**Experiment No.: 1**

**Title:** **Exploring basic Commands of UNIX**

|  |
| --- |
| **Aim and Objective of the Experiment:** |
| To Explore basic commands under Unix/Linux operating system. |

|  |
| --- |
| **COs to be achieved:** |
| **CO1:** Explain the fundamental concepts of operating system Illustrate and analyse the process, threads, process scheduling and thread scheduling. |

|  |
| --- |
| **Theory:** |
| 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 emacs editor, xv (an image viewer), gcc (a C compiler), g++ (a C++ compiler), xfig (a drawing package), latex (a powerful typesetting language) and soffice (Star Office, which is an MS-Office style clone that can read and write Word, Excel and PowerPoint files).  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/Circuit Diagram/Block Diagram:** |
|  |

|  |
| --- |
| **Post Lab Subjective/Objective type Questions:** |
| 1. **Explain any two-text file operation and process management unix command with its syntax.**   Text File Operation Commands –   * + cat (Concatenate and Display File Content): Description: Used to create, view, and concatenate files.   Syntax: cat filename e.g. cat file1.txt   * + grep (Search for a Pattern in a File): Description: Used to search for specific patterns in a file.   Syntac: grep "pattern" filename e.g. grep "error" log.txt  Process Management Commands –   * + ps (Process Status): Displays information about active processes.   Syntax: ps aux e.g. ps -ef   * + kill (Terminate a Process): Sends a signal to terminate a process using its Process ID (PID).   Syntax: kill PID e.g. kill 1234   1. **State the different functions of the operating systems.**    * Process Management      + Schedules processes and handles multitasking.      + Allocates CPU time to different tasks.    * Memory Management      + Allocates and deallocates memory to processes.      + Manages virtual memory and paging.    * File System Management      + Controls file operations like read, write, and access permissions.      + Organizes files in directories.    * Device Management      + Manages input/output (I/O) devices like printers, keyboards, and storage.      + Uses device drivers for communication. |

|  |
| --- |
| **Conclusion:** |
| In this experiment, we explored fundamental UNIX commands for text file operations and process management, such as cat, grep, ps, and kill. This hands-on practice enhanced our understanding of UNIX command-line operations. |

|  |
| --- |
| **Signature of faculty in-charge with Date:** |