“Київський фаховий коледж зв’язку”

Циклова комісія Комп’ютерної та програмної інженерії

**ЗВІТ ПО ВИКОНАННЮ**

**ЛАБОРАТОРНОЇ РОБОТИ №7**

з дисципліни: «Операційні системи»

**Тема:** **“Створення скриптових сценаріїв та визначення апаратної конфігурації системи”**

Виконали студенти

групи КСМ-13а

MathematicainsLivesMatter: Дзизиль Д. Є., Чех І. В.

Перевірив викладач

Сушанова В. С.

Київ 2023

**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 (Windows 7).

3. Віртуальна машина – Virtual Box (Oracle).

4. Операційна система GNU/Linux – CentOS.

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

**Завдання для попередньої підготовки**

***Готував матеріал студент Дзизиль Д.Є.***

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 командного рядка |
| Shebang | Шебанг |
| Interpreter | Інтерпретатор |
| Loop | Цикл |
| Copy Buffer | Буфер обміну |
| Binary Search Path | Шлях бінарного пошуку |
| chmod | Зміна прав доступу |

***Готував матеріал студент Дзизиль Д.Є.***

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

A shell script is a simple text file containing a set of commands to be executed in the operating system shell. Scripting allows you to automate various tasks using a scripting language such as Bash, sh, or others.

The key features of scripting are:

* Command execution: You can list any command supported by the shell in the script, one after the other.
* Access to functional programs: Scripts have access to all commands and possible command combinations, including conditional statements, loops, and changes.
* Task automation: Scripts are used to automate repetitive or routine tasks, which saves a lot of time for the user.
* Use of variables: Scripts can use variables to store and manipulate data.
* Conditional statements: Scripts can make decisions based on conditions and perform actions accordingly.
* Loops: Scripts can perform the same actions multiple times using loops.
* Parameters and arguments: Scripts can use parameters and arguments to modify behavior at runtime.

Scripting is widely used to automate system administration, file processing, software deployment, and many other tasks in the operating system shell. It demonstrates convenience and performance in working with Linux and other systems that support command-line programming.

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

Creating and editing scripts:

Choose a Text Editor (Notepad, Sublime Text or Visual Studio Code), then open it and write your script, which is a set of instructions written in the selected programming language. Save the script with the appropriate file extension for the language you're using (Python — "py", JavaScript — ".js"). Open your command prompt or terminal, navigate to the directory where your script is saved, and run it by typing the appropriate command (Python — "python script.py").

It's essential to test and debug your script to ensure that it works correctly. Programming often involves writing, testing, debugging, and refining your script until it achieves the desired functionality.

* 1. Які основні компоненти материнської плати ви знаєте?

The main components of a motherboard: Central Processing Unit (CPU), Chipset, Expansion Slots, Memory Slots, Input/Output Ports, BIOS/UEFI, CMOS Battery, Power Connectors, Storage Device Connectors, Cooling Components, SATA Ports, Network Interface Controller,Audio Components.

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

MBR (Master Boot Record) and GPT (GUID Partition Table) are two different partition structures used on storage devices like hard disk drives (HDDs) and solid-state drives (SSDs) to organize and manage partitions and load operating system partitions.

MBR is commonly used on older systems and devices with limited storage. It limits the number of partitions to four and supports partition sizes up to 2 TB. However, the limitation of MBR in size and its individual partitions makes it less practical for modern large drives.

On the other hand, GPT has become more prevalent on modern systems, especially computers with large hard drives or SSDs. It supports a larger number of partitions (up to 128 per disk) and allows you to create large partitions (over 2 TB). In addition, it provides better resistance to data corruption and supports modern features such as UEFI for booting the OS.

Therefore, GPT is a more modern and advanced partition system and is suitable for modern computers and servers.

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

Mounting or "mounting" is an essential operation used in operating systems to make file systems or storage devices available and usable.

The mounting process is important for several reasons:

* Access to Stored Data (It allows users and programs to access files and data stored on various media such as hard drives, flash drives, CD/DVDs, among others).
* Isolation of File Systems (Mounting enables users to isolate different file systems and avoid conflicts between them).
* Availability of External Devices (Users can connect and disconnect external devices like USB drives or CD/DVD drives, mount them to access data, and then unmount them when they are done).
* Network Access (In network operating systems, mounts are used to connect to network resources accessible over the network, such as files on other computers).

**Хід роботи**

***Готував матеріал студент Чех І.В.***

1. Початкова робота в CLI-режимі в Linux ОС сімейства Linux:

1.1. 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.

1.2. Запустіть віртуальну машину Ubuntu\_PC (якщо виконуєте завдання ЛР через академію netacad)

1.3. 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 | |
| vi | opens the Vi text editor in the terminal for text editing. Use various key combinations to edit and save files. |
| 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 |
| dw | In the Vi or Vim text editor deletes the word at the cursor position. |
| Lab 12 | |
| lscpu | command provides information about your computer's CPU in a Linux or Unix environment. It includes details such as the number of cores, CPU model, and other characteristics. |
| free -m | command in a Linux or Unix terminal shows memory usage in megabytes, including total, used, free, shared, cached, and buffer memory. |
| lspci | command lists PCI devices in a Linux or Unix system, helping you identify hardware components. |
| lsusb | lists USB devices in a Linux or Unix system, helping you identify connected hardware via USB. |

1. Робота в в терміналі

***Готував матеріал студент Чех І.В.***

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)

**Зображення, що містить текст, Шрифт, знімок екрана

Автоматично згенерований опис**

Open the Nano text editor and create a file called MLM.sh.

Зображення, що містить текст, знімок екрана, монітор, програмне забезпечення

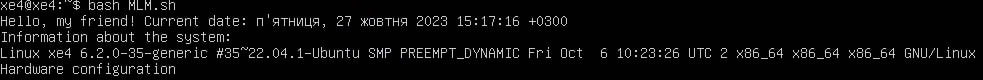
Автоматично згенерований опис

The interface of the Nano editor, in which we enter the necessary commands. Then exit and save.

Зображення, що містить текст, знімок екрана, Шрифт, меню

Автоматично згенерований опис

We try to open the script, nothing comes out, as we can see, it does not have permission. Then we open it through the bash command.



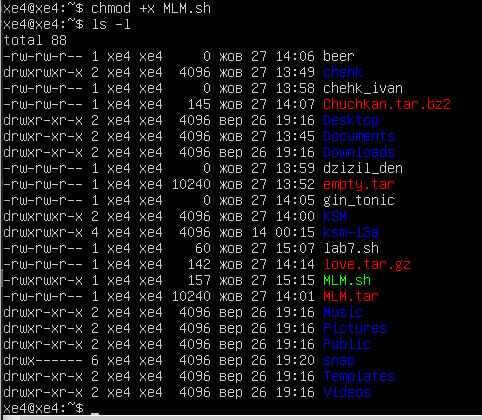
Зображення, що містить текст, знімок екрана

Автоматично згенерований опис

Зображення, що містить текст, знімок екрана

Автоматично згенерований опис

The script is shown to us.



Then we can add permission with the command chmod +x MLM.sh. Let's see if the permission has been added.

And we can run the script again but with a different command.

Зображення, що містить текст, знімок екрана, меню

Автоматично згенерований опис

Result

**Відповіді на контрольні запитання**

***Готував матеріал студент Дзизиль Д.Є.***

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

Manipulating variables and creating branching and looping scenarios are important aspects of programming in many languages. Here's a general overview of how these tasks can be accomplished:

* Declaring variables: To store data in your script, you can create variables by using keywords such as `var`, `let`, or `const`. Once declared, you can assign values to them and use them in your code.
* Branching (if/else): By using the `if/else` construct, you can create branching scenarios in your code. You can check a condition and execute the corresponding block of code based on the result.
* Loops (for/while): Loops allow you to perform certain operations multiple times, creating looping scripts. You can use loops such as `for` and `while` to accomplish this.

It's important to keep in mind that the specific syntax for variables, branching, and loops may differ depending on the programming language you're using. To learn more about these concepts, it's recommended to refer to the documentation and examples specific to your chosen language.

***Готував матеріал студент Дзизиль Д.Є.***

1. В чому відмінність між командами arch та lscpu?

The arch and lscpu commands are two different Linux commands that display information about the processor architecture and operating system. Although both commands provide information about the computer's hardware, they differ in the type of information they provide and their intended purposes.

The arch command provides information about the architecture of the operating system, such as whether it is 32-bit or 64-bit, ARM, MIPS, SPARC, or x86. When you execute the arch command, it will display the architecture of the operating system that is running on your computer.

On the other hand, the lscpu command provides more detailed information about the processor, including the processor model, the number of cores and threads, clock speed, supported extensions, and other processor characteristics. By running the lscpu command, you will get a lot of information about the processor, such as its model name, number of cores, architecture (e.g., x86\_64), and more.

***Готував матеріал студент Дзизиль Д.Є.***

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

To get information about the state of RAM usage, you can use the free command. This command displays various RAM statistics, including the amount of available, free, used, and other memory characteristics.

***Готував матеріал студент Дзизиль Д.Є.***

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

Checking the status of connected peripherals in a Linux terminal system can be easily done using various commands:

* lsusb: This command lists all USB devices that are currently connected to your computer. It provides information about the manufacturer, model, and other characteristics of the USB devices.
* lspci: This command lists all PCI devices that are installed in your computer. It provides information about graphics cards, sound cards, network adapters, and other PCI devices.
* lsblk: This command lists block devices, such as hard drives and SSDs, connected to your system. It provides information about the devices, their partitions, and their mount points.
* hwinfo: This command also provides information about your computer’s hardware, including detailed information about peripheral devices.

***Готував матеріал студент Дзизиль Д.Є.***

1. Які можливості застунку gparted?

GParted, also known as GNOME Partition Editor, is a powerful graphical tool designed to manage disks and partitions in Linux-based operating systems. It offers a wide range of features and capabilities, including creating, deleting, and modifying partitions. With GParted, you can change the sizes and types of partitions, format partitions with different file systems like ext4, NTFS, FAT32, and more. You can also rename partitions, change their location and order, set or remove partition flags, and check and fix file systems such as ext2, ext3, ext4, NTFS and many others.

Moreover, GParted allows you to expand or shrink partitions, perform operations on multiple partitions simultaneously, work with multiple physical disks and drives, and extended LVM (Logical Volume Manager) partitions. Additionally, you can save various operations in a queue for later execution.

In summary, GParted is an essential tool for managing disks and partitions in Linux-based operating systems. Its numerous features and capabilities make it flexible and easy to use.

**Висновки:**

***Готував матеріал студент Чех.І.В***

***На цій лабораторній ми отримали практичні навички з з командною оболонкою Bash. Та ознайомились/навчились писати базові скрипти.***