

| **TITLE: Exploring basic Commands of UNIX: Shell, Processes, Files** |
| --- |

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**AIM:** To Explore basic commands for handling File system under Unix/Linux using shell scripts.(Creating groups, chown , chmod , directory name, tty , diff, umask).

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**Expected Outcome of Experiment:**

**CO 1.** To introduce basic concepts and functions of operating systems.

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**Books/ Journals/ Websites referred:**

1. **Silberschatz A., Galvin P., Gagne G. “Operating Systems Principles”, Willey Eight edition.**
2. **Achyut S. Godbole , Atul Kahate “Operating Systems”, McGraw Hill Third Edition.**
3. **Sumitabha Das “ UNIX Concepts & Applications”, McGraw Hill Second**

**Edition.**

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**Pre Lab/ Prior Concepts:**

An operating system (OS) is a resource manager. It takes the form of a set of software routines that allow users and application programs to access system resources (e.g. the CPU, memory, disks, modems, printers network cards etc.) in safe efficient and abstract way.

* The operating system kernel is in direct control of the underlying hardware. The kernel provides low-level device, memory and processor management functions (e.g. dealing with interrupts from hardware devices, sharing the processor among multiple programs, allocating memory for programs etc.)
* Basic hardware-independent kernel services are exposed to higher-level programs through a library of system calls (e.g. services to create a file, begin execution of a program, or open a logical network connection to another computer).
* Application programs (e.g. word processors, spreadsheets) and system utility programs (simple but useful application programs that come with the operating system, e.g. programs which find text inside a group of files) make use of system calls. Applications and system utilities are launched using a shell (a textual command line interface) or a graphical user interface that provides direct user interaction.

Operating systems can be distinguished from one another by the system calls, system utilities and user interface they provide, as well as by the resource scheduling policies implemented by the kernel.

UNIX has been a popular OS for more than two decades because of its multi-user, multi-tasking environment, stability, portability and powerful networking capabilities.

Linux is a free open source UNIX OS for PCs.

Linux has all of the components of a typical OS :

* **Kernel**

The Linux kernel includes device driver support for a large number of PC hardware devices (graphics cards, network cards, hard disks etc.), advanced processor and memory management features, and support for many different types of file systems. In terms of the services that it provides to application programs and system utilities, the kernel implements most BSD and SYSV system calls, as well as the system calls described in the POSIX.1 specification.

The kernel (in raw binary form that is loaded directly into memory at system startup time) is typically found in the file /boot/vmlinuz, while the source files can usually be found in /usr/src/linux.

* **Shells and GUIs**

Linux supports two forms of command input: through textual command line shells similar to those found on most UNIX systems (e.g. sh - the Bourne shell, bash - the Bourne again shell and csh - the C shell) and through graphical interfaces (GUIs) such as the KDE and GNOME window managers.

* **System Utilities**

Virtually every system utility that you would expect to find on standard implementations of UNIX has been ported to Linux. This includes commands such as ls, cp, grep, awk, sed, bc, wc, more, and so on. These system utilities are designed to be powerful tools that do a single task extremely well (e.g. grep finds text inside files while wc counts the number of words, lines and bytes inside a file). Users can often solve problems by interconnecting these tools instead of writing a large monolithic application program.

* **Application programs**

Linux distributions typically come with several useful application programs as standard. Examples include the emacseditor, xv (an image viewer), gcc (a C compiler), g++ (a C++ compiler), xfig (a drawing package), latex (a powerful typesetting language) and soffice (StarOffice, which is an MS-Office style clone that can read and write Word, Excel and PowerPoint files).

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Description of Commands and options:

**DOS commands:** Attrib, dir, at, chkdsk, shutdown, tree, create a batch file, output and input redirection

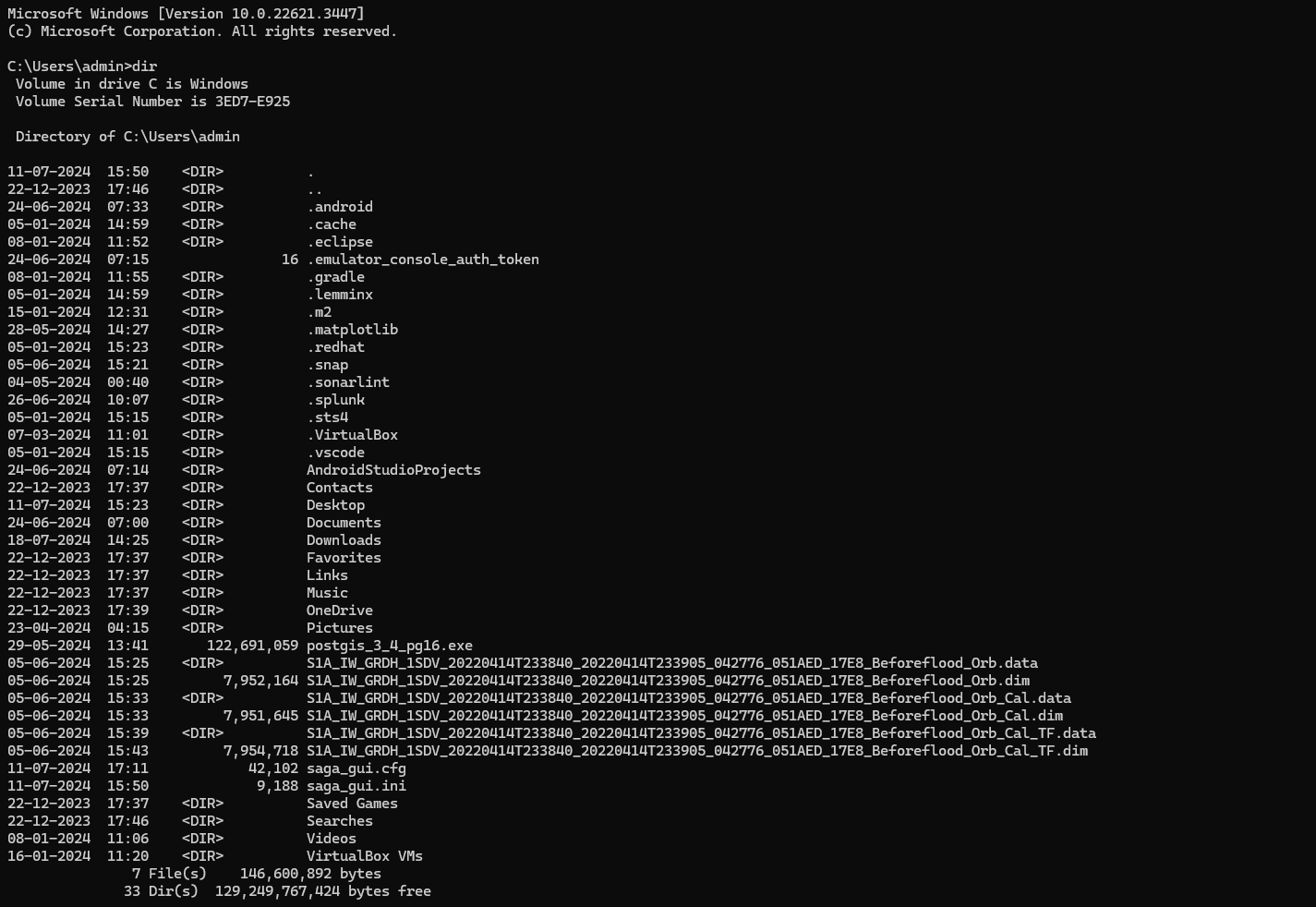
**Windows utilities**: msconfig, defragmenter, performance monitor, task manager, registry editor, event viewer, process explorer

Unix Commands:

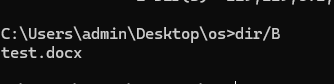
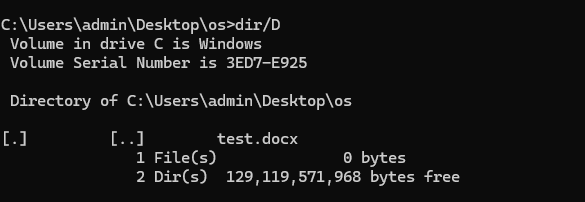
1. Unix file operations: ls, cp, rm , mv, chmod, chown ,chgrp
2. Text file operations in Unix : cat , more , less , head, tail , grep
3. Unix directory management commands : cd, pwd , ln, mkdir, rmdir
4. Unix system status commands: hostname, w, uname
5. Process management: ps, top, kill
6. Unix users commands: whoami , id, groups, passwd , who, last

**Implementation details:**

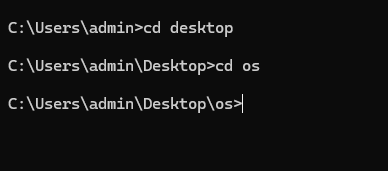
**Basic and Important DOS (Windows Command Prompt) Commands :**

1. dir-- List contents of a directory

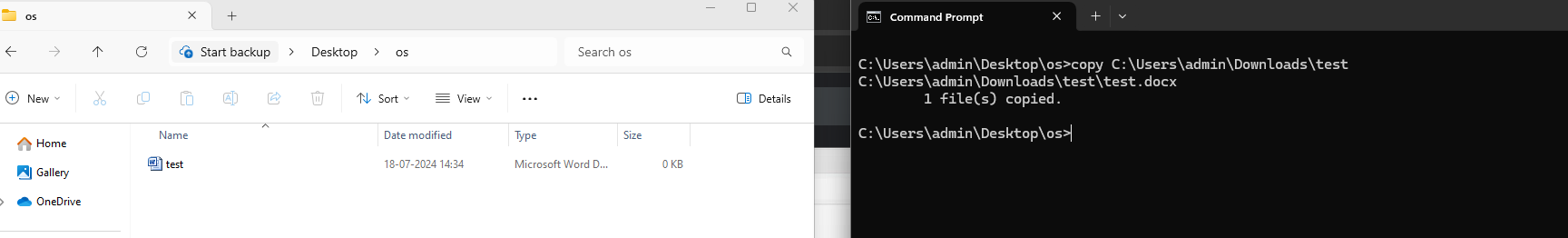
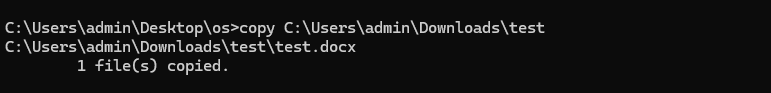
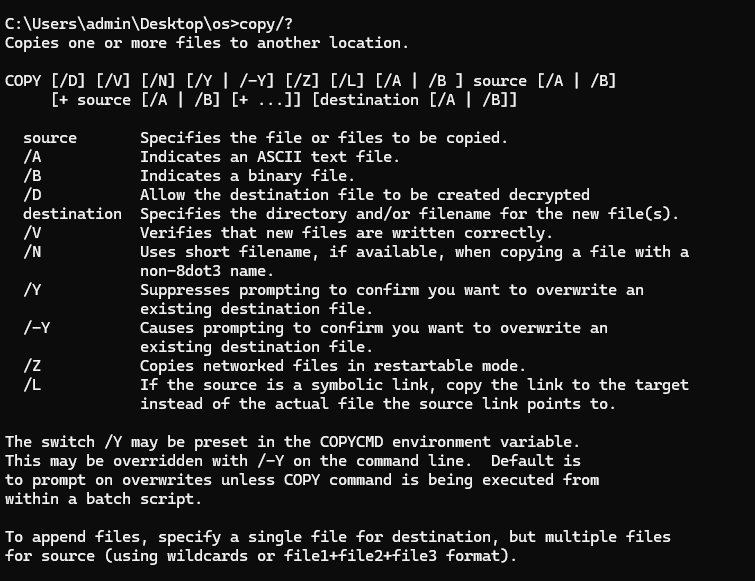




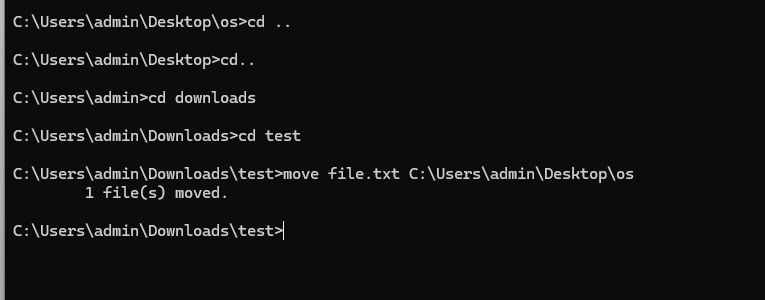
1. cd - Change directory.

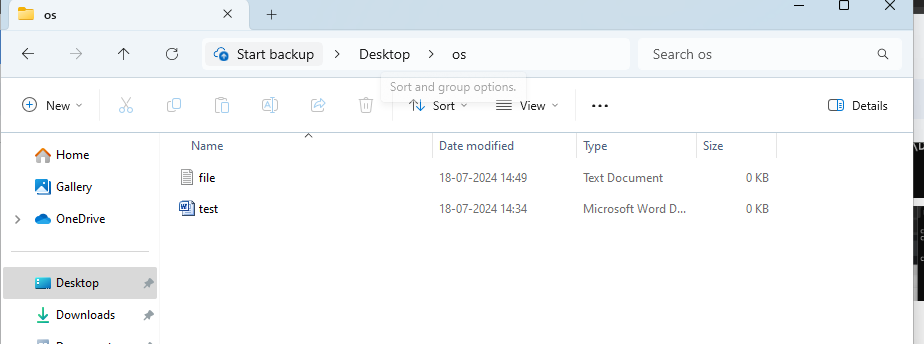
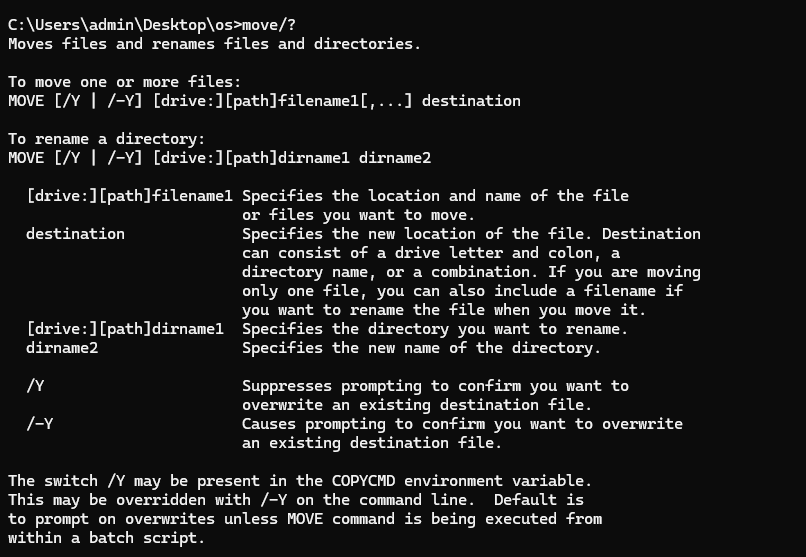


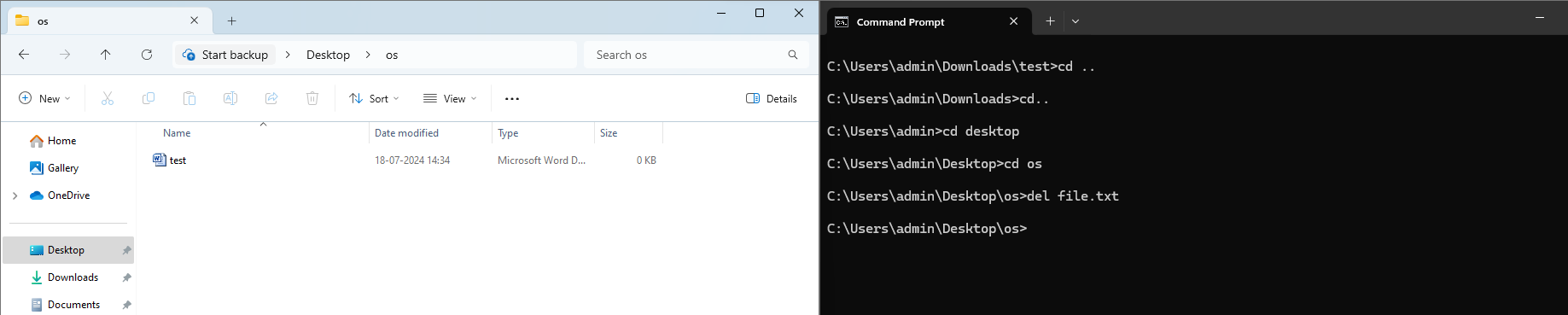
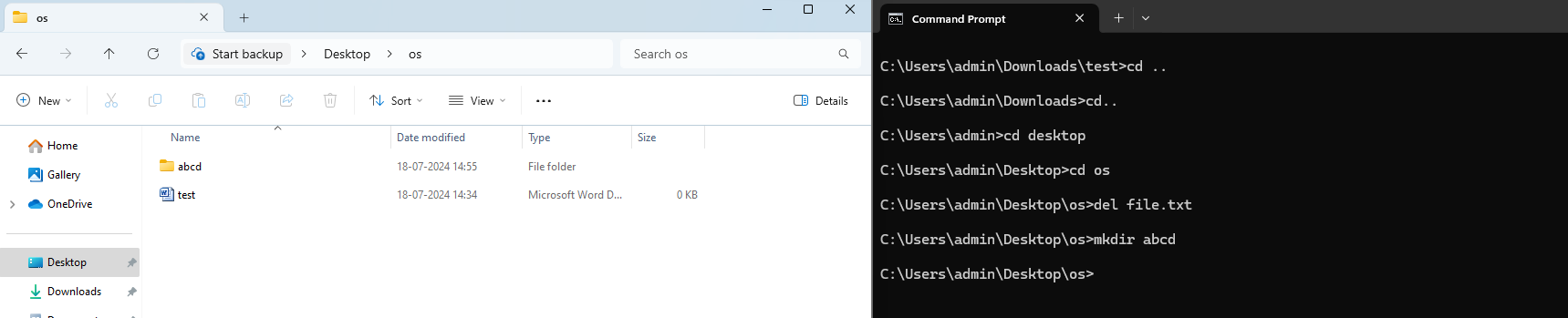
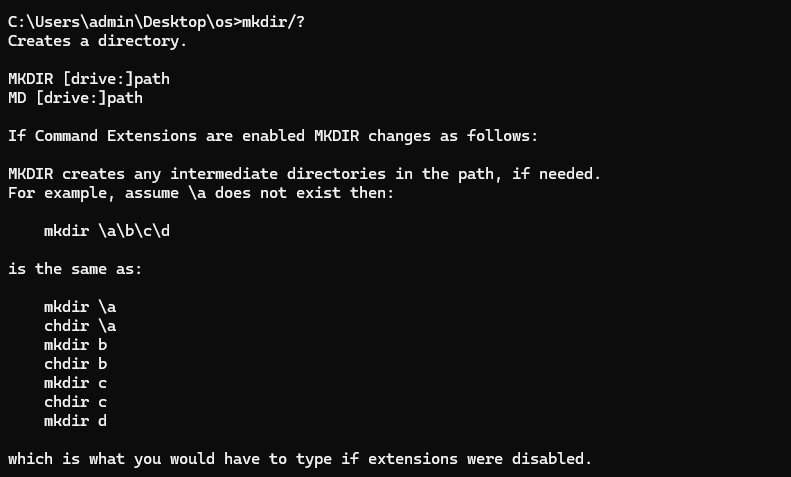
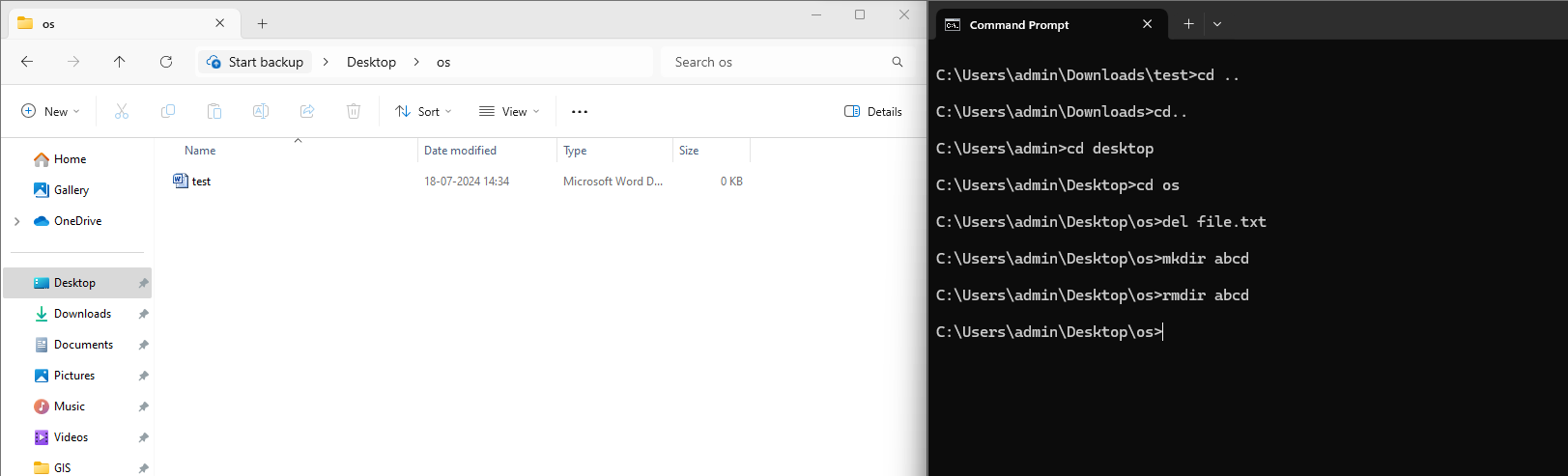
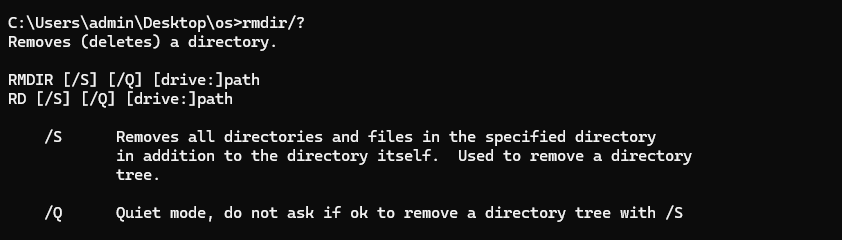
1. copy - Copy files.

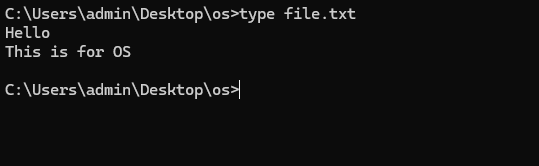
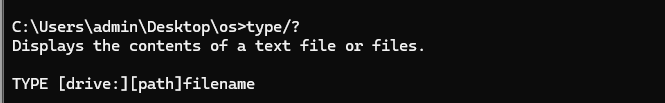
  
  
  
  


1. move - Move files.

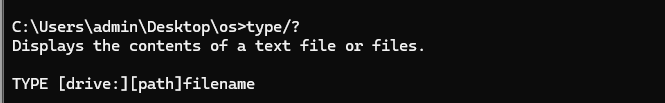


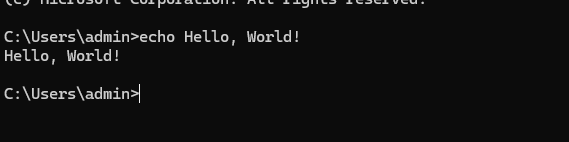
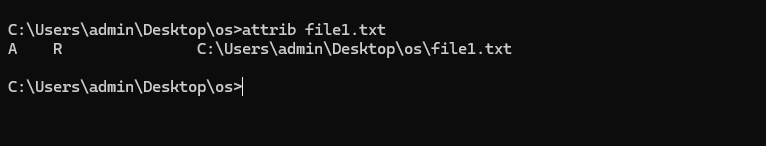
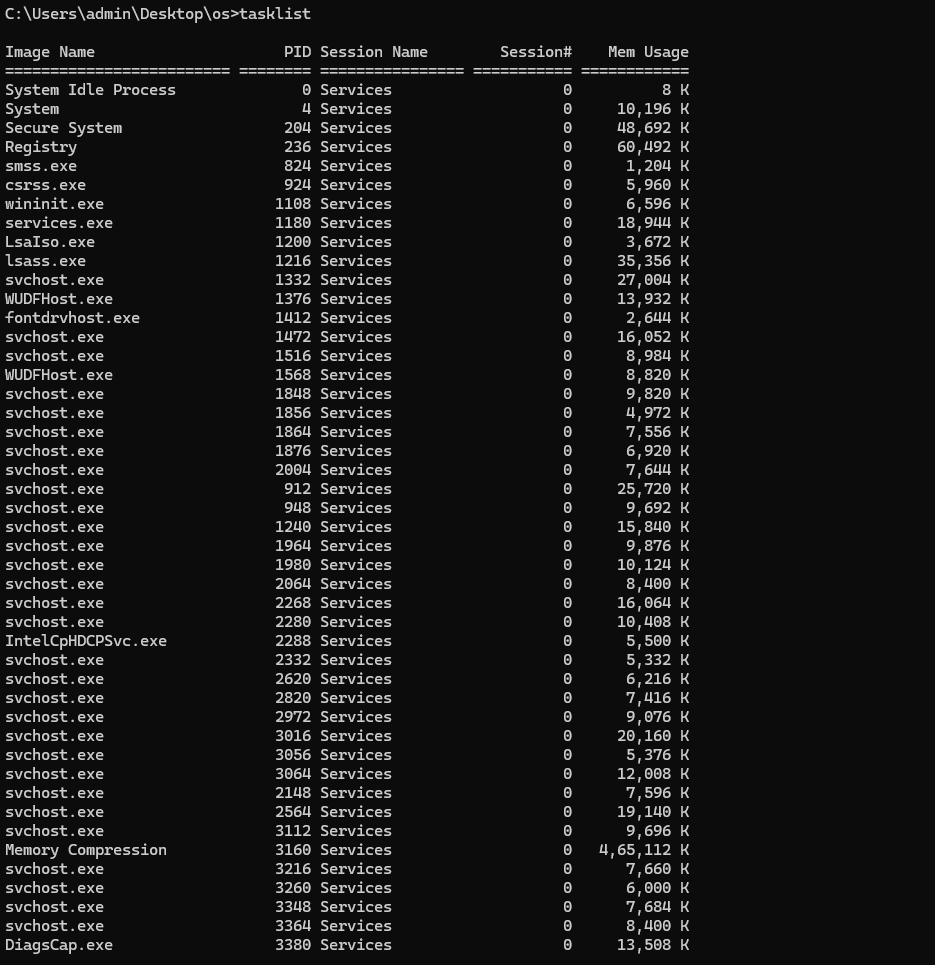
1. del - Delete files.  
   
2. mkdir - Create a new directory  
   
3. rmdir - Remove a directory  
   
4. type - Display the contents of a file

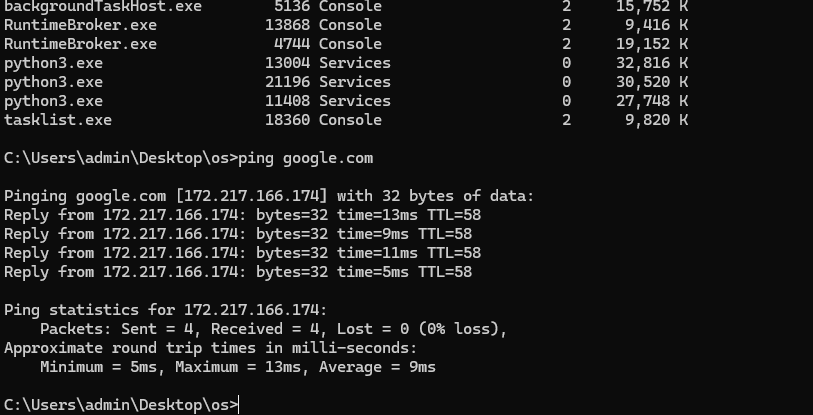
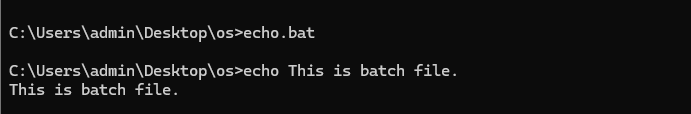
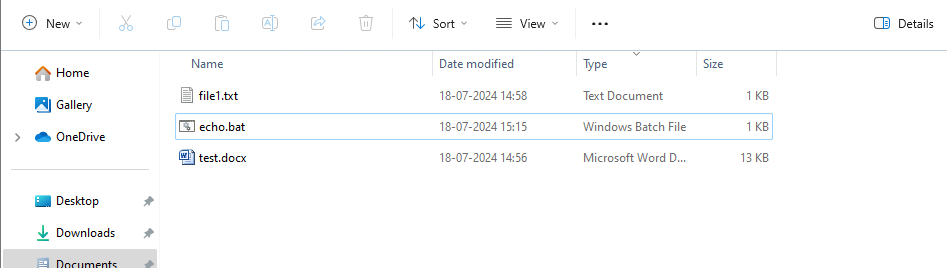
  


1. ren - Rename files.

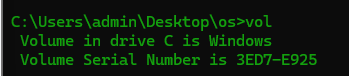
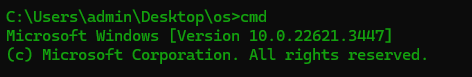
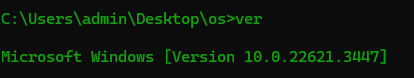


1. cls - Clear the screen.

  
  
  
  
11) echo - Display messages or turn command echoing on/off.  
  
  
12) exit - Exit the Command Prompt.  
  
13) attrib - Display or change file attributes.  
  
14) tasklist - Display a list of currently running processes

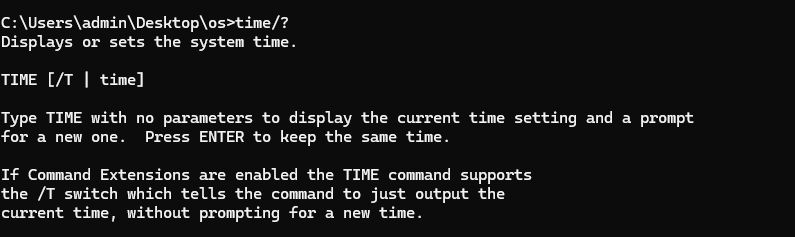
15) ping - Test connectivity to another networked device using ICMP echo.   
16) Create a batch file and execute   
  
  
  
17) Color2

  
18) Vol

  
19) Cmd  
  
20) Ver  
  


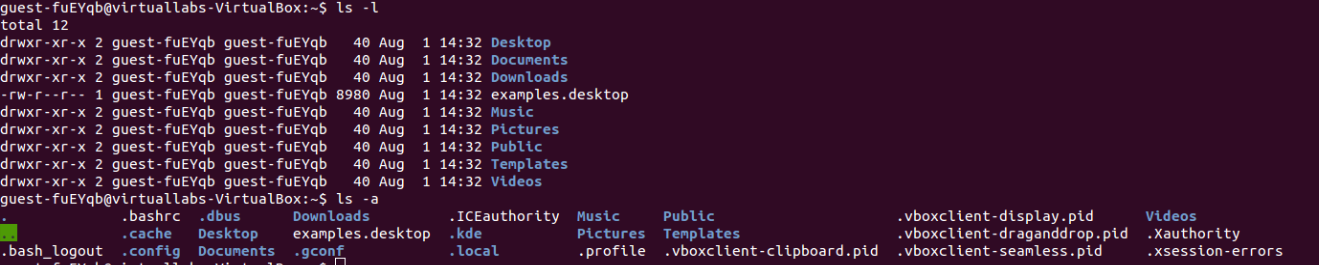
21) Help



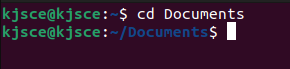
22) Time  
  


Basic and Important Unix (Linux/macOS) Commands:

1. ls - List directory contents.



2. cd - Change directory.



3. cp - Copy files and directories.



4. mv - Move/rename files and directories.



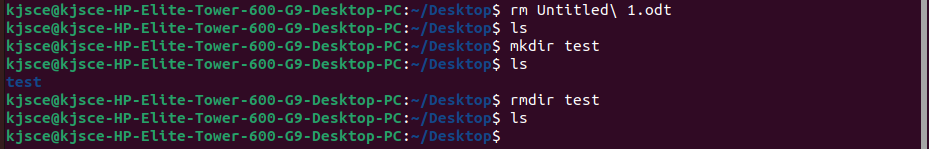
5. rm - Remove files and directories.



6. mkdir - Create directories.



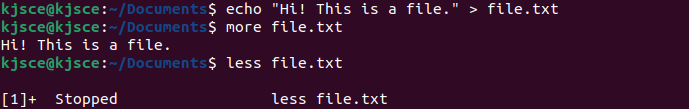
7. rmdir - Remove empty directories.

8. cat - Concatenate and display files.



9. more or less - View file contents one screen at a time.

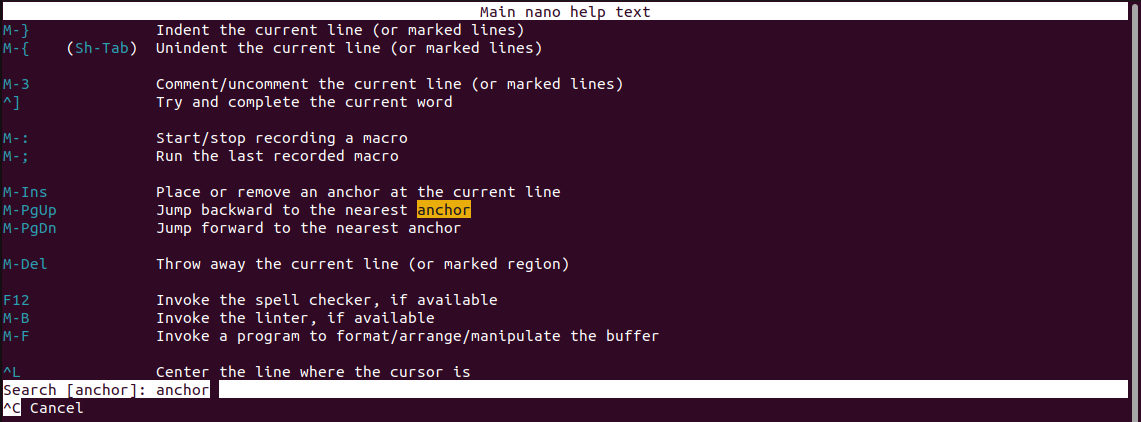


10. touch - Create an empty file or update file timestamps.



11. nano or vi (Linux) / TextEdit (macOS) - Text editors for creating and editing Files.



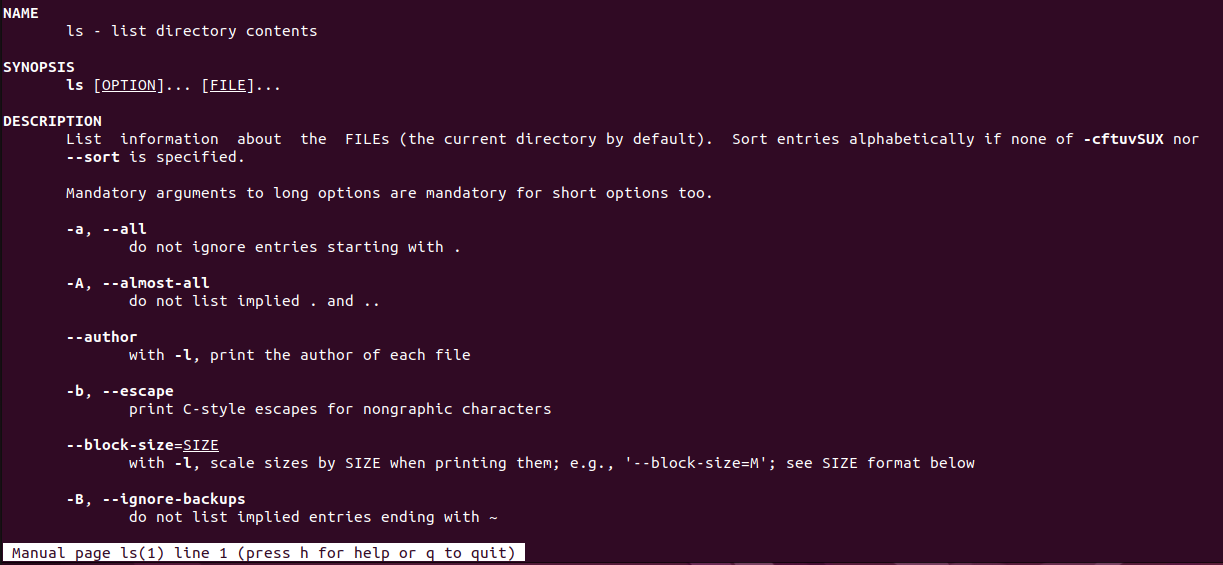


12. pwd - Print working directory.

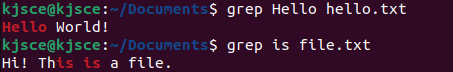


13. man - Display manual pages for commands.





14. grep - Search for patterns in files.



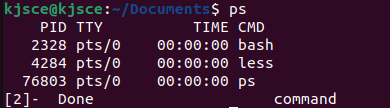
15. chmod - Change file permissions.



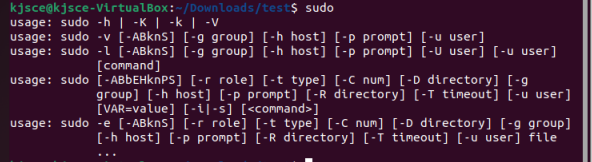
16. Run a command in background

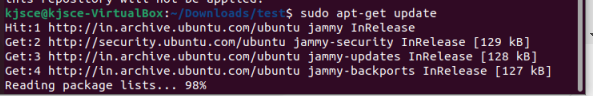
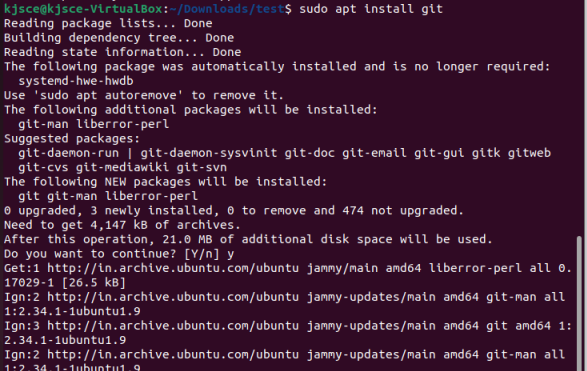
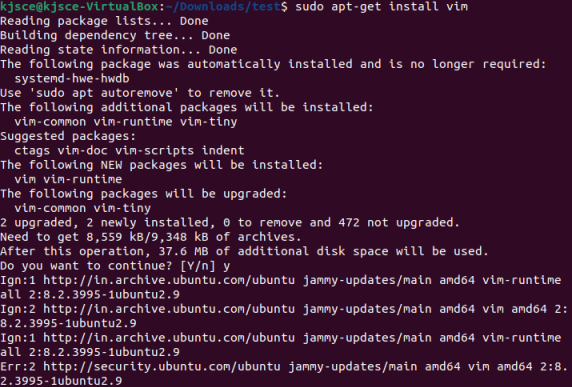


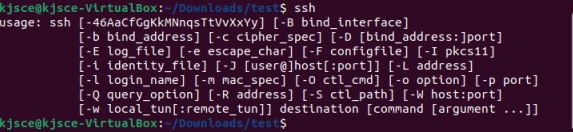
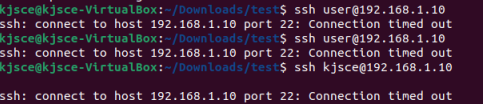
17. Suspend some commands and re-schedule them (Ctrl-z, ps, fg %sequence no of suspended process)

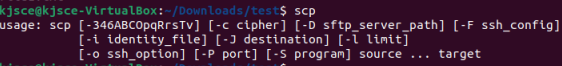


Important Unix Commands (Advanced):

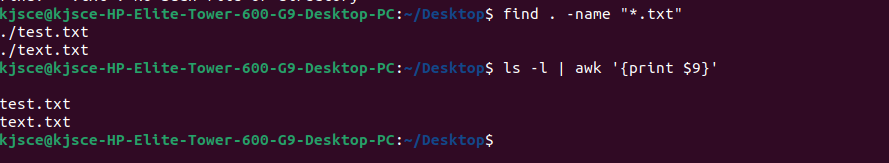
1. sudo - Execute a command with superuser (root) privileges.  
  


2. apt-get (Linux) / brew (macOS) - Package management commands for installing and managing software packages.  
  
  
  


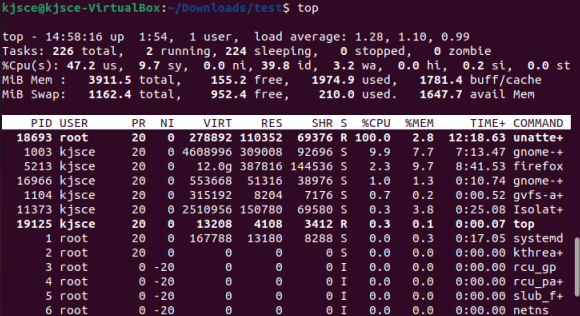
3. ssh - Secure Shell for remote login and command execution.  
  


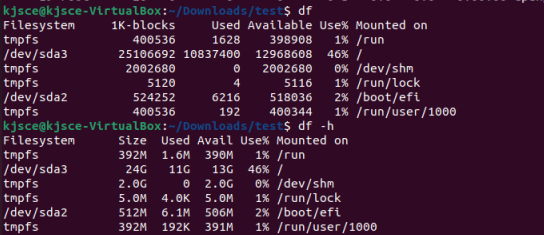
4. scp - Securely copy files between machines.  


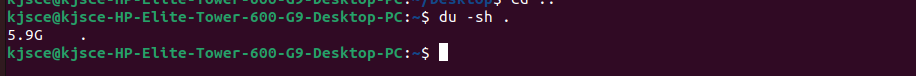
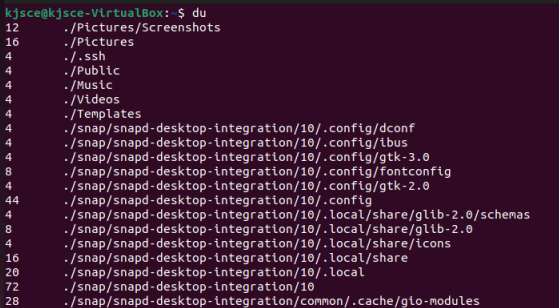
5. find - Search for files in a directory hierarchy.  

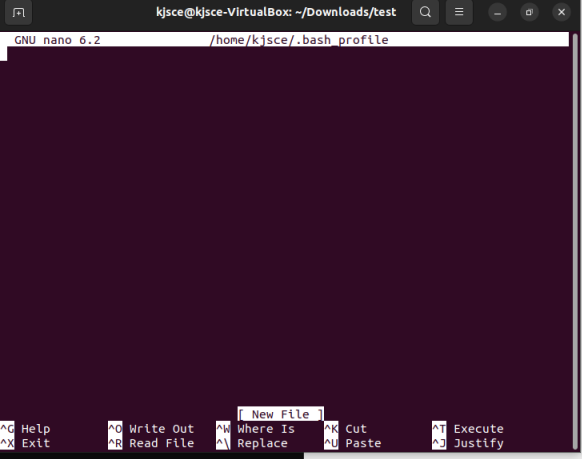
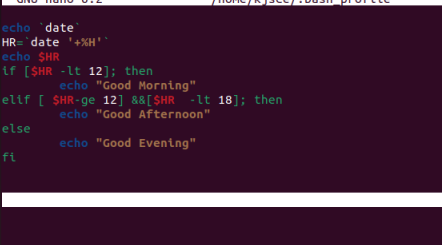
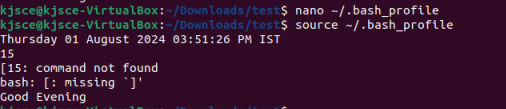

6. awk - Pattern scanning and processing language.  
  


7. sed - Stream editor for text transformations.  


8. top - Display Linux tasks and system status in real-time.  


9. df - Report file system disk space usage.  


10. du - Estimate file space usage.  
  
  


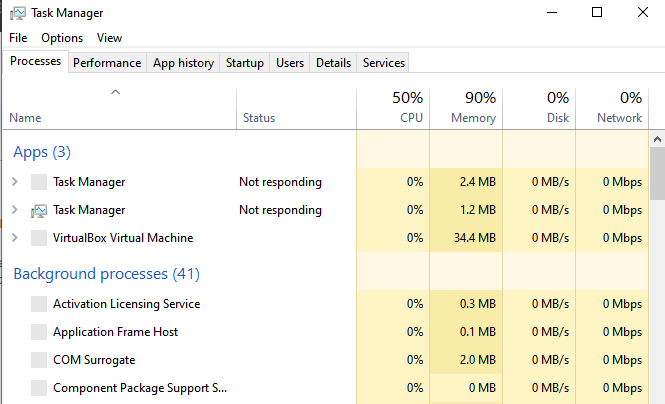
11. Edit .bash\_profile file to print greeting based on time of the day. Execute the same.  
  
  


12. Run commands in multiple user profiles on same machine. (Hint: use hot keys)  
Ctrl+Alt+F1 to Ctrl+Alt+F6 -Switch to different console  
Ctrl+Alt+T- open new terminal

D. List of Windows OS Utilities for Exploration:

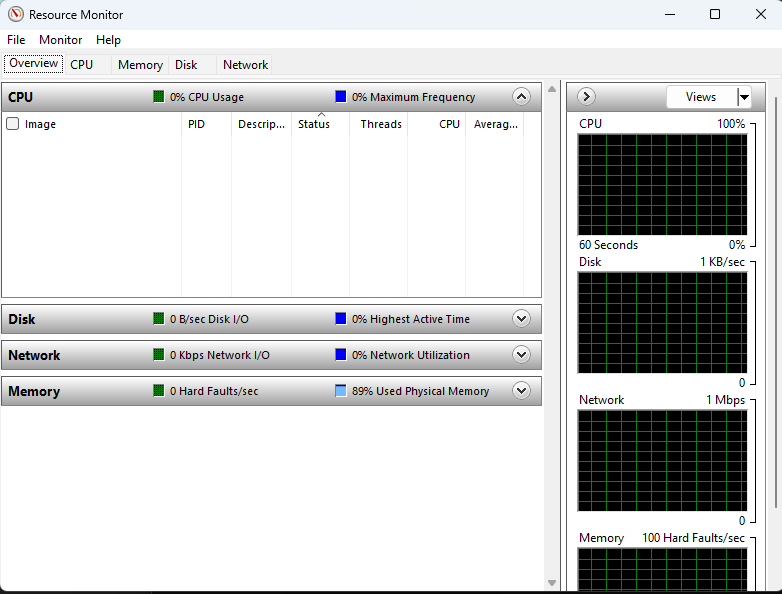
1. Task Manager :

- Purpose : Monitor system performance, view running processes, manage startup programs,and analyze resource usage (CPU, memory, disk, network).

- Usage : Identify resource-heavy processes, terminate unresponsive applications, and monitor system health.  
  


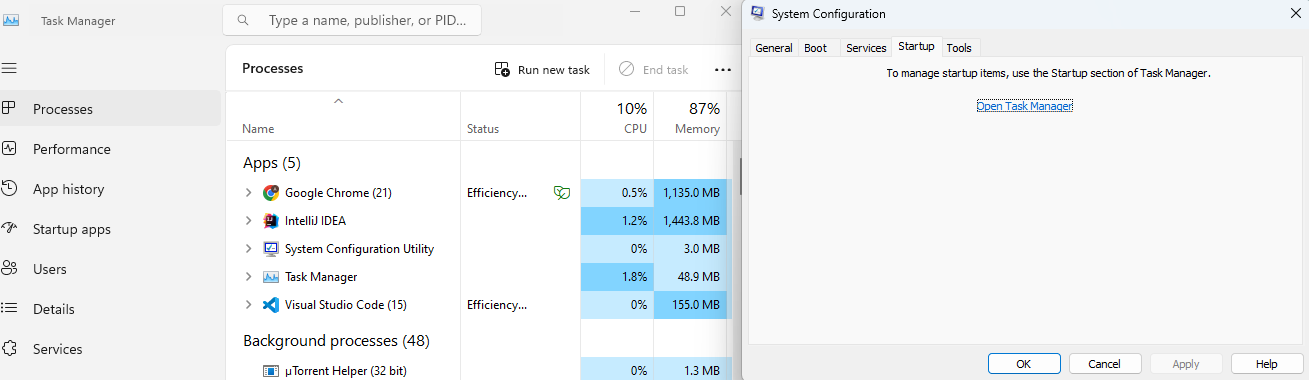
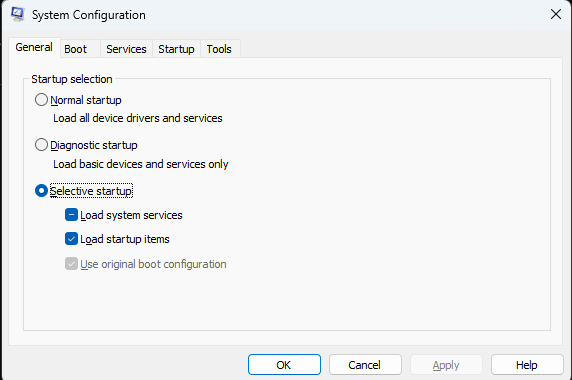
2. Resource Monitor (resmon):

- Purpose : Provides detailed real-time information about CPU, memory, disk, and network usage by processes and services.

- Usage : Monitor specific process performance, analyze disk activity, network usage, andidentify resource bottlenecks.  


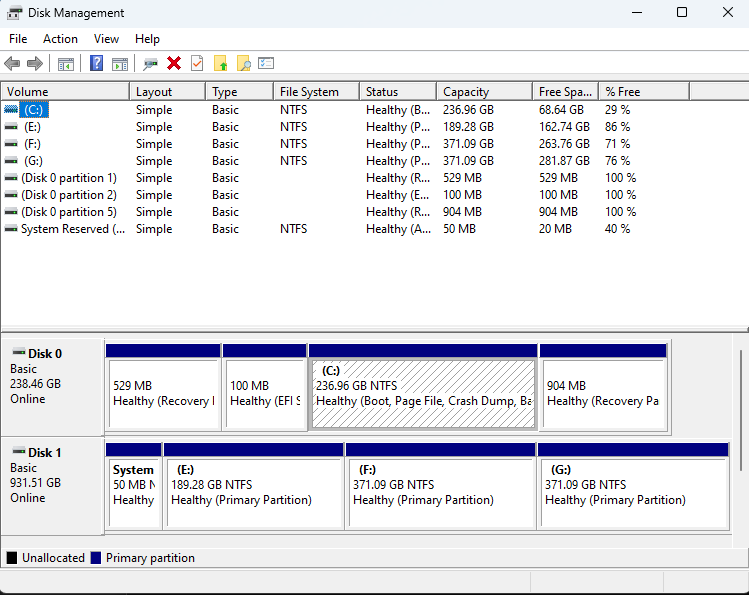
3. System Configuration (msconfig) :

- Purpose : Configure system startup, services, and startup programs.

- Usage : Enable/disable startup programs, manage boot options, and troubleshoot startup Issues.   
  


4. Disk Management :

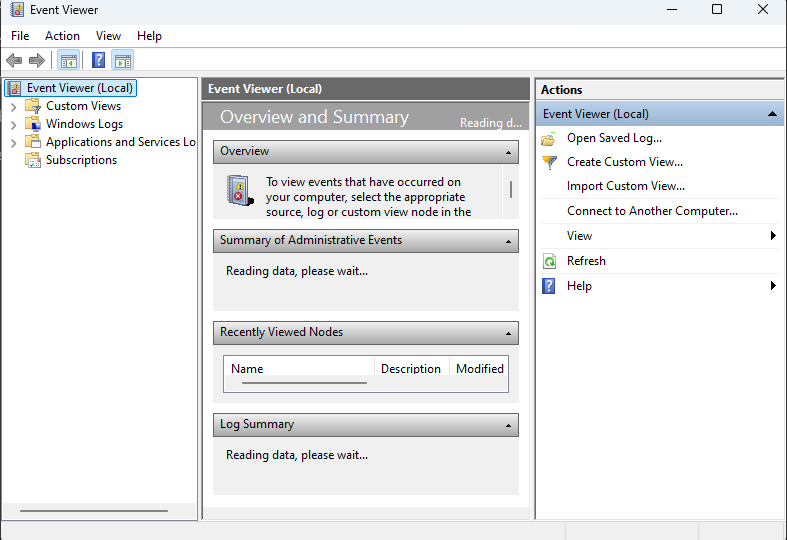
- Purpose : Manage disk partitions, format disks, assign drive letters, and create/delete volumes.

- Usage : Partition disks, extend/shrink volumes, convert disks between different formats ,fragmentation/defragmentation(e.g., MBR to GPT).  


5. Event Viewer :

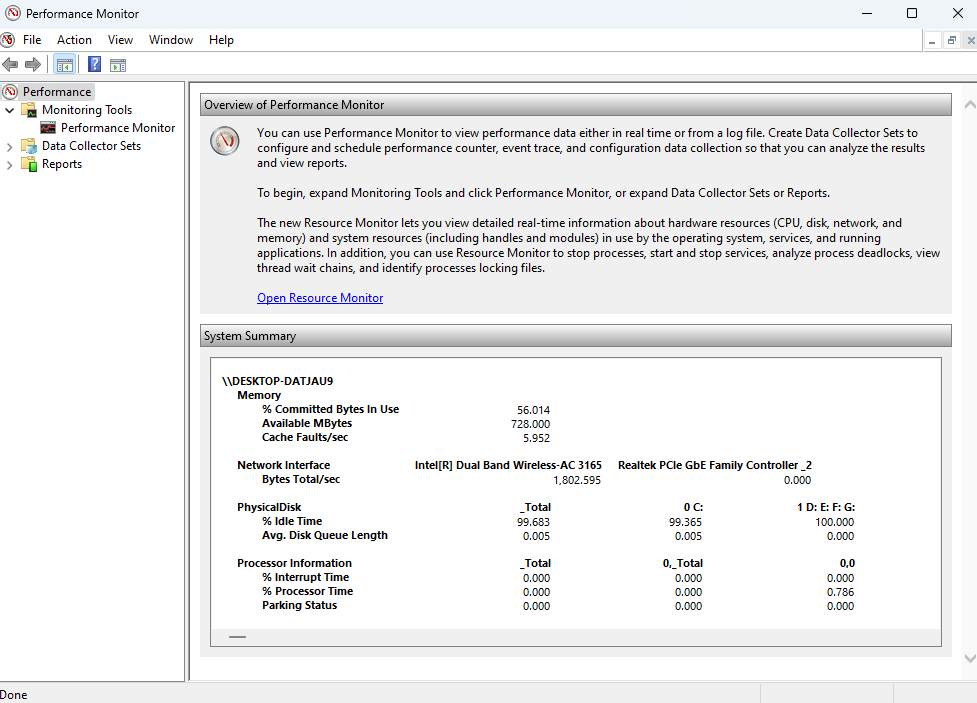
- Purpose : View system logs, application logs, and security logs to diagnose system and application issues.

- Usage : Investigate system errors, warnings, and information events to troubleshoot

problems and monitor system health.  


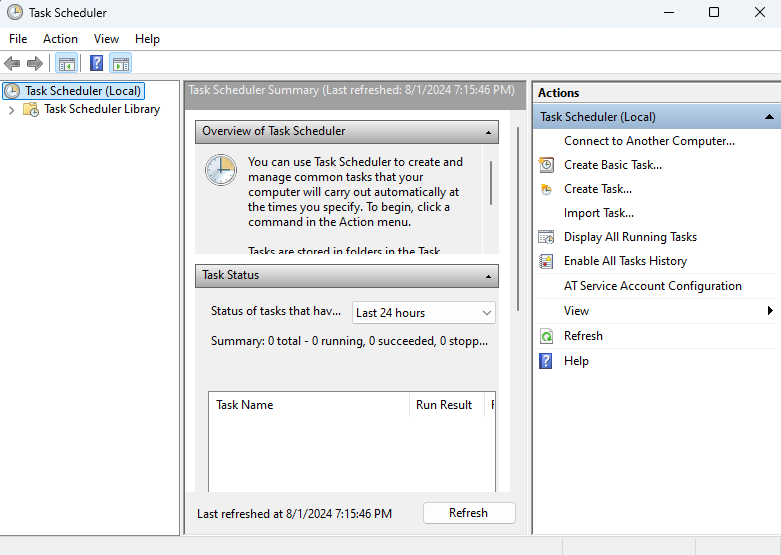
6. Performance Monitor :

- Purpose : Monitor and analyze system performance counters (CPU, memory, disk, network) over time.

- Usage : Create performance logs, set alerts based on performance thresholds, and analyze system performance trends.  


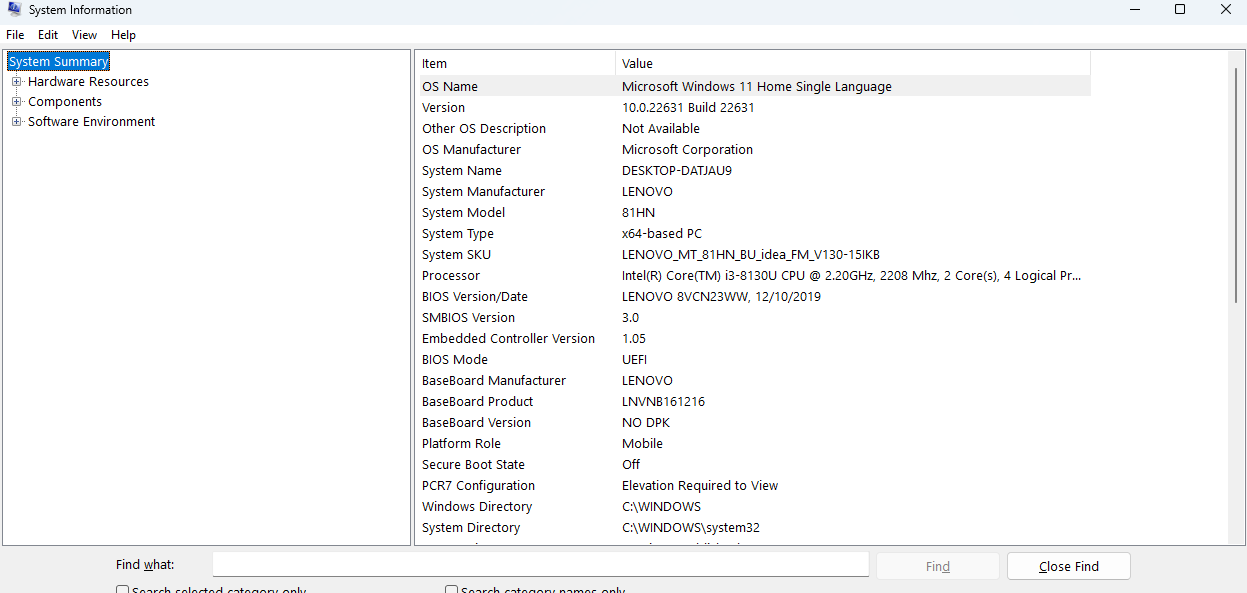
7. Task Scheduler :

- Purpose : Automate tasks and programs to run at specified times or events.

- Usage : Schedule disk cleanup, backups, maintenance tasks, and script executions.  


8. System Information (msinfo32) :

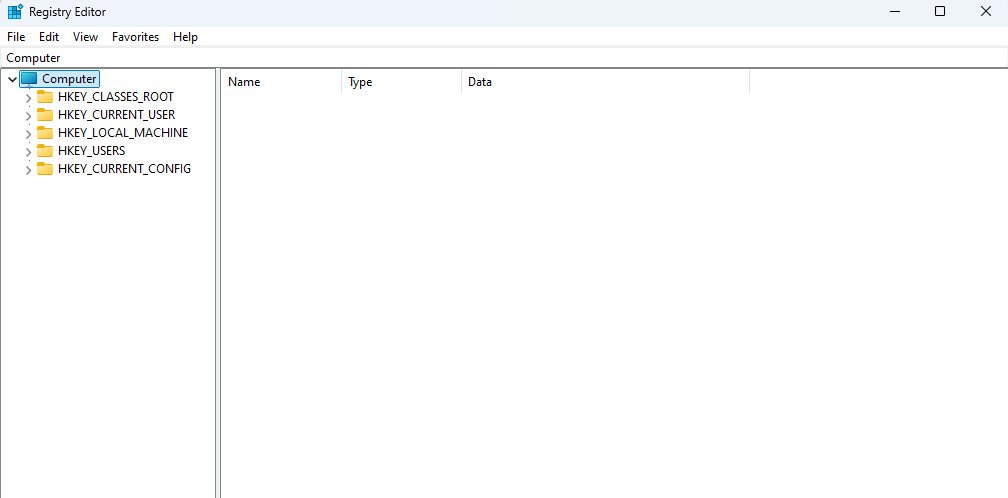
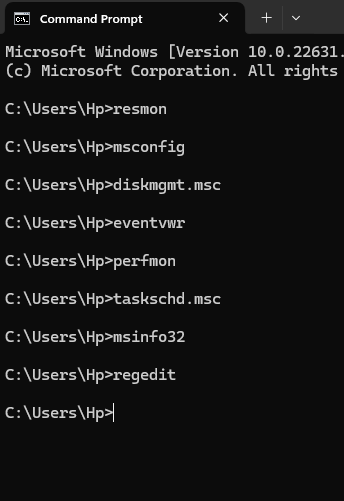
- Purpose : Provides detailed information about system hardware, components, software environment, and system settings.

- Usage : Retrieve system specifications, hardware details, installed software, and system configuration information.  


9. Registry Editor (Regedit) :

- Purpose : Edit and manage the Windows registry, which stores system configuration

settings and options.

- Usage : Modify registry keys and values, troubleshoot system settings, and configure system behaviors.  
  


**Conclusion:**

Explored different commands and OS utilities of Windows and Unix.

**Post Lab Descriptive Questions**

1. Explain how do you read and interpret syntax of any OS command.
2. Explain different functions of the operating systems.
3. What are the default permissions assigned by Unix for Directory.
4. Give the difference between DOS and WINDOWS.
5. Explain Booting Process.

**ANSWERS:**

### 1.The syntax of an OS command typically follows a structured format that includes the command itself, options, and arguments. Here’s a general way to interpret it:

* Command: The base instruction (e.g., ls, mkdir, cp).
* Options: These modify the behavior of the command. They often begin with a dash (e.g., -l for long format in ls, -r for recursive in cp).
* Arguments: These specify the targets or inputs for the command (e.g., filenames, directories).

Example Command: cp -r source\_dir destination\_dir

* cp: The command to copy files/directories.
* -r: Option to copy directories recursively.
* source\_dir: Argument specifying the source directory to copy.
* destination\_dir: Argument specifying where to copy the source directory.

To interpret the syntax:

* Identify the command.
* Recognize any options and their purpose.
* Determine the arguments and their roles.

### 2. Operating systems (OS) perform a variety of critical functions to manage computer hardware and software. Key functions include:

* Process Management: Manages the execution of processes including scheduling, creating, and terminating processes.
* Memory Management: Oversees allocation and deallocation of memory space for processes.
* File System Management: Handles the storage, retrieval, and organization of files on disk drives.
* Device Management: Manages hardware devices and their interactions with the system.
* Security and Access Control: Ensures system security by managing user permissions and protecting against unauthorized access.
* User Interface: Provides an interface for users to interact with the computer (CLI or GUI).

### 3. In Unix-like systems, directories have default permissions assigned:

* Read (r): Allows users to list the contents of the directory.
* Write (w): Allows users to add or remove files from the directory.
* Execute (x): Allows users to access the directory and its contents.

When a directory is created, it typically has the following default permissions:

* Owner: Read, write, and execute (rwx)
* Group: Read and execute (r-x)
* Others: Read and execute (r-x)

For example, a directory might have permissions displayed as drwxr-xr-x.

### 4.DOS (Disk Operating System) and Windows represent different generations and types of operating systems:

* DOS:
  + Text-based, single-tasking OS.
  + Command-line interface (CLI).
  + Limited multitasking capabilities.
  + File system: FAT12, FAT16.
* Windows:
  + Graphical user interface (GUI) based OS.
  + Multi-tasking and multi-user capabilities.
  + Supports advanced features like virtual memory, networking, and modern file systems (e.g., NTFS).
  + Examples include Windows 10, Windows 11.

### 5. The booting process is the sequence of operations that a computer performs to start up and load the operating system:

* Power-On Self Test (POST): When the computer is turned on, the BIOS/UEFI performs POST to check the hardware components and ensure they are functioning properly.
* Load BIOS/UEFI Firmware: The firmware initializes the hardware and prepares the system to load the OS.
* Bootloader: The BIOS/UEFI locates and executes the bootloader from the bootable storage device (e.g., hard drive, SSD). Common bootloaders include GRUB (for Linux) and Windows Boot Manager.
* Load Kernel: The bootloader loads the operating system kernel into memory. The kernel is responsible for managing system resources and hardware.
* Initialize System Services: The kernel initializes system services and drivers necessary for the OS to function properly.
* Start User Interface: The OS loads the user interface (CLI or GUI), allowing user interaction with the system.

**Date: \_\_\_\_\_1/8/24\_\_\_\_\_\_\_\_ Signature of faculty in-charge**