
| bash sample.sh | Run bash script. |
| chmod a+x | Give some new rules to script. |
| Lab12 Understanding Computer Hardware | |
| lscpu | In order to determine the type of CPU execute the lscpu command. |
| head -n 20 /proc/cpuinfo | Use the head command with the -n option to list the first 20 lines of the cpuinfo file. |
| head -n 20 /proc/cpuinfo | Use the head command with the -n option to list the first 20 lines of the cpuinfo file. |
| free -m  free -g | To discover how much RAM and swap space is being used, use the free command. The output shows the amount of memory in megabytes when the -m option is used and in gigabytes when the -g option is used: |
| lspci | To see what devices are connected to the PCI bus, use the lspci command. |
| lspci -k | lspci command with the -k option shows devices along with the kernel driver and modules used |
| lsusb | Attempt to list the USB connected devices. |
| lsmod | For hardware to function, the Linux kernel usually loads a driver or module. Use the lsmod command to view the currently loaded modules. |
| fdisk | The fdisk command is useful for identifying and manipulating disk storage resources on a system. Without the -l option, the fdisk command enters an interactive mode that is typically used to modify partitions on a disk device. |
| fdisk -l | The -l option lists the partition tables for the specified devices and then exits. |

1. Створіть скриптові сценарії з виводом текстових повідомлень для користувача (продемонструйте скріншоти):

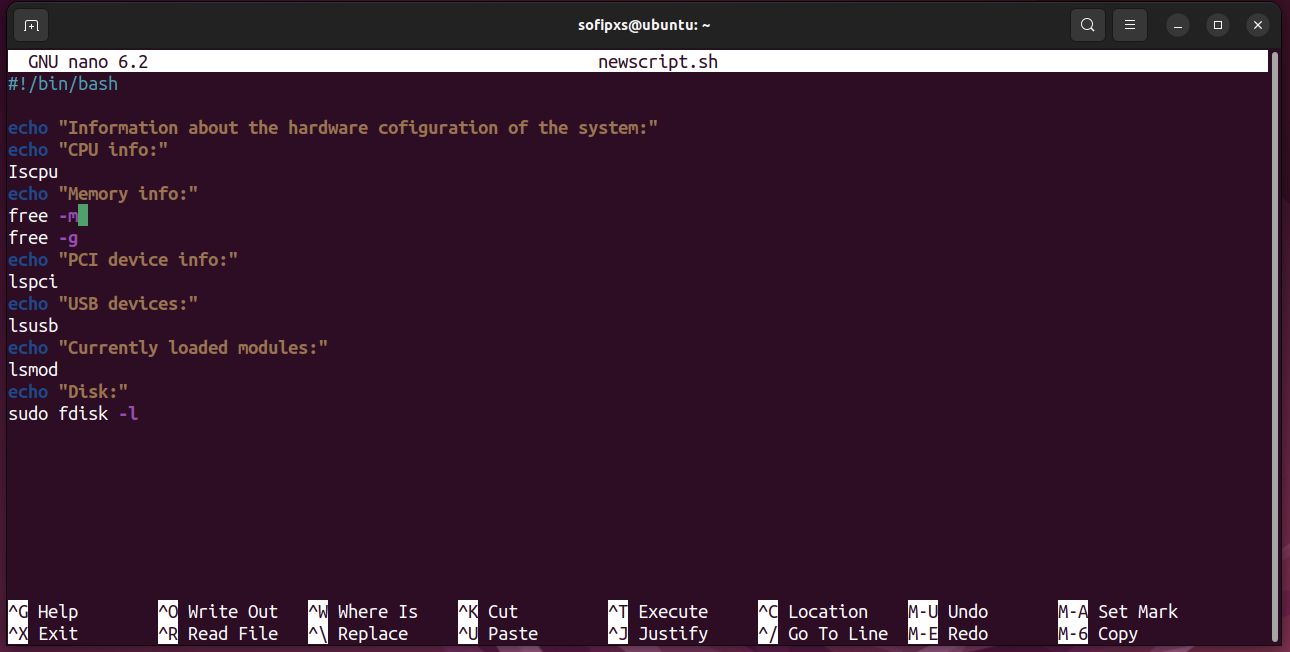
* сценарій має виводити привітання до поточного користувача вказуючи поточну дату та інформацію про поточну систему;

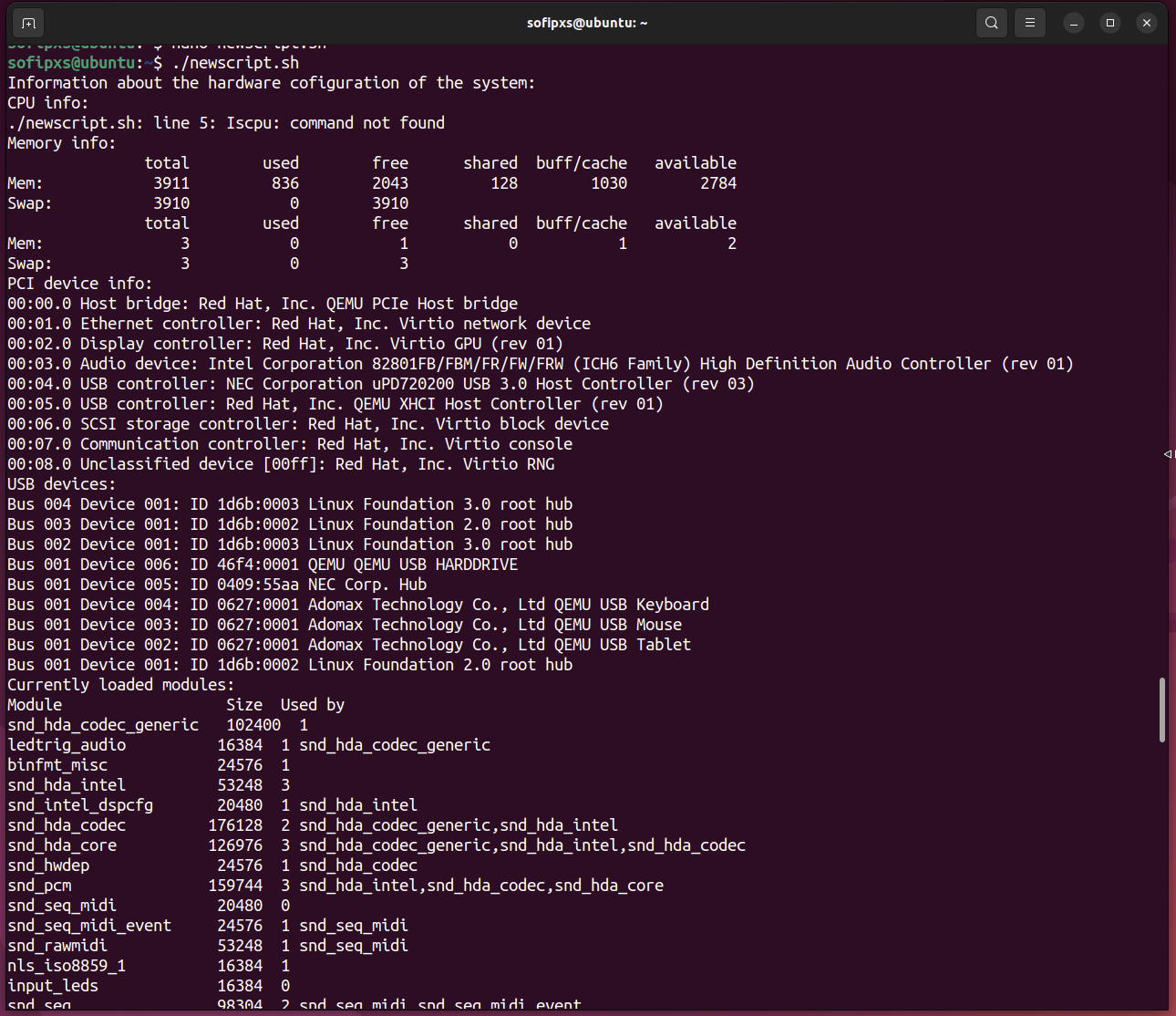




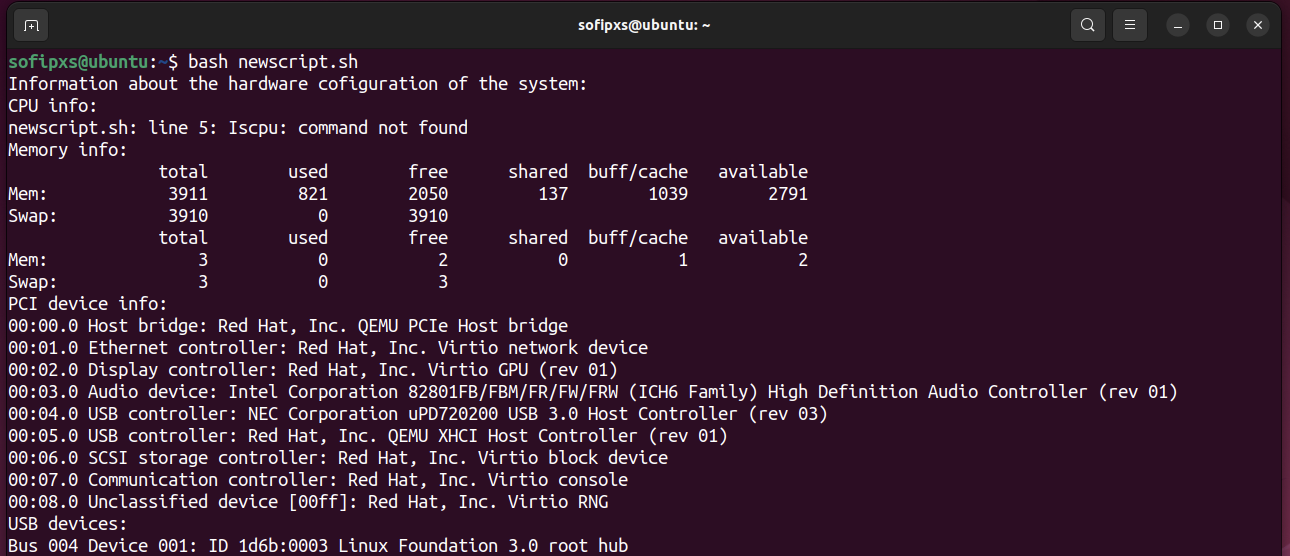


* \*сценарій має виводити інформацію про апаратну конфігурацію поточної системи (використовуйте команди розглянуті в Lab 12: Understanding Computer Hardware);

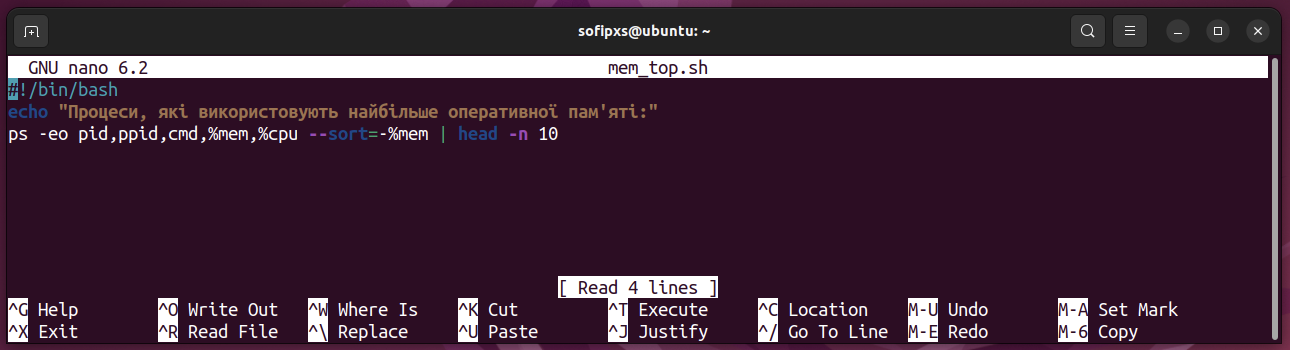


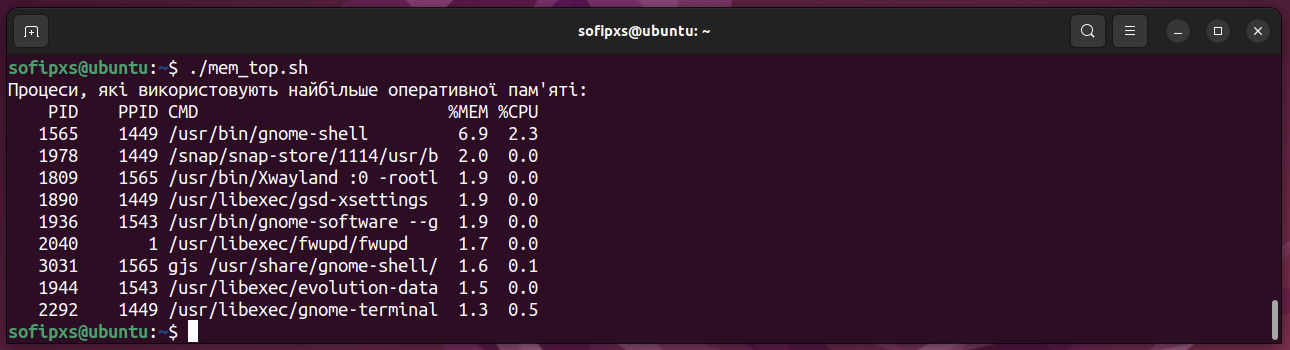


Another way to run the program is to type bash before the filename:

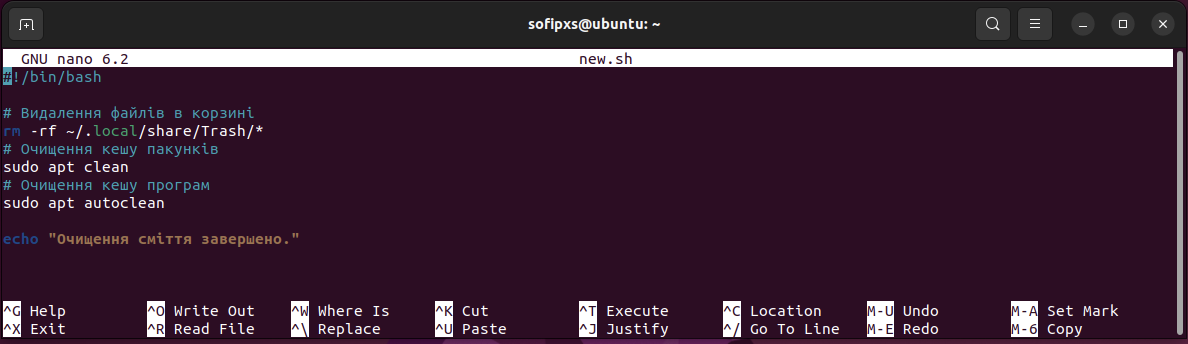


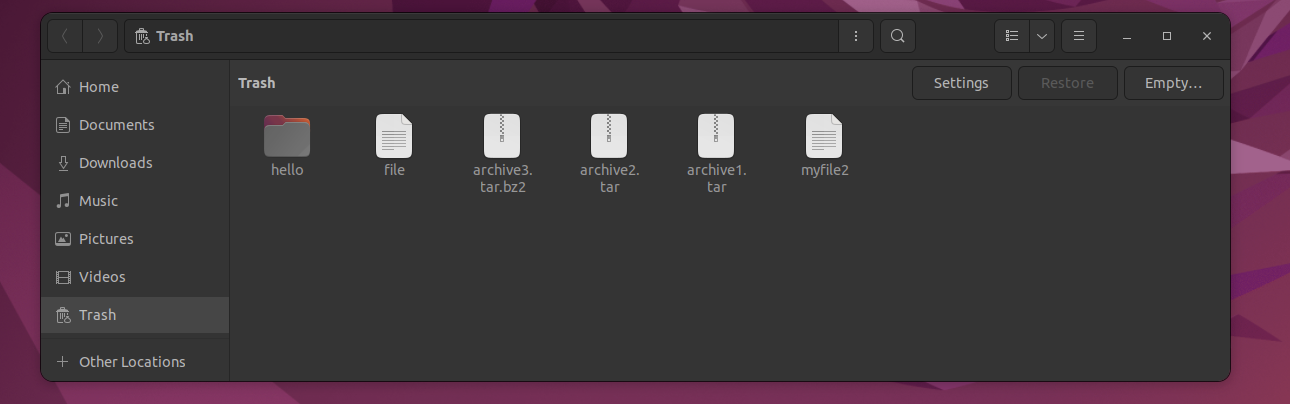
* \*\*наведіть свій приклад скриптового сценарію.

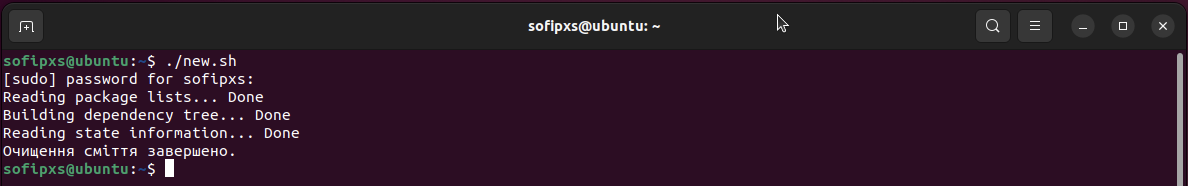


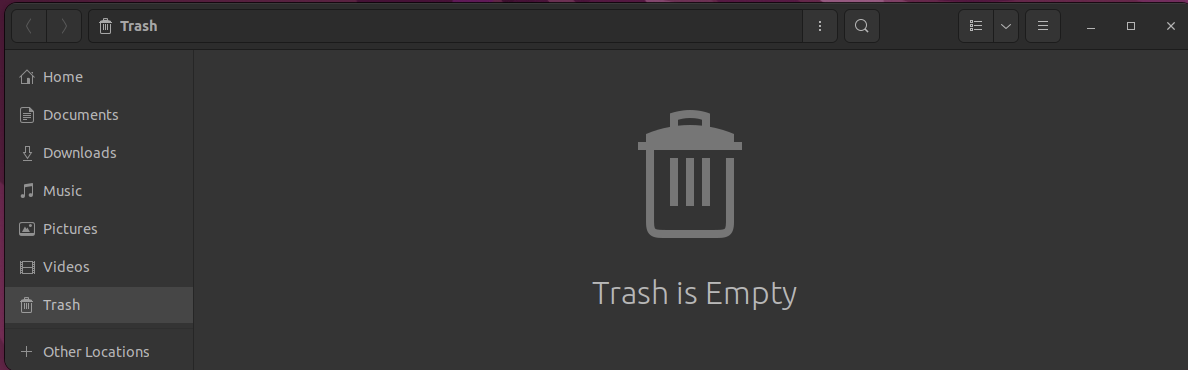


The second script, which deletes files from the Recycle Bin, clears the package cache and also clears the application cache.





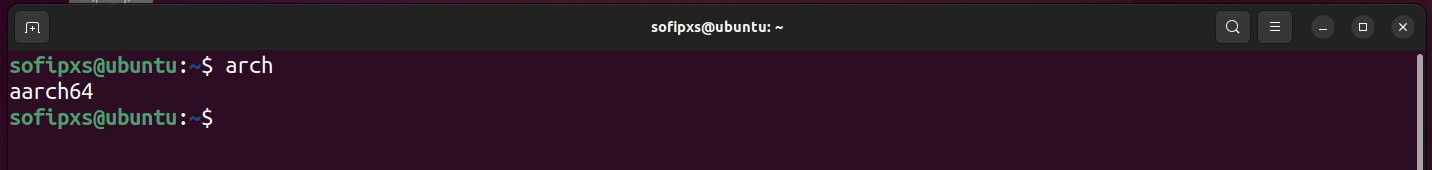




**Відповіді на контрольні запитання:**

1. В чому відмінність між командами arch та lscpu?

* The arch command in Linux is used to print the computer’s architecture. It’s a simple command with no options or arguments.

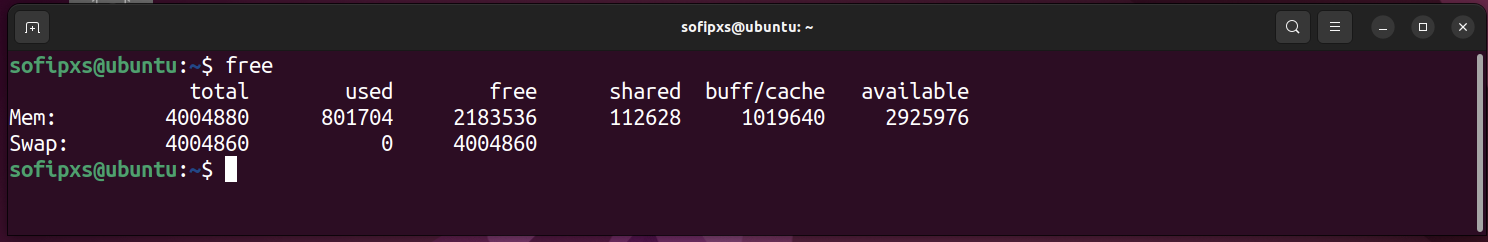


* The lscpu command, on the other hand, provides a detailed list of information about the CPUs in the system. In this case, the output details the CPU architecture, cores, threads, CPU family, model, etc.



1. Якою командою можна отримати інформацію про стан використання RAM поточною системою?

The free command provides information on total, used, free, shared, buffer/cache, and available RAM and swap space at the moment you run the command:



1. \*Яким чином у скриптах можна опрацьовувати змінні та створювати розгалужені та циклічні сценарії?

In command-line scripts such as Bash, variables are manipulated by assigning values and using them in commands and expressions. To create branched scripts, conditional statements such as *if, elif,* and *else* are used to perform different actions depending on the conditions. Loop scripts are created using loops, such as *for, while*, which allow you to execute a certain block of code several times. Such constructs allow you to automate and control the execution of scripts, ensuring their flexibility and functionality.

Simple use of variables:

#!/bin/bash

ANIMAL="penguin"

echo "My favorite animal is a $ANIMAL"

A basic if statement looks like this:

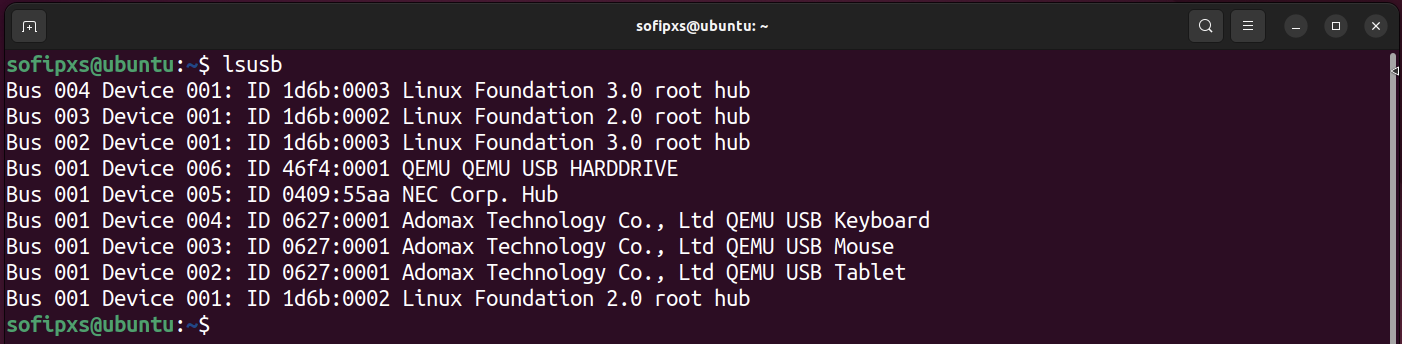
if somecommand; then

  # do this if somecommand has an exit code of 0

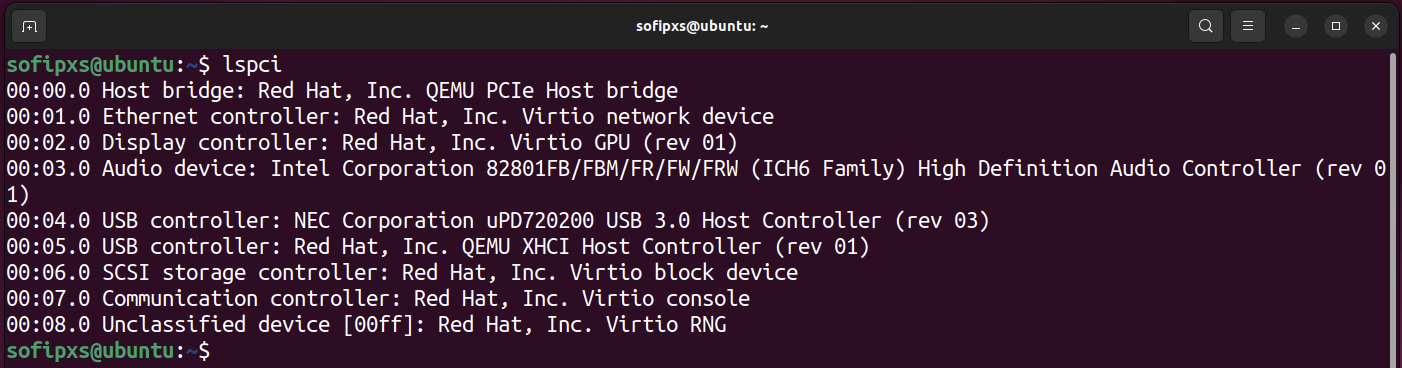
fi

1. \*Які команди для перегляду стану підключення периферійних пристроїв можна використати в терміналі?

* lsusb: This command displays a list of all USB devices connected to the computer along with their IDs.



* lspci: This command displays information about all PCI devices that are connected to the system.



* lsblk: Displays a list of all block devices, such as hard drives and USB drives, along with information about their size and mount points.
* lshw: This command provides detailed information about all hardware components of the system, including peripherals.

1. \*\*Які можливості застунку gparted?

The gparted program is a GNOME partition editor designed to create, rearrange, and remove partitions on a disc. A disc device can be partitioned into one or more partitions. The gparted application allows you to reorder the partitions on a disc device, while preserving the data stored on the partitions. It supports many file system types including ext2, ext3, ext4, fat16, fat32, ntfs, and others. With gparted, you can perform the following tasks:

* Create a partition table on a disc device.
* Enable or disable partition flags, including the boot and hide flags.
* Perform actions on partitions, including creating, deleting, resizing, moving, verifying, labelling, copying, and pasting partitions.

**Висновки:**

During the laboratory work, I got acquainted with the basic concepts of working with the Bash command shell and writing scripts in the GNU/Linux environment. I gained practical skills in creating and editing scripts. The main components of the motherboard, the essence of the mount operation, and ways to obtain information about the system's hardware configuration were discussed.