

“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»**

Виконав(ла/ли) студент(ка/и)
групи **РПЗ-03**
Кошіль Владислав та
Фещенко Євгеній.
Перевірив викладач
Сушанова В.С.

Київ 2022

Мета роботи:

1. Отримання практичних навиків роботи з командною оболонкою Bash.
2. 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 **Фещенко Євгеній**.

Прочитайте короткі теоретичні відомості до лабораторної роботи та зробіть невеликий словник базових англійських термінів з питань класифікації віртуальних середовищ.

Термін англійською	Термін українською
Hash	Хеш
Directory	Каталог
Invoke	Викликати
Prompt	Запит

Дайте відповіді на наступні питання:

Готував матеріал студент **Кошіль Владислав**.

1. Охарактеризуйте поняття скриптового сценарію у командній оболонці.

In a command shell, a scripting script is a series of commands and instructions written in a scripting language that can be executed one after the other in a batch. The scripting language used depends on the command shell being used, but some common scripting languages are Bash, PowerShell, and Python.

The scripting script can be used to automate repetitive tasks, perform system maintenance, or run complex commands that require multiple steps. For example, a script could be written to copy files from one directory to another, rename files, delete files, or create directories.

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

To create and edit scripts, you will need a text editor or integrated development environment (IDE) that supports the scripting language you plan to use. Some editors have built-in support for scripting languages, while others may require plugins or extensions to enable support.

Once you have your text editor or IDE set up, you can start creating and editing scripts by writing code in the scripting language of your choice. You can save your scripts as plain text files with the appropriate file extension for the scripting language you are using.

To run a script, you need to open a command shell and navigate to the directory where the script is saved. Then, you can execute the script by typing its filename followed by any necessary command-line arguments and pressing enter. The command shell will interpret and execute the commands in the script, producing output as it goes.

3. Які основні компоненти материнської плати ви знаєте?

The main components of a motherboard include:

- CPU Socket
- RAM Slots
- Chipset
- BIOS
- Expansion Slots:
- SATA/IDE Connectors
- Power Connectors:
- USB Connectors
- Ethernet/LAN Port
- Audio Connectors

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

MBR is an older partitioning scheme that has been used since the early days of DOS and Windows operating systems. It allows for up to four primary partitions or three primary partitions and one extended partition. MBR partition scheme is typically used for hard disks that are smaller than 2TB.

GPT, on the other hand, is a newer partitioning scheme that was introduced with the UEFI (Unified Extensible Firmware Interface) standard. It supports larger disk sizes and allows for up to 128 partitions. GPT partition scheme is used for hard disks that are larger than 2TB, and for the latest versions of Windows, macOS and Linux.

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

Mounting is the process of making a file system accessible to the operating system by attaching it to a directory tree.

Mounting is necessary because the operating system needs to be able to access multiple file systems simultaneously, and each file system may have its own file hierarchy and naming conventions.

Хід роботи

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

*Готував матеріал студент **Фещенко** **Евгеній**.*

а) Запустіть віртуальну машину VirtualBox, оберіть CentOS та запустіть її.

б) Запустіть віртуальну машину Ubuntu_PC

г) Запустіть свою операційну систему сімейства Linux

2. Опрацюйте всі приклади команд. Створіть таблицю для опису цих команд:

Готував матеріал студент **Кошіль Владислав**.

Назва команди	Її призначення та функціональність
nano	A minimalistic text file editor on the command line. Allows you to view and edit files.
gedit	A text file editor with a graphical interface. Allows you to view and edit files from a more user-friendly interface.
lspci	Command to display information about PCI (Peripheral Component Interconnect) devices on the computer.
lsmod	Command to display a list of loaded Linux kernel modules. Kernel modules allow you to add functionality to the system.
lsusb	Command to display information about USB (Universal Serial Bus) devices on the computer.
vi	Command-line text file editor. Allows you to view and edit files that do not have a graphical interface.

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

Готував матеріал студент **Кошіль Владислав**.

а) Сценарій має виводити привітання до поточного користувача вказуючи поточну дату та інформацію про поточну систему;

```
#!/bin/bash
```

```
# Отримання імені поточного користувача
USER_NAME=$(whoami)
```

```
# Отримання поточної дати та часу
CURRENT_DATE=$(date +"%d-%m-%Y")
CURRENT_TIME=$(date +"%H:%M:%S")
```

```
# Виведення привітання
echo "Hello $USER_NAME! Today $CURRENT_DATE at $CURRENT_TIME."
```

```
# Виведення інформації про поточну систему
echo "Detailed info about your system"
uname -a
```

```
quadfordt@Ubuntu:~$ echo "Hello, $USER_NAME! Today $CURRENT_DATE at $CURRENT_TIME."
Hello, quadfordt! Today 11-04-2023 at 17:36:55.
quadfordt@Ubuntu:~$ echo "Detailed info about your system"
Detailed info about your system
quadfordt@Ubuntu:~$ uname -a
Linux Ubuntu 5.19.0-38-generic #39-Ubuntu SMP PREEMPT_DYNAMIC Fri Mar 17 17:33:16 UTC 2023 x86_64 x86_64 x86_64 GNU/Linux
```

б)Сценарій має виводити інформацію про апаратну конфігурацію поточної системи.

```
#!/bin/bash
```

```
# Виведення заголовка
```

```
echo "Hardware configuration:"
```

```
# Виведення інформації про процесор
```

```
echo "CPU:"
```

```
lscpu | grep "Model" | sed 's/^.\{14\}//'
```

```
lscpu | grep "Arch" | sed 's/^.\{14\}//'
```

```
lscpu | grep "Core" | sed 's/^.\{14\}//'
```

```
# Виведення інформації про пам'ять
```

```
echo "Memory:"
```

```
free -h | grep "Memory:" | awk '{print $2}'
```

```
# Виведення інформації про жорсткий диск
```

```
echo "HDD:"
```

```
lsblk | grep "disk" | awk '{print $1 " (" $4 "B)}'
```

```
quadfordt@Ubuntu:~$ echo "Hardware configuration:"
Hardware configuration:
quadfordt@Ubuntu:~$ echo "CPU:"
CPU:
quadfordt@Ubuntu:~$ lscpu | grep "Model" | sed 's/^.\{14\}//'
Intel(R) Core(TM) i5-6200U CPU @ 2.30GHz
quadfordt@Ubuntu:~$ lscpu | grep "Arch" | sed 's/^.\{14\}//'
x86_64
quadfordt@Ubuntu:~$ lscpu | grep "Core" | sed 's/^.\{14\}//'
Intel(R) Core(TM) i5-6200U CPU @ 2.30GHz
quadfordt@Ubuntu:~$ echo "Memory:"
Memory:
quadfordt@Ubuntu:~$ free -h | grep "Memory:" | awk '{print $2}'
4G
quadfordt@Ubuntu:~$ echo "HDD:"
HDD:
quadfordt@Ubuntu:~$ lsblk | grep "disk" | awk '{print $1 " (" $4 "B)}'
sda (25GB)
quadfordt@Ubuntu:~$ _
```

Відповіді на контрольні запитання

*Готував матеріал студент **Фещенко Євгеній та Кошіль Владислав.***

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

Here are some common ways to work with variables in scripts:

Defining Variables: Variables can be defined in scripts using the appropriate syntax for the scripting language

being used.

Assigning Values: Once a variable is defined, values can be assigned to it using the appropriate syntax.

Manipulating Values: Variables can be manipulated in scripts using various operators and functions provided by the scripting language.

To create branched and looping scenarios in scripts, the following constructs are typically used:

If-Else Statements: If-else statements are used to create branching scenarios in scripts. They allow the script to execute different commands depending on the value of a variable or the result of a condition.

Switch Statements: Switch statements are another way to create branching scenarios in scripts. They allow the script to execute different commands based on the value of a variable.

Loops: Loops are used to repeat a set of commands multiple times in a script. There are several types of loops that can be used in scripts, including for loops, while loops, and do-while loops.

2. В чому відмінність між командами arch та lscpu?

Here are the main differences between these two commands:

Purpose: The "arch" command displays the machine architecture name of the system, while the "lscpu" command displays detailed information about the CPU architecture and other processor-related information.

Output: The output of the "arch" command is a single line of text indicating the machine architecture name of the system, such as "x86_64" for a 64-bit Intel or AMD processor. On the other hand, the output of the "lscpu" command is a detailed report containing information about the CPU architecture, including the processor family, model, clock speed, cache size, and more.

Usage: The "arch" command is generally used to determine the machine architecture of a system, which can be useful for determining which software packages or binaries are compatible with the system. The "lscpu" command, on the other hand, is used to obtain detailed information about the processor, which can be useful for performance tuning, optimization, and troubleshooting.

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

Here are some examples:

Linux:

"free": displays the amount of free and used memory in the system, including the total amount of RAM, the amount of used RAM, the amount of free RAM, and the amount of memory used for buffers and cache.

"top": displays real-time information about system processes, including their memory usage.

macOS:

"top": displays real-time information about system processes, including their memory usage.

"vm_stat": displays virtual memory statistics, including page ins and outs, faults, and more.

Windows:

"tasklist": displays a list of currently running processes, including their memory usage.

"taskmgr": opens the Windows Task Manager, which provides real-time information about system processes, including their memory usage.

4. Які команди для перегляду стану підключення периферійних пристроїв можна використати в

термінали?

Here are some examples of commands that can be used to view the connection status of peripherals in different scenarios:

USB devices on Linux:

"lsusb": lists all USB devices connected to the system.

Network devices on Linux:

"ifconfig": displays information about all network interfaces on the system.

"ip link": displays information about network interfaces, including their state and MAC address.

Serial devices on Linux:

"ls /dev/tty*": lists all serial devices connected to the system.

Disk drives on Linux:

"lsblk": lists all disk drives connected to the system, including their partition information.

USB devices on Windows:

"wmic path Win32_USBControllerDevice": lists all USB devices connected to the system.

Network devices on Windows:

"ipconfig": displays information about all network interfaces on the system.

Disk drives on Windows:

"diskpart": displays information about all disk drives connected to the system.

5. Які можливості застунку gparted?

Some of the main features of GParted include:

- Graphical User Interface: GParted provides a user-friendly graphical interface for managing partitions on storage devices.
- Partition Management: GParted can be used to create, resize, move, copy, and delete partitions on hard drives and other storage devices.
- File System Support: GParted supports a wide range of file systems, including ext2, ext3, ext4, NTFS, FAT32, and more.
- Live CD/USB Support: GParted can be run from a Live CD or USB drive, allowing users to manage partitions without installing GParted on their system.
- Multi-Language Support: GParted is available in multiple languages, making it accessible to users around the world.
- Disk Cloning: GParted allows users to clone an entire disk or partition to another disk or partition.

Висновок: In this lab, we learned how to create scripted scenarios and determine the hardware configuration of the system.