|  |  |
| --- | --- |
| **Sr. No.** | **Aim Of the Practical** |
| 1 | **Study Practical:** **Unix Architecture** Here is a basic block diagram of a Unix system −  Unix Architecture  The main concept that unites all the versions of Unix is the following four basics −   * **Kernel** − The kernel is the heart of the operating system. It interacts with the hardware and most of the tasks like memory management, task scheduling and file management. * **Shell** − The shell is the utility that processes your requests. When you type in a command at your terminal, the shell interprets the command and calls the program that you want. The shell uses standard syntax for all commands. C Shell, Bourne Shell and Korn Shell are the most famous shells which are available with most of the Unix variants. * **Commands and Utilities** − There are various commands and utilities which you can make use of in your day to day activities. **cp**, **mv**, **cat**and **grep**, etc. are few examples of commands and utilities. There are over 250 standard commands plus numerous others provided through 3rd party software. All the commands come along with various options. * **Files and Directories** − All the data of Unix is organized into files. All files are then organized into directories. These directories are further organized into a tree-like structure called the **filesystem**.     **B. Types of OS**  **C. Flavors of LINUX** |
| 2 | Study of Unix Architecture and the following Unix commands with option:   |  |  | | --- | --- | | **User Access:** | login, logout, passwd, exit | | **Help:** | man, help | | **Directory:** | mkdir, rmdir, cd, pwd, ls, mv | | **Editor:** | vi, gedit, ed, sed | | **File Handling / Text Processing:** | cp, mv, rm, sort, cat, pg, lp, pr, file, find, more, cmp, diff, comm, head, tail, cut, grep, touch, tr, uniq | | **Security and Protection:** | chmod, chown, chgrp, newgrp | | **Information:** | learn, man, who, date, cal, tty, calendar, time, bc, whoami, which, hostname, history, wc | | **System Administrator:** | su or root, date, fsck, init 2, wall, shut down, mkfs, mount, unmount, dump, restor, tar, adduser, rmuser | | **Terminal:** | echo, printf, clear | | **Process:** | ps, kill, exec | | **I/O Redirection** (<, >, >>), **Pipe** ( | ), \*, gcc | |  1. **User Access**   **//Here Sudo is for superuser access**  **sudo login-----login to a different user in the terminal**    **logout -----used to logout of user in terminal just like exit**  **gnome-session-quit -----logout the user**    **passwd-----used to change password of the user which is currently active**    **exit ------logout the user in the terminal which is logged in by login command same as logout**     1. Also used to close the terminal window 2. **Help**   **man-----gives the manual page of command that is typed after it**  Eg. man login    **help-----when written in the form help (command) it gives info about the specific command**  Eg. help logout     1. **Directory**   **mkdir-----used to create a folder**  Eg. mkdir samplefolder    **rmdir-----used to delete a folder**  Eg. rmdir samplefolder    **cd-----changes the present working directory**    **pwd-----displays the present working directory**    **ls-----displays the list of files in current directory**    **mv ------ move (rename) files**     1. **Editor**   **vi----- a programmer’s text editor**    **gedit------ opens a new gedit file(text file)**  **ed----- open a line oriented text editor in terminal**  **sed-----stream editor for transforming text**   1. **File Handling and Text Processing**   **cp-----it is used to copy a file or a folder**  cp source\_filename destination\_directory      **mv----- used to move a file or folder**  mv source\_filename destination\_directory  eg. mv sample tutorial  **rm------used to delete a file**  Eg. rm sample  **Sort-----used to sort lines in a file**  \*original file    \*sort sample    **Cat------Used to concat data of multiple files**  \*sample file    \*sample1 file    \*cat sample sample1    **pg ----- browse pagewise through text files**  **lp-----used to print files**  \*lp sample    Didn’t printed as no printer available  **Pr-----Convert text files for printing**  \*pr sample    **File-----Determine file type**  \*file sample    **Find-----Searches for file in the directory**  \*find sample    **More----- file perusal filter for crt viewing**  \*more sample    **cmp ----- compare two files byte by byte**  \*cmp sample sample1    **diff ------ compare files line by line**  \*diff sample sample1    **comm ----- compare two sorted files line by line**  **head ----- output the first part of files**  **tail ----- output the last part of files**  **cut----- displays the bytes at specific positions from every line in a file**  \*cut -b 1 sample    the reason on this output is that h is first byte in first line , m is first byte in second line, I is first byte in third line    **Grep -----print lines that contains a specific pattern of text**  \*grep is sample    **touch ----- change file timestamps**  **tr ----- translate or delete characters or modify them**    **uniq ------ report or omit repeated lines**       1. **Security and Protection**   **Chmod-----used to change read and write permissions of a file**  \*chmod u=r sample  //this command makes the file only readable i.e user=read in command,,,,, group=g, others=o, w=write, x=xecute  chmod u=r sample    **chown ----- change file owner**  Eg. sudo chown root sample  Here root is name of user  Sample is name of file  Sudo is for superuser access        The file is now under root owner  **Chgrp-----Used the changed the group of file**  **Newgrp-----Used to login to a new group**   1. **Information**   **who ----- show who is logged on**  **\***who    **Date-----gives the current date and time**    **Cal-----gives the calendar**    **tty ----- print the file name of the terminal connected to standard input**    **Calendar ----- reminder service**    **time ------ run programs and summarize system resource usage**    **bc ----- An arbitrary precision calculator language**    **Whoami-----gives the name of user which is logged in**    **Which-----locates the location of a command**    **Hostname-----shows the name of the host**      **History-----gives the history of all commands you typed**    **wc ----- print newline, word, and byte counts for each file**  **\***wc sample     1. **System Administrator (need to write sudo su before using all commands)**   **Su or root----- gives you superuser or root rights for commands**    **fsck ----- check and repair a Linux filesystem**  **init-----init basically defines how the system has to boot up**  how to write:- init 1 , init 2 ,init 3   |  |  |  | | --- | --- | --- | | **Run level** | **Name** | **Description** | | 0 | Halt | Shuts down the system | | 1 | Single-user mode | Mode for administrative tasks. | | 2 | Multi-User Mode | Does not configure network interfaces and does not export networks services | | 3 | Multi-User Mode with Networking | Starts the system normally | | 4 | Not used / user definable | For special purposes | | 5 | Start the system normally with GUI display manager | Run level 3 + display manager | | 6 | Reboot | Reboots the system | | s or S | Single-user mode | Not configure network interface or start daemon |   **wall ----- write a message to all users**  **shutdown ----- Halt, power-off or reboot the machine**    **mkfs ----- build a Linux filesystem**    **mount ----- mount a filesystem**  **adduser-----used to add a new user or group to the os**    **userdel-----used to delete a user**     1. **Terminal**   **Echo-----used to print a message**    **Printf----- format and print data (mostly used in scripting with variables)**  **Clear-----clear the terminal screen**  **After typing clear**     1. **I/O Redirection**  |  |  | | --- | --- | | ***command*** **> *file*** | Redirection of output | | ***command*** **>> *file*** | Appends output to specified file | | ***command*** **< *file*** | Redirection of input | | ***command*** **2 > *file*** | Redirection of error messages (**stdin**, **stdout** and **stderr** are associated with numerical values 0, 1 and 2 - called file descriptors) |  Some Examples Using Redirection Commands To send the output produced by the **who** command to the file called **wholist**:   |  |  | | --- | --- | | *Type:* | **who > wholist** |   To create a file called **demo** :   |  |  | | --- | --- | | *Type:* | **cat > demo** |   To run the program **myprog** taking input from the file called **data** and sending the output to the file called **results** :   |  |  | | --- | --- | | *Type:* | **myprog < data > results** | |
| 3. | 1. Write a shell script which calculatesnth Fibonacci number where n will be provided as input when prompted.   **Solution:**  echo "Enter the number of terms you want "  read n  a=0  b=1  echo -n $a $b  for(( i=2;i<$n;i++ ))  do  c=$(( $a + $b ))  a=$b  b=$c  echo -n " " $c  done  echo " "  **Output:**     1. Write a shell script which takes one number from user and finds factorial of a given number.   **Solution:**  echo -n "enter number::"  read no1  fact=1  for(( i=1;i<=$no1;i++ ))  do  fact=$(( $fact \* $i ))  done  echo "Factorial is "$fact  **Output:**     1. Write a shell script to sort the number in ascending order. (Using array).   echo "enter the size of the array"  read n  echo "enter Numbers in array:"  for (( i = 0; i < $n; i++ ))  do  read nos[$i]  done  for (( i = 0; i < $n ; i++ ))  do  for (( j = 0; j < $i; j++ ))  do  if [ ${nos[$i]} -lt ${nos[$j]} ]; then  t=${nos[$i]}  nos[$i]=${nos[$j]}  nos[$j]=$t  fi  done  done  echo -e "\nSorted Numbers "  for (( i=0; i < $n; i++ ))  do  echo -n ${nos[$i]}" "  done  echo ""  **Output:** |
| 4 | 1. Write a shell script which will take a file name from the user and finds that whether the file is there or not in a current working directory and displays the appropriate message. 2. Write a shell script which compares two files given by the user and if both files are same then delete the second one, if not then merge the two files in a new file. |