

**NRI INSTITUTE OF INFORMATION  
SCIENCE  
& TECHNOLOGY BHOPAL**



**DEPARTMENT OF INFORMATION  
TECHNOLOGY**

**LAB MANUAL**

**Open Source Software LAB  
(Linux and R)  
(IT-407)**

**INFORMATION TECHNOLOGY (IT)**

## **EXPERIMENT NO: 1.**

**NAME OF THE EXPERIMENT:** Write a Shell Script that accepts a file name, starting and ending line numbers as arguments and displays all lines between the given line numbers.

**AIM:** By using bash shell for writing the shell scripts, first create any file in that file read any line number as starting and ending line number and print all lines between that line numbers.

### **ALGORITHM:**

Step 1: Create a file with 5-6 lines of data

File can be created by vi text.txt or  
cat text.txt Step 2: Now write a shell script  
with

vi prg1.sh

step3: Check the no of arguments for

shell script if 0 arguments then

print no arguments else if 1

argument then print 1 argument

else if 2 arguments then print 2 arguments

else check for file is there or not (if file is not there print file does  
not exists) else sed -ne "\$2,'\$3' p' \$1

sed is one of powerful filter (stream editor)

-e default option (script on command line)

-n suppresses automatic output

\$2 first line number passed \$3 2<sup>nd</sup> line number passed

p is a print command (prints current content to the pattern space).

\$1 is name of file from which we are printing data between the line  
numbers.

### **SOURCE CODE**

```
_[lakshmi@localhost ~]$
```

```
vi prg1.sh if [ $# -eq 0 ]
```

```
then
```

```
echo no
```

```
arguements
```

```
elif [ $# -eq
```

```
1 ]
```

```
then
```

```
echo one
```

```
arguement
```

```
elif [ $# -eq
```

```
2 ]
```

```
then
echo two
arguements
else
i
f

[

!

-
e

$
1

]

t
h
e
n
echo file does
not exist else
sed -ne
"$2','$3' p'
$1 fi
fi
```

## EXPECTED OUTPUT

```
lakshmi@localhost:~  
[lakshmi@localhost ~]$ vi prg1.txt  
[lakshmi@localhost ~]$ vi prg1.sh  
[lakshmi@localhost ~]$ vi prg1.sh  
[lakshmi@localhost ~]$ sh prg1.sh text.txt 2 5  
fine thank u  
bye  
hgfdghfd  
jff  
[lakshmi@localhost ~]$ █
```

## Viva Questions

1. Who developed UNIX?

- |                    |                                        |
|--------------------|----------------------------------------|
| 1. Ken Thompson    | 2. Dennis Ritchie,                     |
| 3. Douglas Maclroy | 4. <b>Ken Thompson, Dennis Ritchie</b> |

2. What is 'ps' command for?

- |                                |                                     |
|--------------------------------|-------------------------------------|
| 1. prints the status           | 2. <b>prints the process status</b> |
| 3. prints the execution status | 4. none                             |

3. Which command is used to terminate the process?

- |                |                |
|----------------|----------------|
| 1. ps          | 2. who         |
| 3. <b>grep</b> | 4. <b>kill</b> |

4. Shell is an interface between user and

- |                  |             |
|------------------|-------------|
| 1. <b>Kernel</b> | 2. hardware |
| 3. I/O           | 4. none     |

5. Which is of the following is not a type shell

- |            |               |
|------------|---------------|
| 1. korn    | 2. bourne     |
| 3. C-shell | 4. <b>All</b> |

## **EXPERIMENT NO: 2**

**NAME OF THE EXPERIMENT:** Write a shell script that deletes all lines containing a specific word in one or more files supplied as arguments to it.

**AIM:** By using bash shell for writing the shell scripts, first create any file with 5-10 lines of data, then specify any particular word, if that word is present in any lines of file those lines will be deleted.

### **ALGORITHM:**

Step 1: Create a file with 5-6 lines of data

    Create the 2 file f1 and f2 as vi f1 and vi f2

Step2: Now write a shell script with

    vi prg2.sh

step3: Check the no of arguments for shell script

    if 0 arguments then print no arguments

    else pattern=\$1 (word will be stored in pattern)

    for fname in \$\*

    for every filename in given files

    if it is a file if [ -f \$fname ] then

        print DELETING \$pattern FROM \$fname

        sed '/\$pattern/d' \$fname

        sed acts as filter if word is a file in any line that will be deleted

        '/' is used to represent regular expressions

        '/d' is a delete command in sed

        else print file NOT FOUND

### **SOURCE CODE:**

```
[lakshmi@localhost ~]$ vi prg2.sh
if [ $# -eq 0 ]
then
echo NO ARGUMENTS
else
pattern=$1
shift
for fname in $*
do
if [ -f $fname ]
then
echo DELETING $pattern FROM $fname
sed '/$pattern/d' $fname
```

```
else
echo $fname NOT FOUND
fi
done
fi
```

### **EXPECTED OUTPUT**

```
akshmi@localhost ~]$ sh prg2.sh
ARGUMENTS
akshmi@localhost ~]$ vi prg2.sh
akshmi@localhost ~]$ vi prg2.sh
akshmi@localhost ~]$ sh prg2.sh hai f1 f2
LETING hai FROM f1
ne
e

LETING hai FROM f2
ce color
ce day

akshmi@localhost ~]$ █
```

### **Viva Questions**

1. Which command is use for the copy in Unix?
  1. copy
  2. cp
  3. cpy
  4. none
- 2 .What is stand for IPC?
  1. Inter procedure communication
  2. Inter process communication
  3. Inter part compare
  4. None
- 3 .What is the stand for FIFO?
  1. First in First out
  2. File in File out
  3. First Inter First Out
  4. None

## **EXPERIMENT: 3**

**NAME OF THE EXPERIMENT:** Write a shell script that displays a list of files in current directory to which the user has read, write and execute permissions.

**AIM:** In the current directory we are able know which files have read, write and execute permissions.

### **ALGORITHM**

Step1: Here we are working with current directory

Shell script will be written at command line

`ls -l|grep '^rwx'`

ls is list command (list all the files in current directory)

-l long listing the files

grep acts as filter

it doesn't print the files which does not have read, write and execute permissions

^ represents negation.

### **SOURCE CODE:**

```
$ls -l | grep '^rwx'
```

### **EXPECTED OUTPUT**

```

[lakshmi@localhost ~]$ ls -l|grep '^.rwx'
[lakshmi@localhost ~]$ ls -l
total 36
-rw-rw-r-- 1 lakshmi lakshmi 26 Jun 29 14:42 f1
-rw-rw-r-- 1 lakshmi lakshmi 39 Jun 29 14:41 f2
-rw-rw-r-- 1 lakshmi lakshmi 22 Jun 29 15:01 f3
-rw-rw-r-- 1 lakshmi lakshmi 212 Jun 29 14:34 prg1.sh
-rw-rw-r-- 1 lakshmi lakshmi 203 Jun 29 14:58 prg2.sh
-rw-rw-r-- 1 lakshmi lakshmi 234 Jun 29 15:06 prg3.sh
-rw-rw-r-- 1 lakshmi lakshmi 159 Jun 29 15:23 prg4.sh
-rw-rw-r-- 1 lakshmi lakshmi 211 Jun 29 14:15 prg.txt
-rw-rw-r-- 1 lakshmi lakshmi 54 Jun 29 14:16 text.txt
[lakshmi@localhost ~]$ ls -l|grep '^.rw'
-rw-rw-r-- 1 lakshmi lakshmi 26 Jun 29 14:42 f1
-rw-rw-r-- 1 lakshmi lakshmi 39 Jun 29 14:41 f2
-rw-rw-r-- 1 lakshmi lakshmi 22 Jun 29 15:01 f3
-rw-rw-r-- 1 lakshmi lakshmi 212 Jun 29 14:34 prg1.sh
-rw-rw-r-- 1 lakshmi lakshmi 203 Jun 29 14:58 prg2.sh
-rw-rw-r-- 1 lakshmi lakshmi 234 Jun 29 15:06 prg3.sh
-rw-rw-r-- 1 lakshmi lakshmi 159 Jun 29 15:23 prg4.sh
-rw-rw-r-- 1 lakshmi lakshmi 211 Jun 29 14:15 prg.txt
-rw-rw-r-- 1 lakshmi lakshmi 54 Jun 29 14:16 text.txt
[lakshmi@localhost ~]$ chmod u+x f1
[lakshmi@localhost ~]$ ls -l|grep '^.rwx'
-rwxrw-r-- 1 lakshmi lakshmi 26 Jun 29 14:42 f1
[lakshmi@localhost ~]$ chmod g+x f1

```

---

```

[lakshmi@localhost ~]$ ls -l|grep '^.rwx'
-rwxrwxr-- 1 lakshmi lakshmi 26 Jun 29 14:42 f1
[lakshmi@localhost ~]$ █

```

## **METHOD 2:**

```

echo "enter the directory name"
read dir
if [ -d $dir ]
then
cd $dir
ls > f
exec < f
while read line
do
if [ -f $line ]
then
if [ -r $line -a -w $line -a -x $line ]
then
echo "$line has all permissions"
else
echo "files not having all permissions"
fi
fi
done
fi

```

## **EXPECTED OUTPUT**

```
student@ubuntu:~$sh prg3.sh
enter the directory name
dir1
ff has all permissions
files not having permissions
```

### **Method3:**

```
echo "List of Files which have Read, Write and Execute Permissions in Current Directory"
for file in *
do if [ -r $file -a -w $file -a -x $file ]
then
echo $file
fi
done
```

**INPUT:** sh prog3.sh

**OUTPUT:** List of Files which have Read, Write and Execute Permissions in Current Directory f1 f2 f3

### **Viva Questions**

1. What are states that the page can be in, after causing a page fault?

1. On a swap device and not in memory
2. On the free page list in the main memory,
3. In an executable file,
4. all

2. At what mode the fault handler executes?

1. execution mode
2. kernel mode
3. operation mode
4. none

3. Which one data structure is used for Demand Paging?

1. Page table entries,
2. Disk block descriptors,
3. Page frame data table (pfdata),
4. Sw

## **EXPERIMENT NO: 4**

**NAME OF THE EXPERIMENT:** Write a shell script that receives any number of file names as arguments checks if every argument is a file or directory, when it is a file, report no of lines in it.

**AIM:** In this shell script we will supply the file names or directory names as arguments if it is a file name then it prints no of lines in file, if it is a directory then it prints it is a directory file.

### **ALGORITHM:**

Step1: check files and directories in the current directory by ls command

Step2:create a shell script

```
vi prg3.sh
```

step3: Check the no of arguments for shell script

```
if 0 arguments then print no arguments
```

```
for fname in $*
```

```
for every filename in given files
```

```
if it is a file if [ -f $fname ] then
```

```
print it is a regular file
```

```
count=`wc -l $fname`
```

```
wc is used to count no of lines words characters
```

```
-l is used only to print lines.
```

```
print count
```

```
else
```

```
print it is a directory file.
```

### **SOURCE CODE**

```
[lakshmi@localhost ~]$ vi prg3.sh
if [ $# -eq 0 ]
then
echo NO ARGUMENTS
else
for fname in $*
do
if [ -f $fname ]
then
echo $fname IS A REGULAR FILE
count=`wc -l $fname`
echo NUMBER OF LINES IN $fname ARE :
echo $count
else
echo $fname IS A DIRECTORY FILE
fi
done
fi
```

## EXPECTED OUTPUT

```
lakshmi@localhost ~
[lakshmi@localhost ~]$ sh prg3.sh f3
f3 IS A REGULAR FILE
NUMBER OF LINES IN f3 ARE :
3 f3
[lakshmi@localhost ~]$ ls
f1 f2 f3 prg1.sh prg2.sh prg3.sh prg4.sh prg.txt text.txt
[lakshmi@localhost ~]$ sh prg3.sh lakshmi
lakshmi IS A DIRECTORY FILE
[lakshmi@localhost ~]$
```

## Viva Questions

1. What is stand for BSS in Unix?

- |                            |                            |
|----------------------------|----------------------------|
| 1. Block Started by Symbol | 2. Black Started by Symbol |
| 3. Black Started by Symbol | 4. none                    |

2. Which one is best in action between fork() and vfork()?

- |           |            |
|-----------|------------|
| 1. fork() | 2. vfork() |
| 3. both   | 4. none    |

3. Which symbol will be used with grep command to match the pattern pat at the beginning of a line?

- |                      |          |
|----------------------|----------|
| A. ^pat              | B. \$pat |
| C. pat\$             | D. pat^  |
| E. None of the above |          |

## **EXPERIMENT NO: 5**

**NAME OF THE EXPERIMENT:** 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.

**AIM:** In this shell script we will supply the file names as arguments, any word that is present in first argument file can counted and printed if it present in other argument files.

### **ALGORITHM:**

Step1: Check the no of arguments for shell script  
if 0 arguments then print no arguments  
step2: else translate each word in the first file is to be on separate line  
which will be stored in temp file  
step3: for i in \$\*  
for every filename in given files  
step 4: translate each word in the file is to be on separate line  
which will be stored in temp1 file  
step5: count no of lines in temp file assign it to j  
step6: initialize j=1  
step 7: while i<j  
extract the line that are common in both the file by using  
head and tail commands  
then apply the filter grep to count and print the lines  
which are common to files  
increment j  
step 8: end

### **SOURCE CODE**

```
if [ $# -eq 0 ]
then
echo "no arguments"
else
tr " " "\n"<$1>temp
shift
for i in $*
do
tr "" "\n"<$i>temp1
y=`wc -l < temp`
j=1
while [ $j le $y ]
do
x=`head -n $j temp | tail -l`
c=`grep -c "$x" temp1`
echo $x $c
j=`expr $j + 1`
```

done  
done  
fi

## EXPECTED OUTPUT

```
[lakshmi@localhost labprograms]$ sh laxmi5.sh sampletext2 sampletext3
hair 3
laxmi5.sh: line 12: [: 1+1: integer expression expected
[lakshmi@localhost labprograms]$ sh laxmi5.sh sampletext2 sampletext3 sampletext4
4
hair 3
laxmi5.sh: line 12: [: 1+1: integer expression expected
hair 2
laxmi5.sh: line 12: [: 1+1: integer expression expected
[lakshmi@localhost labprograms]$
```

## VIVA QUESTIONS

### 1. What is Shell Scripting ?

Shell scripting, in Linux or Unix, is programming with the shell using which you can automate your tasks. A shell is the command interpreter which is the interface between the User and the kernel. A shell script allows you to submit a set of commands to the kernel in a batch. In addition, the shell itself is very powerful with many properties on its own, be it for string manipulation or some basic programming stuff.

### 2. The command "cat file" gives error message "--bash: cat: Command not found". Why?

It is because the PATH variable is corrupt or not set appropriately. And hence the error because the cat command is not available in the directories present PATH variable.

### 3. How to find the length of a string in Linux?

```
$ x="welcome" $ echo ${#x} 7
```

### 4. What are the different timestamps associated with a file?

- Modification time:- Refers to the time when the file is last modified.
- Access time :- The time when the file is last accessed.
- Changed time :- The time when the attributes of the file are last changed.

### 5. How to get the list of files alone in a directory in Linux?

```
$ ls -lrt | grep ^-
```

## EXPERIMENT NO: 6

**NAME OF THE EXPERIMENT:** \_Write a shell script to list all of the directory files in a directory.

**AIM:** \_In this shell script we will supply the directory name as an argument then our program will return us the list of directories in the given directory

### ALGORITHM:

Step1: enter the name of the directory

    Read dir

Step2: if it is a directory

    Then list the files present in that directory

    By using ls command with -p option to list all directory files in a given directory

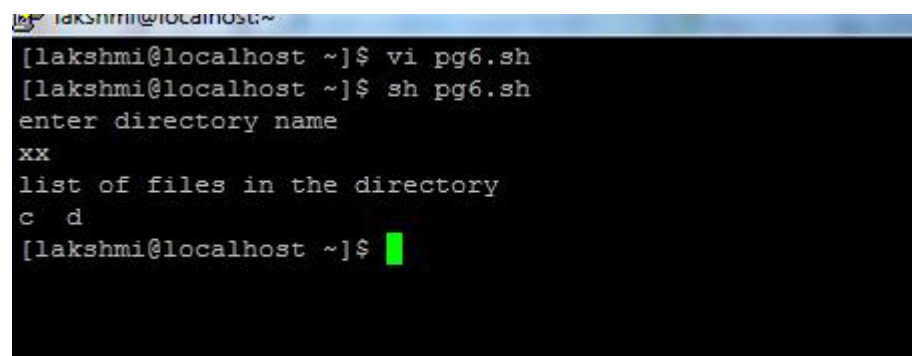
Step 3: else enter the directory name

Step 4: stop

### SOURCE CODE

```
echo "enter directory name"
read dir
if [ -d $dir ]
then
echo "list of files in the directory"
ls $dir
else
echo "enter proper directory name"
fi
```

### EXPECTED OUTPUT:



```
lakshmi@localhost:~$ vi pg6.sh
[lakshmi@localhost ~]$ sh pg6.sh
enter directory name
xx
list of files in the directory
c d
[lakshmi@localhost ~]$
```

## **Viva Questions**

**1. What are the different timestamps associated with a file?**

1)Modification time 2) Access time 3)Changed time

**2. How to get the list of files alone in a directory in Linux?**

`$ ls -lrt | grep ^-`

**3. Which command is used to sort the lines of data in a file in reverse order**

A.sort    B.sh    C.st    D.sort -r

## EXPERIMENT NO: 7

**NAME OF THE EXPERIMENT:** Write a shell script to find factorial of a given number.

**3.AIM:** By using bash shell we will write a shell script to find the factorial of a given number.

## ALGORITHM

Step 1: read any number to find factorial

Step 2: initialize fact=1 and i=1

Step 3: while i less than

```
do
    fact=fact* i
    i=i+1
done
```

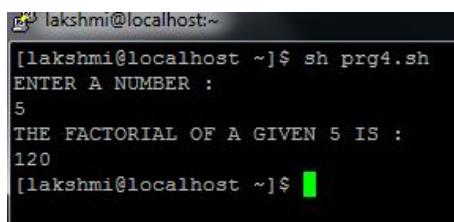
step 4:print fact

step 5:stop.

## SOURCE CODE:

```
echo ENTER A NUMBER :
read n
echo THE FACTORIAL OF A GIVEN $n IS :
fact=1
i=1
while [ $i -le $n ]
do
fact=`expr $fact \* $i`
i=`expr $i + 1`
done
echo $fact
```

## EXPECTED OUTPUT



```
lakshmi@localhost:~
[lakshmi@localhost ~]$ sh prg4.sh
ENTER A NUMBER :
5
THE FACTORIAL OF A GIVEN 5 IS :
120
[lakshmi@localhost ~]$
```

## Viva Questions

1. Which command is used to display the top of the file?

- A. cat    B. head    C. more    D. grep  
E. None of the above

2. Which command is used to copy all files having the string chap and any two characters after that to the progs directory?

- A. cp chap?? Progs    B. cp chap\* progs    C. cp chap[12] /progs/\*.\*    D. cp chap?? /progs/\*

3. Which command is used to change protection mode of files starting with the string emp and ending with 1,2, or 3?

- A. chmod u+x emp[1-3]    B. chmod 777 emp\*  
C. chmod u+r ??? emp    D. chmod 222 emp?

## **EXPERIMENT NO: 8**

**NAME OF THE EXPERIMENT:** Write awk script to count no of lines in a file that do not contain vowels.

**AIM:** By using bash shell we will write awk script to count the number of lines that do not contain vowels

### **ALGORITHM:**

Step 1: create a file with 5-10 lines of data

Step 2: write an awk script by using grep command to filter the lines that do not contain vowels

```
awk ' $0 !~/aeiou/ ,print $0}' file1
```

step3: count=count+1

step4:print count

step5:stop

### **5. SOURCE CODE:**

```
awk ' $0 !~/aeiou/ ,print $0}' file1
```

### **6. EXPECTED OUTPUT:**

```
$vi file1
```

```
pvwxyz sssx
```

```
hhhh kkkk
```

```
zzzzz bbbbb
```

```
$ awk -f prg.awk file1
```

```
3
```

### **Viva Questions**

1. Which command is used to remove a directory?

A. rd      B. rmdir      C. dldir      D. rdir

E. None of the above

2. Which of the following keys is used to replace a single character with new text?

A. S      B. s      C. r      D. C

E. None of the above

3. Which command is used to extract specific columns from the file?

A. cat      B. cut      C. grep      D. paste

## **EXPERIMENT NO: 9**

**NAME OF THE EXPERIMENT:** Write awk script to count no of characters, words, lines in a given file.

**AIM:** By using bash shell we will write awk script to count the number of lines, words, characters in a given file.

### **ALGORITHM**

Step 1: create a file with 5-10 lines of data

Step 2: write an awk script

    find the length of file

    store it in x

step3: count the no of fields (NF), store it in y

step4: count the no of records (NR), store it in NR

step5: print NR,x,y

step6: stop

### **SOURCE CODE:**

```
awk 'x+=length($0); y+=NF} end ,print NR, x,y}' filename
```

### **EXPECTED OUTPUT**

```
vi filename.txt
```

```
hai how r u
```

```
file thank u
```

```
bye
```

```
3 28 8
```

### **VIVA QUESTIONS:**

**1. How to find the last modified file or the newest file in a directory?**

```
$ ls -lrt | grep ^- | awk 'END{print $NF}'
```

**2. How to access the 10th command line argument in a shell script in Linux?**

\$1 for 1st argument, \$2 for 2nd, etc... For 10th argument, \${10}, for 11th, \${11} and so on.

**3. How to find the sum of all numbers in a file in Linux?** \$ awk '{x+=\$0}END{print x}'  
file

**4. How to delete a file which has some hidden characters in the file name?**

Since the rm command may not be able to delete it, the easiest way to delete a file with some hidden characters in its name is to delete it with the find command using the inode number of the file.

```
$ ls -li
```

## **EXPERIMENT NO: 10**

**NAME OF THE EXPERIMENT:** \_Write C program that makes a copy of a file using standard I/O and System calls.

**AIM:** \_By using bash shell we will write awk script to count the number of lines, words, characters in a given file.

### **ALGORITHM:**

```
Step 1:define BUFSIZE 1024
        char buf[BUFSIZE];
step 2: if (argc!=3)
        printf("Usage: cp <src> <dst>\n");
        return 1;
step 3:src = open(argv[1]);
step 4: if (src==-1)
        print "Unable to open %s\n", argv[1]
        return 1;
step 5: creat(argv[2]);
        dst = open(argv[2]);
step 6: if (dst==-1)
        print Unable to create %s\n", argv[2]
        return 1;
step 7: while ((amount = read(src, buf, BUFSIZE))>0)
        write(dst, buf, amount);
step 8: close(src);
        close(dst);
```

### **SOURCE CODE:**

```
#include "syscall.h"
#include "stdio.h"
#include "stdlib.h"
#define BUFSIZE 1024
char buf[BUFSIZE];
int main(int argc, char** argv) {
    int src, dst, amount;
    if (argc!=3) {
        printf("Usage: cp <src> <dst>\n");
        return 1;
    }
    src = open(argv[1]);
    if (src==-1) {
```

```
printf("Unable to open %s\n", argv[1]);
return 1;
}
creat(argv[2]);
dst = open(argv[2]);
if (dst==-1) {
printf("Unable to create %s\n", argv[2]);
return 1;
}
while ((amount = read(src, buf, BUFSIZE))>0) {
write(dst, buf, amount);
}
close(src);
close(dst);
return 0;
}
```

### **EXPECTED OUTPUT:**

```
[lakshmi@localhost labprograms]$ vi prg10.c
[lakshmi@localhost labprograms]$ cc prg10.c
[lakshmi@localhost labprograms]$ ./a.out sampletext2 file10
Data copied from sampletext2 to file10
[lakshmi@localhost labprograms]$
```

### **Viva Questions**

**1. Using the grep command, how can you display or print the entire file contents?**

```
$ grep '.*' file
```

**2. What is the difference between a local variable and environment variable in Linux?**

A local variable is the one in which the scope of the variable is only in the shell in which it is defined. An environment variable has scope in all the shells invoked by the shell in which it is defined.

