**Week-1**

Session-1

1. Log into the system

**Sol:** $ login: <user name>

$ password: \*\*\*\*\*\*

1. Use vi editor to create a file called myfile.txt which contains some text.

**Sol:** $ vi myfile.txt

**[Press i to go in insert mode]**

~ Unix is Case Sensitive

~ Never leave the Computer without logging out when you are working in a time sharing or network environments.

Press <Esc>

: wq <ENTER>

1. correct typing errors during creation.

**Sol:** $ vi myfile.txt

**[Press i to go in insert mode]**

~ unix is Case Sensitive

Press<Esc>

: wq <ENTER>

1. Save the file

Press<Esc>

: wq <ENTER>

1. logout of the system

**Sol: $**exit

# Session 2

1. Log into the system

**Sol:** $ login: <user name>

$ password: \*\*\*\*\*\*

1. open the file created in session 1

**Sol:** $ vi myfile

~ Unix is Case Sensitive

~ Never leave the Computer without logging out. ~ Shell Programming c) Add some text

**Sol:** $ vi myfile

~ Unix is Case Sensitive

~ Never leave the Computer without logging out when you are working in a time sharing or network environments. ~ Shell Programming

1. Change some text **Sol:** $ vi myfile

~ Unix is Case Sensitive

~ Shell Programming

1. Delete some text

**Sol:** $ vi myfile

~ Unix is Case Sensitive

~ Shell Programming

1. Save the Changes

**Sol: press [Esc]**

: wq

1. Logout of the system

**Sol: $**exit

**Week-2**

1. Log into the system

**Sol:** $ login: <user name>

$ password: \*\*\*\*\*\*

1. Use the cat command to create a file containing the following data. Call it mytable use tabs to separate the fields.

1425 Ravi 15.65

4320 Ramu 26.27

6830 Sita 36.15

1450 Raju 21.86

**Sol: $ cat > mytable**

1425 <tab> Ravi <tab> 15.65 <tab>

4320 <tab> ramu <tab> 26.27 <tab>

6830 <tab> sita <tab> 36.15 <tab>

1450 <tab> Raju <tab> 21.86 <tab>

Press Ctrl+D to quit enter

1. Use the cat command to display the file, mytable.

**Sol: $ cat mytable**

|  |  |  |
| --- | --- | --- |
| 1425 | Ravi | 15.65 |
| 4320 | ramu | 26.27 |
| 6830 | sita | 36.15 |
| 1450 | Raju | 21.86 |

1. Use the vi command to correct any errors in the file, mytable.

**$ vi mytable**

|  |  |  |
| --- | --- | --- |
| 1425 | Ravi kumar | 15.65 |
| 4320 | ramu Singh | 26.27 |
| 6830 | sita kumara | 36.15 |
| 1450 | Raju singh | 21.86 |

**press [Esc]**

: wq [Enter]

1. Use the sort command to sort the file mytable according to the first field. Call the sorted file mytable

**$ sort -k 1n mytable**

1425 Ravi kumar 15.65

1450 Raju singh 21.86

4320 ramu Singh 26.27

6830 sita kumara 36.15

1. Print the file mytable

**Sol: $ lpr mytable**

1. Use the cut and paste commands to swap fields 2 and 3 of mytable. Call it mytable

$ cut –f1 mytable > temp1

$ cut –f2 mytable > temp2

$ cut –f3 mytable > temp3

$ paste temp1 temp3 temp2 > mytable

1. Print the new file mytable

**Sol: $ lpr mytable**

1. Logout of the system.

**Sol: $**exit

**Result: Week-2 questions are demonstrated with suitable examples successfully.**

**Week-3**

1) a) Login to the system

**Sol:** $ login: <user name>

$ password: \*\*\*\*\*\*

1. Use the appropriate command to determine your login shell **Sol:** $echo “the login shell is $SHELL”

1. Use the /etc/passwd file to verify the result of step b. **Sol:** $cat /etc/passwd

1. Use the who command and redirect the result to a file called myfile1. Use the more command to see the contents of myfile1. Soln: $who > myfile1

$more myfile1

1. **Use the date and who commands in sequence (in one line) such that the output of date will display on the screen and the output of who will be redirected to a file called myfile2. Use the more command to check the contents of myfile2.**

$date; who>myfile2

$more myfile2

2) a) Write a sed command that deletes the first character in each line in a file. **Sol:** $cat > sample my name is Ramu

I am studying B.Tech III year I am learning UNIX

^Z(press ctrl+Z) $sed "s/.//" sample

1. Write a sed command that deletes the character before the last character in each line in a file.

**Sol: $**sed "s/.\(.\)$/\1/" sample

1. Write a sed command that swaps the first and second words in each line in a file.

## sed -e "s/\([^ ]\*\) \*\([^ ]\*\)/\2 \1 /g" sample

**Week-4**

1. **Pipe your /etc/passwd file to awk, and print out the home directory of each user.**

cat /etc/passwd | awk “ { print $6}‟

1. **Develop an interactive grep script that asks for a word and a file name and then tells how many lines contain that word.**

Cat>filename

Day by day week by end

Week by week month by end

Month by month year by end

But friendship is never end

$ vi grep.sh

echo "Enter the pattern to be searched: "

read pattern

echo "Enter the file to be used: " read filename

echo "Searching for $pattern from file $filename"

echo "The selected records are: " grep "$pattern" $filename echo "The no.of lines contains the word( $pattern ) :" grep -c "$pattern" $filename

Output :

$ sh grep.sh

Enter the pattern to be searched: by

Enter the file to be used: filename Searching for by from filename The selected records are:

Day by day week by end

Week by week month by end

Month by month year by end

The no. of lines contains the words ( by ) :3

**WEEK 5**

1. Write a shell script that takes a command –line argument and reports on whether it is directory, a file, or something else.
2. 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.
3. Write a shell script that determines the period for which a specified user is working on the system.

* 1. Write a shell script that takes a command –line argument and reports on whether it is directory, a file, or something else.

echo " enter file" read str if test -f $str

then echo "file exists and it is an ordinary file" elif test -d $str then echo "directory file" else echo "not exists" fi

* 1. 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.

vi week5B.sh

echo –n “Enter file name”

read filename if [! –f $filename ]

then

echo “filename $filename does not exist”

exit 1 fi

tr ‘[:lower:] ‘ ‘[:upper:]’ < filename

* 1. a shell script that determines the period for which a specified user is working on the system.

echo "Enter the USER NAME : "

read user last $user

**NOTE:**

**Last command:** The last command reads listing of last logged in users from the system file called /var/log/wtmp

**Week-6**

1. Write a shell script that accepts a file name starting and ending line numbers as arguments and displays all the lines between the given line numbers.
2. Write a shell script that deletes all lines containing a specified word in one or more files supplied as arguments to it.

* 1. echo "enter the filename"

read fname

echo "enter the starting line number"

read s

echo "enter the ending line number"

read n

sed -n $s,$n\p $fname | cat > newline

cat newline

* 1. Write a shell script that deletes all lines containing a specified word in one or more files supplied as arguments to it.

echo “enter file name” read file echo “enter word”

read

word

echo

“

file before removing

**$word**

:

”

cat

**$file**

grep

-

v

-

i

**$word**

**$file**

>

test

mv

test

**$file**

echo

“

file after removing

**$word**

:

”

cat

**$file**

**Week**

**-**

**7**

**a) Write a shell script that computes the gross salary of a employee according to the following rules:**

**i)If basic salary is < 1500 then HRA =10% of the basic and DA =90% of the basic.**

**ii)If basic salary is >=1500 then HRA =Rs500 and DA=98% of the basic.**

$ vi gsalary.sh echo "enter basic salary" read bsal if [ $bsal -lt 1500 ]

then

gsal=$((bsal+((bsal/100)\*10)+(bsal/100)\*90)) echo "The gross salary is $gsal" fi

if [ $bsal -ge 1500 ]

then

gsal=$((bsal+500+((bsal/100)\*98))) echo "The gross salary is $gsal" fi

**b) Write a shell script that accepts two integers as its arguments and computes the value of first number raised to the power of the second number.**

vi pow.sh

echo "Enter the integer value :"

read int1

echo "Enter the power of that integer:" read int2 pv=$int1 i=1

while [ $i -lt $int2 ]

do

pv=`expr $pv \\* $int1` i=` expr $i + 1 ` done

echo "The value of first number to the power of the second number :"

echo "$pv"

**Week 8**

**a) Write an interactive file-handling shell program. Let it offer the user the choice of copying, removing, renaming, or linking files. Once the user has made a choice, have the program ask the user for the necessary information, such as the file name, new name and so on.**

**vi Krishna.txt**

No one is waste, No one is greate

Without work nothing is possible

With hard work nothing is impossible

So do your work perfectly

**vi Vivekananda.txt** Be a Hero.

Always Say, 'I have no fear'.

**vi week8a.sh** while true do

echo "\*\*\*\*\*\*\*MENU\*\*\*\*\*\*\*\*\*"

echo

" 1. List of files.

1. Copying files.
2. Removing files.
3. Renaming files.
4. Linking files.

press [CTRL+C] TO EXIT" echo "enter your choice " read ch

case "$ch" in

1. ) echo "The list of file names." ls -l || echo "These are file";;

1. echo "Enter the old filename."

read ofile echo "Enter the new file name." read nfile cp $ofile $nfile && echo "Copied sucessfully." || echo "Copied is not possible." ;;

1. ) echo "Enter the file name to remove." read rfile

rm -f $rfile && echo "Successfully removed." ;;

1. ) echo "Enter the old file name."

read ofile

echo "Enter the new file name." read nfile

mv $ofile $nfile && echo "The file $ofile name renamed to $nfile." || echo "You cann't Rename the file.".;;

1. ) echo "Enter the original filename." read ofile

echo "Enter the new filename to link a file." read lfile

ln $ofile $lfile && echo "Creat the linking file Sccessfully." || echo "You cann't Linking the file.";; \* )

echo "Invalid option." Echo " Enter correct choice." esac done

**b. Write shell script that takes a login name as command – line argument and reports when that person logs in**

vi week8b.sh

echo "Enter the user name:"

read user

last $user && echo "these are the datails of user $user"

**c) Write a shell script which receives two file names as arguments. It should check whether the two file contents are same or not. If they are same then second file should be deleted.**

vi week8c.sh

echo "enter the first file name" read file1

echo "enter the second file name" read file2

cmp $file1 $file2 && rm $file2 if [ -e $file1 ]

then if [ ! -e $file2 ]

then

echo " the two files contents are same. so $file2 is deleted" else echo " the two file contents are not same and $file2 not deleted" fi else

echo "$file1 is not existed" fi

**Week-9**

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

echo "The list of file names in the current directory:" for file in \* do

if [ -f $file ] then

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

then

ls -l $file fi fi

done

**b) Develop an interactive script that ask for a word and a file name and then tells how many times, that word occurred in the file.**

Cat>sample.txt

Day by day week by end

Week by week month by end

Month by month year by end

But friendship is never end

**Press Ctrl+d**

$ vi grep.sh

echo "Enter the pattern to be searched: "

read pattern

echo "Enter the file to be used: " read filename

echo "Searching for $pattern from file $filename" echo "The selected records are: " grep "$pattern" $filename echo "The no.of lines contains the word( $pattern ) :" grep -c "$pattern" $filename

**Output :**

$ sh grep.sh

Enter the pattern to be searched: by

Enter the file to be used: sample.txt Searching for by from sample.txt  **The selected records are:**

Day by day week by end

Week by week month by end

Month by month year by end

The no. of lines contains the words ( by ) :3

**Week-10**

**Write a shell script to perform the following string operations: i) To extract a sub-string from a given string.**

echo "Enter the string"

read str strlen=${#str} echo "the length of the given string is:$strlen" echo "Enter the position of substring in main string" read s1

echo "Enter the ending position of substring in main string" read f1

echo $str|cut -c$s1-$f1

**ii) To find the length of a given string**

str="Hare Ram Singh" strlen=${#str} echo "lenth of $str is:$strlen"