**p1:Learn the use of basic UNIX commands –**

**a. To access information using date, history, man, who, whoami, uptime, finger,cal.**

**b. To display contents of files using cat, vi, more, head, tail, grep, cmp, wc**

**c. To manage files using cat, cp, ls, mv,rm, chmod, find**

**d. Process utilities using ps, pid, ppid, tty, time, kill, exit**

**e. Directory handling utilities using cd, mkdir, rmdir, mv, pwd**

while true

do

echo "implementing 10 unix commands"

echo "1.listing"

echo "2.display"

echo "3.calender"

echo "4.date"

echo "5.tty"

echo "6.user name"

echo "7.who"

echo "8.time taken by a process"

echo "9.make a directory"

echo "10.find a file"

echo "\*.exit"

echo "enter your choice"

read choice

case $choice in

1)echo "list of files using commands"

ls -l;;

2)echo "welcome to unix";;

3)cal;;

4)date;;

5)tty;;

6)uname;;

7)who;;

8)time sleep 2;;

9)echo "enter directory name"

read name ls -l;;

10)echo "enter a file"

read file

find $file;;

\*) exit;;

esac

done



p2:**Write a shell script that displays list of all the files in the current directory to which the user has read, write and execute permissions**

echo "The list of files with read, write and execute permission"

for file in \*

do

if [ -f $file ]

then

if [ -r $file -a -w $file -a -x $file ]

then

ls -l $file

fi

fi

done



**p3:Write a shell script that accepts a list of file names as its arguments, count and reports the occurrence of each word that is present in the first argument file on other argument files.**

echo "Enter the number of files:"

read n

echo "Enter the file names:"

read file

set $file

for i in `cat $1`

do

echo "word= $i"

echo "-----------"

for j in $@

do

echo "$j = `grep -o -i "$i" $j | wc -l`"

done

echo "-----------"

done



**p4:Write a shell script that accepts one or more file name as arguments and converts all of them to uppercase, provided they exist in the current directory.**

echo “Enter the file name:”

read file

for x in $file

do

if [ ! -f $x ]

then

echo “file not found!”

continue

fi

tr ‘[a-z]’ ‘[A-Z]’ < $x

done



**p5:Write grep commands to the following:**

**a. To select the lines from a file that has exactly 2 characters.**

**b. To select the lines from a file that has more than 2 blank spaces.**

while true

do

echo "enter the filename"

read file

echo "1. to select the lines from a file that has exactly 2 char"

echo "2. To select the lines from a file that has more than 2 spaces"

echo "3.exit"

echo "Enter your choice"

read ch

case $ch in

1) echo "lines that have only 2 char"

grep -n '\<..\>' $file;;

2) echo "lines that has more than 2 spaces are"

grep '[[:space:]]\{2,\}' $file;;

\*) exit;;

esac

done



p6:**Write a shell script which accepts two file names as arguments. Compare the contents. If they are same, then delete the second file.**

echo "Enter the first filename:"

read file1

echo "Enter the second filename:"

read file2

if [ -f $file1 ]

then

if cmp $file1 $file2

then

echo "The contents are same. Deleting the second file"

rm $file2

else

echo "The contents are different"

fi

else

echo "File/s not found"

fi



**p7:Write a shell script**

**a. to count number of lines in a file that do not contain vowels.**

**b. to count number of characters, words, lines in a given file.**

echo "Enter the filename:"

read file

count=0

awk '$0!~/[aeiou]/{ count++ }END{printf "The number of lines without vowels are: %d\n",count}' $file

echo "Enter the filename:"

read file

chars=`cat $file | wc -c`

words=`cat $file | wc -w`

lines=`grep -c "." $file`

echo "The file has $chars characters, $words words and $lines lines.

**p8:Write a shell script to list all the files in a given directory.**

echo "Enter the directory name:"

read dir

for file in "${dir}/"\*

do

echo $file

done



**p10:Write a shell script to read three text files in the current directory and merge them into a single file and returns a file descriptor for the new file.**

if [ $# -eq 3 ]

then

cat $1 $2 $3 > merged.txt

echo "Files merged and named merged.txt"

cat merged.txt

else

echo "Supply exactly three files"

fi



cat merged.txt