**Shri G.S. Institute of Technology and Science**

**INDORE**



**Department of Information Technology and Application**

**MCA First Year Semester I January -June 2025**

**Lab Assignment I**

**CT 10709: Operating System**

SUBMITED TO SUBMITTED BY

Ms. Shweta Gupta SHIV ARORA

Mr. Shashank Sharma Enrolment No.

0801CA241133

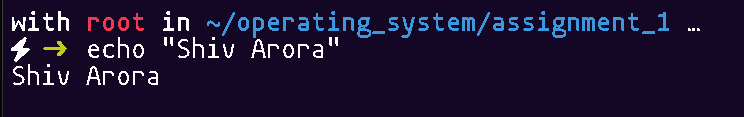
Question 1) What are the Internal commands Explain it? give 10 commands with syntax examples (with screenshots)

**Internal Commands**: Internal commands are built into the shell and do not require external executables. They are faster because they are executed directly by the shell without spawning new processes. Internal commands handle fundamental operations and within the shell environment, like navigating directories, managing processes, or setting variables. Since they are part of the shell, they are always available without needing to be installed separately. Examples include cd (change directory), echo (display a message), exit (exit the shell), and history (show command history).

Examples

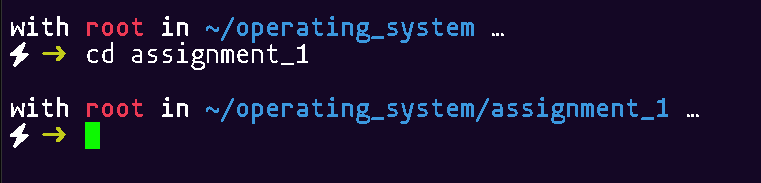
1. Echo -> Displays a line of text or variable value to the terminal.

Syntax: echo [option] [string]



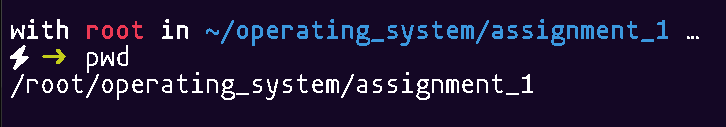
1. cd -> Changes the current working directory.

Syntax: cd [directory]



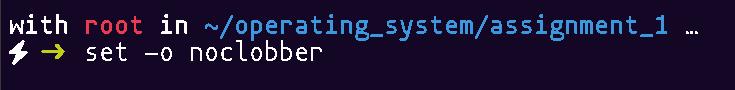
1. pwd -> Prints the current working directory path.

Syntax: pwd



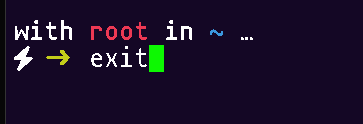
1. set -> Displays or sets environment variables or shell options.

Syntax: set [options]



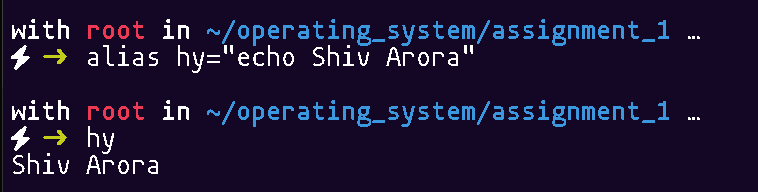
1. exit -> Exits the current shell or terminal session.

Syntax: exit [status]



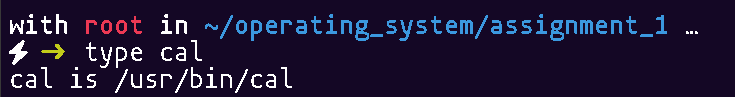
1. alias -> Creates a shortcut for a command or series of commands.

Syntax: alias name='command'



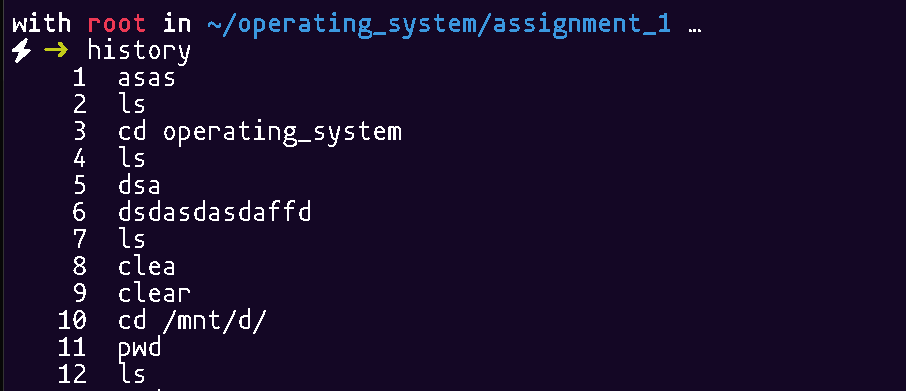
1. type -> Displays information about how a command is interpreted (built-in, alias, or executable).

Syntax: type command\_name



1. history -> Shows the list of previously executed commands.

Syntax: history



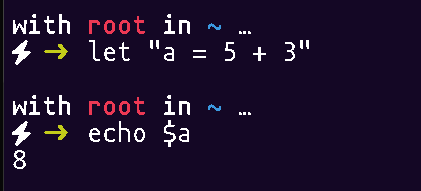
1. times -> Displays user and system times for the current shell session.

Syntax: times



1. let-> Evaluates arithmetic expressions and assigns the result to a variable.

Syntax: let "expression"



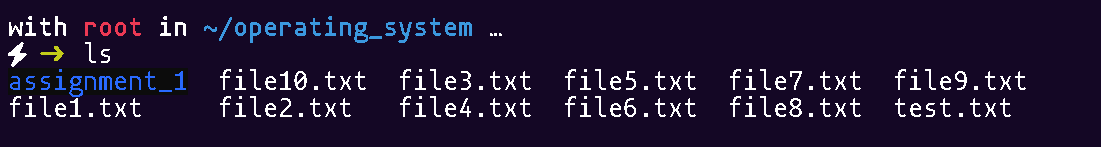
Question 2) What are External Commands define it? Explain any 10 commands with syntax & examples?

**External Commands**: These are commands that are not part of the shell itself and are stored as separate executable files on the system. When you run an external command, the shell spawns a new process to execute it. External commands are typically stored in directories like /bin, /usr/bin, or /sbin, and they can be called from anywhere in the system if their paths are included in the $PATH environment variable. Examples include ls (lists files), cp (copies files), and grep (searches through files). These commands often offer more complex functionality and are compiled programs.

Examples

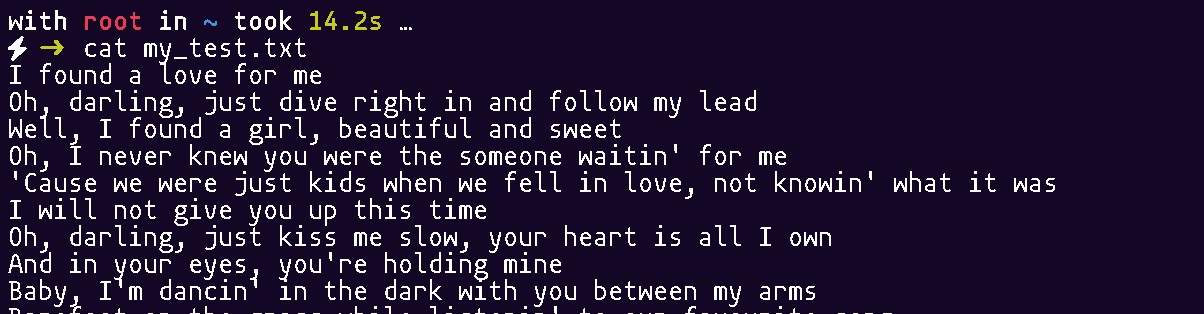
1. **ls** -> Lists files and directories in the current directory.

Syntax: ls [options] [directory]



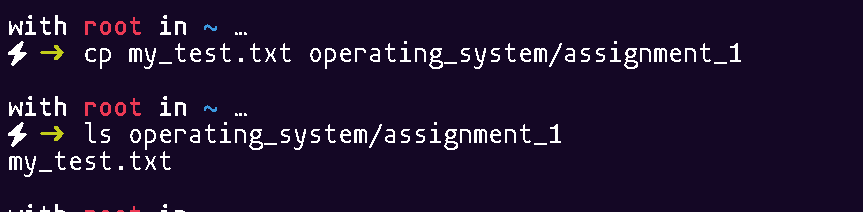
1. **cat** -> Displays the content of files.

Syntax: cat [file]



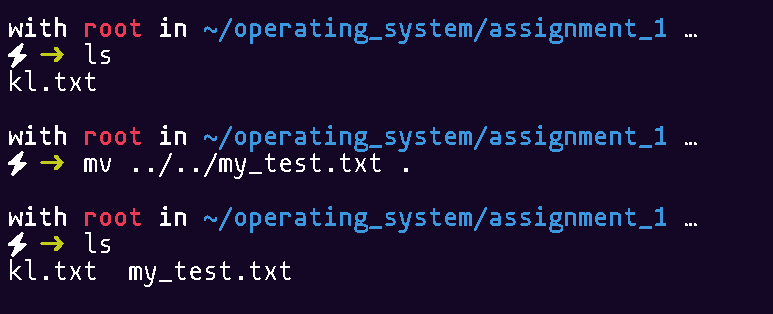
1. **cp** -> Copies files or directories from one location to another.

Syntax: cp [source] [destination]



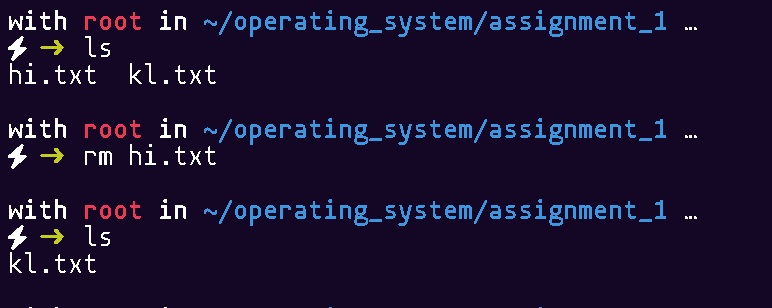
1. **mv** -> Moves or renames files or directories.

Syntax: mv [source] [destination]



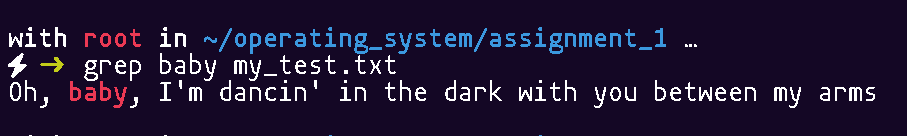
1. **rm** -> Removes files or directories.

Syntax: rm [options] [file]



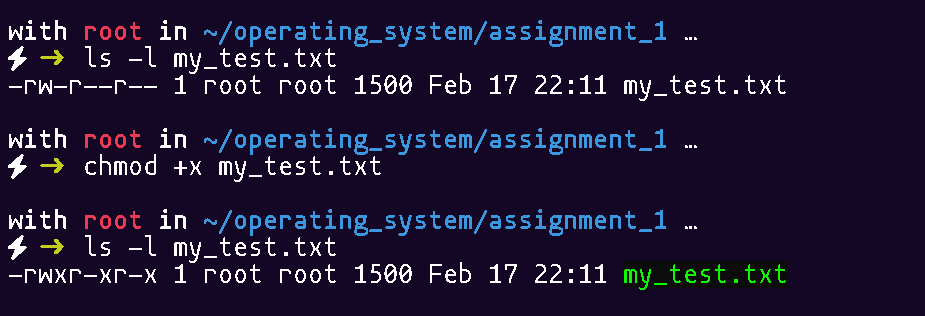
1. **grep** -> Searches for a specified pattern within files.

Syntax: grep [options] "pattern" [file]



1. **chmod** -> Changes the permissions of a file or directory.

Syntax: chmod [permissions] [file]



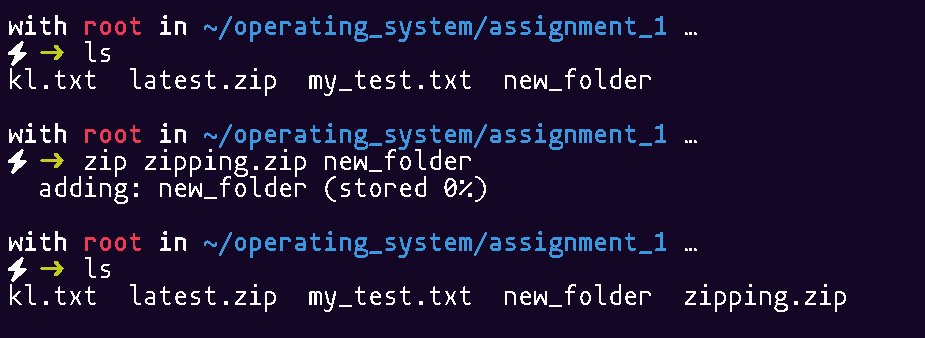
1. **wget** -> Downloads files from the internet.

Syntax: wget [URL]



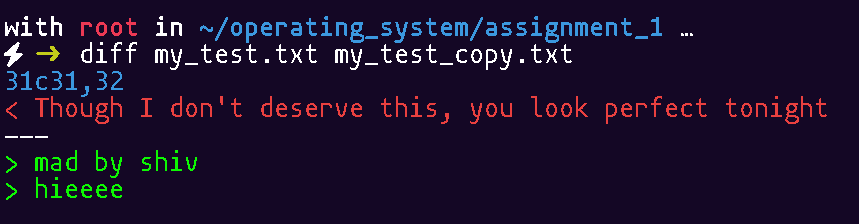
1. zip -> Compresses files into a ZIP archive.

Syntax: zip [options] [archive-name] [files]



1. diff -> Compares two files line by line and shows the differences.

Syntax: diff [file1] [file2]



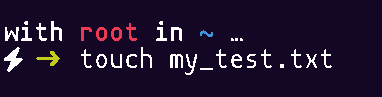
Question 3) What are the file manipulation commands explain it 1? Give 5 commands with syntax & examples?

**File Manipulation Commands**: These commands are specifically designed for working with files and directories. They allow you to create, delete, copy, move, and manipulate files or directories within the file system. File manipulation commands are crucial for organizing, managing, and accessing files. For instance, touch creates new empty files or updates file timestamps, cp copies files or directories, mv moves or renames them, and rm deletes files or directories. Other commands, like mkdir (make directories) and rmdir (remove empty directories), are also essential in managing the file system.

Examples

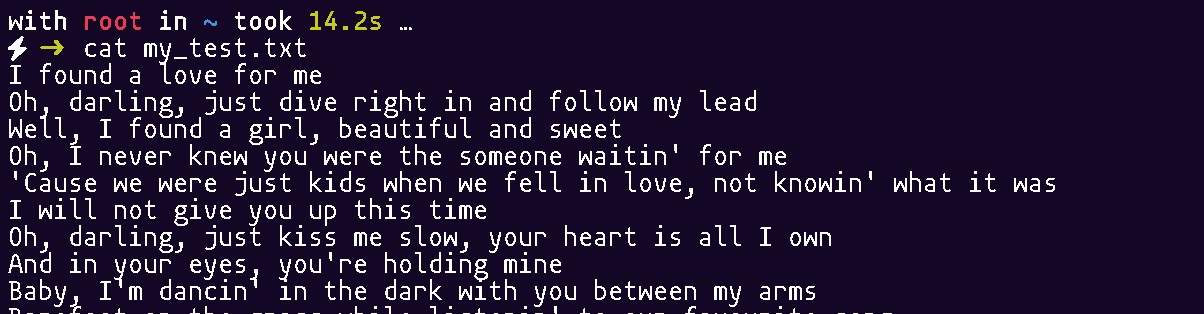
1. touch -> Creates an empty file if it doesn’t exist, or updates the timestamp of an existing file.

Syntax: touch [filename]



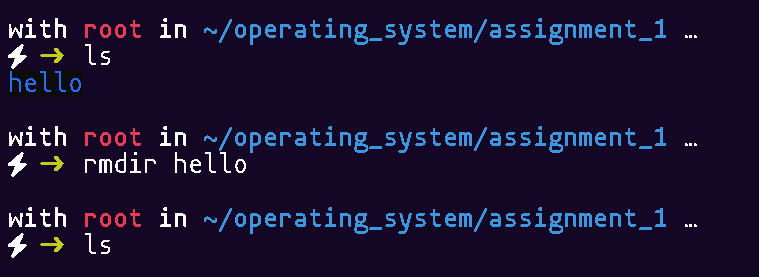
1. cat -> Displays the content of files.

Syntax: cat [filename]



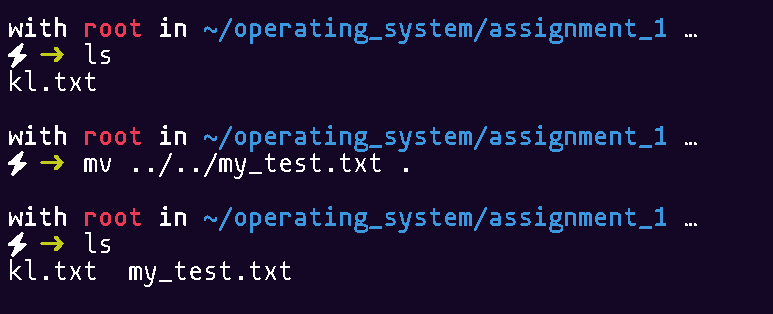
1. rmdir -> Removes empty directories.

Syntax: rmdir [directory]



1. mv -> Moves or renames files or directories

Syntax: mv [source] [destination]



1. rm -> Removes files or directories

Syntax: rm [filename]

