Name: Sarvesh Surve
Roll No. 73
Class/ Sem: SE/ IV
Experiment No.2
Linux shell script 2.1 Write shell scripts to do the following: a. Display OS version, release number, kernel version b. Display top 10 processes in descending order c. Display processes with highest memory usage. d. Display current logged in user and log name. Display current shell, home directory, operating system type, current path setting, current working directory
Date of Performance:
Date of Submission:
Marks:
Sign:

Aim: Linux shell script 2.1 Write shell scripts to do the following:

Objective:

a. Display OS version, release number, kernel version
b. Display top 10 processes in descending order

c. Display processes with highest memory usage.

d. Display current logged in user and log name.

e. Display current shell, home directory, operating system type, current path setting, current working directory

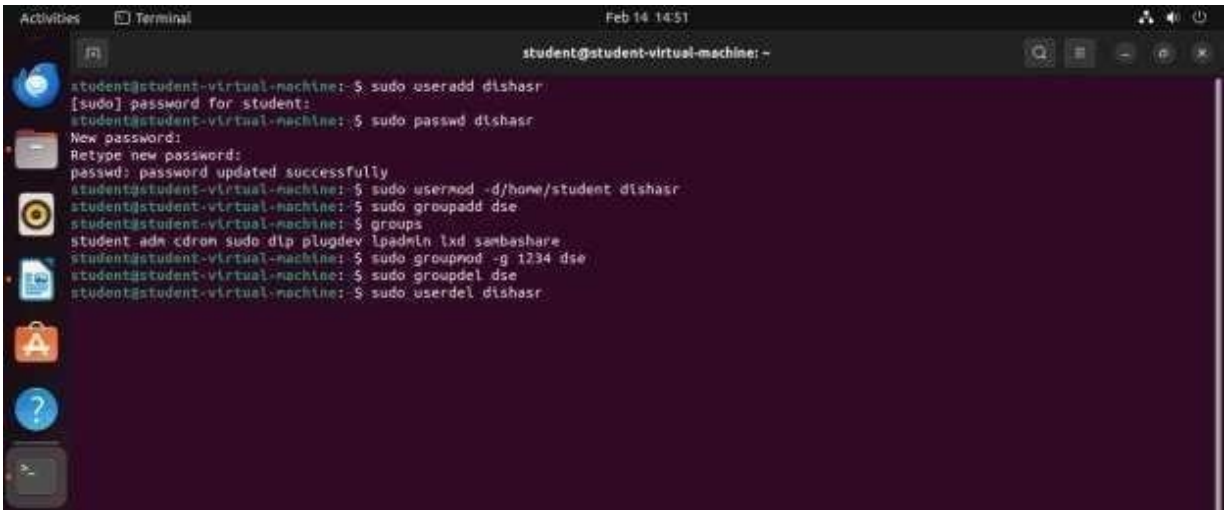
Theory:

Shell is a user program, or its environment is provided for user interaction. It is a command prompt within Linux where you can type commands. It is a program that takes your commands from the keyboard and gives them to the OS to perform. Shell is not part of system KERNAL but it uses system KERNAL to execute programs, create files,etc. A Shell Script is a text file that contains a sequence of commands for a UNIX based OS. It is called a Shell Script because it combines into a "Script" in a single file a sequence of commands, that would otherwise have to be presented to the system from a keyboard one at a time. A Shell Script is usually created for command sequences for which a user has a repeated need. You initiate the sequence of commands in Shell Script by simply entering the name of the Shell Script on a command line.
Types of Shell Script :-

1. sh - Simple Shell
2. bash - Bourne Again Shell
3. ksh - Korne Shell
4. csh - C Shell
5. ssh - Secure Shell

To use a particular Shell type the Shell name at the command prompt. Eg:- $csh - It will switch the current Shell to C Shell. To view the current Shell that is being used, type echo $ SHELL at the command prompt.

Output:


Activities
Terminal
Feb 14 14:51
student@student-virtual-machine: ~
student@student-virtual-machine: $ sudo useradd dishasr
[sudo] password for student:
student@student-virtual-machine: $ sudo passwd dishasr
New password:
Retype new password:
passwd: password updated successfully
student@student-virtual-machine: $ sudo usermod -d/home/student dishasr
student@student-virtual-machine: $ sudo groupadd dse
student@student-virtual-machine: $ groups
student adm cdrom sudo dip plugdev lpadmin lxd sambashare
student@student-virtual-machine: $ sudo groupmod -g 1234 dse
student@student-virtual-machine: $ sudo groupdel dse
student@student-virtual-machine: $ sudo userdel dishasr

Conclusion:

What is a Shell ?

A shell is a command-line interface (CLI) program that serves as the primary interface between a user and anoperating system. It allows users to interact with the system by typing commands in a text-based environment rather than relying solely on graphical user interfaces (GUIs). Shells interpret user input, execute commands, and manage the execution of programs. They provide functionalities such as command execution, input/output redirection, piping (connecting the output of one command to the input of another), scripting (automating sequences of commands), and environment variable manipulation. Shells also typicallyprovide features like command history, tab completion, and customizable prompt settings to enhance user productivity. Common examples of shells include Bash (Bourne Again SHell), Zsh (Z shell), and PowerShell, each with its own syntax, features, and extensions. Overall, shells play a crucial role in enablingusers to interact with the operating system efficiently and perform various tasks ranging from simple file manipulation to complex system administration tasks.

Name different types of shell

There are several types of shells available in Unix-like operating systems, each with its own features andcapabilities.

1. **Bourne Shell (sh):** Developed by Stephen Bourne at AT&T Bell Labs, the Bourne Shell was one of the earliest Unix shells. It provides basic command-line interface functionality and scripting capabilities. While it lacks some of the advanced features of later shells, it remains a fundamental component of Unix systems.

2. **Bourne-Again Shell (bash):** Created as an enhanced version of the Bourne Shell, bash has become oneof the most widely used shells in Unix-like operating systems. It offers features such as command-line editing, history, aliases, and programmable completion, making it powerful for both interactive use and scripting.

3. **C Shell (csh):** Developed by Bill Joy at the University of California, Berkeley, the C Shell provides a syntax similar to the C programming language. It offers interactive features like command-line history and aliases. While popular among some users, its scripting capabilities are considered less robust compared to bash.

4. **Korn Shell (ksh):** Developed by David Korn at AT&T Bell Labs, the Korn Shell combines features from the Bourne Shell and the C Shell, offering a rich set of programming features for scripting. It providesadvanced scripting constructs, command-line editing, and job control, making it popular among power usersand system administrators.

5. **Z Shell (zsh):** Zsh is an extended version of the Bourne Shell with additional features such as advanced tab completion, spelling correction, and customizable prompts. It aims to provide an interactive experience with enhanced usability and productivity for users.

6. **Fish Shell (fish):** Fish, short for "friendly interactive shell," is designed to be user-friendly and intuitivefor interactive use. It offers features like syntax highlighting, autosuggestions, and powerful scripting capabilities while focusing on simplicity and ease of use.