**Batch: B2 Roll No.: 16010121194**

**Experiment No. 01**

**Grade: AA / AB / BB / BC / CC / CD /DD**

**Signature of the Staff In-charge with date**

|  |
| --- |
| **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

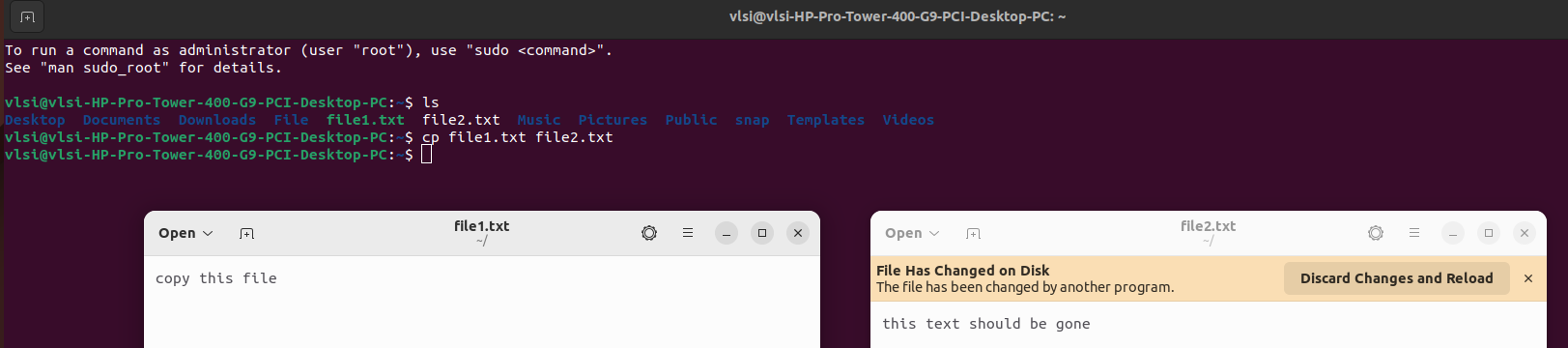
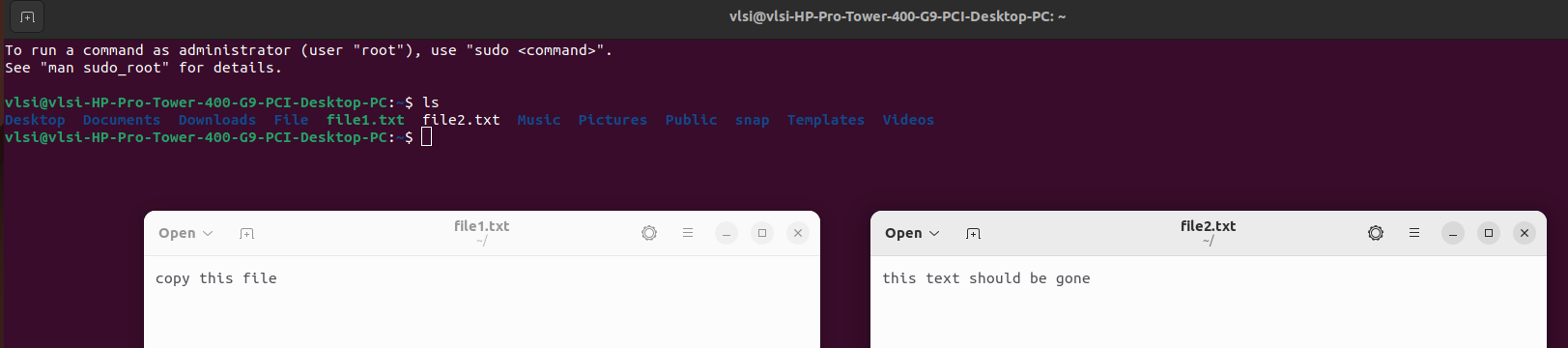
Unix Commands:

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 ,utime
5. Process management: ps, top, kill,nice
6. Unix users commands: whoami , id, groups, passwd , who, last

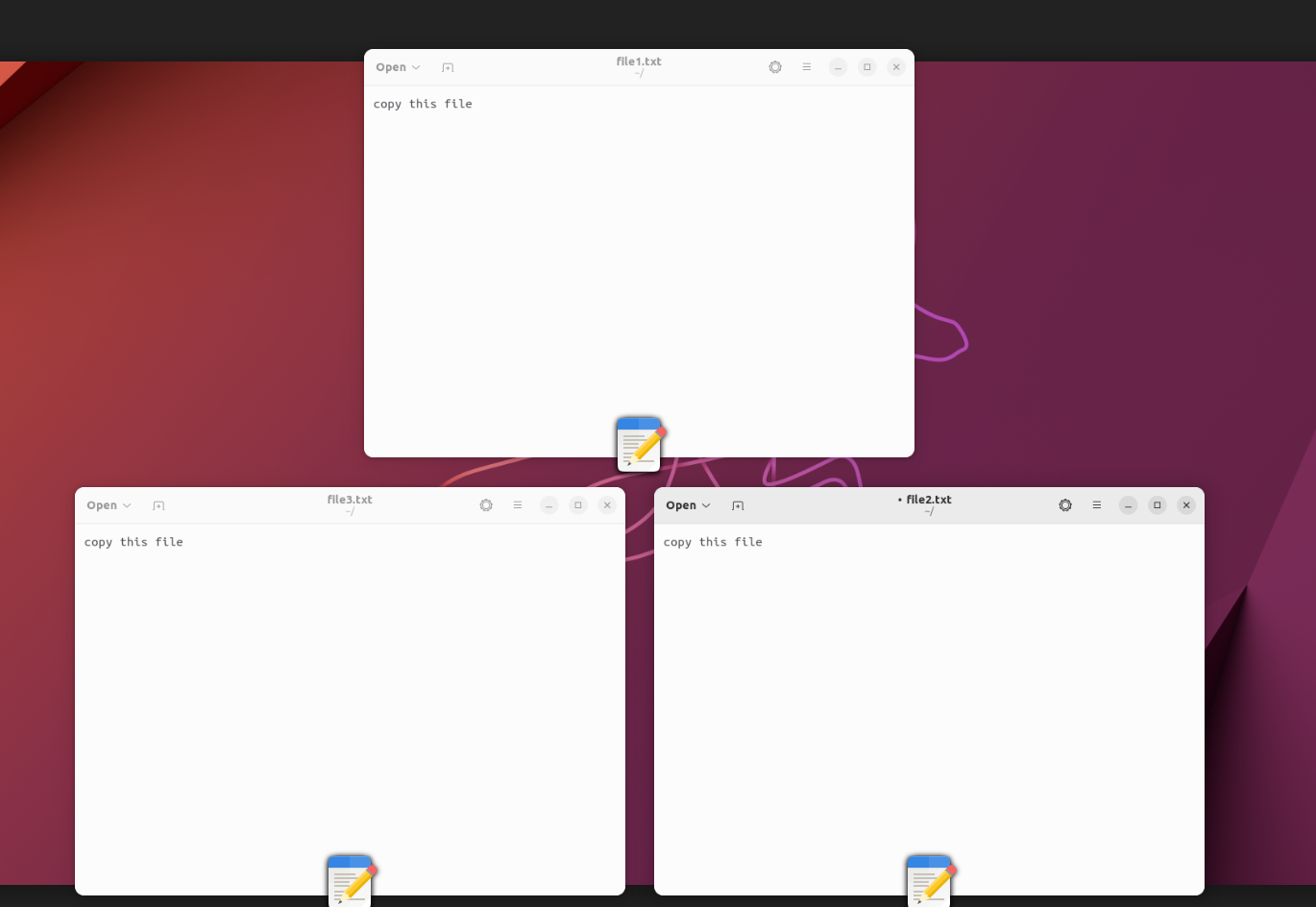
**Implementation details:**

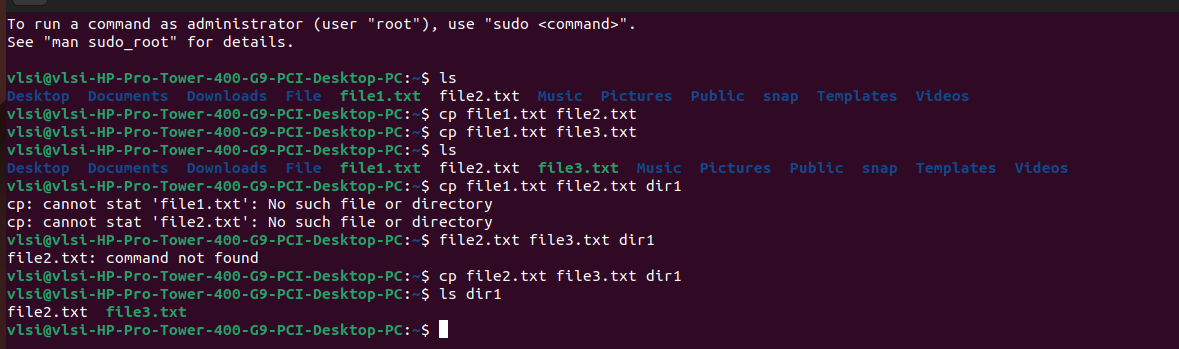
**Ubuntu**

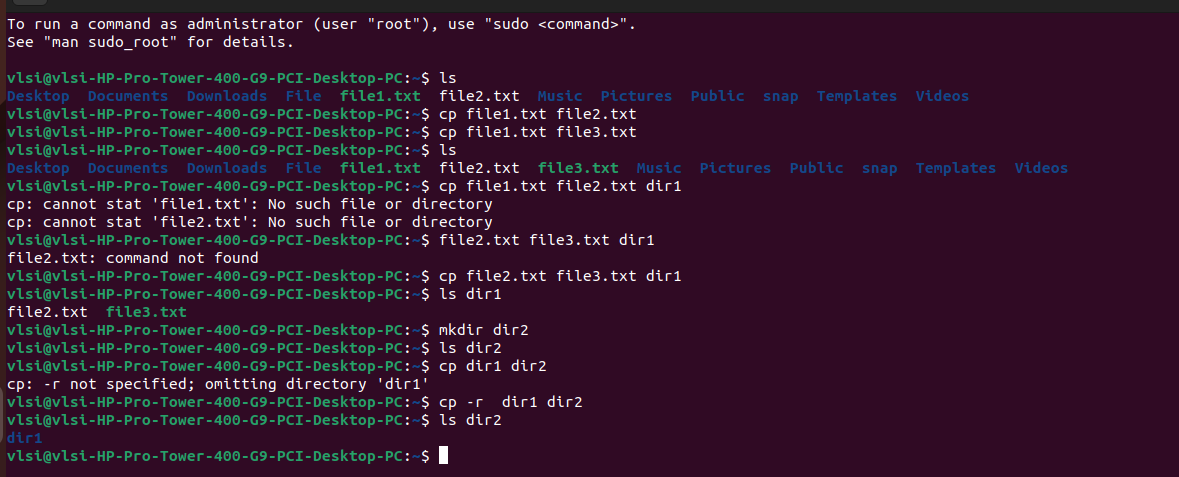
**cp command:**

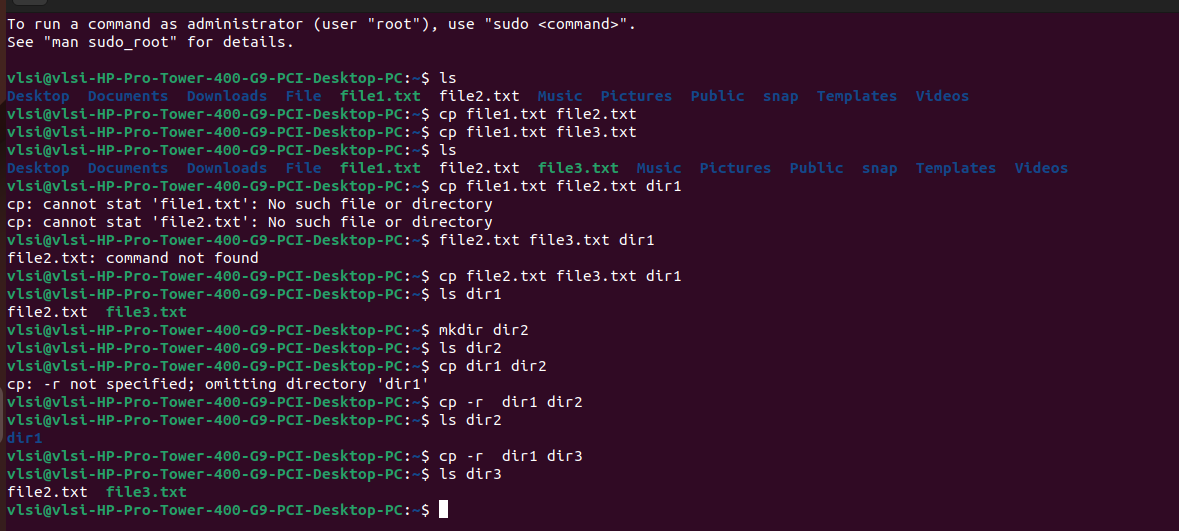
****

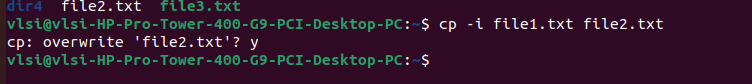
****



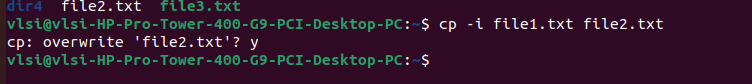




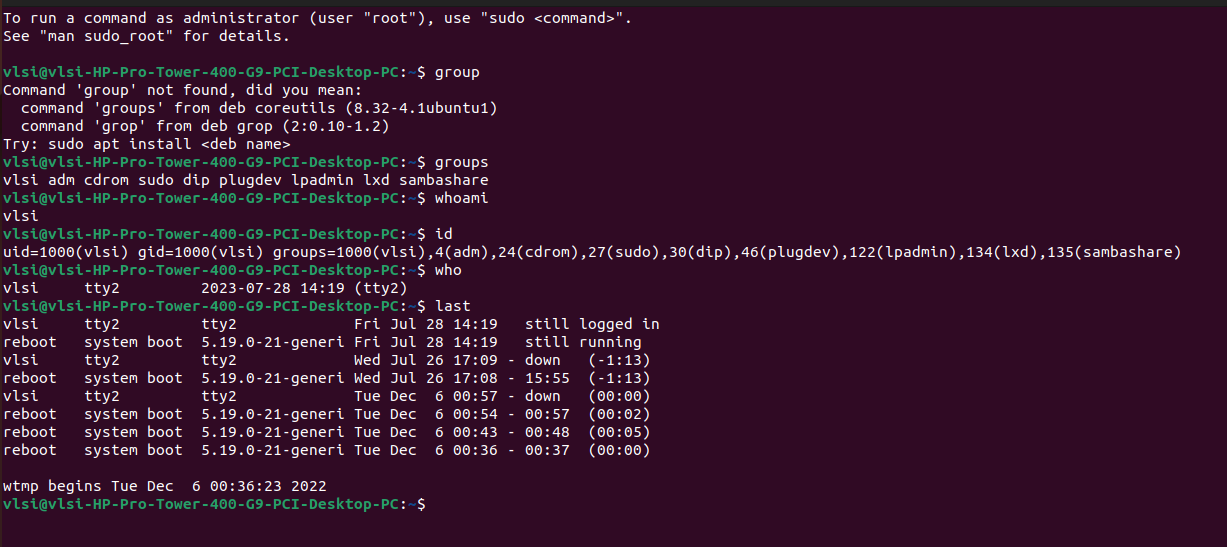




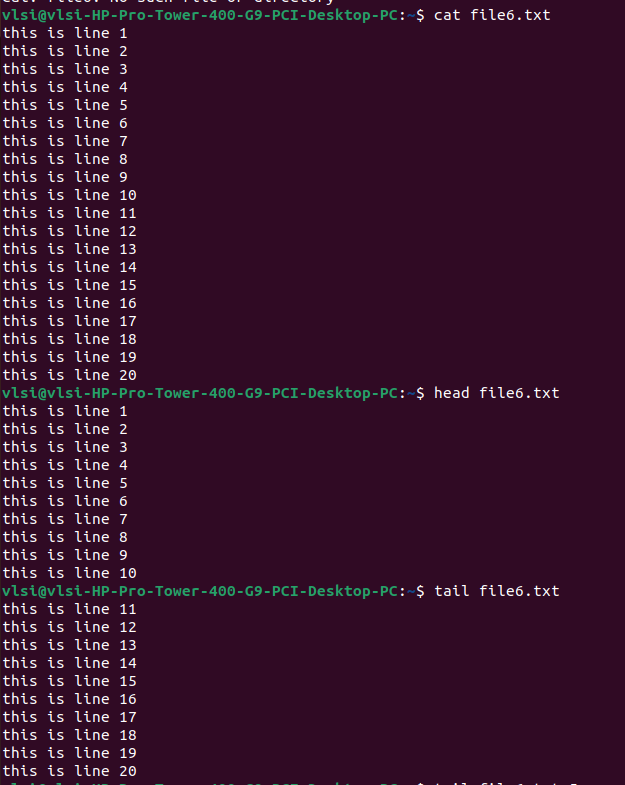
**cat command:**

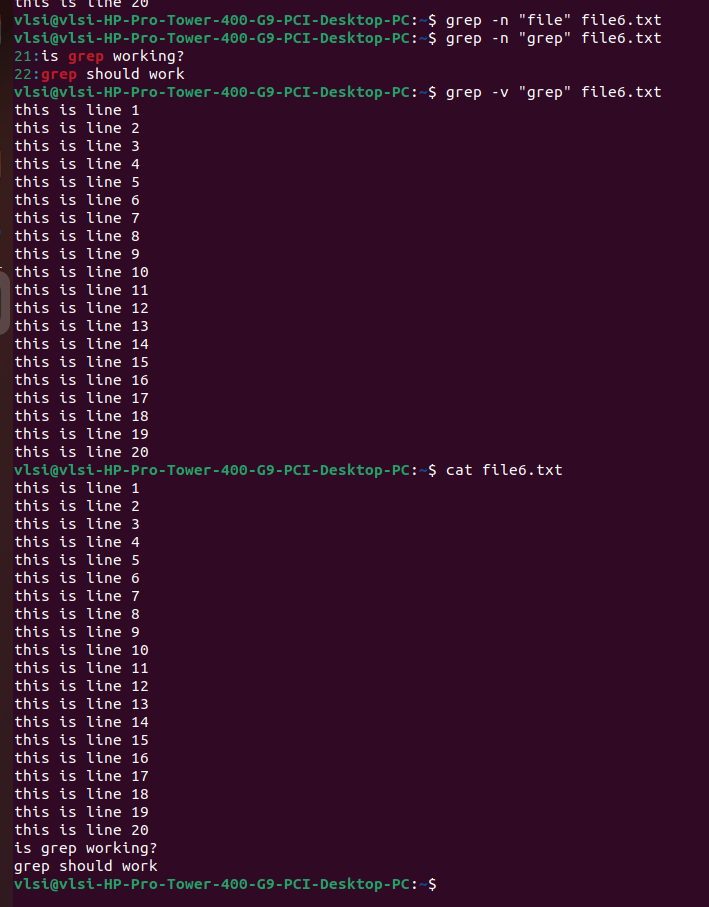


**User commands:**

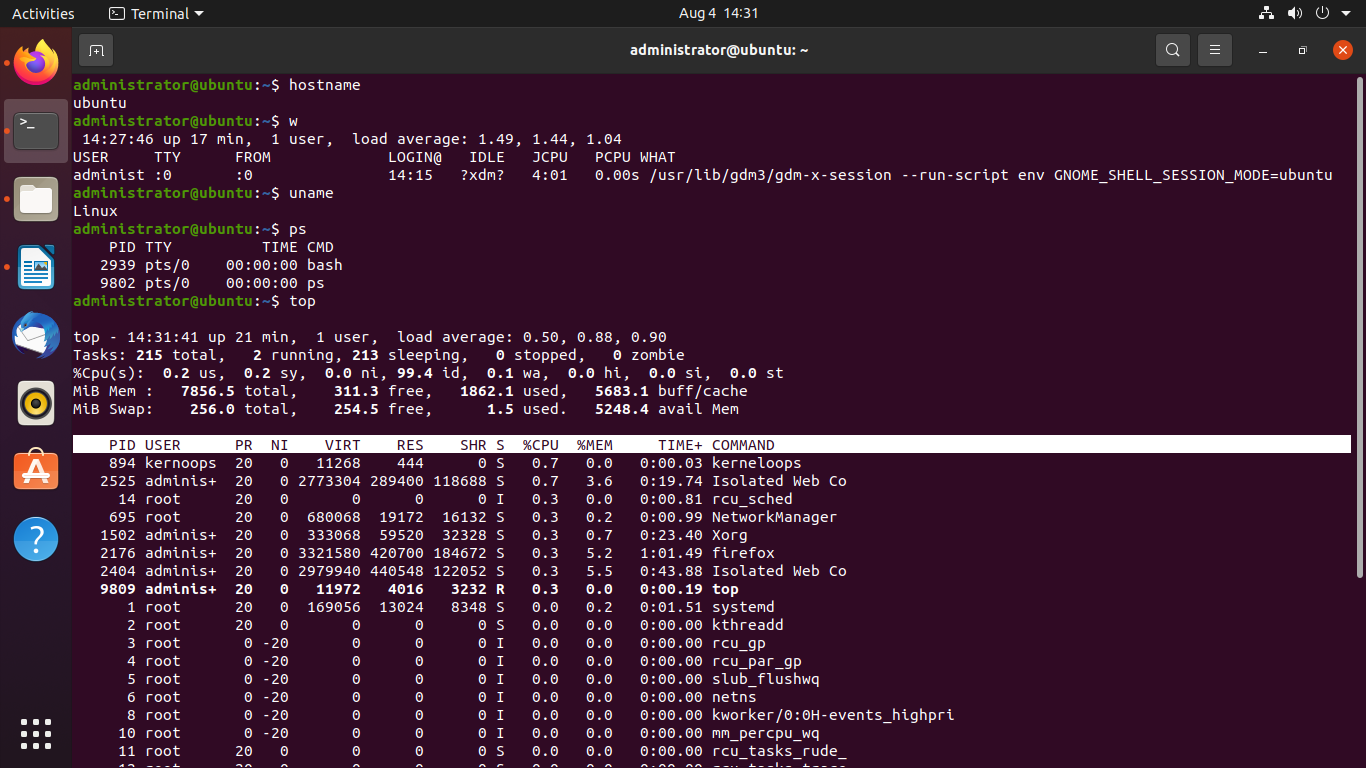


**Text file operations:**

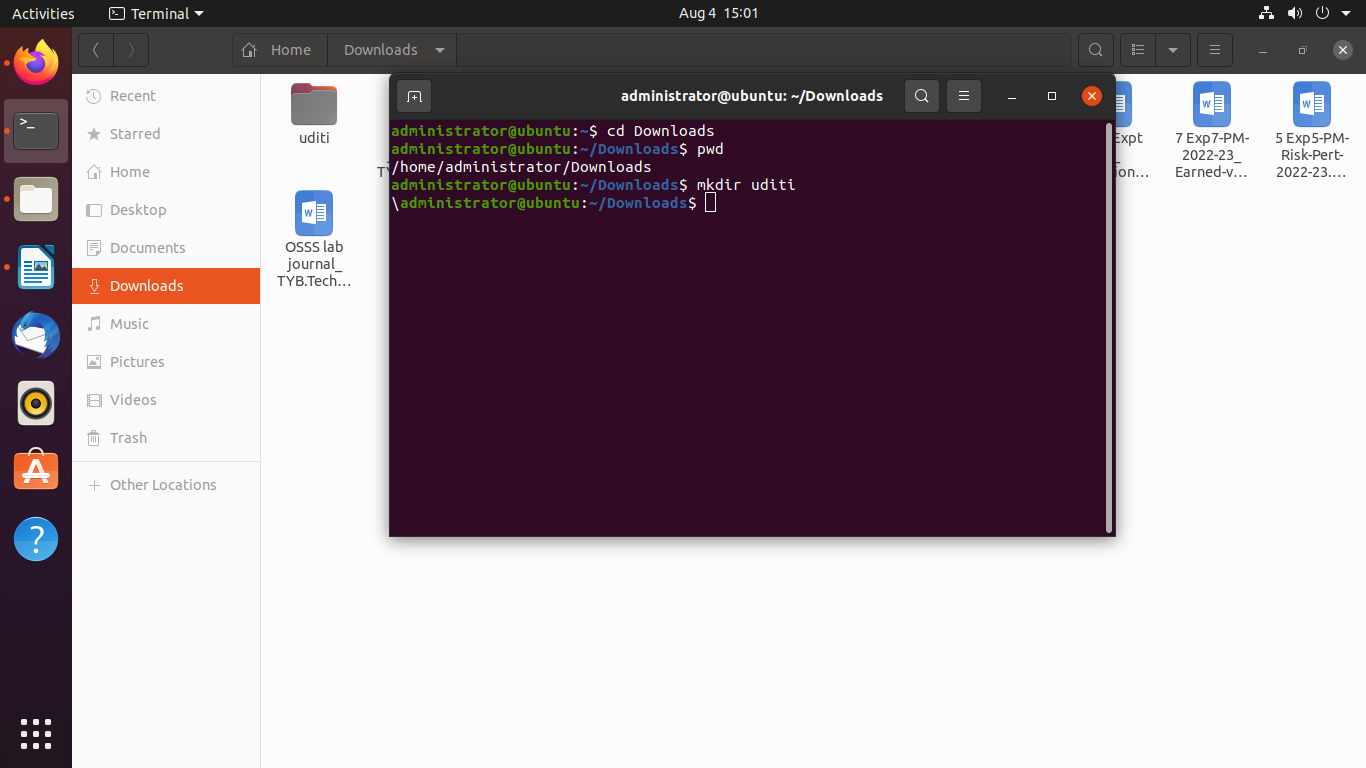


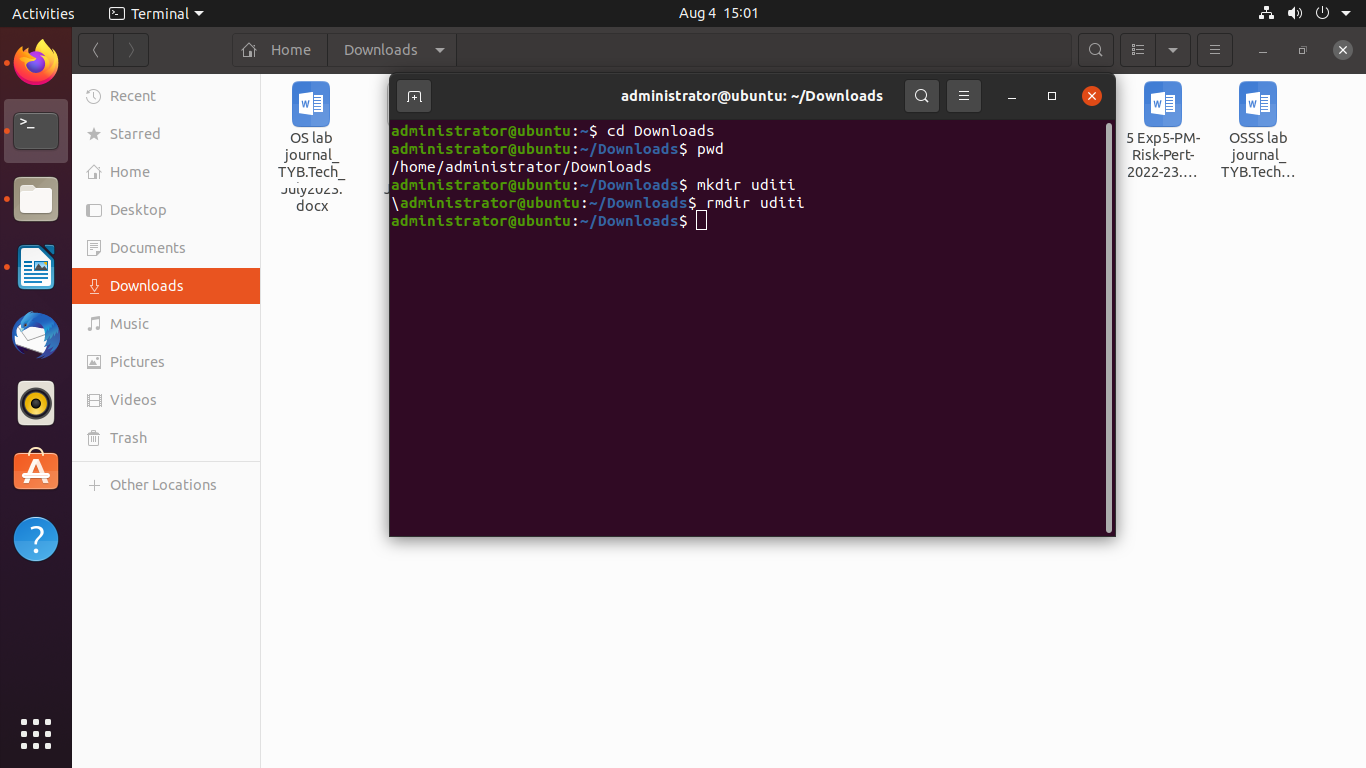


**System status and process management commands:**



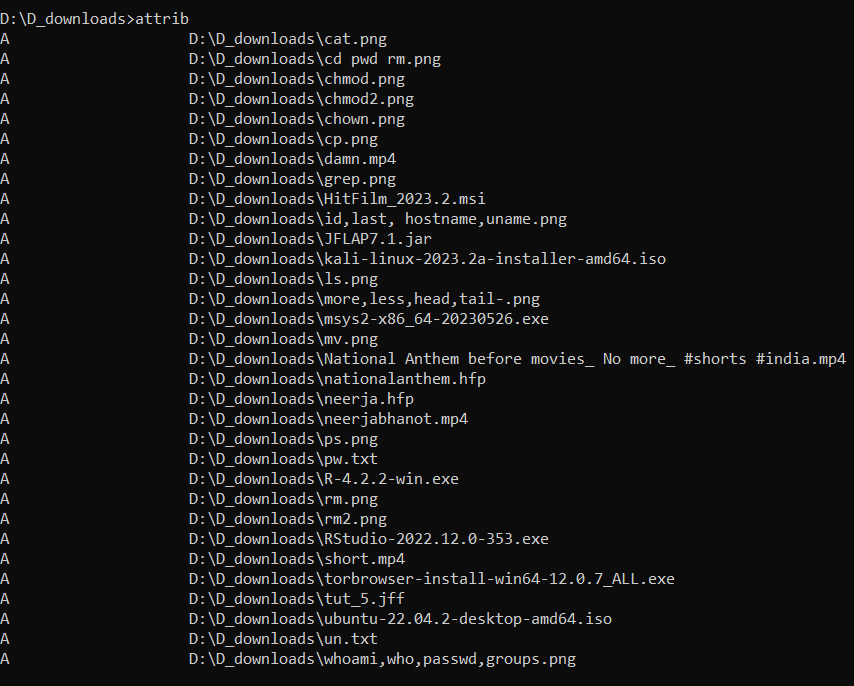
**Unix directory management commands:**



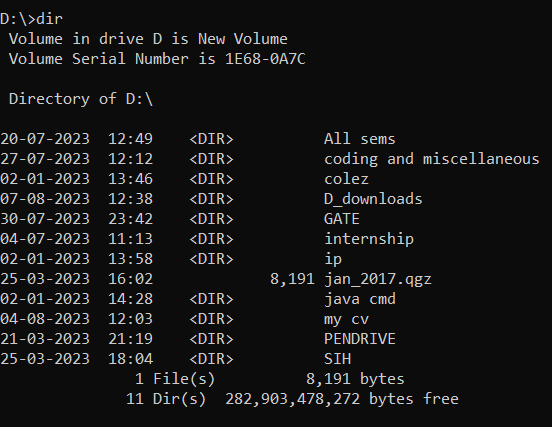


**2. DOS commands:** Attrib, dir, at, chkdsk, shutdown, tree, create a batch file, output and input redirection

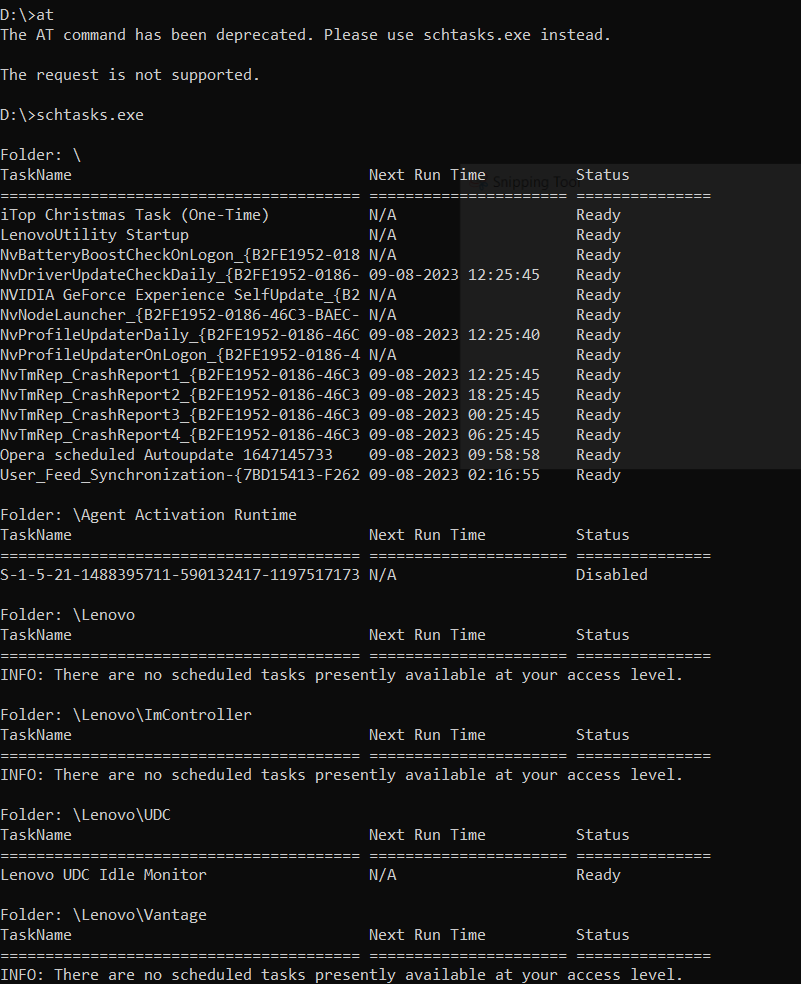
* attrib



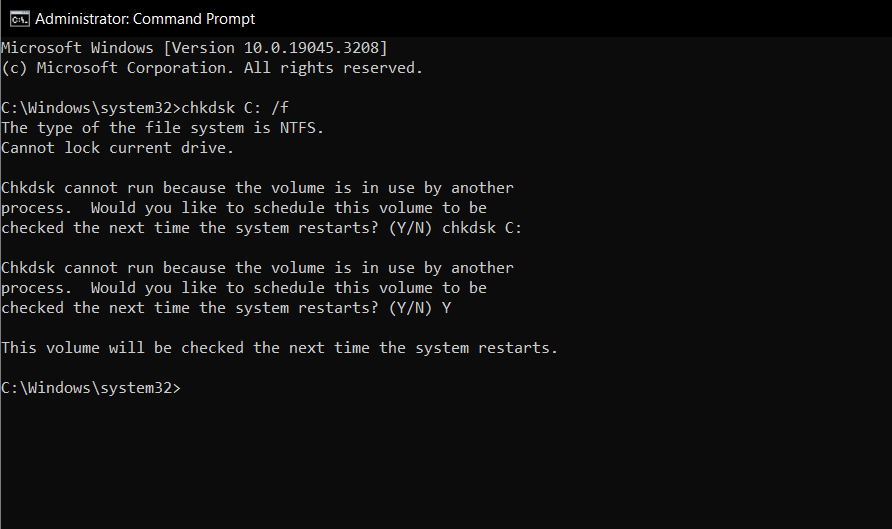
* dir

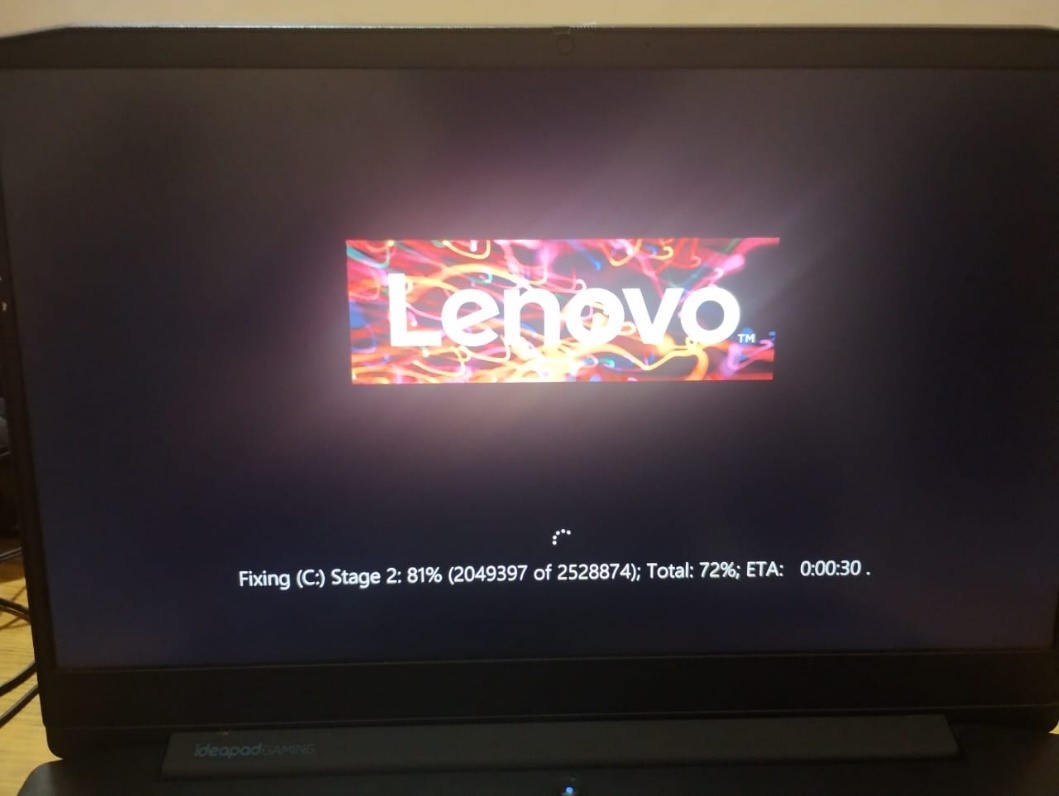


* at

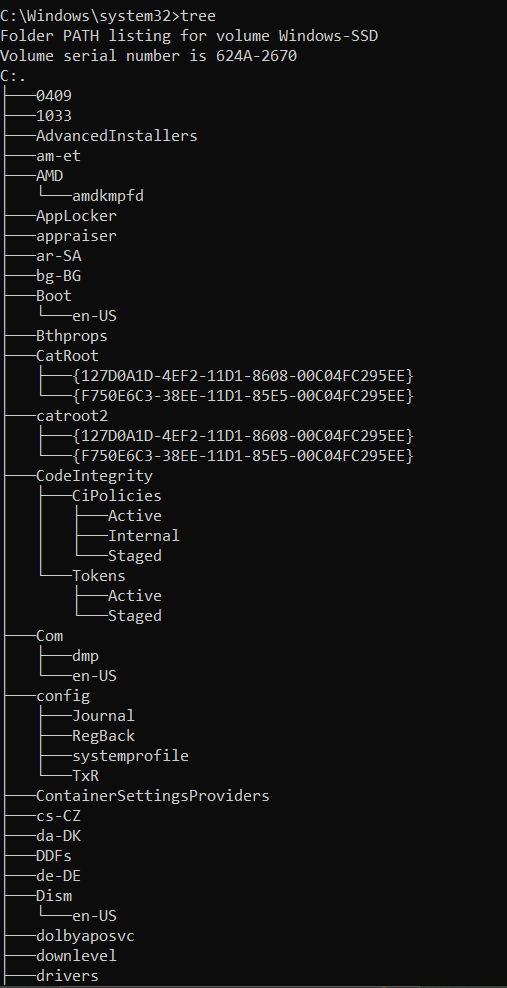


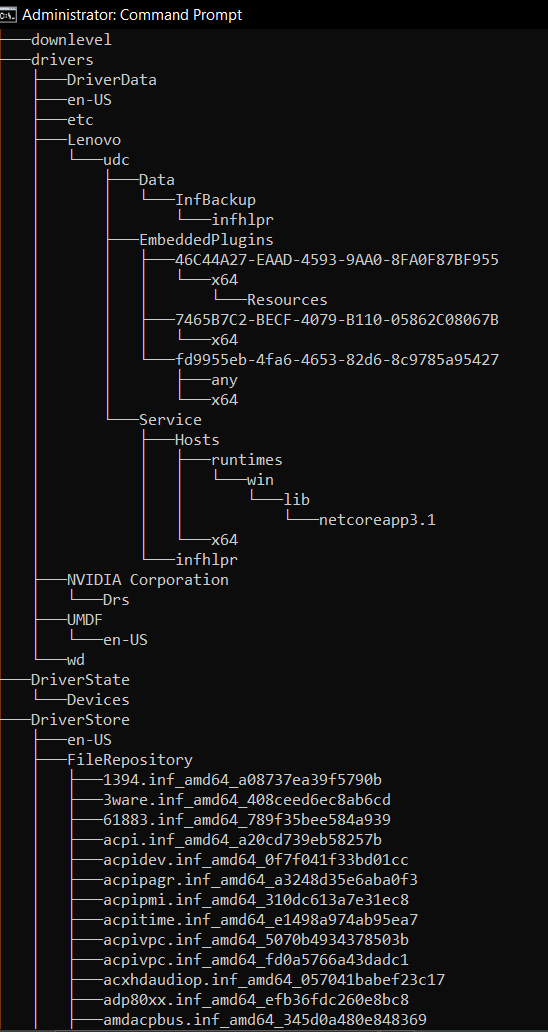
* chkdsk



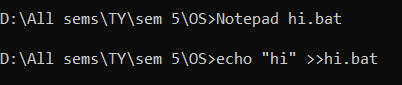


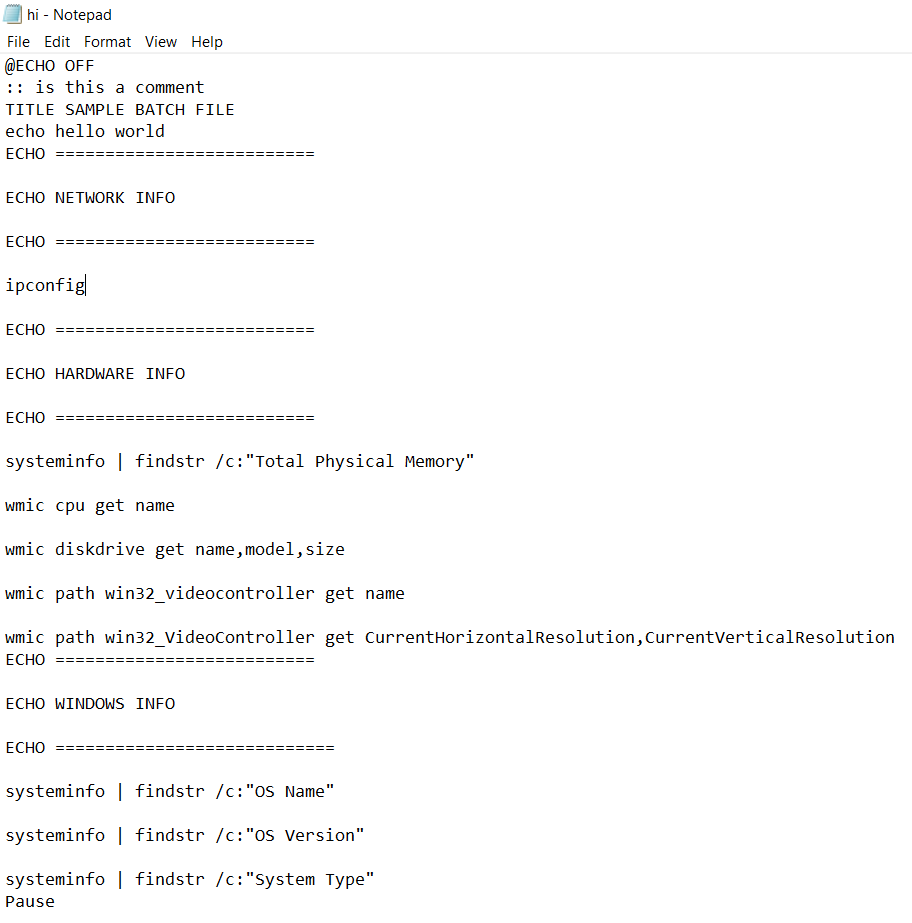
* tree

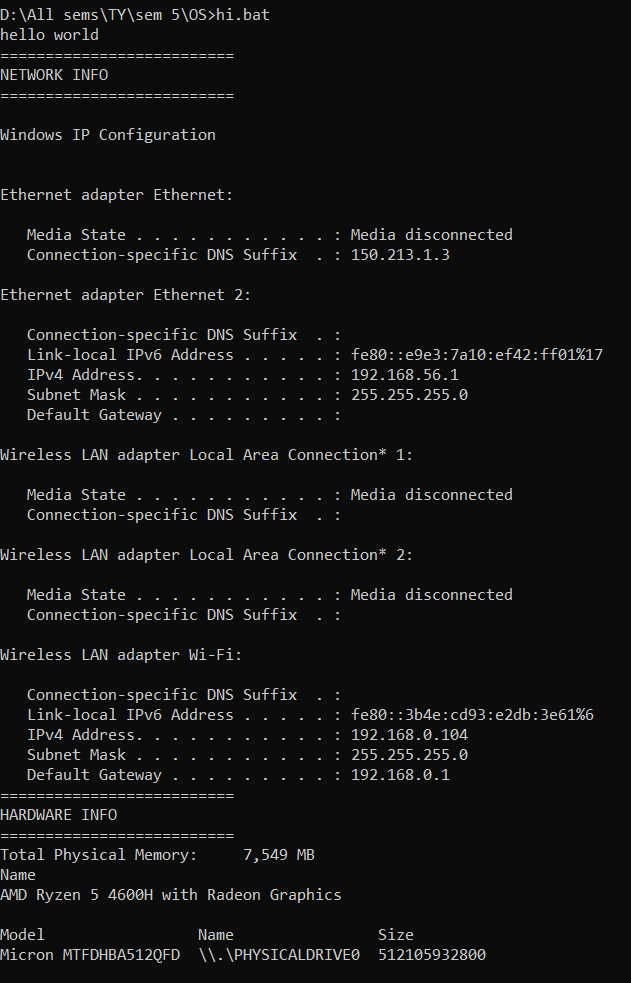


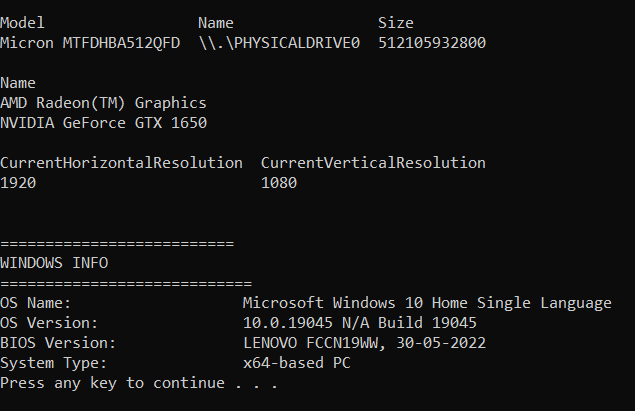


* create a batch file



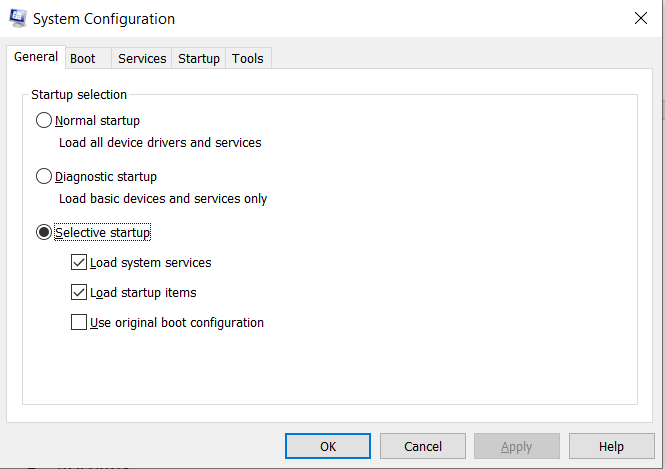




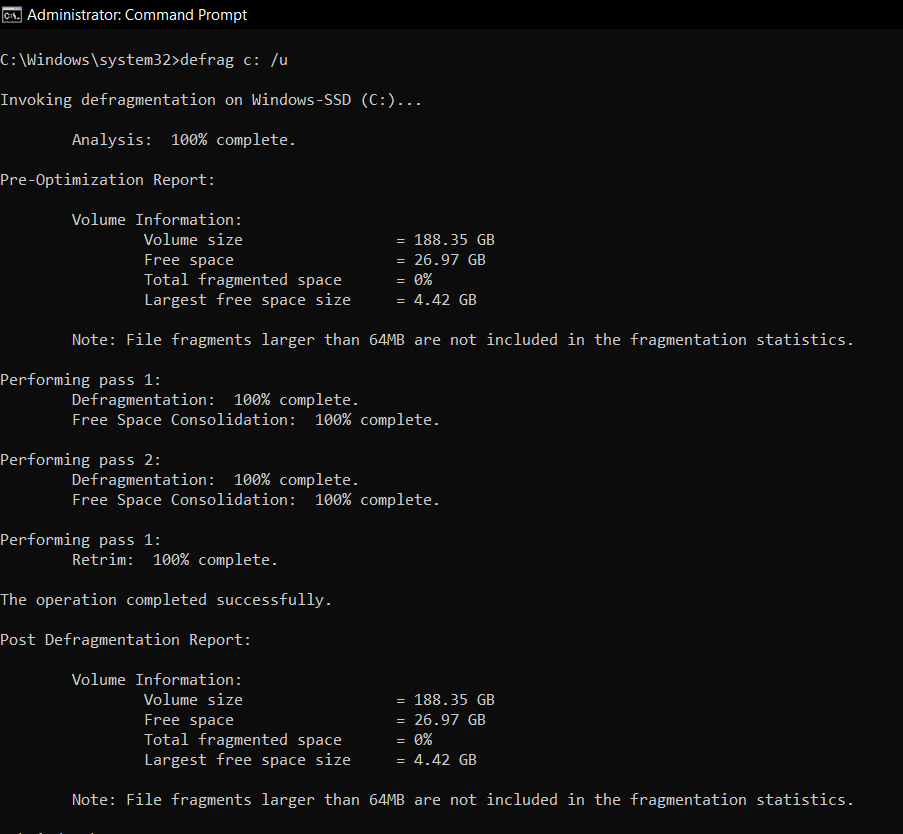


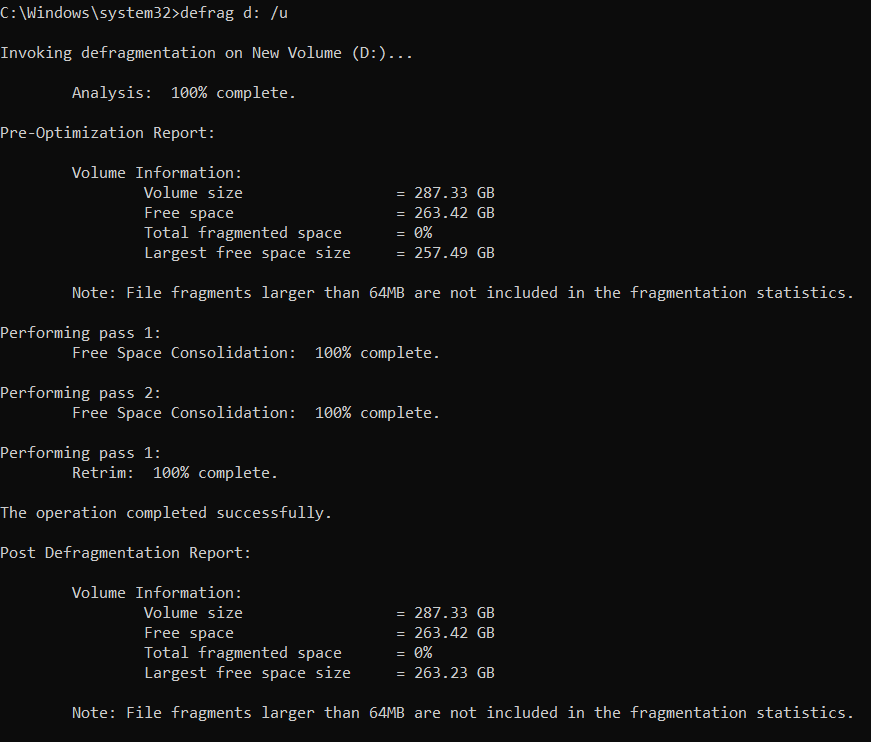
1. Windows utilities: msconfig, defragmenter, performance monitor, task manager, registry editor, event viewer, process explorer

* msconfig

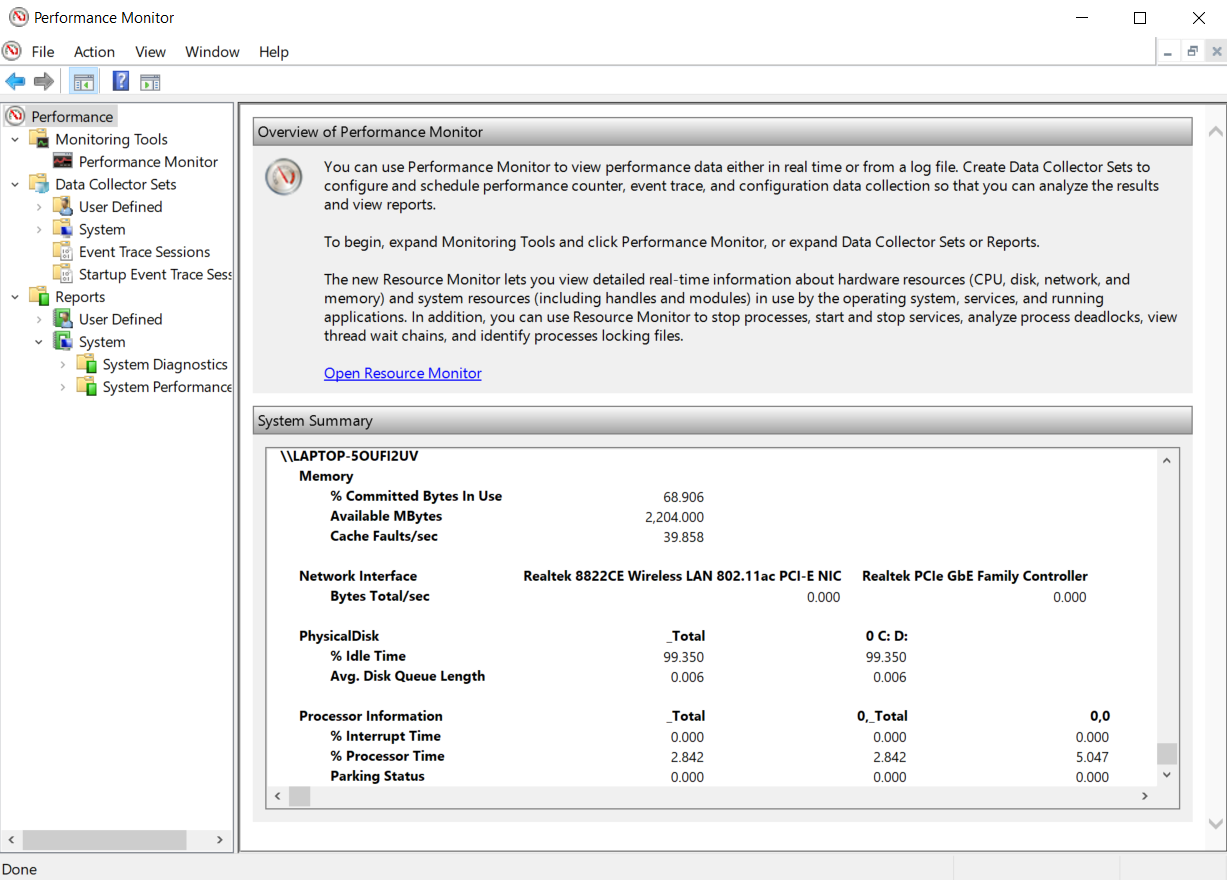


* defragmenter

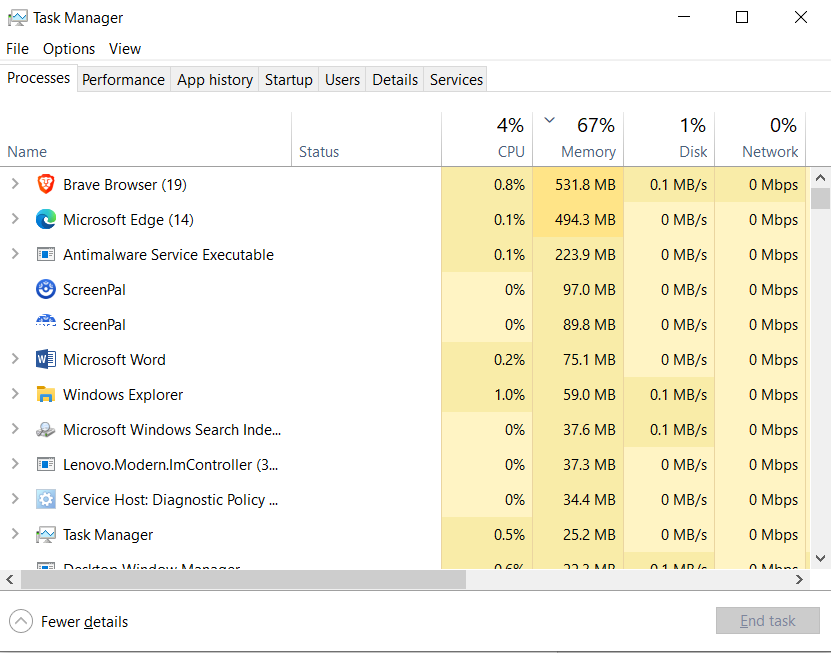




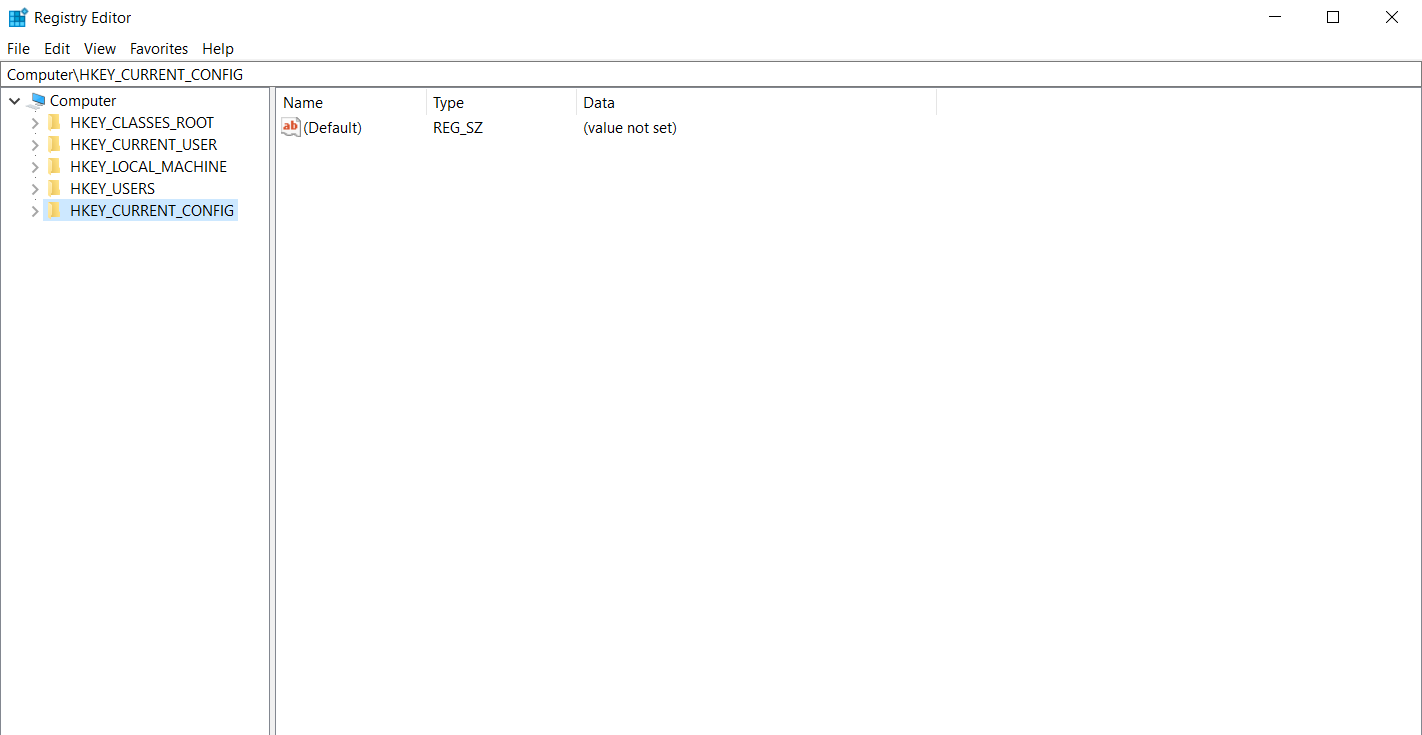
* performance monitor



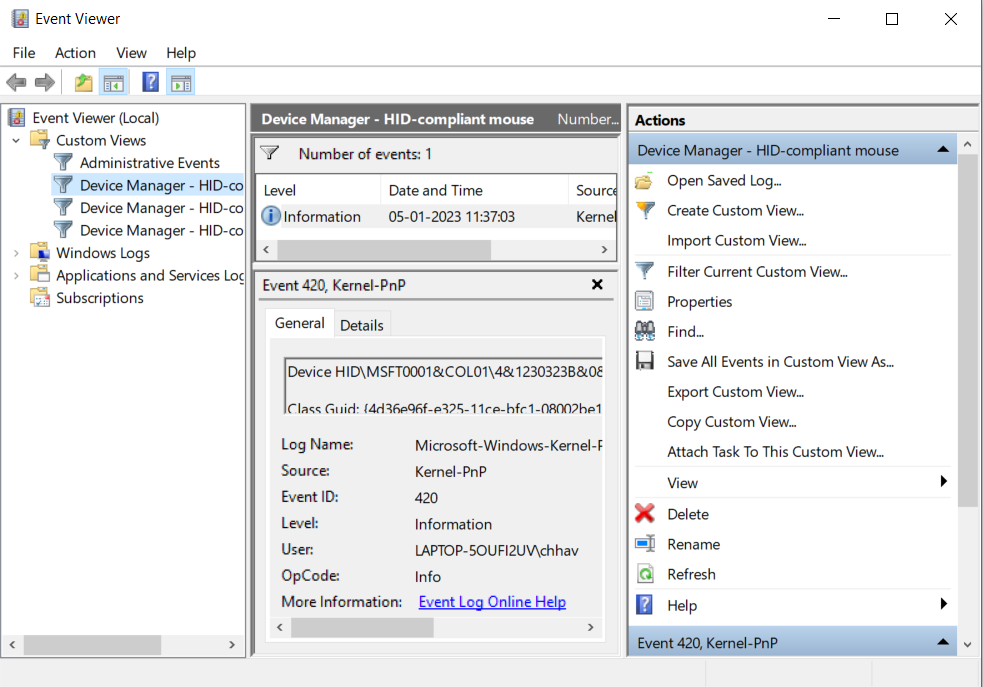
* task manager



* registry editor



* event viewer



**Conclusion:**

Used and understood various UNIX commands, DOS commands, Windows utilities. Learnt about shells and kernel programming. All commands were executed successfully.

**Post Lab Descriptive Questions**

1. Explain how do you read and interpret syntax of any OS command.

a. Identify the command: Know the name of the command you want to use.

b. Check the manual: Most operating systems provide a manual page (man page) for each command. Type 'man <command\_name>' in the terminal to access it.

c. Understand the options: Commands often have various options to modify their behavior. Options are typically preceded by a dash (-) or double dash (--).

d. Arguments: Commands may require additional arguments (e.g., filenames, paths) to execute correctly.

e. Order matters: Some commands expect specific order of options and arguments.

f. Practice and experimentation: Try out the command with different options and arguments to understand its behavior better.

1. Explain different functions of the operating systems.

a. Process management: Scheduling, creation, and termination of processes.

b. Memory management: Allocating and managing memory resources for processes.

c. File system management: Handling file creation, deletion, and organization.

d. Device management: Managing communication with hardware devices.

e. User interface: Providing a means for users to interact with the system (e.g., GUI or command-line interface).

f. Security: Enforcing access control and protecting the system from unauthorized access.

g. Networking: Facilitating communication between computers and devices over a network.

h. Error handling: Detecting and resolving errors to maintain system stability.

i. Software management: Installing, updating, and removing software packages.

1. What are the default permissions assigned by Unix for Directory.

The default permissions for a directory in Unix (and Unix-like systems) are typically 755. This means:

Owner: Full permissions (read, write, execute).

Group: Read and execute permissions.

Others: Read and execute permissions.

1. Give the difference between DOS and WINDOWS.

* DOS is a single-user, command-line-based OS primarily used in the early days of personal computers.
* Windows is a multi-user, graphical user interface (GUI) based OS developed by Microsoft.
* DOS is text-based and lacks a GUI, while Windows provides a visual desktop environment.
* Windows supports multitasking, allowing multiple programs to run simultaneously, whereas DOS is single-tasking.
* Windows supports more advanced hardware and networking capabilities compared to DOS.

1. Explain Booting Process.

a. Power-on self-test (POST): The computer's hardware is checked to ensure it's functioning correctly.

b. Bootloader: A small program, like GRUB or NTLDR, is loaded from the boot device (e.g., hard drive) into memory.

c. Kernel loading: The bootloader loads the OS kernel into memory.

d. Initialization: The kernel initializes essential system components and drivers.

e. User space initiation: The kernel starts the init process, which initializes the rest of the system, services, and user space processes.

f. Graphical User Interface (GUI): In Windows and many modern Linux systems, the GUI is loaded, allowing user interaction. In some Linux distributions, a display manager may handle the login process.

g. User login: The user logs in, and the operating system becomes fully functional, ready to run applications and handle user requests.

**Date: \_\_\_\_\_\_\_\_\_\_\_\_\_ Signature of faculty in-charge**