

“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 РПЗ-13а
DOMINO: Яковенко
Н.Ю.,
Чурюмова К.А. та
Скряга П.В.
Перевірила викладач
Сушанова В.С.

Київ 2024

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

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 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

Матеріал підготував Яковенко Н.

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.

English	Ukrainian
behavior accordingly	поводитися відповідно
executable	виконуваний файл
permissions	дозволи
buffer	буфер
script	скрипт
to execute the script	для виконання скрипта
variables	змінні
shell	оболонка

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:
 - 4.1. *Охарактеризуйте поняття скриптового сценарію у командній оболонці.
A shell script is a set of commands that are executed sequentially to perform certain tasks or automate certain processes. Scripts can contain conditions, loops, variables, and other programming constructs to control the execution of commands.
 - 4.2. *Яким чином створюються та редагуються скрипти, що треба зробити щоб запустити скрипт?
Scripts can be created and edited using text editors such as Nano, Vim, or any text editor. To run a script, you first need to set its executive privileges using the command `chmod +x script_name.sh`, where `script_name.sh` is the name of your script. After that, you can run the script by entering the path to it in the command line.

4.3. **Які основні компоненти материнської плати ви знаєте?

The main components of the motherboard include the processor, RAM, connectors for connecting a disc drive (SATA, M.2), connectors for connecting other devices (USB, HDMI, Ethernet), BIOS or UEFI, power supply system (including capacitors, voltage regulators), and other important components.

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

The MBR (Master Boot Record) and GPT (GUID Partition Table) are two different formats for allocating data on storage devices such as hard disk drives (HDDs) or solid-state drives (SSDs). MBR is traditionally used for BIOS systems, while GPT is used for UEFI systems. Both of these formats are used to store information about disc partitions and their attributes.

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

The mount operation is the process of attaching a file system stored on a storage device (for example, a hard drive or a flash drive) to a specific point in the file system hierarchy of the operating system. This is necessary for the operating system to access the contents of the device. Mounting allows the operating system to see and work with the contents of the device as if it were a regular part of the file system.

5. 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:

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.

Матеріал підготувала Чурюмова К.

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
vi myfile	Create a new file
wq	to 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	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
u	Undo the last operation
2dw	Delete two words
xxxx	Delete four characters, one at a time
p	Whatever was lasted deleted or yanked can be “pasted”. Paste the deleted lines below the current line
J	Join two lines, the current and the next
yw	Copy (or “yank”) the current word
:%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
a	Enter insert mode.
o	Open a blank line below the current line
: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 . You'll notice the cursor moves to the beginning of the first instance of the word line as shown in image below
n	Search for the next instance of the word line
?line	Search backward for the word line . You'll notice the cursor moves to the beginning of the previous instance of the word line as shown in image below
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
lscpu	In order to determine the type of CPU execute the lscpu command
Head -n	Use the head command with the -n option to list the first 20 lines of the cpuinfo file
Free -m	The output shows the amount of memory in megabytes when the -m
Free -g	in gigabytes when the -g option is used
lspci	To see what devices are connected to the PCI bus
Lspci -k	to show 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

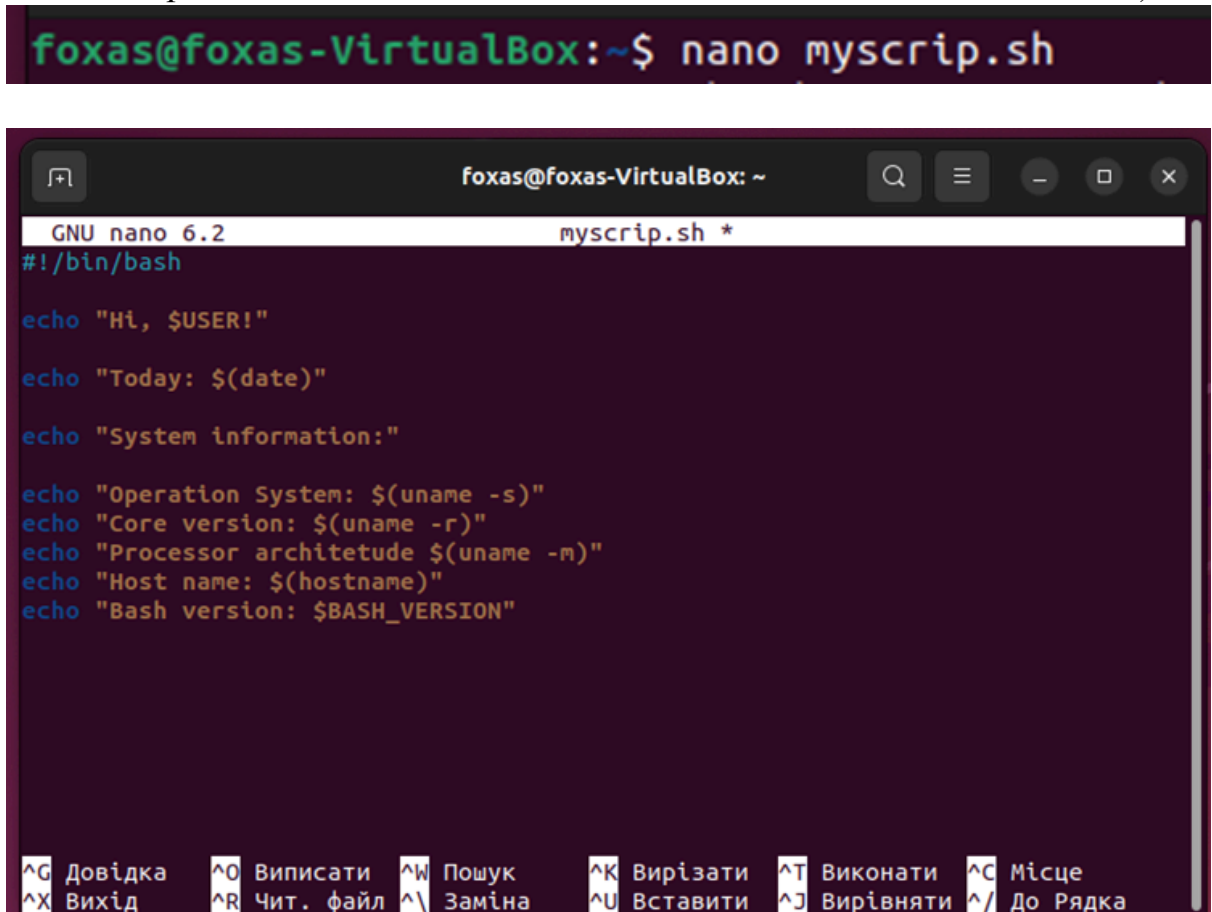
fdisk	command to list the disk devices
Fdisk -l	The -l option lists the partition tables for the specified devices and then exits.

Примітка: Скріншоти виконання команд в терміналі можна **не представляти**, достатньо коротко описати команди в таблиці.

Матеріал підготував Скрыга П.

3. 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;

First create a script with the command: script_name.sh. A window opens in which we write the code of our script, in our case we greet the user, show the current date and brief information about the system (OS name, kernel version, processor architecture, hostname and BASH version)



The image shows a terminal window with a dark background. At the top, the prompt is 'foxas@foxas-VirtualBox:~\$'. The user has entered 'nano myscrip.sh', opening the nano text editor. The editor's title bar shows 'GNU nano 6.2' and 'myscrip.sh *'. The file content is as follows:

```
#!/bin/bash

echo "Hi, $USER!"

echo "Today: $(date)"

echo "System information:"

echo "Operation System: $(uname -s)"
echo "Core version: $(uname -r)"
echo "Processor archititude $(uname -m)"
echo "Host name: $(hostname)"
echo "Bash version: $BASH_VERSION"
```

At the bottom of the terminal, there is a row of keyboard shortcuts in Ukrainian: ^G Довідка, ^O Виписати, ^W Пошук, ^K Вирізати, ^T Виконати, ^C Місце, ^X Вихід, ^R Чит. файл, ^_ Заміна, ^U Вставити, ^J Вирівняти, ^/ До Рядка.

Next, run the `chmod +x script_name.sh` command to grant the script execution rights. And run the script itself

```
foxas@foxas-VirtualBox:~$ chmod +x myscrip.sh
foxas@foxas-VirtualBox:~$ ./myscrip.sh
Hi, foxas!
Today: четвер, 28 березня 2024 21:52:55 +0200
System information:
Operation System: Linux
Core version: 6.5.0-26-generic
Processor archititude x86_64
Host name: foxas-VirtualBox
Bash version: 5.1.16(1)-release
foxas@foxas-VirtualBox:~$
```

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

We do the same as in the first task, but in the code we write all the commands that were in the 12th lecture of the Cisco tests

```
foxas@foxas-VirtualBox:~$ nano myscript.sh
```

```
foxas@foxas-VirtualBox: ~
GNU nano 6.2 myscript.sh
#!/bin/bash
echo "Hardware configuration information"
echo "CPU information:"
lspcu
echo "Memory information:"
free -m
free -g
echo "PCI device information:"
lspci
echo "USB devise information:"
lsusb
echo "DISK information:"
sudo fdisk -l
```

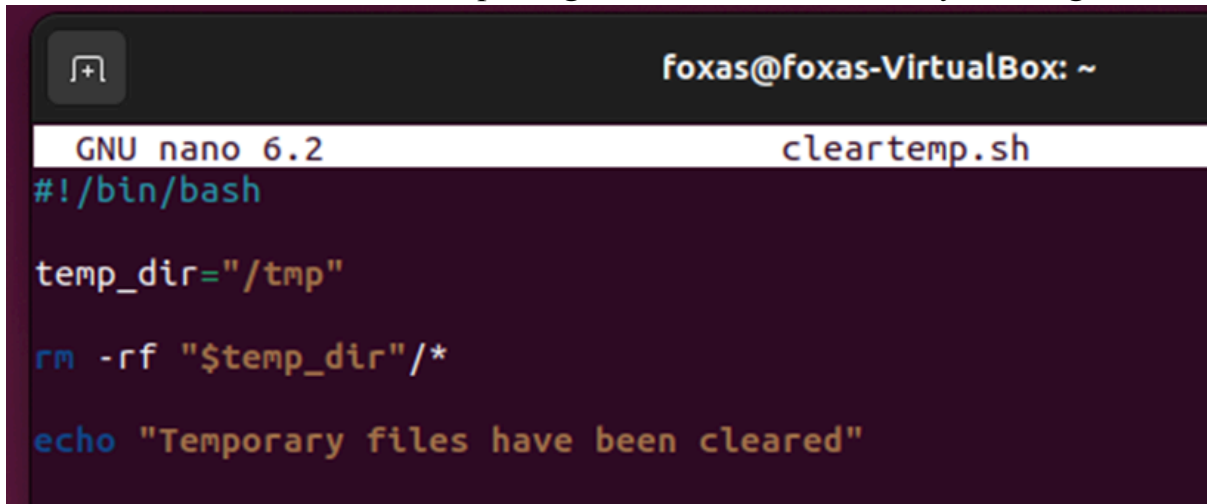

We also grant execution rights and receive information about (CPU, Memory, PCI and USB Device and DISK)

```
foxas@foxas-VirtualBox: ~  
foxas@foxas-VirtualBox:~$ chmod +x myscript.sh  
foxas@foxas-VirtualBox:~$ ./myscript.sh  
Hardware configuration information  
CPU information:  
./myscript.sh: рядок 4: lspci: команду не знайдено  
Memory information:  
Пам.:          загалом      використ.   вільна     спільна буфери/кеш  дост.  
Своп.:          8390        1165       5294       52      1930      6915  
Пам.:          2047         0         2047  
Своп.:          8          1          5          0          1          6  
Пам.:          8          1          5          0          1          6  
Своп.:          1          0          1  
PCI device information:  
00:00.0 Host bridge: Intel Corporation 440FX - 82441FX PMC [Natoma] (rev 02)  
00:01.0 ISA bridge: Intel Corporation 82371SB PIIX3 ISA [Natoma/Triton II]  
00:01.1 IDE interface: Intel Corporation 82371AB/EB/MB PIIX4 IDE (rev 01)  
00:02.0 VGA compatible controller: VMware SVGA II Adapter  
00:03.0 Ethernet controller: Intel Corporation 82540EM Gigabit Ethernet Controller (rev 02)  
00:04.0 System peripheral: InnoTek Systemberatung GmbH VirtualBox Guest Service  
00:05.0 Multimedia audio controller: Intel Corporation 82801AA AC'97 Audio Controller (rev 01)  
00:06.0 USB controller: Apple Inc. KeyLargo/Intrepid USB  
00:07.0 Bridge: Intel Corporation 82371AB/EB/MB PIIX4 ACPI (rev 08)  
00:0b.0 USB controller: Intel Corporation 82801FB/FBM/FR/FW/FRW (ICH6 Family) USB2 EHCI Controller  
00:0d.0 SATA controller: Intel Corporation 82801HM/HEM (ICH8M/ICH8M-E) SATA Controller [AHCI mode] (rev 02)  
USB device information:  
Bus 002 Device 001: ID 1d6b:0002 Linux Foundation 2.0 root hub  
Bus 001 Device 002: ID 80ee:0021 VirtualBox USB Tablet  
Bus 001 Device 001: ID 1d6b:0001 Linux Foundation 1.1 root hub  
DISK information:  
[sudo] пароль до foxas:  
Диск /dev/loop0: 4 KiB, 4096 байтів, 8 секторів  
Одиниці: секторів з 1 * 512 = 512 байтів  
Розмір сектора (логічного/фізичного): 512 байтів / 512 байтів  
Розмір введення-виведення (мінімальний/оптимальний): 512 байтів / 512 байтів  
  
Диск /dev/loop1: 63,45 MiB, 66531328 байтів, 129944 секторів  
Одиниці: секторів з 1 * 512 = 512 байтів  
Розмір сектора (логічного/фізичного): 512 байтів / 512 байтів  
Розмір введення-виведення (мінімальний/оптимальний): 512 байтів / 512 байтів  
  
Диск /dev/loop2: 63,91 MiB, 67010560 байтів, 130880 секторів  
Одиниці: секторів з 1 * 512 = 512 байтів  
Розмір сектора (логічного/фізичного): 512 байтів / 512 байтів  
Розмір введення-виведення (мінімальний/оптимальний): 512 байтів / 512 байтів  
  
Диск /dev/loop3: 73,88 MiB, 77463552 байтів, 151296 секторів  
Одиниці: секторів з 1 * 512 = 512 байтів  
Розмір сектора (логічного/фізичного): 512 байтів / 512 байтів  
Розмір введення-виведення (мінімальний/оптимальний): 512 байтів / 512 байтів
```


- **наведіть свій приклад скриптового сценарію.

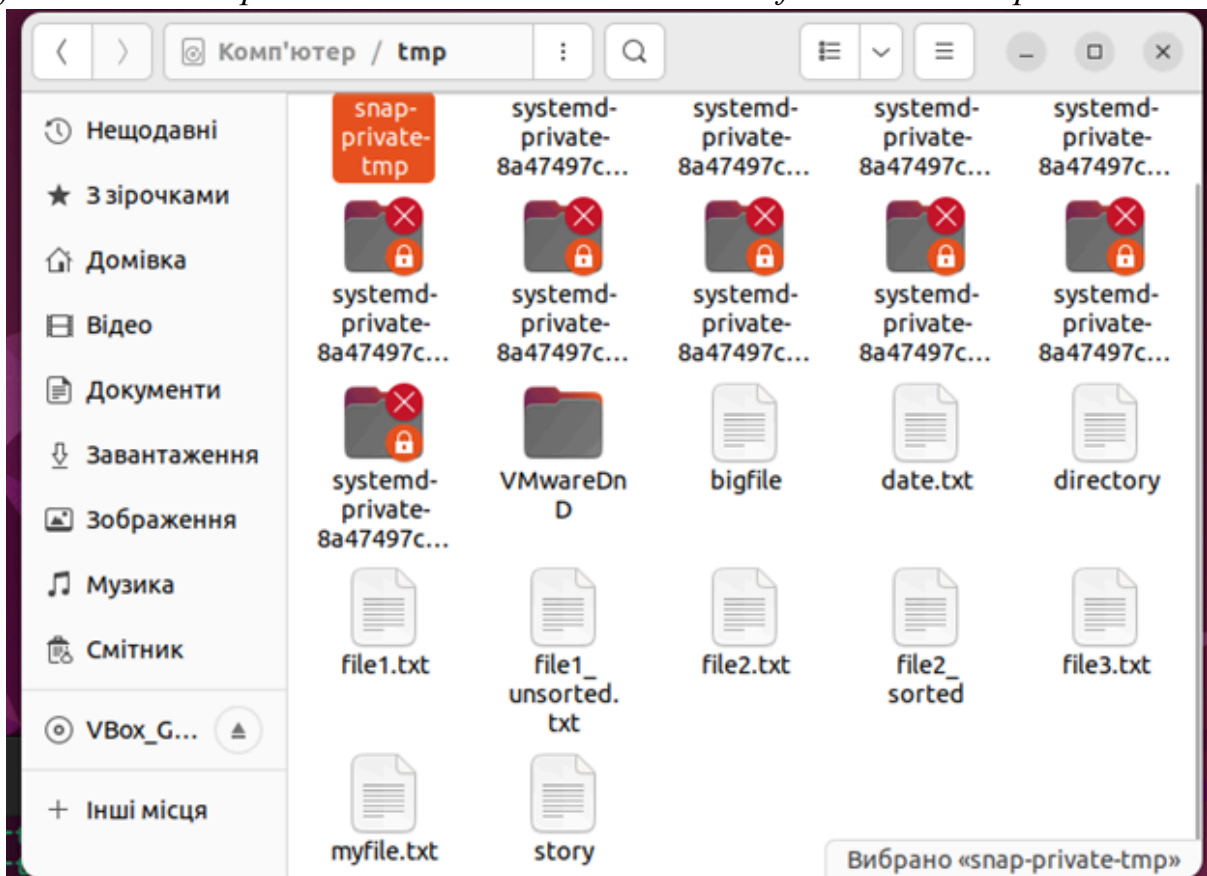
Let's make a very easy but very useful script that will clean up unnecessary temporary system files.

To do this, create a script again and write the following code:



```
foxas@foxas-VirtualBox: ~  
GNU nano 6.2 cleartemp.sh  
#!/bin/bash  
  
temp_dir="/tmp"  
  
rm -rf "$temp_dir"/*  
  
echo "Temporary files have been cleared"
```

This code should delete all unnecessary temporary files (all files that the system needs will be left untouched). To test the program, let's transfer the text files from the previous lab and see if the script works.

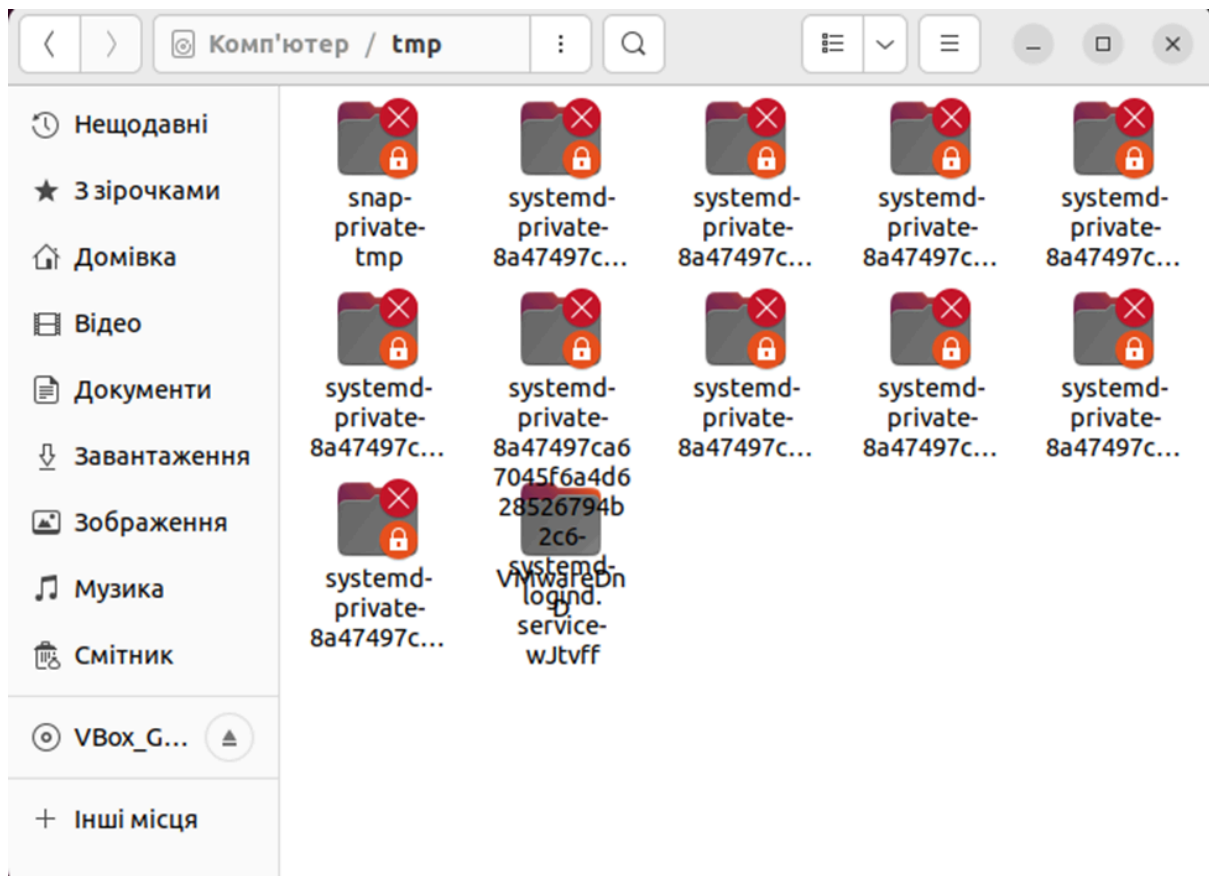


Run the script

```

foxas@foxas-VirtualBox:~$ nano cleartemp.sh
foxas@foxas-VirtualBox:~$ chmod +x cleartemp.sh
foxas@foxas-VirtualBox:~$ ./cleartemp.sh
rm: не вдалося вилучити '/tmp/snap-private-tmp-1': дію заборонено
rm: не вдалося вилучити '/tmp/systemd-private-8a47497ca67045f6a4d628526794b2c6-color.service-WY3NvU': дію заборонено
rm: не вдалося вилучити '/tmp/systemd-private-8a47497ca67045f6a4d628526794b2c6-fwupd.service-xfk5Cj': дію заборонено
rm: не вдалося вилучити '/tmp/systemd-private-8a47497ca67045f6a4d628526794b2c6-ModemManager.service-0reqbC': дію заборонено
rm: не вдалося вилучити '/tmp/systemd-private-8a47497ca67045f6a4d628526794b2c6-power-profiles-daemon.service-Ek19H4': дію заборонено
rm: не вдалося вилучити '/tmp/systemd-private-8a47497ca67045f6a4d628526794b2c6-switcheroo-control.service-DwHGM4': дію заборонено
rm: не вдалося вилучити '/tmp/systemd-private-8a47497ca67045f6a4d628526794b2c6-systemd-logind.service-wJtvff': дію заборонено
rm: не вдалося вилучити '/tmp/systemd-private-8a47497ca67045f6a4d628526794b2c6-systemd-oomd.service-ZtNQvB': дію заборонено
rm: не вдалося вилучити '/tmp/systemd-private-8a47497ca67045f6a4d628526794b2c6-systemd-resolved.service-Y6SUu1': дію заборонено
rm: не вдалося вилучити '/tmp/systemd-private-8a47497ca67045f6a4d628526794b2c6-systemd-timesyncd.service-bdVNu6': дію заборонено
rm: не вдалося вилучити '/tmp/systemd-private-8a47497ca67045f6a4d628526794b2c6-upower.service-hyr5W5': дію заборонено
rm: не вдалося вилучити '/tmp/VMwareDnD': дію заборонено
Temporary files have been cleared
foxas@foxas-VirtualBox:~$

```



As a result, all unnecessary files are deleted. Use! You can come up with an infinite number of useful scripts. But this is the first useful idea that came to my mind.

Контрольні запитання:

1. В чому відмінність між командами `arch` та `lscpu`?

The 'arch' command displays the architecture of the machine you're using. The 'lscpu' will retrieve specific information about the CPU configuration. Commonly used options include -b for machine-readable output, -c for displaying online CPUs, and -p for displaying output in a parsable format.

To illustrate, here are the results of both commands

```
foxas@foxas-VirtualBox: ~  
foxas@foxas-VirtualBox:~$ arch  
x86_64  
foxas@foxas-VirtualBox:~$ lscpu  
Архітектура: x86_64  
Режими ЦП: 32-bit, 64-bit  
Розміри адрес: 48 bits physical, 48 bits virtual  
Порядок байтів: Little Endian  
Процесори: 4  
Список робочих ЦП: 0-3  
Ідентифікатор виробника: AuthenticAMD  
Назва моделі: AMD Ryzen 5 3600 6-Core Processor  
Сімейство процесорів: 23  
Модель: 113  
Потоків на ядро: 1  
Ядер на сокет: 4  
Сокети: 1  
Модифікація: 0  
ВогоMIPS: 8400.03  
Прапорці: fpu vme de pse tsc msr pae mce cx8 apic sep mtrr pge m  
ca cmov pat pse36 clflush mmx fxsr sse sse2 ht syscall  
nx mmxext fxsr_opt rdtscp lm constant_tsc rep_good no  
pl nonstop_tsc cpuid extd_apicid tsc_known_freq pni pc  
lmulq dq ssse3 cx16 sse4_1 sse4_2 movbe popcnt aes rdra  
nd hypervisor lahf_lm cmp_legacy cr8_legacy abm sse4a  
misalignsse 3dnowprefetch ssbd vmcall fsgsbase bmi1 b  
mi2 rdseed clflushopt arat  
Virtualization features:  
Виробник гіпервізора: KVM  
Тип віртуалізації: повна  
Caches (sum of all):  
L1d: 128 KiB (4 instances)  
L1i: 128 KiB (4 instances)  
L2: 2 MiB (4 instances)  
L3: 128 MiB (4 instances)  
NUMA:  
Вузли NUMA: 1  
Процесори вузла0 NUMA: 0-3  
Vulnerabilities:  
Gather data sampling: Not affected  
Itlb multihit: Not affected  
L1tf: Not affected  
Mds: Not affected  
Meltdown: Not affected  
Mmio stale data: Not affected  
Retbleed: Mitigation; untrained return thunk; SMT disabled  
Spec rstack overflow: Vulnerable: Safe RET, no microcode  
Spec store bypass: Not affected  
Spectre v1: Mitigation; usercopy/swapgs barriers and __user pointe  
r sanitization  
Spectre v2: Mitigation; Retpolines, STIBP disabled, RSB filling, P  
BRSE-eIBRS Not affected  
Srbds: Not affected  
Tsx async abort: Not affected  
foxas@foxas-VirtualBox:~$ S
```

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

To discover how much RAM and swap space is being used, use the free command: free -m, free -g

```
foxas@foxas-VirtualBox:~$ free
              загалом      використ.   вільна   спільна буфери/кеш   дост.
Пам.:      8592116    1232372    5021164    54168    2338580    7030608
Своп.:      2097148         0    2097148
foxas@foxas-VirtualBox:~$ free -m
              загалом      використ.   вільна   спільна буфери/кеш   дост.
Пам.:      8390      1203    4902      52      2283      6865
Своп.:      2047         0    2047
foxas@foxas-VirtualBox:~$ free -g
              загалом      використ.   вільна   спільна буфери/кеш   дост.
Пам.:         8         1         4         0         2         6
Своп.:         1         0         1
```

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

Scripts use your system's shell tools. For example, to process variables, you use constructs such as `my_variable="Hello, world!"` and `echo $my_variable` for output, and to create loops and branches, you use the `if`, `for`, `while`, etc. constructs.

For example:

if [condition]; then

//commands that are executed if the condition is met

else

//commands that are executed if the condition is not met

fi

4. *Які команди для перегляду стану підключення периферійних пристроїв можна використати в терміналі? *To see what devices are connected to the PCI bus, use the `lspci` command, for USB connected devices: `lsusb`*

```
foxas@foxas-VirtualBox:~$ lspci
00:00.0 Host bridge: Intel Corporation 440FX - 82441FX PMC [Natoma] (rev 02)
00:01.0 ISA bridge: Intel Corporation 82371SB PIIX3 ISA [Natoma/Triton II]
00:01.1 IDE interface: Intel Corporation 82371AB/EB/MB PIIX4 IDE (rev 01)
00:02.0 VGA compatible controller: VMware SVGA II Adapter
00:03.0 Ethernet controller: Intel Corporation 82540EM Gigabit Ethernet Controller (rev 02)
00:04.0 System peripheral: InnoTek Systemberatung GmbH VirtualBox Guest Service
00:05.0 Multimedia audio controller: Intel Corporation 82801AA AC'97 Audio Controller (rev 01)
00:06.0 USB controller: Apple Inc. KeyLargo/Intrepid USB
00:07.0 Bridge: Intel Corporation 82371AB/EB/MB PIIX4 ACPI (rev 08)
00:0b.0 USB controller: Intel Corporation 82801FB/FBM/FR/FW/FRW (ICH6 Family) USB2 EHCI Controller
00:0d.0 SATA controller: Intel Corporation 82801HM/HEM (ICH8M/ICH8M-E) SATA Controller [AHCI mode] (rev 02)
foxas@foxas-VirtualBox:~$ lsusb
Bus 002 Device 001: ID 1d6b:0002 Linux Foundation 2.0 root hub
Bus 001 Device 002: ID 80ee:0021 VirtualBox USB Tablet
Bus 001 Device 001: ID 1d6b:0001 Linux Foundation 1.1 root hub
```

5. **Які можливості застунку gparted?

GParted enables you to easily manage your disk partitions:

- *Create partition tables, (e.g., msdos or gpt)*
- *Create, move, copy, resize, check, label, set new UUID, and delete partitions*
- *Enable and disable partition flags, (e.g., boot or hidden)*
- *Align partitions to mebibyte (MiB) or traditional cylinder boundaries*
- *Attempt data rescue from lost partitions*

GParted supports the following actions on file systems:

File system	Detect	Read	Create	Grow	Shrink	Move	Copy	Check	Label	UUID	Required software
apfs	✓	✗	✗	✗	✗	✓	✓	✗	✗	✗	
bitlocker	✓	✗	✗	✗	✗	✓	✓	✗	✗	✗	
btrfs	✓	✓	✓	✓✓ [2]	✓✓ [2]	✓	✓	✓	✓ [10]	✓ [15]	btrfs-progs / btrfs-tools
exfat	✓ [9]	✓	✓	✗	✗	✓	✓	✓	✓	✓	exfatprogs
ext2	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	e2fsprogs
ext3	✓	✓	✓	✓✓	✓	✓	✓	✓	✓	✓	e2fsprogs
ext4	✓ [7]	✓	✓	✓✓	✓	✓	✓	✓	✓	✓	e2fsprogs v1.41+
File system	Detect	Read	Create	Grow	Shrink	Move	Copy	Check	Label	UUID	Required software
f2fs	✓ [13]	✓	✓	✓	✗	✓	✓	✓	✗	✗	f2fs-tools
fat16	✓	✓	✓	✓ [4]	✓ [4]	✓	✓	✓	✓ [6]	✓ [6]	dosfstools
fat32	✓	✓	✓	✓ [4]	✓ [4]	✓	✓	✓	✓ [6]	✓ [6]	dosfstools
hfs	✓	✓	✓	✗	✓ [4]	✓	✓	✗	✗	✗	hfsutils
hfs+	✓	✓	✓	✗	✓ [4]	✓	✓	✓	✗	✗	hfsprogs
jfs	✓	✓	✓	✓✓ [2]	✗	✓	✓	✓	✓	✓	jfsutils
linux-raid	✓	✓	✗	✗	✗	✓	✓	✗	✗	✗	mdadm
File system	Detect	Read	Create	Grow	Shrink	Move	Copy	Check	Label	UUID	Required software
linux-suspend	✓	✗	✗	✗	✗	✓	✓	✗	✗	✗	
linux-swap	✓	✓	✓	✓	✓	✓	✓	✗	✓ [8]	✓ [8]	util-linux
luks	✓	✓	✗	✓✓	✗✓	✓	✓	✗	✗	✗	cryptsetup , dmsetup
lvm2 pv	✓	✓	✓	✓✓	✓✓	✓	✗	✓	✗	✗	lvm2
minix	✓	✗	✓	✗	✗	✓	✓	✓	✗	✗	util-linux
nilfs2	✓ [12]	✓	✓	✓✓ [2]	✓✓ [2]	✓	✓	✗	✓	✓	nilfs-utils / nilfs-tools
ntfs	✓	✓	✓	✓	✓	✓	✓ [1]	✓	✓	✓ [11]	ntfs-3g / ntfsprogs
File system	Detect	Read	Create	Grow	Shrink	Move	Copy	Check	Label	UUID	Required software
refs	✓ [14]	✗	✗	✗	✗	✓	✓	✗	✗	✗	
reiser4	✓	✓	✓	✗	✗	✓	✓	✓	✗	✗	reiser4progs
reiserfs	✓	✓	✓	✓✓	✓	✓	✓	✓	✓	✓	reiserfsprogs / reiserfs-utils
udf	✓	✓	✓	✗	✗	✓	✓	✗	✓	✓	New udftools , Old udftools
ufs	✓	✗	✗	✗	✗	✓	✓	✗	✗	✗	
xfs	✓	✓	✓	✓✓ [2]	✗ [3]	✓	✓ [5]	✓	✓	✓	xfsprogs , xfsdump
zfs	✓	✗	✗	✗	✗	✓	✓	✗	✗	✗	
File system	Detect	Read	Create	Grow	Shrink	Move	Copy	Check	Label	UUID	Required software



Supported offline and online. For example, unmounted and mounted.



Not supported offline. Supported online only.



Supported offline only.



Not supported.

Conclusion

In the course of this lab, we learned how to create scripted scenarios and perform basic actions with them.