

Assessment 2 by abid hussain

by Abid Hussain

Submission date: 26-Jul-2023 11:26AM (UTC+0100)

Submission ID: 210697741

File name: 44005_Abid_Hussain_Assessment_2_by_abid_hussain_503754_443253824.docx (1.09M)

Word count: 3401

Character count: 17702

Title Page

¹Programme:	CertHE/BSc Applied Computing
Module:	Computer Architecture & Operating Systems
Module code:	ACCA4022
Lecturer(s):	Md Shantanu Islam Mobolanle Bello
Assignment Title:	Assessment-2 Operating Systems
STUDENT	ABID HUSSAIN 2225754

Contents

Introduction.....	3
1. Operating Systems and their key functions.....	3
Brief Introduction.....	3
Evolution of OS	3
Types of Operating System.....	4
Common Functions of Modern General Operating Systems	5
2. Use of CLI commands in a Windows / Mac OS.....	7
Examples of Commands	9
3. Use of CLI commands in a Linux OS.....	11
Examples.....	11
4. Use of Batch scripts in a Windows / Mac OS.....	13
Examples.....	14
5. Use of Bash scripts in a Linux OS.....	16
Examples.....	17
Conclusion	19
References.....	20

Introduction

This task examines the essential characteristics of Operating Systems (OS), which constitute the backbone of modern computing and are the subject of the paper's exploration. It recounts the history of operating systems from their early days to the present day, focusing on their crucial role in the management of hardware resources, user interactions, and the overall stability of the system. This paper covers a wide range of operating systems, including single-user, multi-user, real-time, distributed, and mobile operating systems. Additionally, it examines the usage of Command Line Interface (CLI) commands in operating platforms such as Windows, macOS, and Linux, in addition to the scripting capabilities of Windows batch files and Linux Bash scripts.

1. Operating Systems and their key functions

- What is an Operating System, brief introduction and evolution?

Brief Introduction

An Operating System (OS) is a piece of software that runs on a computer and acts as an intermediary between the hardware and the programmes used on that machine. In order to ensure the smooth and effective operation of a computer or other computing device, it maintains and organises the many system resources, such as the central processing unit (CPU), memory, storage, and input/output devices.

The kernel is the most fundamental part of a computer system, performing data processing at the hardware level. Controls for input/output, memory, and processes are all handled by it.

The shell is the topmost layer of an operating system and controls user input. The shell is responsible for interacting with the operating system, and it does so by reading commands from the user or a shell script. A shell script is a file containing a set of system commands (Deitel, Deitel, and Choffnes, 2004).

Evolution of OS

The story begins with the first operating systems, which were developed in the 1950s and 1960s. The article continues by highlighting the development of multiprogramming and timesharing systems in the 1960s and 1970s, which made it possible for numerous people to use a computer at once.

Moving into the 1980s, the article discusses the development of microprocessors and personal computers, which in turn led to the creation of graphical operating systems like Microsoft's

Disc Operating System (DOS) and Apple's Macintosh System Software (OS). In the 1990s, GUIs like those found in Microsoft Windows and Mac OS gained broad popularity.

Linux gained popularity in the server and development communities ¹⁷ in the late 1990s and 2000s, coinciding with the advent of network-centric operating systems. The 2010s heralded the arrival of mobile operating systems like Android and iOS, made specifically for portable electronic devices (Hansen, 2001).

¹⁶ Types of Operating System

The following is a list of the several types of operating systems that you should be familiar with:

Operating systems designed for ¹ a single user and a single task at a time are called "single-user, single-tasking" systems. They are common in embedded systems and antiquated computers.

These operating systems are designed for usage by a single user, but they have the capacity to handle several tasks or processes at once. They rapidly transition between activities, creating the impression that they are multitasking when in reality only one activity is being carried out at any given time.

Multi-user operating systems are able to accommodate numerous users at once. By logging in, each user can launch their own processes in isolation from one another.

The term "real-time operating system" refers to an OS that must process data and react to inputs in a very short amount of time. Robotics, industrial automation, and aerospace systems are only a few examples of popular uses where precision and speed are required.

In contrast to interactive systems, operating systems that perform work in batches are known as "batch processing" systems. Tasks are submitted to the system, run in order, and the results are generated automatically.

Time-sharing operating systems allow for numerous users at once by seamlessly switching between tasks, giving each user the impression of having the system to themselves. These operating systems are widely used in today's PCs.

Distributed operating systems are OSes that are built to function over a network of computers while giving the impression to users and programmes that they are accessing a single, unified resource. They make it easier for people to pool their resources and work together online.

OS Network: Operating systems designed specifically for use in a networked environment, managing data and providing services to a wide variety of connected devices and users. Common features include a centralised user management system, print services, and file sharing.

Smartphones, tablets, and other mobile devices require specialised operating systems called mobile operating systems (OS). In addition to enhancing functionality and reducing energy usage, they also provide intuitive touch-screen user interfaces.

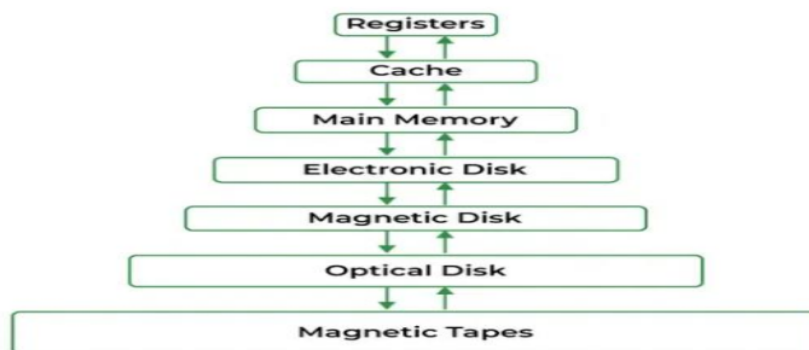
Virtualization operating systems, often known as hypervisors, allow numerous virtual machines (VMs) to run on a single physical machine while maintaining their own isolated environments.

Cloud operating systems, or Cloud OS, are OSes designed specifically for use with cloud computing platforms; they allow users and programmes to take use of the cloud's scalability and adaptability (Hansen, 1973).

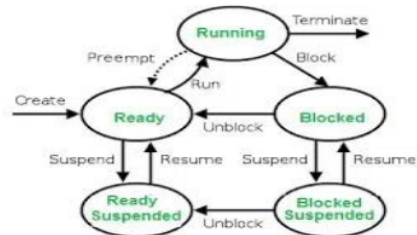
- 1. **Introduce and analyse the key and common functions of modern general Operating Systems.**

Common Functions of Modern General Operating Systems

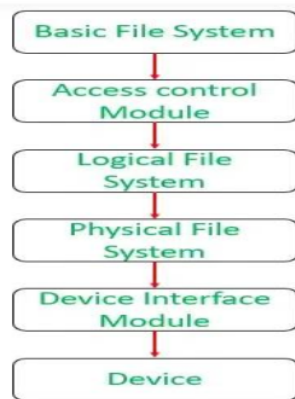
- Primary memory is a huge collection of data that can be accessed through its address, and it is managed by the operating system. It's a quick storage option that the CPU can go to right away. Operating systems handle memory allocation and deallocation, making sure that one program's memory is not used by another. It monitors primary memory, allots memory addresses, and controls processes in multiprogramming, among other memory management tasks.



- Processor management is another important function of the operating system, as it allocates jobs to the processor and ensures that each process receives enough time to function properly. The OS is also in charge of device management, which includes allocating and releasing resources for all connected devices.



- Information, user permissions, and file status are all managed through the operating system's file management system. The operating system (OS) is the software layer between the user and the computer's physical components.



- The OS is also in charge of user interfaces like command interpreters, system booting, and performance management. It keeps track of the amount of time it takes for the system to respond to service requests. The operating system (OS) is responsible for basic computer functions such as handling peripheral devices like the mouse, keyboard, and printer, as well as for work accounting, error-detecting assistance, coordination with other software and users.



- In addition to providing network connectivity and managing communication between computers on a network, network management is an integral part of any operating system. It also includes safety features like firewalls to prevent hacking and keep data private (Garrido, and Schlesinger, 2008).

1

2. Use of CLI commands in a Windows / Mac OS.

- **The Use of CLI in Windows / Mac OS**

The **CLI** relies on textual commands to carry out tasks. Certain requirements for syntax and line breaks mean that errors will be generated if you try to use these commands incorrectly.

In the case of command line interfaces included in application bundles, the commands may be specific to that bundle. But in terms of semantics and consistency, all CLIs are on the same line (Raybaut, 2009). The following are some basic commands that are used in a Windows / Mac OS.

Directory and File Listings

Use `dir` or `dir /b` (for a truncated list).

Directory Reorganization

Use `cd directory_path>`.

Making a List or Directory

Use `Mkdir directory_name>`

Getting Rid of Data

Use `del` or `erase`

Use `rm file_name>` in macOS

When a Directory Is Deleted

`rmdir directory_name>` or `rd directory_name>`

For Mac OS, use `rm -r "directory name."`

Data Duplication

Use cp source_file> destination_path>

File Renaming

Ren old_name> new_name>

In the Mac, use mv "old_name" "new_name."

Providing Access to System Details

Use Obtain-ComputerInfo

In Mac OS, use system_profiler

An IP or Website is Pinned

ping website_or_IP>

Verifying IP Addresses and Network Interfaces

Use ipconfig or ipconfig /all

In Mac OS, use ifconfig

- 1. Give at least two examples with brief explanation and show screenshots.

Examples of Commands

1. The IP configuration information of all network interfaces on the Windows system can be viewed by using the “**ipconfig command**”, which is used to show the information. It gives information about each network device, including its IP address, subnet mask, default gateway, and DNS servers, among other things. The screenshot below shows the command in detail.

```
Command Prompt
C:\Users\AA>ipconfig

Windows IP Configuration

Ethernet adapter Ethernet:

    Media State . . . . . : Media disconnected
    Connection-specific DNS Suffix  . :

Wireless LAN adapter Local Area Connection* 1:

    Media State . . . . . : Media disconnected
    Connection-specific DNS Suffix  . :

Wireless LAN adapter Local Area Connection* 2:

    Media State . . . . . : Media disconnected
    Connection-specific DNS Suffix  . :

Ethernet adapter VMware Network Adapter VMnet1:

    Connection-specific DNS Suffix  . :
    Link-local IPv6 Address . . . . . : fe80::cc3d:beb9:f563:6958%13
    IPv4 Address. . . . . : 172.18.16.1
    Subnet Mask . . . . . : 255.255.255.0
    Default Gateway . . . . . :

Ethernet adapter VMware Network Adapter VMnet8:

    Connection-specific DNS Suffix  . :
    Link-local IPv6 Address . . . . . : fe80::8d6e:14d0:76b1:c2c3%17
    IPv4 Address. . . . . : 192.168.127.1
    Subnet Mask . . . . . : 255.255.255.0
    Default Gateway . . . . . :

Wireless LAN adapter WiFi:

    Connection-specific DNS Suffix  . :
    Link-local IPv6 Address . . . . . : fe80::7256:2646:d08a:20c3%11
    IPv4 Address. . . . . : 192.168.118.142
    Subnet Mask . . . . . : 255.255.255.0
    Default Gateway . . . . . : 192.168.118.106
```

2. On a Windows computer, you can use the `dir` command to view a list of the files and folders that are located in the current directory. It offers details such as the file names, their sizes, and the timestamps at which they were created for the directories and files. The below is the screenshot of the command used in windows operating system.

```
Command Prompt
C:\Users\AA>dir
Volume in drive C has no label.
Volume Serial Number is 4E5F-4B6B

Directory of C:\Users\AA

02/07/2023  21:25    <DIR>          .
02/07/2023  21:25    <DIR>          ..
21/02/2023  12:13    <DIR>          .docker
19/03/2023  13:43    <DIR>          .eclipse
14/01/2023  07:23    <DIR>          .icesoft
19/03/2023  08:07    <DIR>          .matplotlib
19/05/2023  09:02             170 .packettracer
04/04/2023  09:30          34,901 .pdfbox.cache
09/05/2023  15:34           1,243 .qtenvrc
25/07/2023  07:50    <DIR>          .VirtualBox
04/01/2023  10:05    <DIR>          3D Objects
19/05/2023  09:03    <DIR>          Cisco Packet Tracer 8.2.0
04/01/2023  10:05    <DIR>          Contacts
08/07/2023  12:04    <DIR>          Desktop
23/07/2023  01:10    <DIR>          Documents
23/07/2023  23:57    <DIR>          Downloads
04/01/2023  10:05    <DIR>          Favorites
05/01/2023  07:11    <DIR>          GNS3
04/01/2023  10:05    <DIR>          Links
04/01/2023  10:05    <DIR>          Music
25/07/2023  07:07    <DIR>          OneDrive
21/04/2023  18:48    <DIR>          Pictures
04/01/2023  10:05    <DIR>          Saved Games
04/01/2023  10:07    <DIR>          Searches
23/03/2023  18:02    <DIR>          Videos
25/07/2023  07:50    <DIR>          VirtualBox VMs
02/07/2023  21:25             0 whoami
          4 File(s)          36,314 bytes
          23 Dir(s)  60,225,019,904 bytes free

C:\Users\AA>
```

3. AAUse¹ of CLI commands in a Linux OS.

- The Use of CLI Commands Using a Linux OS

Through the use of the Terminal on a computer running the Linux operating system, it is possible to communicate with the Command Line Interface (CLI). Users are able to accomplish a wide variety of responsibilities in a time-efficient manner thanks to Linux's robust and flexible command-line environment (Sobell, 2013). The following is a list of some of the fundamental CLI commands in Linux:

- The "ls" command is what is used to generate a list of the files and directories that are located in the working directory at the moment.
- The "cd" command is used to navigate to a different directory than the one that is currently being utilized.
- It is possible to determine the IP address of the machine by using the "ifconfig" program.
- When you want to create a new directory, you use the "mkdir" command.
- The "rm" command can be used to delete (remove) directories or files from a computer.
- To copy files or directories, you can use the "cp" command on your command line.
- The "pwd" command is what is used to display the complete path of the working directory that is currently being used.
- Sending ICMP echo request packets is what the "ping" command does when it's being used to test whether or not a network is connected to a certain host or IP address.
- The "ps" command is what is used to provide information regarding the processes that are currently operating.

- Give at least two examples with brief explanation and show screenshots.

Examples

1. Linux users can customize and view information about their system's many network interfaces by using the ifconfig command. The term "interface configuration" refers to the function that enables users to examine and make changes to the settings of a network interface. These parameters include IP addresses, netmasks, broadcast addresses, and more. The below screens shot show the command in detail.

- 1
2. The `ls` command is what is used to generate a list of the files and directories that are located in the working directory at the moment. It gives information on the names of files, their sizes, and the permissions associated with each file. The screenshot below shows the command in detail.

1

4. Use of Batch scripts in a Windows / Mac OS.

- Document the use of batch scripts using a Windows / Mac OS.

Mac operating systems utilizes shell scripts, written in a language like Bash, which is a Unix shell. Shell scripts can be generated with a text editor such as nano or Visual Studio Code and have the extension.sh at the end of the filenames (Robbins, et al, 2010). The following is a simple illustration of a shell script for MAC operating system:

```
#!/bin/bash
# This is my first macOS shell script

# Display the message
echo "Hello, World!"

# List files in the current directory
ls

# Pause the script to see the output
read -p "Press Enter to continue..."
```

#!/bin/bash: This is referred to as a shebang, because it identifies the shell that will be used to parse the script (/bin/bash in this instance).

This sign, referred to as the hash mark (#), denotes a comment within the shell script.

echo is a command that causes text to be displayed on the Terminal.

ls is a command that lists the files and directories that are located in the directory that is currently being worked in.

read: Before continuing with the execution of the script, this command awaits further instruction from the user.

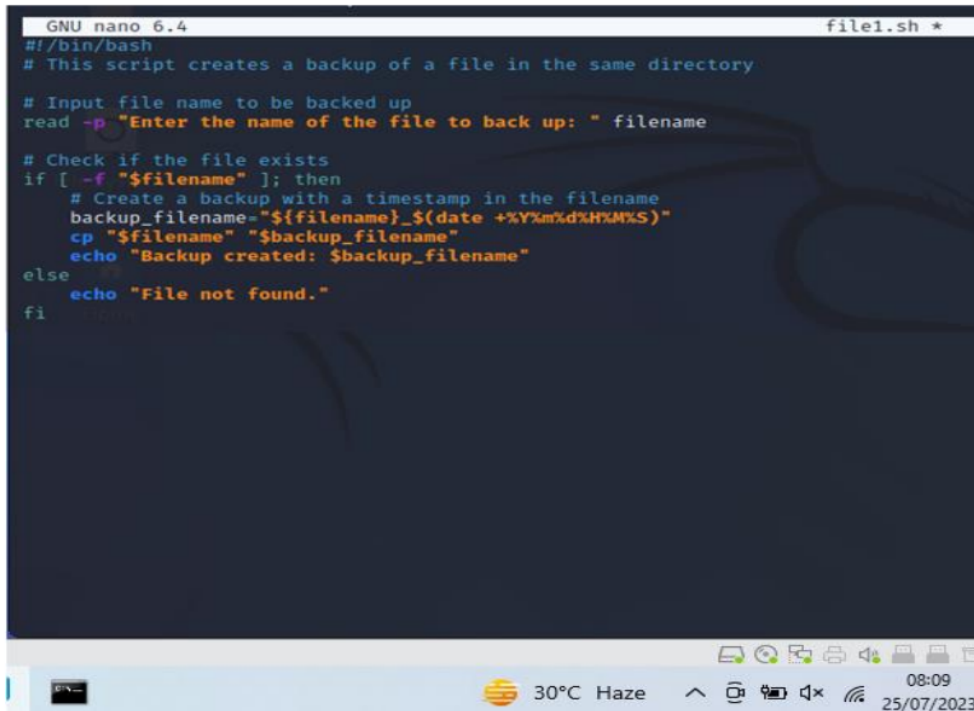
Execution of a Shell Script in macOS:

The script should be saved with a.sh extension (for example, myscript.sh), then the Terminal should be opened, navigated to the path of the script, and the chmod command should be used to make it executable.

- 1 Give at least two examples with brief explanation and show screenshots.

Examples

1. An Easy Script to Back Up Files: The below screenshot represent a shell script that is used to backup files.



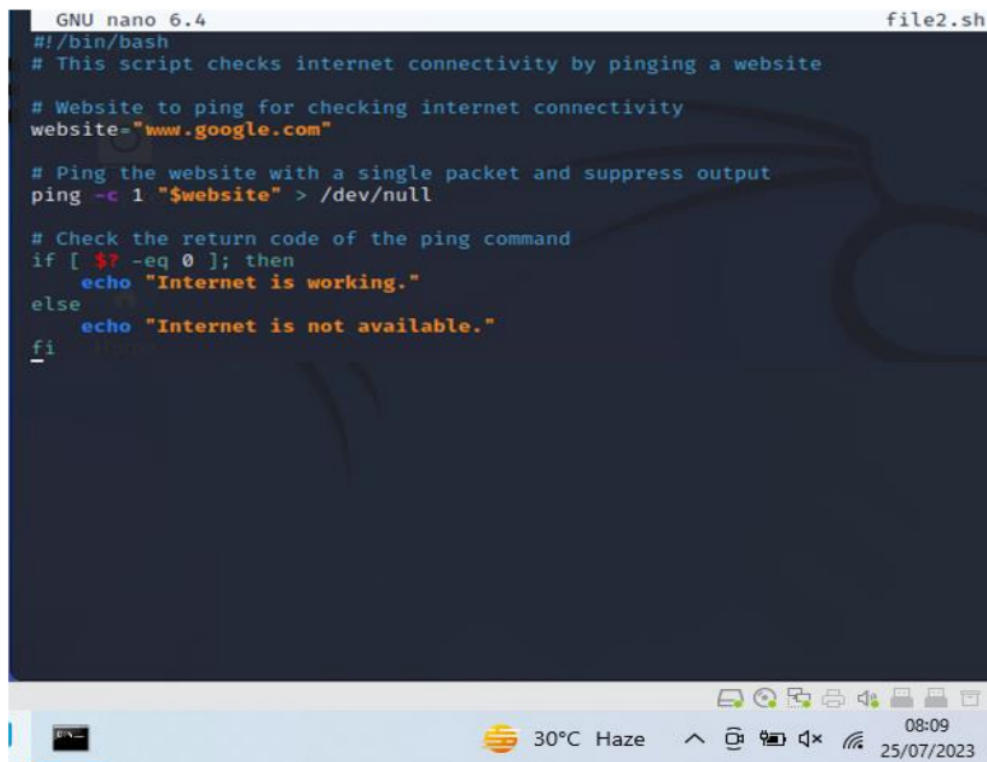
```
GNU nano 6.4 file1.sh *
#!/bin/bash
# This script creates a backup of a file in the same directory

# Input file name to be backed up
read -p "Enter the name of the file to back up: " filename

# Check if the file exists
if [ -f "$filename" ]; then
    # Create a backup with a timestamp in the filename
    backup_filename="${filename}_${date +%Y%m%d%H%M%S}"
    cp "$filename" "$backup_filename"
    echo "Backup created: $backup_filename"
else
    echo "File not found."
fi
```

- `#!/bin/bash`: This is the whole enchilada, calling upon the shell (`/bin/bash`) to carry out the script's instructions.
- `read -p`: User input is saved in the `filename` variable when using this command. If `[-f "$filename"]`: The existence of the input file is verified using an if statement (`-f` verifies a standard file).
- `restore_filename="$(date +%Y%m%d%H%M%S)"`: In this line, the current time and date are appended to the original filename, creating a variable called `backup_filename` with the format `YYYYMMDDHHMMSS`.
- `cp` For example: `"$filename" "$backup_filename"` Using the backup filename, this command will make a duplicate of the source file.
- `echo`: This command causes the Terminal to show text.

2. Connectivity Test through: The screenshot below represents a sample shell script that is used to check network connectivity.



The screenshot shows a terminal window with the title bar 'GNU nano 6.4' and 'file2.sh'. The script content is as follows:

```
#!/bin/bash
# This script checks internet connectivity by pinging a website

# Website to ping for checking internet connectivity
website="www.google.com"

# Ping the website with a single packet and suppress output
ping -c 1 "$website" > /dev/null

# Check the return code of the ping command
if [ $? -eq 0 ]; then
    echo "Internet is working."
else
    echo "Internet is not available."
fi
```

The terminal window has a taskbar at the bottom showing system icons, temperature (30°C), weather (Haze), time (08:09), and date (25/07/2023).

- `#!/bin/bash` for the Internet: This is the whole enchilada, calling upon the shell (`/bin/bash`) to carry out the script's instructions.
- `website="www.google.com"`: This line assigns the target website's URL to the variable `website`.
- To see if a website is reachable online, use the ping command: `ping -c 1 "$website" > /dev/null`. With `-c 1`, only one packet is transmitted, and `> /dev/null`, all output is muted.
- The value of the `?` special variable is the exit status of the previous command run. Any value other than 0 indicates a failure.
- If the ping operation was successful (return code 0), as determined by the conditional statement `if [ $? -eq 0 ]`; then, the next step will be taken.
- `echo`: This Terminal command prints forth text.

1

5. Use of Bash scripts in a Linux OS.

- Document the use of bash scripts using a Linux OS.

In most Linux versions, the default shell is Bash, hence shell scripts written in this language are known as Bash scripts. You can use these scripts to automate a wide range of Linux-based processes, from executing commands to carrying out specific tasks. Linux's Bash scripting language allows for fast task automation, configuration management, and the execution of a wide variety of system functions. If your scripts perform activities that could change or delete data, you should always proceed with extreme caution and check them multiple times. Be sure your scripts are doing what they're supposed to by testing them extensively (Blum, 2008). This section details how to run Bash scripts in Linux.

Making a Bash Program

- Launch your preferred text editor and make a new file with the .sh extension (for example, myscript.sh) and save it.

Primitive Syntax

- To add comments to the script, use the # symbol. The interpreter will not process comments; however, they can be helpful in understanding the code.
- Variables: In both variable declarations and usage (e.g., name="John"), no spaces should be present surrounding the = sign. The \$ symbol can be substituted for the variable's name to retrieve its value (e.g., echo \$name).
- Substituting a Command: To run a command and store its output, surround the command with backslashes (') or \$(), as in result=\$(ls) or result='ls'.
- Predicate Clauses: Conditional command execution can be achieved with the help of if, else, and fi.
- For looping through lists or continuing an action until a certain condition is met, you can use the for, while, or until statements.

Bash script execution

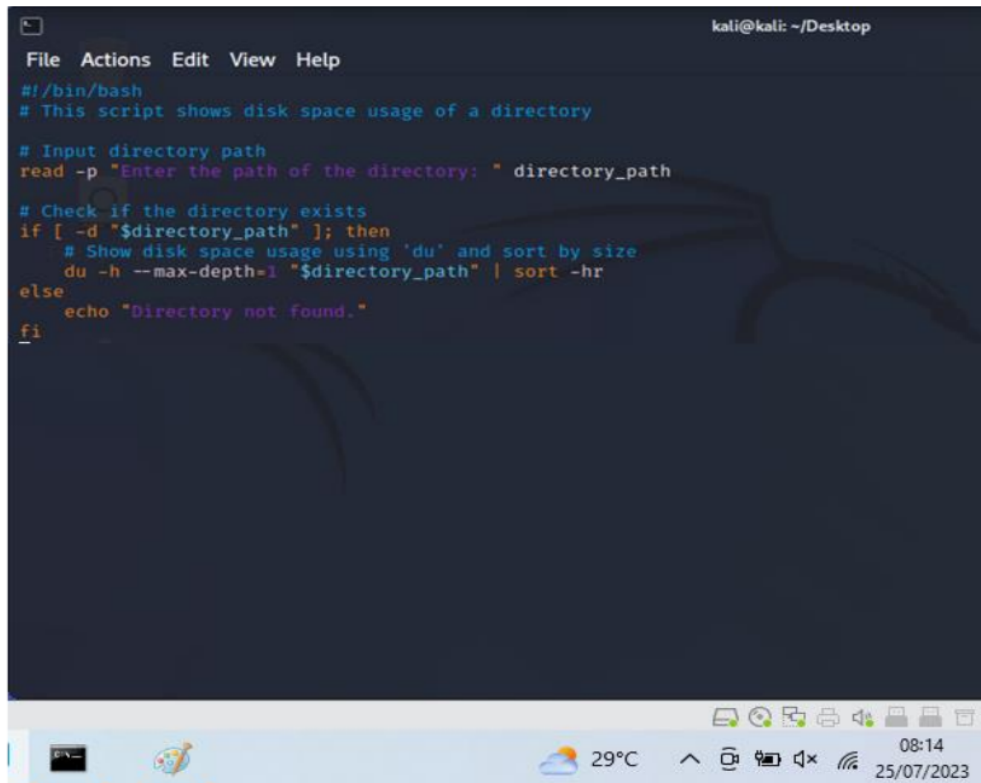
- Launch the Terminal, go to the script's directory, and run the script by typing its filename.

- 1. Give at least two examples with brief explanation and show screenshots.

Examples

1. Use of Disk Space

The below script lists the size-ordered disk consumption of the current directory and its subdirectories.



```
kali@kali: ~/Desktop
File Actions Edit View Help
#!/bin/bash
# This script shows disk space usage of a directory

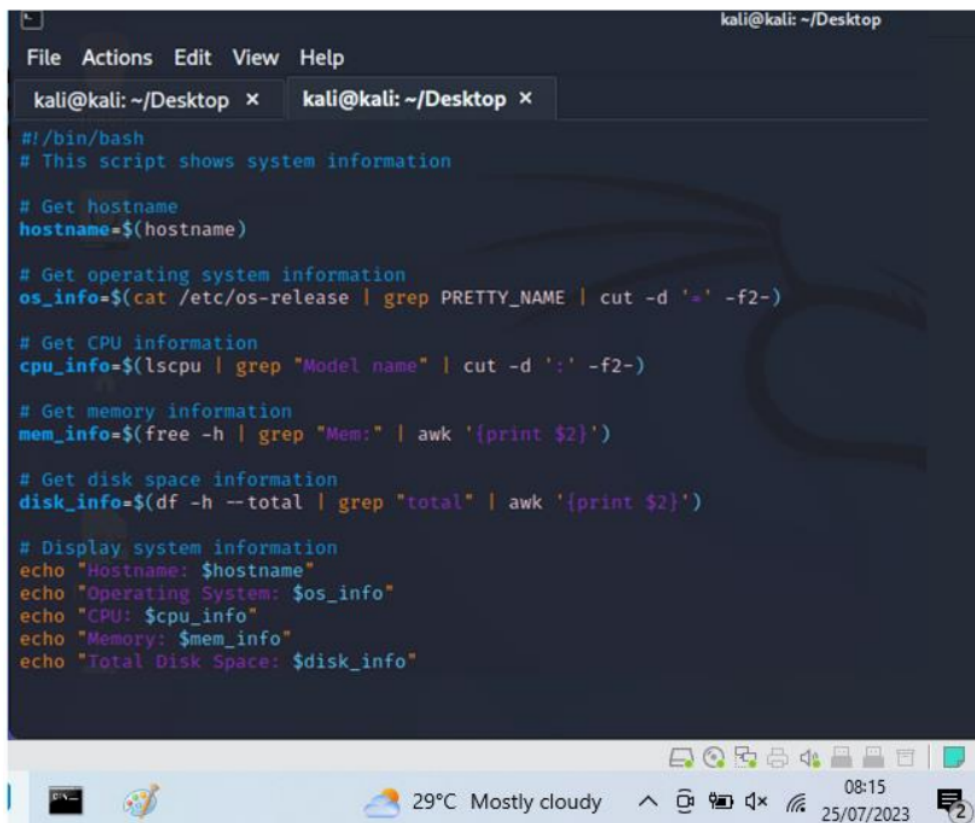
# Input directory path
read -p "Enter the path of the directory: " directory_path

# Check if the directory exists
if [ -d "$directory_path" ]; then
    # Show disk space usage using 'du' and sort by size
    du -h --max-depth=1 "$directory_path" | sort -hr
else
    echo "Directory not found."
fi
```

- The script will ask for a directory path when run.
- When determining how much space a given directory and its immediate subdirectories take up on disk, the du command is used if the directory already exists. If you choose the --max-depth=1 option, only the first level of results will be displayed.
- Using the sort command's -hr parameters, the output is formatted for human consumption and sorted from largest to smallest.

2. Details About the System

- This script shows basic system statistics such as the hostname, OS, CPU, RAM, and disk space.
- The script utilizes Linux commands to retrieve system data.
- The hostname command displays the current hostname of the system.
- Information on the operating system can be retrieved using the `cat /etc/os-release` command, and then the `grep` and `cut` commands can be used to extract the necessary data.
- The `lscpu` tool displays CPU data, while the `grep` and `cut` commands allow you to narrow down the list of possible models.
- RAM consumption can be viewed with `free -h`, and the overall size of RAM can be extracted with `grep` and `awk`.
- Disk utilization can be viewed with `df -h --total`, and the total disk space can be retrieved with the `grep` and `awk` tools.

A screenshot of a Kali Linux desktop environment. The top panel shows the window title 'kali@kali: ~/Desktop' and a menu bar with 'File', 'Actions', 'Edit', 'View', and 'Help'. Below the menu bar, there are two tabs, both labeled 'kali@kali: ~/Desktop'. The main area of the window is a terminal with a dark background and light-colored text. The terminal shows a script for displaying system information. The script starts with a shebang line '#!/bin/bash' and a comment '# This script shows system information'. It then defines variables for hostname, operating system information, CPU information, memory information, and disk space information. Finally, it displays the system information using the 'echo' command. The desktop environment at the bottom shows a taskbar with various icons, including a weather icon showing '29°C Mostly cloudy', a clock showing '08:15', and a date showing '25/07/2023'. There is also a notification icon with the number '2'.

Conclusion

This task investigates the fundamental principles of Operating Systems (OS) as well as the practical uses of Command Line Interface (CLI) scripting in settings running Windows, macOS, and Linux. Operating systems operate as middlemen between computer hardware and computer applications. Their responsibilities include resource management, user interface creation, and ensuring that the system runs smoothly. They are essential to the operation of any computing environment because they make interaction between users more seamless and improve resource consumption. CLI commands offer users a way to communicate with the operating system that is both straightforward and powerful. As a result, users are able to do a wide variety of activities in an effective manner. Linux provides users with a command-line environment that is both strong and adaptable, giving them the ability to easily control files, directories, processes, and networking.

Batch scripts offer a straightforward method to automate work by executing a series of commands in succession. This helps to simplify operations that are repetitive and streamline tasks that are more complex. Within a Linux system, bash scripts, which are written in a scripting language that is native to Linux, provide a method that is both versatile and reliable for increasing productivity and automating operations. This paper, in its whole, offers a complete overview of operating systems as well as the practical applications of command line scripting across a variety of platforms.

6

References

3

Deitel, H.M., Deitel, P.J. and Choffnes, D.R., 2004. Operating systems. Pearson/Prentice Hall.

Hansen, P.B., 2001. The evolution of operating systems. In Classic operating systems: from batch processing to distributed systems (pp. 1-34). New York, NY: Springer New York.

3

4

Hansen, P.B., 1973. Operating system principles. Prentice-Hall, Inc..

Garrido, J.M. and Schlesinger, R., 2008. Principles of modern operating systems. Jones & Bartlett Learning.

9

Raybaut, P., 2009. Spyder-documentation. Available online at: pythonhosted.org.

5

Sobell, M.G., 2013. A practical guide to Linux commands, editors, and shell programming. Prentice Hall.

2

Robbins, L., Hansen, M.E., Kleypas, J.A. and Meylan, S.C., 2010. CO2calc: A user-friendly seawater carbon calculator for Windows, Mac OS X, and iOS (iPhone) (p. 17). US Department of the Interior, US Geological Survey.

1

Blum, R., 2008. Linux command line and shell scripting bible (Vol. 481). John Wiley & Sons.

Assessment 2 by abid hussain

ORIGINALITY REPORT

16%

SIMILARITY INDEX

7%

INTERNET SOURCES

3%

PUBLICATIONS

13%

STUDENT PAPERS

PRIMARY SOURCES

1

Submitted to University of Wales, Lampeter

Student Paper

10%

2

Submitted to City University

Student Paper

1%

3

mts.intechopen.com

Internet Source

1%

4

Submitted to Edith Cowan University

Student Paper

<1%

5

Submitted to University of Plymouth

Student Paper

<1%

6

Submitted to Liverpool John Moores University

Student Paper

<1%

7

www.tafe.sa.edu.au

Internet Source

<1%

8

patents.google.com

Internet Source

<1%

9

academic.oup.com

Internet Source

<1%

10	idoc.pub Internet Source	<1 %
11	www.coursehero.com Internet Source	<1 %
12	A.F.B. Abu Kasim, M.S.C. Chan, E.J. Marek. "Performance and failure analysis of a retrofitted Cessna aircraft with a Fuel Cell Power System fuelled with liquid hydrogen", Journal of Power Sources, 2022 Publication	<1 %
13	lists.ovirt.org Internet Source	<1 %
14	Submitted to Pathfinder Enterprises Student Paper	<1 %
15	ebin.pub Internet Source	<1 %
16	ipfs.io Internet Source	<1 %
17	link.springer.com Internet Source	<1 %

Exclude quotes Off

Exclude matches Off

Exclude bibliography Off