“Kyiv Vocation College of Communication”
Cyclical Commission of Computer Engineering

REPORT ON EXECUTION
LABORATORY WORK №7

on the discipline: "Operating Systems"

Topic: "Scripting Scenario Creation and System Hardware Configuration Definition"

Performed by student
group RPZ-13a
Kateryna Hranat
Checked by teacher
V.S. Sushanova

Performed by student group RPZ-13a Kateryna Hranat

Kyiv 2024

Work objectives:

1. Acquiring practical skills in working with the Bash command shell.
2. Introduction to basic operations when working with scripting scenarios.

Material Support for Classes:

1. IBM PC type computers.
2. Windows operating system and VirtualBox virtual machine (Oracle).
3. GNU/Linux operating system (any distribution).
4. Cisco Networking Academy website netacad.com and its online courses on Linux.

Performed by student group RPZ-13a Kateryna Hranat

Tasks for Preliminary Preparation:

1. Read the brief theoretical information for the laboratory work and create a small dictionary of basic English terms related to the purpose of commands and their parameters.

English terms	**Ukrainian terms**
Shell Scripts	Скрипти оболонки
Variables	Змінні
Conditionals	Умовні конструкції
Loops	Цикли
Shebang	Шебанг
Hashbang	Хешбанг

2. Study the materials of the online course "NDG Linux Essentials" from Cisco:
   - Chapter 11 - Basic Scripting
   - Chapter 12 - Understanding Computer Hardware

Complete✔

3. Complete testing in the NDG Linux Essentials course on the following topics:
   - Chapter 11 Exam
   - Chapter 12 Exam

Complete✔

4. Based on the material covered, provide answers to the following questions:

   4.1. *Characterize the concept of a script in a command shell.

   Concept of a Script in Command Shell:

   A script in a command shell is a file containing a sequence of commands executed consecutively or in a specific order when the script is launched. They allow automating routine tasks or executing a sequence of commands from a single program or system, aiding in consistency and execution speed.

4.2. *How are scripts created and edited, and what needs to be done to execute a script?

Script Creation, Editing, and Execution:

Scripts can be created and edited using text editors such as Vim, Emacs, or Nano, or through integrated shell editors such as Bash or PowerShell ISE. To execute a script, you need to use a command that specifies the path to the script file and invokes the command shell interpreter, for example:

```
bash script.sh
```

or

```
./script.sh
```

4.3. **What are the main components of a motherboard that you know?

Main Components of a Motherboard:

The main components of a motherboard include the processor (CPU), random-access memory (RAM), chipset, connectors for device attachment (SATA, USB, PCI-E), BIOS or UEFI, power connectors (ATX or EPS), input-output controllers (I/O controllers), and interfaces for connecting peripheral devices (e.g., keyboards, mice, monitors).

4.4. **Briefly describe for which devices the concepts of MBR and GPT are applicable.

MBR and GPT for Devices:

MBR (Master Boot Record) and GPT (GUID Partition Table) are methods of partitioning and organizing data on storage devices. MBR is used for BIOS-based systems, whereas GPT is a more modern standard used with UEFI. MBR can be used for HDDs and SSDs, but GPT is typically used for modern devices working with UEFI.

4.5. **What is the essence of the mounting operation, and why is it necessary?

Essence of Mounting Operation:

The mounting operation is the process of attaching a file system located on a device (such as a hard disk or USB drive) to a specific point in the file system of the operating system. This is necessary to access files and folders on that device. For example, to access the contents of a USB drive, you must first mount it to a specific directory in your file system and then unmount it after use.

5. Prepare an initial version of the report in electronic form:
    - Title page, topic, and purpose of the work
    - Glossary of terms
    - Answers to points 4.1 - 4.5 from the tasks for preliminary preparation

Complete✔

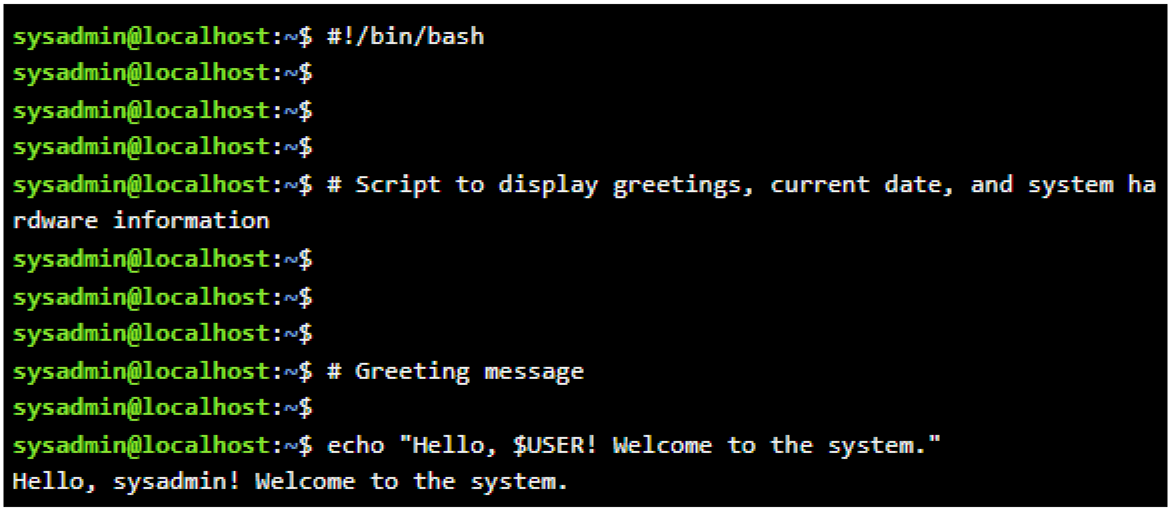
Progress of Work:

1. Initial work in CLI mode in a Linux operating system of the Linux family:
   1.1. Start your Linux-based operating system (if you are using your own PC and have it installed) and open the terminal.
2. Work through all the command examples provided in the lab assignments of the NDG Linux Essentials - Lab 11: Basic Scripting and Lab 12: Understanding Computer Hardware. Create a table to describe these commands.

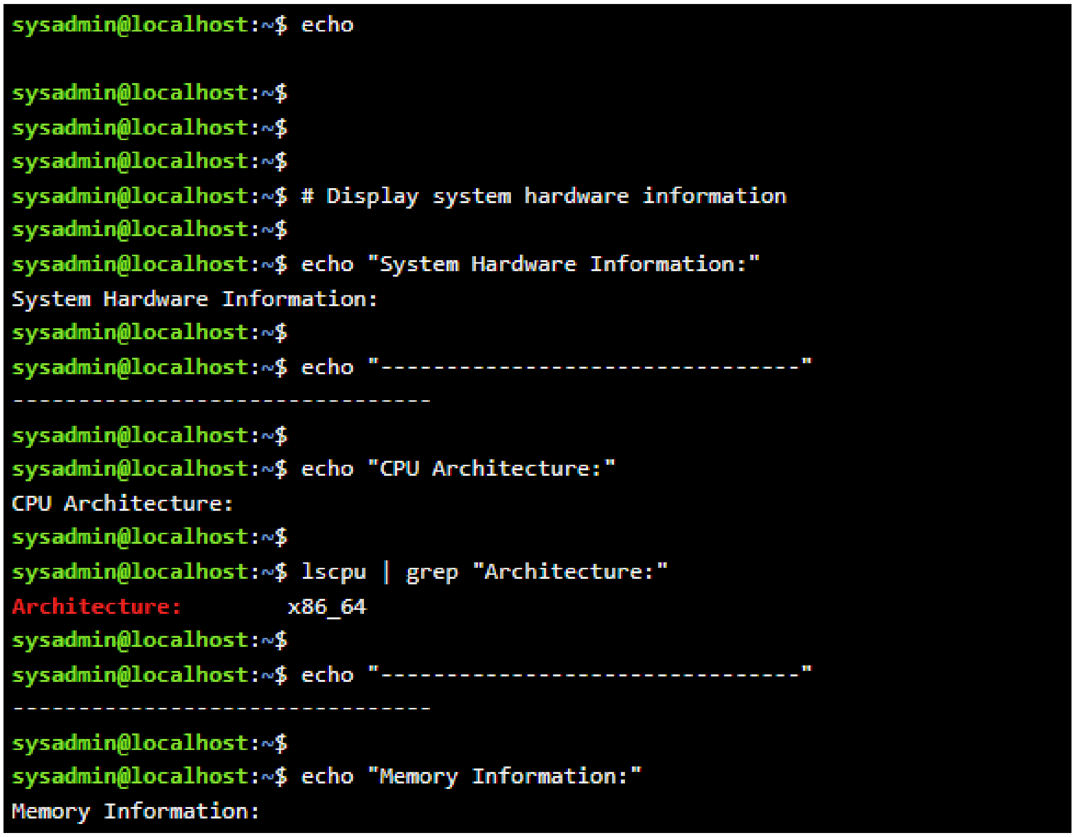
Command	Description
vi myfile	To create a new file
dw	Delete the word
u	Undo the last operation
2dw	Delete two words
xxxx	Delete four characters, one at a time

4u	Undo the last 4 operations and recover the deleted characters
14x	Delete 14 characters
dd	Delete the current line
p	Paste the deleted lines below the current line
yw	Copy (or “yank”) the current word
lspci	Display information about PCI buses and devices.
lsusb	Display information about USB buses and devices.
lscpu	Display information about CPU architecture.
lsblk	List block devices.
lsmod	Show the status of modules in the Linux Kernel.
hwinfo	Display detailed hardware information.

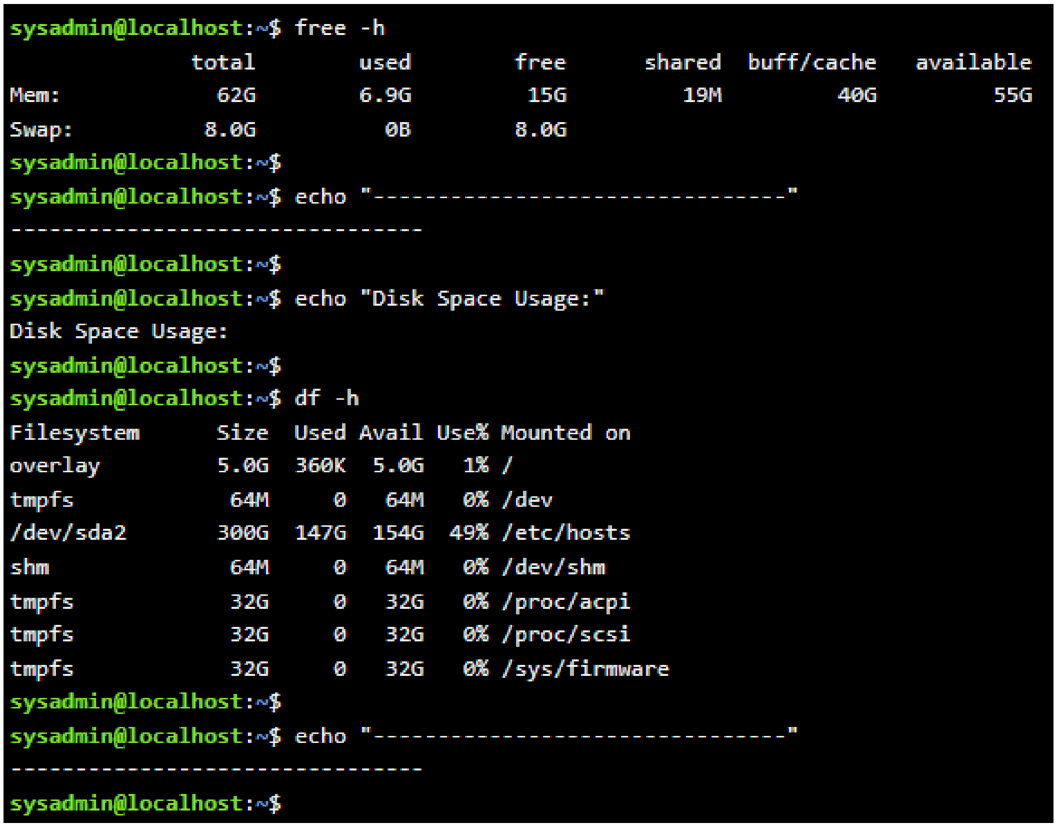
3. Create script scenarios with output of text messages for the user (demonstrate screenshots):
    - *The script should output a greeting to the current user indicating the current date and information about the current system.
    - *The script should output information about the hardware configuration of the current system (use the commands discussed in Lab 12: Understanding Computer Hardware).
    - **Provide your example of a script scenario.



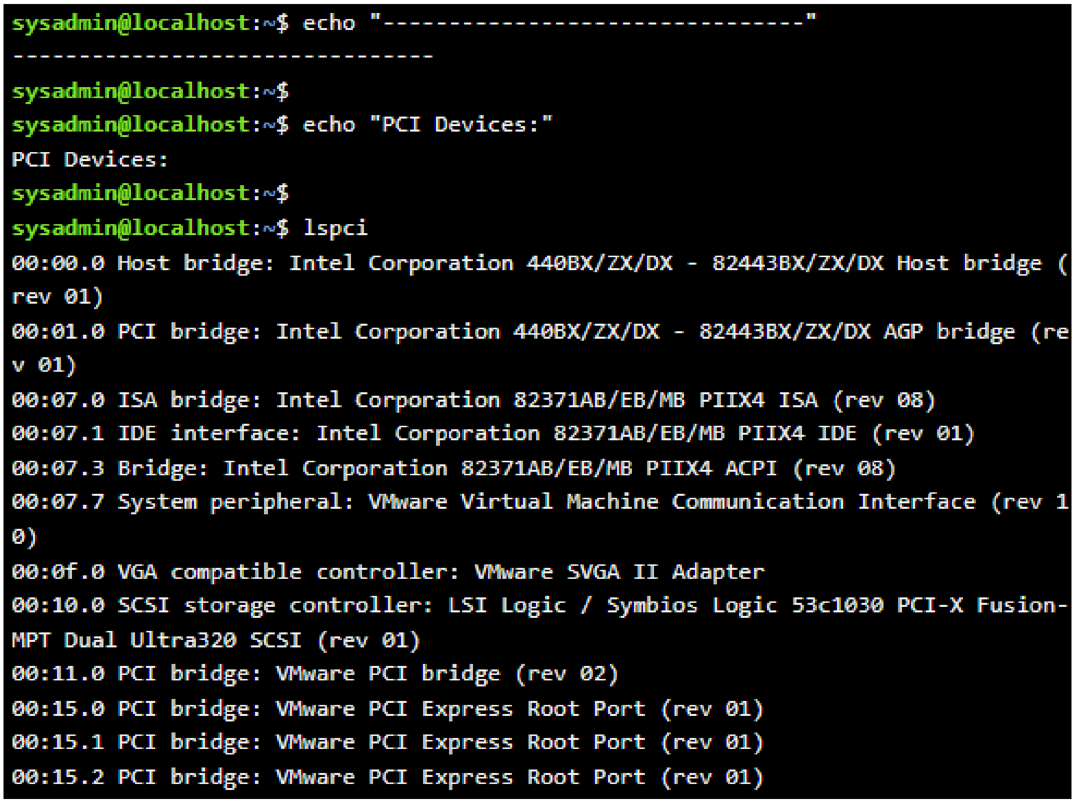
```
sysadmin@localhost:~$ #!/bin/bash
sysadmin@localhost:~$
sysadmin@localhost:~$
sysadmin@localhost:~$
sysadmin@localhost:~$ # Script to display greetings, current date, and system ha
rdware information
sysadmin@localhost:~$
sysadmin@localhost:~$
sysadmin@localhost:~$
sysadmin@localhost:~$ # Greeting message
sysadmin@localhost:~$
sysadmin@localhost:~$ echo "Hello, $USER! Welcome to the system."
Hello, sysadmin! Welcome to the system.
```



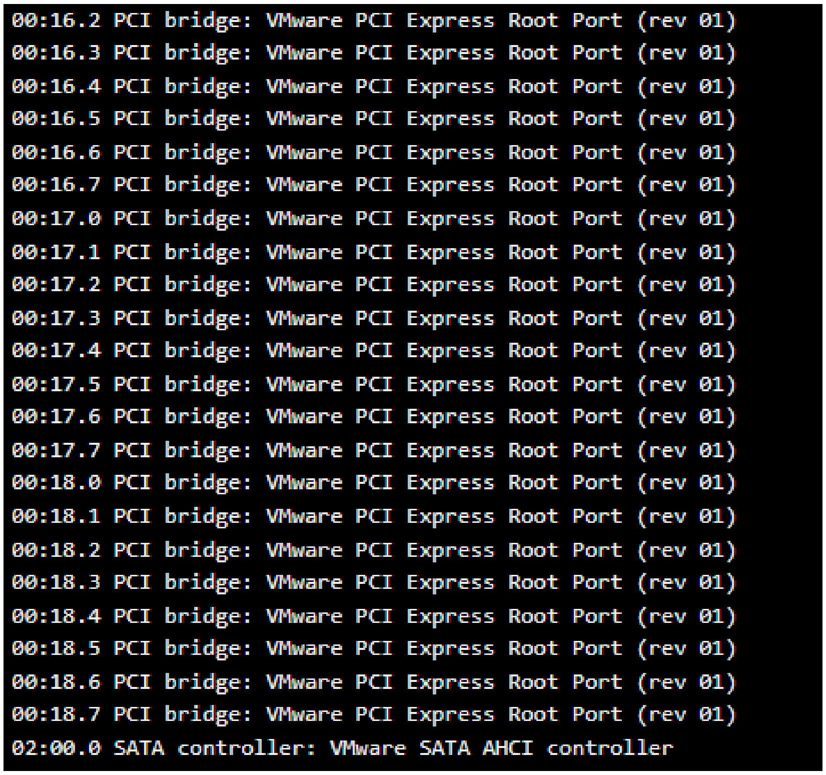
```
sysadmin@localhost:~$ echo

sysadmin@localhost:~$
sysadmin@localhost:~$
sysadmin@localhost:~$
sysadmin@localhost:~$ # Display system hardware information
sysadmin@localhost:~$
sysadmin@localhost:~$ echo "System Hardware Information:"
System Hardware Information:
sysadmin@localhost:~$
sysadmin@localhost:~$ echo "--------------------------------"
--------------------------------
sysadmin@localhost:~$
sysadmin@localhost:~$ echo "CPU Architecture:"
CPU Architecture:
sysadmin@localhost:~$
sysadmin@localhost:~$ lscpu | grep "Architecture:"
Architecture:        x86_64
sysadmin@localhost:~$
sysadmin@localhost:~$ echo "--------------------------------"
--------------------------------
sysadmin@localhost:~$
sysadmin@localhost:~$ echo "Memory Information:"
Memory Information:
```



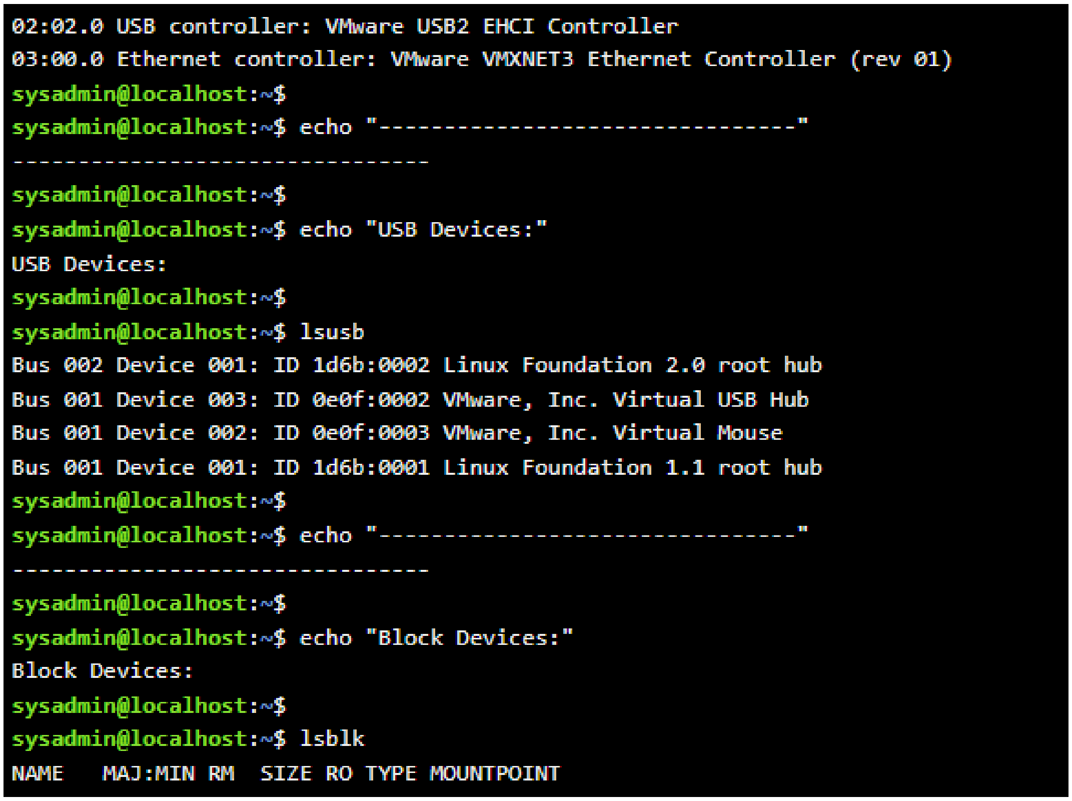
```
sysadmin@localhost:~$ free -h
              total        used        free      shared  buff/cache   available
Mem:            62G        6.9G         15G         19M         40G         55G
Swap:          8.0G          0B        8.0G
sysadmin@localhost:~$
sysadmin@localhost:~$ echo "--------------------------------"
--------------------------------
sysadmin@localhost:~$
sysadmin@localhost:~$ echo "Disk Space Usage:"
Disk Space Usage:
sysadmin@localhost:~$
sysadmin@localhost:~$ df -h
Filesystem      Size  Used Avail Use% Mounted on
overlay         5.0G  360K  5.0G   1% /
tmpfs            64M     0   64M   0% /dev
/dev/sda2       300G  147G  154G  49% /etc/hosts
shm              64M     0   64M   0% /dev/shm
tmpfs            32G     0   32G   0% /proc/acpi
tmpfs            32G     0   32G   0% /proc/scsi
tmpfs            32G     0   32G   0% /sys/firmware
sysadmin@localhost:~$
sysadmin@localhost:~$ echo "--------------------------------"
--------------------------------
sysadmin@localhost:~$
```



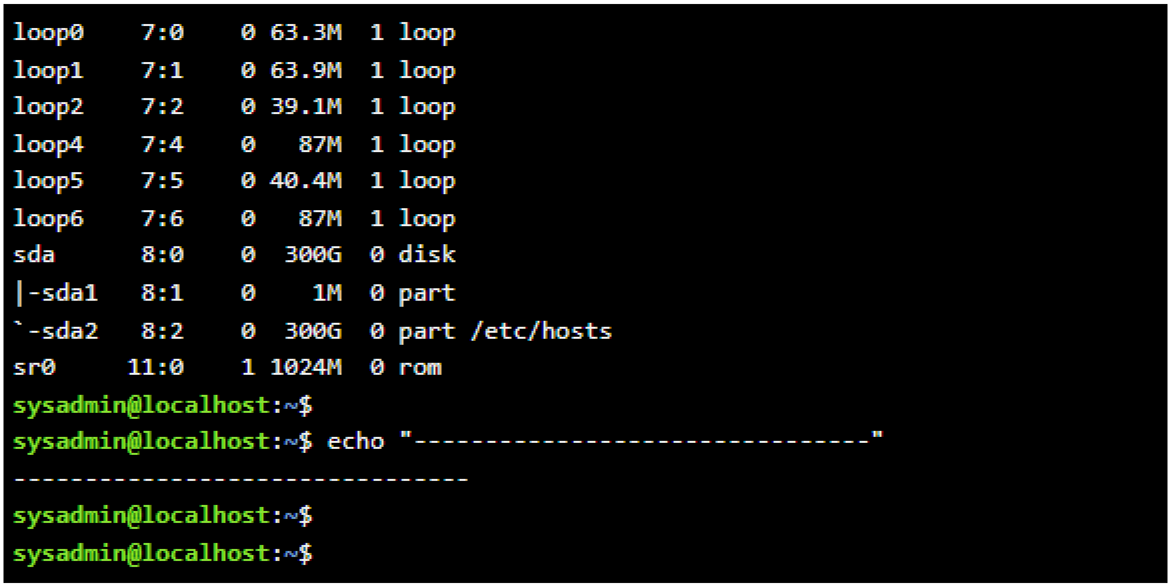
```
sysadmin@localhost:~$ echo "--------------------------------"
--------------------------------
sysadmin@localhost:~$
sysadmin@localhost:~$ echo "PCI Devices:"
PCI Devices:
sysadmin@localhost:~$
sysadmin@localhost:~$ lspci
00:00.0 Host bridge: Intel Corporation 440BX/ZX/DX - 82443BX/ZX/DX Host bridge (
rev 01)
00:01.0 PCI bridge: Intel Corporation 440BX/ZX/DX - 82443BX/ZX/DX AGP bridge (re
v 01)
00:07.0 ISA bridge: Intel Corporation 82371AB/EB/MB PIIX4 ISA (rev 08)
00:07.1 IDE interface: Intel Corporation 82371AB/EB/MB PIIX4 IDE (rev 01)
00:07.3 Bridge: Intel Corporation 82371AB/EB/MB PIIX4 ACPI (rev 08)
00:07.7 System peripheral: VMware Virtual Machine Communication Interface (rev 1
0)
00:0f.0 VGA compatible controller: VMware SVGA II Adapter
00:10.0 SCSI storage controller: LSI Logic / Symbios Logic 53c1030 PCI-X Fusion-
MPT Dual Ultra320 SCSI (rev 01)
00:11.0 PCI bridge: VMware PCI bridge (rev 02)
00:15.0 PCI bridge: VMware PCI Express Root Port (rev 01)
00:15.1 PCI bridge: VMware PCI Express Root Port (rev 01)
00:15.2 PCI bridge: VMware PCI Express Root Port (rev 01)
```



```
00:16.2 PCI bridge: VMware PCI Express Root Port (rev 01)
00:16.3 PCI bridge: VMware PCI Express Root Port (rev 01)
00:16.4 PCI bridge: VMware PCI Express Root Port (rev 01)
00:16.5 PCI bridge: VMware PCI Express Root Port (rev 01)
00:16.6 PCI bridge: VMware PCI Express Root Port (rev 01)
00:16.7 PCI bridge: VMware PCI Express Root Port (rev 01)
00:17.0 PCI bridge: VMware PCI Express Root Port (rev 01)
00:17.1 PCI bridge: VMware PCI Express Root Port (rev 01)
00:17.2 PCI bridge: VMware PCI Express Root Port (rev 01)
00:17.3 PCI bridge: VMware PCI Express Root Port (rev 01)
00:17.4 PCI bridge: VMware PCI Express Root Port (rev 01)
00:17.5 PCI bridge: VMware PCI Express Root Port (rev 01)
00:17.6 PCI bridge: VMware PCI Express Root Port (rev 01)
00:17.7 PCI bridge: VMware PCI Express Root Port (rev 01)
00:18.0 PCI bridge: VMware PCI Express Root Port (rev 01)
00:18.1 PCI bridge: VMware PCI Express Root Port (rev 01)
00:18.2 PCI bridge: VMware PCI Express Root Port (rev 01)
00:18.3 PCI bridge: VMware PCI Express Root Port (rev 01)
00:18.4 PCI bridge: VMware PCI Express Root Port (rev 01)
00:18.5 PCI bridge: VMware PCI Express Root Port (rev 01)
00:18.6 PCI bridge: VMware PCI Express Root Port (rev 01)
00:18.7 PCI bridge: VMware PCI Express Root Port (rev 01)
02:00.0 SATA controller: VMware SATA AHCI controller
```



```
02:02.0 USB controller: VMware USB2 EHCI Controller
03:00.0 Ethernet controller: VMware VMXNET3 Ethernet Controller (rev 01)
sysadmin@localhost:~$
sysadmin@localhost:~$ echo "--------------------------------"
--------------------------------
sysadmin@localhost:~$
sysadmin@localhost:~$ echo "USB Devices:"
USB Devices:
sysadmin@localhost:~$
sysadmin@localhost:~$ lsusb
Bus 002 Device 001: ID 1d6b:0002 Linux Foundation 2.0 root hub
Bus 001 Device 003: ID 0e0f:0002 VMware, Inc. Virtual USB Hub
Bus 001 Device 002: ID 0e0f:0003 VMware, Inc. Virtual Mouse
Bus 001 Device 001: ID 1d6b:0001 Linux Foundation 1.1 root hub
sysadmin@localhost:~$
sysadmin@localhost:~$ echo "--------------------------------"
--------------------------------
sysadmin@localhost:~$
sysadmin@localhost:~$ echo "Block Devices:"
Block Devices:
sysadmin@localhost:~$
sysadmin@localhost:~$ lsblk
NAME   MAJ:MIN RM  SIZE RO TYPE MOUNTPOINT
```



```
loop0    7:0    0 63.3M  1 loop
loop1    7:1    0 63.9M  1 loop
loop2    7:2    0 39.1M  1 loop
loop4    7:4    0   87M  1 loop
loop5    7:5    0 40.4M  1 loop
loop6    7:6    0   87M  1 loop
sda      8:0    0  300G  0 disk
|-sda1   8:1    0    1M  0 part
`-sda2   8:2    0  300G  0 part /etc/hosts
sr0     11:0    1 1024M  0 rom
sysadmin@localhost:~$
sysadmin@localhost:~$ echo "--------------------------------"
--------------------------------
sysadmin@localhost:~$
sysadmin@localhost:~$
```

Control questions

1. What is the difference between the commands 'arch' and 'lscpu'?\
   The "arch" command displays the machine architecture of the current system, while the "lscpu" command provides detailed information about the CPU architecture and its capabilities.
2. Which command can be used to obtain information about the usage of RAM by the current system?
   The "free" command can be used to obtain information about the RAM usage by the current system.
3. *How can variables be processed and branching and looping scenarios be created in scripts?
   Variables in scripts can be processed by assigning values to them using the assignment operator "=", and branching and looping scenarios can be created using conditional statements (if-else) for branching and loops like "for" or "while" for iteration.
4. *What commands can be used in the terminal to view the status of connected peripheral devices?

Commands such as "lsusb" and "lspci" can be used in the terminal to view the status of connected peripheral devices. "lsusb" lists USB devices, while "lspci" lists PCI devices.

5. **What are the capabilities of the 'gparted' utility?
   The "gparted" utility is a graphical partition editor that allows users to create, resize, delete, move, and manage disk partitions on a variety of filesystems. It provides a user-friendly interface for managing disk partitions and file systems.

Conclusions:

Shell scripting is a powerful tool for automating tasks and executing commands in a sequential or specific order. It allows users to streamline repetitive tasks and ensure consistency and efficiency in their workflows.
Understanding hardware components and system configurations is essential for effective system management and troubleshooting. Commands such as "arch" and "lscpu" provide valuable information about system architecture and CPU details, aiding in system analysis and optimization.
Manipulating variables and creating branching and looping scenarios in scripts expand the capabilities of shell scripting, enabling users to create more complex and versatile automation solutions tailored to their specific needs.
Monitoring peripheral device connections through terminal commands helps users manage hardware resources and diagnose connectivity issues promptly, ensuring smooth operation and optimal performance of peripheral devices.
The "gparted" utility offers comprehensive capabilities for disk partitioning and management, allowing users to create, resize, move, and delete partitions with ease. It provides a user-friendly interface for managing disk partitions efficiently, contributing to better system organization and storage management.