“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»**

Виконавли студенти

групи РПЗ-03

Команда: Губенко Є.О.,

Заїка С.В. та Кресан Р.А.

Перевірив викладач

Сушанова В.С.

Київ 2022

**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 Губенко Є.О***

1. Прочитайте короткі теоретичні відомості до лабораторної роботи та зробіть невеликий словник базових англійських термінів з питань призначення команд та їх параметрів.
2. Вивчіть матеріали онлайн-курсу академії Cisco “NDG Linux Essentials”:
   * Chapter 11 - Basic Scripting
   * Chapter 12 - Understanding Computer Hardware
3. Пройдіть тестування у курсі NDG Linux Essentials за такими темами:
   * Chapter 11 Exam
   * Chapter 12 Exam
4. На базі розглянутого матеріалу дайте відповіді на наступні питання:
   1. Охарактеризуйте поняття скриптового сценарію у командній оболонці.

*Скриптовий сценарій у командній оболонці є файлом, який містить послідовність команд і інструкцій, що виконуються послідовно при запуску скрипту. Скриптові сценарії зазвичай використовуються для автоматизації повторюваних задач або для запуску складних процесів з декількох команд.*

*У скриптовому сценарії можна використовувати всі команди, які підтримуються командною оболонкою, такі як змінні, розгалуження, цикли та функції. Скрипти зазвичай зберігаються в текстових файлах з розширенням .sh (для оболонки Bash в Unix-подібних системах) або .bat (для оболонки Windows).*

*Використання скриптових сценаріїв може допомогти зекономити час і зусилля, спростити процес роботи з командною оболонкою, зменшити кількість помилок і забезпечити більш консистентну та передбачувану роботу з системою.*

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

*Для створення скрипту в командній оболонці, спочатку потрібно створити текстовий файл з розширенням .sh (для Unix-подібних систем) або .bat (для Windows). В цьому файлі ми будемо записувати послідовність команд, які хочемо виконати. Далі, відкриваємо файл у текстовому редакторі та записуємо команди, які хочемо виконати, одну за одною, з нового рядка. Після того, як ми написали скрипт, зберігаємо файл з розширенням .sh або .bat.*

*Після створення скрипту, ми можемо редагувати його в будь-який час, відкриваючи файл у текстовому редакторі та вносячи зміни в послідовність команд.*

*Щоб запустити скрипт, ми повинні мати права на виконання файлу. У Unix-подібних системах це можна зробити за допомогою команди chmod +x script.sh, де script.sh - назва файлу з нашим скриптом. Після цього, ми можемо запустити скрипт, введенням команди ./script.sh у командній оболонці.*

*У Windows, файл з розширенням .bat можна запустити просто подвійним клацанням на ньому у провіднику Windows. Але в командній оболонці можна запустити файл, введенням його назви без розширення у командному рядку. Наприклад, якщо назва нашого скрипту script.bat, ми можемо запустити його, введенням команди script у командному рядку.*

* 1. Які основні компоненти материнської плати ви знаєте?

*Материнська плата є одним з основних компонентів комп'ютера, на якій розташовані різноманітні інші компоненти. Основними компонентами материнської плати є:*

* ***Сокет*** *- це місце на материнській платі, де розміщується процесор. В залежності від типу процесора, сокет може бути різного розміру та форми.*
* ***Слоти******для******оперативної******пам'яті*** *- це місця на материнській платі, де розташовуються модулі оперативної пам'яті. У сучасних материнських платах зазвичай використовуються слоти DDR4.*
* ***Слоти розширення*** *- це місця на материнській платі, куди можна підключити різноманітні розширювальні картки, такі як карти звуку, мережеві карти, карти відео тощо. Найпоширенішими типами слотів розширення є PCIe та PCI.*
* ***Контролери вводу-виводу*** *- це компоненти, що керують різноманітними інтерфейсами вводу-виводу, такими як USB, SATA, Ethernet, HDMI та інші.*
* ***Чіпсет*** *- це компонент, який забезпечує зв'язок між іншими компонентами на материнській платі, контролює швидкість передачі даних та інші параметри роботи системи.*
* ***Біос*** *- це програмний компонент, який забезпечує ініціалізацію та налаштування апаратних компонентів при запуску комп'ютера та зберігає основні налаштування системи.*

*Ці компоненти можуть відрізнятися в залежності від конкретної моделі материнської плати та її призначення.*

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

*MBR (Master Boot Record) та GPT (GUID Partition Table) - це дві різні структури даних, які використовуються для організації дискового простору на зовнішніх та внутрішніх жорстких дисках.*

*MBR використовується для старіших операційних систем, таких як Windows XP, які можуть бути встановлені на жорсткі диски об'ємом до 2 ТБ. MBR може містити до чотирьох основних розділів диска або три основних розділів та один розширений, що містить декілька логічних розділів.*

*GPT є більш сучасним стандартом та зазвичай використовується на жорстких дисках об'ємом понад 2 ТБ. Він може містити до 128 розділів, тому є більш гнучким та ефективним. GPT також має більш надійну систему захисту даних та підтримує більше форматів файлових систем, таких як NTFS, FAT32, exFAT та інші.*

*Обидва стандарти використовуються для організації дискового простору на зовнішніх та внутрішніх жорстких дисках, а також на дискетах та інших схожих пристроях зберігання даних.*

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

*Операція монтування - це процес приєднання (підключення) файлової системи до ієрархії файлів операційної системи. Після монтування файлова система стає доступною для читання та запису в операційній системі.*

*Монтування є необхідним, оскільки на більшості операційних систем диски та пристрої зберігання даних представлені як файлові системи, що розташовуються в окремих розділах. Якщо файлову систему не монтувати, то вона не буде доступною для використання. Наприклад, при підключенні зовнішнього жорсткого диска до комп'ютера, спочатку потрібно монтувати його, щоб операційна система змогла зчитувати та записувати файли на цей диск.*

*Операція монтування може виконуватися як автоматично, коли операційна система розпізнає підключений пристрій та монтує його автоматично, так і вручну, коли користувач самостійно вказує директорію, в яку потрібно змонтувати пристрій з файловою системою.*

1. Підготувати в електронному вигляді початковий варіант звіту:

* Титульний аркуш, тема та мета роботи
* Словник термінів
* Відповіді на п.4.1 та п.4.5 з завдань для попередньої підготовки

**Хід роботи**

1. Початкова робота в CLI-режимі в Linux ОС сімейства Linux:
   * 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.
   * Запустіть віртуальну машину Ubuntu\_PC (якщо виконуєте завдання ЛР через академію netacad)
   * Запустіть свою операційну систему сімейства Linux (якщо працюєте на власному ПК та її встановили) та запустіть термінал.

***Готував матеріал студент Заїка С. В.***

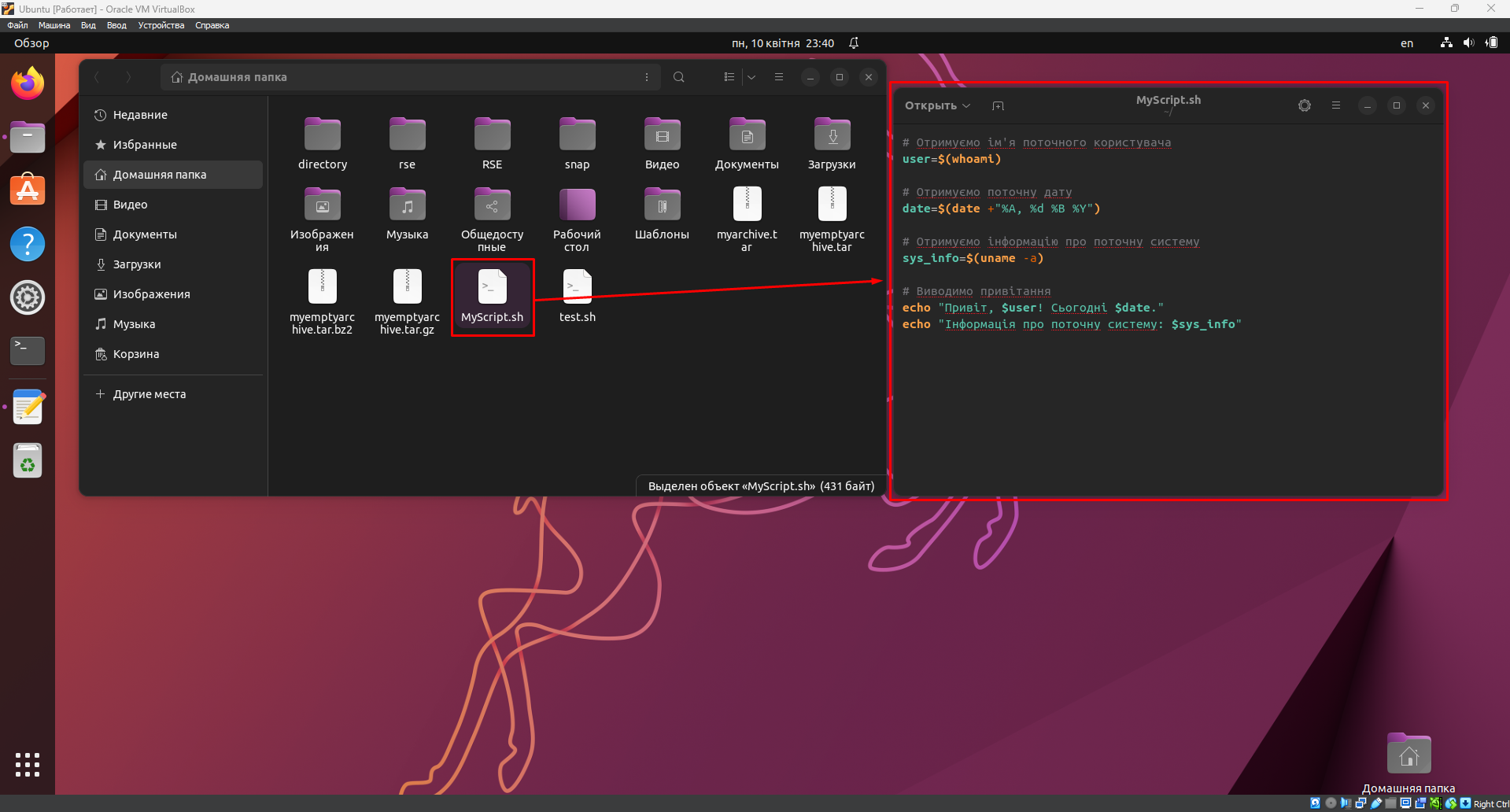
1. Опрацюйте всі приклади команд, що представлені у лабораторних роботах курсу NDG Linux Essentials - Lab 11: Basic Scripting та Lab 12: Understanding Computer Hardware. Створіть таблицю для опису цих команд\*\*\*

|  |  |
| --- | --- |
| **Назва команди** | **Її призначення та функціональність** |
| vi myfile | Команда vi myfile відкриває текстовий редактор Vim (Vi Improved) для редагування файлу з назвою "myfile". Vim є потужним інструментом для редагування тексту в командному рядку, який надає багато корисних функцій, таких як підсвічування синтаксису, автодоповнення та багато іншого. |
| dw | Команда dw є скороченням від "delete word" і використовується у текстовому редакторі Vim для видалення слова, на якому знаходиться курсор. Вим визначає слово як послідовність символів, розділених пробілами, табуляцією або символами нового рядка. |
| xxxx | Команда xxxx не є стандартною командою у текстових редакторах або в операційній системі. Будь ласка, уточніть, про який контекст або програму йдеться, щоб я зміг надати корисну відповідь. |
| :%s/text//g | Команда :%s/text//g виконує пошук та заміну в текстовому редакторі Vim у всьому файлі. Заміна полягає у видаленні всіх входжень підрядка text, за яким слідує пробіл (або інший роздільник) у всьому файлі. Символ % вказує, що пошук та заміна відбудеться в усьому файлі, s - позначає команду заміни, / - роздільник пошукового виразу, // - роздільник замінного виразу, g - вказує, що необхідно виконати заміну у всіх входженнях на рядку, а не тільки на першому. |
| :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. |
| lscpu | Команда lscpu використовується в операційній системі Linux для відображення інформації про центральний процесор (CPU) на системі. Виконання команди виводить детальну інформацію про архітектуру CPU, таку як виробник, модель, кількість ядер та потоків, швидкість тактування, кеш-пам'ять та інші параметри. |
| head -n 20/proc/cpuinfo | Команда head -n 20 /proc/cpuinfo використовується в операційній системі Linux для виведення перших 20 рядків вмісту файлу /proc/cpuinfo, який містить інформацію про центральний процесор (CPU) на системі. |
| lspci | Команда lspci використовується в операційній системі Linux для відображення інформації про PCI (Peripheral Component Interconnect) пристрої, підключені до комп'ютера. Виконання команди виводить список всіх пристроїв, підключених до шини PCI, разом з їхніми ідентифікаторами, виробниками та моделями. |
| lsusb | Команда lsusb використовується в операційній системі Linux для відображення інформації про USB-пристрої, підключені до комп'ютера. Виконання команди виводить список всіх USB-пристроїв, підключених до системи, разом з їхніми ідентифікаторами, виробниками та моделями. |
| fdisc - l | Команда fdisk -l використовується в операційній системі Linux для відображення інформації про розділи диска на системі. Виконання команди виводить список усіх дискових пристроїв на системі та їхніх розділів, що відображаються разом із їхніми параметрами, такими як тип файлової системи та розмір розділу. |

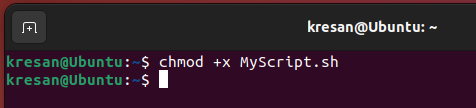
***Готував матеріал студент Кресан Руслан***

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;**

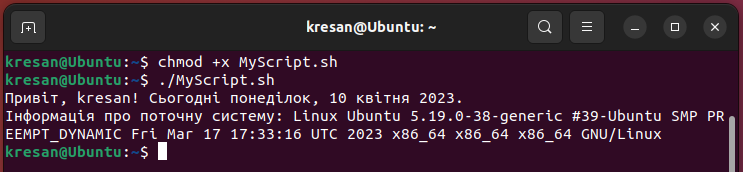
*Для створення скриптового сценарію нам необхідно відкрити текстовий редактор, записати у нього необхідні команди та зберегти файл з розширенням ”****.sh****”.*



*Після цього необхідно відкрити термінал та встановити права виконання на файл, використовуючи команду chmod .*

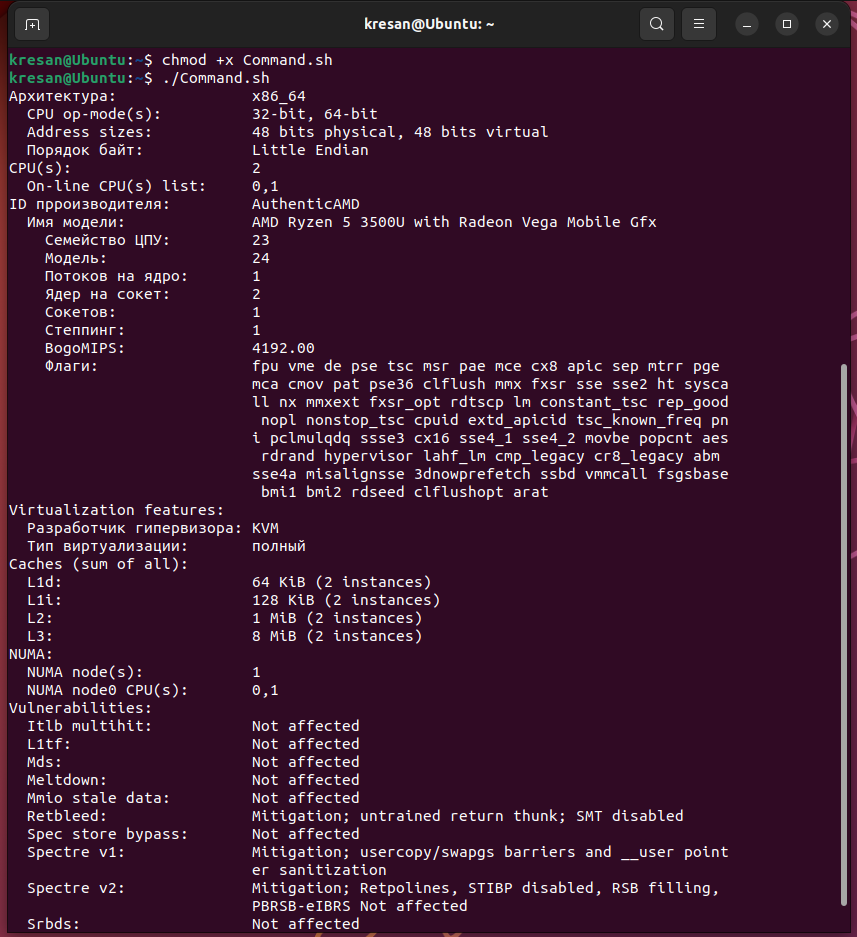


*Далі ми можемо запустити скрипт, виконавши команду ./MyScript.sh.*



* + **сценарій має виводити інформацію про апаратну конфігурацію поточної системи (використовуйте команди розглянуті в Lab 12).**

*Так як і в попередньому завданні ми створюємо файл з розширенням ”****.sh****”, заносимо до нього необхідні команди, встановлюємо право виконання та запускаємо скрипт.*



***Готував матеріал студент Губенко Є.О***

**Відповіді на контрольні запитання**

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

*У скриптах на мові програмування для оболонки Linux, таких як Bash, ви можете опрацьовувати змінні та створювати розгалужені (умовні) та циклічні (циклічні) сценарії. Ось кілька основних підходів:*

* *Змінні: Ви можете створювати змінні у скриптах на мові Bash, присвоюючи значення змінним. Наприклад:*

*name="John"*

*age=30*

*echo "Мене звати $name і мені $age років."*

* *Розгалужені (умовні) конструкції: Ви можете використовувати розгалужені конструкції, такі як if, elif та else, для виконання різних дій залежно від умови. Наприклад:*

*age=30*

*if [ $age -gt 18 ]; then*

*echo "Я доросла особа."*

*else*

*echo "Я дитина."*

*fi*

* *Циклічні (циклічні) конструкції: Ви можете використовувати циклічні конструкції, такі як for, while та until, для виконання блоку коду кілька разів або до виконання певної умови. Наприклад:*

*count=1*

*while [ $count -le 5 ]; do*

*echo "Значення: $count"*

*count=$((count+1))*

*done*

1. В чому відмінність між командами arch та lscpu?

*Команда arch та lscpu - це дві різні команди, що надають різні інформаційні характеристики про операційну систему та апаратне забезпечення комп'ютера.*

*Команда arch повертає інформацію про архітектуру операційної системи, на якій запущений процес, наприклад, x86, x86\_64, arm, arm64 тощо. Для отримання цієї інформації достатньо ввести в термінал команду arch, яка поверне результат у вигляді текстового рядка з назвою архітектури.*

*Наприклад, виконання команди arch на комп'ютері з архітектурою x86\_64 поверне наступний результат:* ***x86\_64***

*Команда lscpu, з іншого боку, повертає детальну інформацію про апаратне забезпечення, таке як кількість процесорів, їх модель, кількість ядер, розмір кеш-пам'яті, а також інші деталі апаратної конфігурації системи. Результат виконання команди lscpu також повертається у вигляді текстового рядка.*

*Наприклад, виконання команди lscpu на комп'ютері з двома процесорами Intel Xeon E5-2697 поверне наступний результат:*

***Architecture: x86\_64***

***CPU op-mode(s): 32-bit, 64-bit***

***Byte Order: Little Endian***

***CPU(s): 48***

***On-line CPU(s) list: 0-47***

***Thread(s) per core: 2***

***Core(s) per socket: 12***

***Socket(s): 2***

***NUMA node(s): 2***

***Vendor ID: GenuineIntel***

***CPU family: 6***

***Model: 63***

***Model name: Intel(R) Xeon(R) CPU E5-2697 v3 @ 2.60GHz***

***Stepping: 2***

***CPU MHz: 2594.251***

***BogoMIPS: 5192.50***

***Virtualization: VT-x***

***L1d cache: 32K***

***L1i cache: 32K***

***L2 cache: 256K***

***L3 cache: 35840K***

***NUMA node0 CPU(s): 0-11,24-35***

***NUMA node1 CPU(s): 12-23,36-47***

*Отже, основна відмінність між командами arch та lscpu полягає в тому, що перша команда повертає інформацію про архітектуру операційної системи, а друга - про апаратне забезпечення комп'ютера. Обидві команди є корисними у визначенні параметрів системи та можуть бути використані для діагностики, настройки та управління комп'ютером.*

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

*У більшості операційних систем для отримання інформації про стан використання RAM поточною системою використовують команду free.*

*Ця команда показує загальну кількість фізичної та віртуальної пам'яті, яку використовує система, а також кількість вільної, вільної для буферизації та вільної для кешування пам'яті. Для використання команди відкрийте термінал і введіть:* ***free***

*Якщо ви хочете отримати більш детальну інформацію про використання RAM, ви можете використати команду top, яка показує інформацію про процеси, що виконуються на вашій системі, включаючи інформацію про використання RAM. Щоб запустити команду top, відкрийте термінал і введіть:* ***top***

*У верхній частині вікна top ви побачите загальну інформацію про систему, включаючи використання пам'яті.*

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

*Для перегляду стану підключення периферійних пристроїв у терміналі можна використати такі команди:*

* *lsusb: ця команда показує список всіх USB-пристроїв, підключених до вашої системи. Якщо пристрій правильно підключено і встановлено його драйвери, то ви можете побачити детальну інформацію про пристрій, таку як ідентифікатор виробника та моделі.*
* *lspci: ця команда показує список всіх PCI-пристроїв, підключених до вашої системи, таких як відеокарти, мережеві адаптери та інші. Вона також показує детальну інформацію про пристрій, таку як його ідентифікатор, назва та виробник.*
* *lsblk: ця команда показує список всіх зберігання на вашій системі, включаючи тверді диски, флешки та інші пристрої зберігання. Вона також показує інформацію про їх розмір та тип файлової системи.*
* *dmesg: ця команда показує журнал подій ядра, що містить інформацію про підключення та відключення периферійних пристроїв, а також інші події, що стосуються їх роботи.*
* *udevadm: ця команда надає можливість взаємодіяти з системою управління пристроями udev, яка відповідає за динамічне створення та видалення файлів пристроїв. Вона дозволяє переглядати інформацію про підключені пристрої, в тому числі їх ідентифікатори, типи та властивості.*

1. Які можливості застунку gparted?

*GParted - це графічний інтерфейс для керування дисками та розділами в Linux і Windows. Він має наступні можливості:*

* *Створення, видалення, редагування та переміщення розділів на жорсткому диску.*
* *Переміщення розділів на жорсткому диску без втрати даних.*
* *Перенесення вільного простору між розділами.*
* *Форматування розділів у різних файлових системах, таких як NTFS, FAT32, ext4, і багатьох інших.*
* *Перейменування розділів.*
* *Перегляд інформації про жорсткий диск та розділи, включаючи типи файлових систем, розміри розділів та їх вміст.*
* *Робота з дисками, що містять різні типи розділів, такі як MBR та GPT.*
* *Перетворення MBR у GPT та навпаки.*
* *Клонування диску або розділу для створення резервної копії даних.*
* *Підтримка багатьох мов і зручний інтерфейс користувача.*

*Загалом, GParted є потужним та зручним інструментом для роботи з дисками та розділами в Linux і Windows, що дозволяє користувачам легко і ефективно керувати своїми даними на жорсткому диску.*

***Готував матеріал студент Заїка С. В.***

**Висновки**

В ході виконання лабораторної роботи ми отримали практичні навички роботи з командною оболонкою Bash, та познайомились з базовими діями при роботі зі скриптовими сценаріями.