

LAB MANUAL

LINUX PROGRAMMING

Year : 2019 - 2020
Course Code : ACS109
Regulations : IARE - R16
Class : B.Tech VI Semester
Branch : CSE

Prepared By

N.M DEEPIKA
Assistant professor



Department of Computer Science & Engineering

INSTITUTE OF AERONAUTICAL ENGINEERING

(Autonomous)

Dundigal – 500 043, Hyderabad



INSTITUTE OF AERONAUTICAL ENGINEERING

(Autonomous)

Dundigal, Hyderabad - 500 043

Vision

To bring forth professionally competent and socially sensitive engineers, capable of working across cultures meeting the global standards ethically.

Mission

To provide students with an extensive and exceptional education that prepares them to excel in their profession, guided by dynamic intellectual community and be able to face the technically complex world with creative leadership qualities.

Further, be instrumental in emanating new knowledge through innovative research that emboldens entrepreneurship and economic development for the benefit of wide spread community.

Quality Policy

Our policy is to nurture and build diligent and dedicated community of engineers providing a professional and unprejudiced environment, thus justifying the purpose of teaching and satisfying the stake holders.

A team of well qualified and experienced professionals ensure quality education with its practical application in all areas of the Institute.

Philosophy

The essence of learning lies in pursuing the truth that liberates one from the darkness of ignorance and Institute of Aeronautical Engineering firmly believes that education is for liberation.

Contained therein is the notion that engineering education includes all fields of science that plays a pivotal role in the development of world-wide community contributing to the progress of civilization. This institute, adhering to the above understanding, is committed to the development of science and technology in congruence with the natural environs. It lays great emphasis on intensive research and education that blends professional skills and high moral standards with a sense of individuality and humanity. We thus promote ties with local communities and encourage transnational interactions in order to be socially accountable. This accelerates the process of transfiguring the students into complete human beings making the learning process relevant to life, instilling in them a sense of courtesy and responsibility.



INSTITUTE OF AERONAUTICAL ENGINEERING

(Autonomous)

Dundigal, Hyderabad - 500 043

COMPUTER SCIENCE AND ENGINEERING

Program Outcomes	
PO1	Engineering knowledge: Apply the knowledge of mathematics, science, engineering fundamentals, and an engineering specialization to the solution of complex engineering problems.
PO2	Problem analysis: Identify, formulate, review research literature, and analyze complex engineering problems reaching substantiated conclusions using first principles of mathematics, natural sciences, and engineering sciences.
PO3	Design/development of solutions: Design solutions for complex engineering problems and design system components or processes that meet the specified needs with appropriate consideration for the public health and safety, and the cultural, societal, and environmental considerations.
PO4	Conduct investigations of complex problems: Use research-based knowledge and research methods including design of experiments, analysis and interpretation of data, and synthesis of the information to provide valid conclusions.
PO5	Modern tool usage: Create, select, and apply appropriate techniques, resources, and modern engineering and IT tools including prediction and modeling to complex engineering activities with an understanding of the limitations.
PO6	The engineer and society: Apply reasoning informed by the contextual knowledge to assess societal, health, safety, legal and cultural issues and the consequent responsibilities relevant to the professional engineering practice.
PO7	Environment and sustainability: Understand the impact of the professional engineering solutions in societal and environmental contexts, and demonstrate the knowledge of, and need for sustainable development.
PO8	Ethics: Apply ethical principles and commit to professional ethics and responsibilities and norms of the engineering practice.
PO9	Individual and team work: Function effectively as an individual, and as a member or leader in diverse teams, and in multidisciplinary settings.
PO10	Communication: Communicate effectively on complex engineering activities with the engineering community and with society at large, such as, being able to comprehend and write effective reports and design documentation, make effective presentations, and give and receive clear instructions.
PO11	Project management and finance: Demonstrate knowledge and understanding of the engineering and management principles and apply these to one's own work, as a member and leader in a team, to manage projects and in multidisciplinary environments.
PO12	Life-long learning: Recognize the need for, and have the preparation and ability to engage in independent and life-long learning in the broadest context of technological change.



INSTITUTE OF AERONAUTICAL ENGINEERING

(Autonomous)

Dundigal, Hyderabad - 500 043

Program Specific Outcomes

PSO1	Professional Skills: The ability to research, understand and implement computer programs in the areas related to algorithms, system software, multimedia, web design, big data analytics, and networking for efficient analysis and design of computer-based systems of varying complexity.
PSO2	Problem-Solving Skills: The ability to apply standard practices and strategies in software project development using open-ended programming environments to deliver a quality product for business success.
PSO3	Successful Career and Entrepreneurship: The ability to employ modern computer languages, environments, and platforms in creating innovative career paths, to be an entrepreneur, and a zest for higher studies.



INSTITUTE OF AERONAUTICAL ENGINEERING

(AUTONOMOUS)

Dundigal, Hyderabad - 500 043

Certificate

This is to certify that it is a bonafied record of Practical work done by Sri/Kum. _____ bearing the Roll No. _____ of _____ class _____ branch in the _____ laboratory during the academic year _____ under our supervision.

Head of the Department

Lecture In-Charge

External Examiner

Internal Examiner

Index

S. No.	List of Experiments	Page No.	Date	Remarks
I	GENERAL PURPOSE UTILITIES COMMANDS			
II	FILE SYSTEM, TEXT PROCESSING COMMANDS AND VI EDITOR			
III	SED, GREP, EGREP, FGREP			
IV	BASIC SHELL SCRIPTING			
V	SHELL SCRIPTING			
VI	INPUT OUTPUT REDIRECTIONS AND COMMAND SUBSTITUTIONS			
VII	AWK SCRIPT			
VIII	PATTERN SCANNING AND PROCESSING SCRIPTS			
IX	PATTERN SCANNING AND PROCESSING SCRIPTS			
X	PROCESS ATTRIBUTES AND USAGE OF FORK()			
XI	USAGE OF PIPES AND NAMED PIPES			
XII	SYNCHRONIZATION AND LOCKING TECHNIQUES			
XIII	CLIENT SEVER IMPLEMENTATION USING SOCKETS AND SHARED MEMORY			

EXPERIMENT 1

1.1 OBJECTIVE

- a) 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.
- b) Illustrate by writing script that will print, message "Hello World, in Bold and Blink effect, and in different colors like red, brown etc using echo commands?

1.2 RESOURCE/REQUIREMENTS

Linux operating system,vi-editor, shell-interpreter.

1.3 PROGRAM LOGIC

1. Read a filename, starting and ending line numbers as arguments
2. Find difference between starting and ending line numbers
3. Test given filename exists or not
4. If exists display between lines to output stream else display file not exists

1.4.a DESCRIPTION / PROCEDURE

```
echo " Enter the file name"
read fname
echo "enter starting line number"
read sl
echo "enter ending line number"
read el
d=`expr $el - $sl`
if [ -f $fname ]
then
    echo "the lines between $sl and $el of given file are"
    head -$el $fname | tail -$d
else
    echo "file doesnt exist"
fi
```

INPUT: sh

```
prog1.sh
enter the file
name file1
enter starting line number
15
enter ending line number
20
```

OUTPUT:

```
# It displays 15 to 20 between lines
```

1.4.b DESCRIPTION / PROCEDURE

```
# clear the screen
clear
echo -e "\033[1m Hello World"
# print bold effect
echo -e "\033[5m Blink"
# blink effect
echo -e "\033[0m Hello World"
# print back to normal
```

```
echo -e "\033[31m Hello World"
#print in Red color
echo -e "\033[32m Hello World"
# Green color
echo -e "\033[33m Hello World"
# See remains on screen
echo -e "\033[34m Hello World"
echo -e "\033[35m Hello World"
echo -e "\033[36m Hello World"
echo -e -n "\033[0m "
# print back to normal
echo -e "\033[41m Hello World"
echo -e "\033[42m Hello World"
echo -e "\033[43m Hello World"
echo -e "\033[44m Hello World"
echo -e "\033[45m Hello World"
echo -e "\033[46m Hello World"
echo -e "\033[0m Hello World"
# Print back to normal
```

1.5 LAB ASSIGNMENT

1. Write a shell script to count number of words present in a file without using commands.
2. Write a menu driven shell script to execute a command as 1.for `ls`,2 for grep and 3 for cat.

1.6 PRE-LAB QUESTIONS

1. Define shell script? What is the difference between shell and kernel.
2. Name few file handling commands present in unix.

1.7 POST-LAB QUESTIONS

1. What is the purpose of case statement?
2. What the difference between break and exit statement?

EXPERIMENT 2

2.1 OBJECTIVE

- a) Write a shell script that deletes all lines containing a specified word in one or more files supplied as arguments to it.
- b) Illustrate by writing script using for loop to print the following patterns?

2.2 RESOURCE/REQUIREMENTS

Linux operating system ,vi-editor, shell-interpreter

2.3 PROGRAM LOGIC

Read file name from command line arguments and display lines inverse of specified word.

2.4.a DESCRIPTION / PROCEDURE

```
if [ $# -ne 0 ]
then
    echo enter the word
    read word
    for fname in
        $* do
        if [ -f $fname ]
        then
            echo the given input filename is:$fname
            grep -v "$word" $fname
        else
            echo its not a
            file fi
        done
        else
            echo "enter atleast one argument as input"
        fi
```

INPUT:

```
sh prog2.sh 3.sh
enter the word
echo
```

OUTPUT:

The given input filename is : 3.sh
It displays all the lines other than pattern matching

2.4.b.i DESCRIPTION / PROCEDURE

```
# do the following for
loop echo "Stars"
# outer loop
for (( i=1; i<=5; i++ ))
do
#inner loop

    for ((j=1; j<=i; j++))
```

```
do
  echo -n " *"
done
echo ""
done
#
```

2.4.b.ii **DESCRIPTION / PROCEDURE**

```
# do the following for loop
```

```
echo "Can you see the following:"
# outer loop
for (( i=1; i<=5; i++ ))
do
#inner loop
  for (( j=1; j<=i; j++ ))
  do
    echo -n "$i"
  done
  echo ""
done
#
```

2.5 **LAB ASSIGNMENT**

1. Read two strings str1 and str2 and check
 - i) Compare two strings
 - ii) Palindrome or not .

2.6 **PRE-LAB QUESTIONS**

1. What are positional parameter and name any two.
2. Write down the syntax of `if` statement.

2.7 **POST-LAB QUESTIONS**

1. What is the purpose of the variable \$? What are the various output it has?

EXPERIMENT 3

3.1 OBJECTIVE

- 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.
- b) Illustrate to redirect the standard input (stdin) and the standard output (stdout) of a process, so that scanf () reads from the pipe and printf () writes into the pipe?

3.2 RESOURCE/REQUIREMENTS

Linux operating system ,vi-editor, shell-interpreter

3.3 PROGRAM LOGIC

Read a list of files from current directory and display the file names to output stream whose files has read, write, execute permissions.

3.4.a DESCRIPTION / PROCEDURE

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 pp2.txt

3.4.b DESCRIPTION / PROCEDURE

```
#include <stdio.h>
#include <unistd.h>
#include <sys/ipc.h>
main()
{
int fd[2];int n=0, i;
pipe(fd);
if (fork() == 0) {
/* create Child process */
close(1) ; dup(fd[1]) ;
close(fd[0]);
/* try not read from the pipe in this example.So close fd[0]. */ for
(i=0; i < 10; i++) {printf("%d\n",n);
/* Now that stdout has been redirected, printf automatically writes into the pipe. */ n++; }
} else {/* Parent process */close(0) ; dup(fd[0]) ;
/* Redirect the stdin of this process to the pipe*/
close(fd[1]);
/* will not write into the pipe.So we close fd[1]. */
for (i=0; i < 10; i++) {scanf("%d",&n);
/* Now that stdin has been redirected, scanf automatically reads from the pipe. */
```

```
printf("n = %d\n",n);
/* try stdout of this has not changed . So this will be shown in the terminal.    */ sleep(1);
}}
```

3.5 LAB ASSIGNMENT

1. Read a file name from command line and check it's a file or not.
2. Read a file name from command line and check if it read and write permission or not.

3.6 PRE-LAB QUESTIONS

1. What is the difference between \$* and \$@.
2. How to read a variable ,assign ,and access it

3.7 POST-LAB QUESTIONS

1. How to check if file is existing, it has read, write and execution permission.

EXPERIMENT 4

4.1 OBJECTIVE

- a) Write a shell script that receives any number of file names as arguments checks if every argument supplied is a file or a directory and reports accordingly. Whenever the argument is a file, the number of lines on it is also reported.
- b) Illustrate by writing c program where process forks to a child, and create a child process by using forks and suddenly terminates itself?

4.2 RESOURCE/REQUIREMENTS

Linux operating system ,vi-editor, shell-interpreter

4.3 PROGRAM LOGIC

Read a list of files from current directory and display the file names to output along with number of lines of each file

4.4.a DESCRIPTION / PROCEDURE

```
echo enter the name
for fname in
* do
if test -f
$name then
echo "file" $fname
echo "number of lines" `cat $fname | wc -
l` else if test -d $fname
then
echo "dir" $fname
fi
fi
done
```

INPUT:

```
sh prog4.sh
```

OUTPUT:ent

```
er the name
file 3.sh
number of lines 9
```

4.4.b DESCRIPTION / PROCEDURE

```
#include <stdio.h>
/* for fork() */
#include <sys/types.h>
#include <unistd.h>
/* for wait*() */
#include <sys/wait.h>
int main() {
pid_t mypid,
childpid; int status;
mypid = getpid();
printf("Hi. I'm the parent process. My pid is %d.\n", mypid);
childpid = fork();
```

```

if ( childpid == -1 ) {
    perror("Cannot proceed. fork() error");
    return 1;
}
if (childpid == 0) {
    printf("Child 1: I inherited my parent's pid as %d.\n", mypid);
    mypid = getpid();
    printf("Child 1: getppid() tells my parent is %d. My own pid instead is %d.\n", getppid(), mypid);
    /* forks another child */
    childpid = fork();
    if ( childpid == -1 ) {
        perror("Cannot proceed. fork() error");
        return 1;
    }
    if (childpid == 0) {
        /* this is the child of the first child, thus "Child 2" */
        printf("Child 2: I inherited my parent's PID as %d.\n", mypid);
        mypid = getpid();
        printf("Child 2: getppid() tells my parent is %d. My own pid instead is %d.\n", getppid(),
            mypid);
        childpid = fork();
        if ( childpid == -1 ) {
            perror("Cannot proceed. fork() error");
            return 1;
        }
        if (childpid == 0) {
            /* "Child 3" sleeps 30 seconds then terminates 12, hopefully before its parent "Child 2" */
            printf("Child 3: I inherited my parent's PID as %d.\n", mypid);
            mypid = getpid();
            printf("Child 3: getppid() tells my parent is %d. My own pid instead is %d.\n", getppid(),
                mypid);
            sleep(30);
            return 12;
        } else /* the parent "Child 2" suddenly returns 15 */ return 15; }
    } else {
        /* this is still "Child 1", which waits for its child to exit */
        while ( waitpid(childpid, &status, WNOHANG) == 0 ) sleep(1);
        if ( WIFEXITED(status) ) printf("Child1: Child 2 exited with exit status %d.\n",
            WEXITSTATUS(status));
        else printf("Child 1: child has not terminated correctly.\n");
    }
}
}
return 0;
}

```

4.5 LAB ASSIGNMENT

1. Write a shell script to count number of txt,c and shell programs present in current directory.
2. Write a shell script to count number of only files present in current directory.

4.6 PRE-LAB QUESTIONS

1. How to write arithmetic multiplication operator in shell.
2. Write down the syntax for nested if statement.

4.7 POST-LAB QUESTIONS

1. What is means by relation operator, name any three relation operators present in shell.
2. What is meant by logic operator, and Explain about each operator.

EXPERIMENT 5

5.1 OBJECTIVE

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

5.2 RESOURCE/REQUIREMENTS

Linux operating system ,vi-editor, shell-interpreter

5.3 PROGRAM LOGIC

Read list of file names and counts and report the occurrence of each word that is present in the first argument file on other argument files using comm and grep commands.

5.4 DESCRIPTION / PROCEDURE

echo Enter file name:

read file1

read file2

a=`comm -2 \$file1 \$file2`

b=`grep -c \$a \$file2`

echo Words contained in file one occurred in file two \$b times

grep -n \$a \$file2

INPUT: sh

prog5.sh

Enter file

name: f1

myfile

OUTPUT:

Words contained in file one occurred in file two 3 times

1:myfile contains

5:myfile

8:myfile

5.5 LAB ASSIGNMENT

1. Write a script to reverse a given string.
2. Write a script to copy list of file into specified directory.

5.6 PRE-LAB QUESTIONS

1. What is difference between comm and diff commands
2. What is the significance of using single and double quotation in echo statement.
3. What are environmental variables.

5.7 POST-LAB QUESTIONS

1. What is difference between grep, egrep and fgrep commands
2. How to copy and paste lines in a vi-editor
3. What are different types of shell and how to move from one shell to other shell

EXPERIMENT 6

6.1 OBJECTIVE

Write a shell script to list all of the directory files in a directory.

6.2 RESOURCE/REQUIREMENTS

Linux operating system ,vi-editor, shell-interpreter

6.3 PROGRAM LOGIC

1. Read a directory
2. Test given directory is directory file and exist using test options -d
3. If its directory and exist display all the sub directories and files to output stream
4. else display its not directory or not exists.

6.4 DESCRIPTION / PROCEDURE

```
echo " Enter dir name "  
read dir  
if [ -d $dir ]  
then  
    echo " Files in $dir are  
    " ls $dir  
else  
    echo " Dir does not exist"  
fi
```

INPUT:s

```
h Lp6.sh  
Enter dir name  
Presanna
```

OUTPUT:

```
Files in prasanna  
are 3.sh  
4.sh  
pp2.txt
```

6.5 LAB ASSIGNMENT

1. Write a shell script to count number of words present in a file without using commands.
2. Write a menu driven shell script to execute a command as 1.for `ls` ,2 for grep and 3 for cat.

6.6 PRE-LAB QUESTIONS

1. Name few test commands present in unix.
2. Write down the syntax case statement

6.7 POST-LAB QUESTIONS

1. What is the purpose of shift statement.
2. What the difference is between break and exit statement.

EXPERIMENT 7

7.1 OBJECTIVE

Write a shell script to find factorial of a given number.

7.2 RESOURCE/REQUIREMENTS

Linux operating system ,vi-editor, shell-interpreter

7.3 PROFRAM LOGIC

1. Read a filename
2. take a variable i for count and execute expression in for loop ,increment the variable till loop ends,
3. Display the factorial of given number to output stream.

7.4 DESCRIPTION / PROCEDURE

```
echo Factorial
echo Enter number:
read n
fact=1
i=1
for((i=1;i<=n;i++))
do
    fact=`expr $fact \* $i`
done
echo Factorial of $n is $fact
```

INPUT:

```
sh p7.sh
Factorial
Enter number:5
```

OUTPUT:

```
Factorial of 5 is 120
```

7.5 LAB ASSIGNMENT

1. Write a shell script for menu driven to execute a command like 1 for ls,2 for ls -l, etc..
2. Write a shell script to find the sum of digits.

7.6 PRE-LAB QUESTIONS

1. What is the use of while loop.
2. How you do command substitution in shell script.
3. How do you access command line arguments from within a shell script?

7.7 POST-LAB QUESTIONS

1. Illustrate the difference between while and for loop.
2. Which operator is used to check string is Null .
3. What is the name of the variable which counts number of arguments passed?

EXPERIMENT 8

8.1 OBJECTIVE

Write an awk script to count the number of lines in a file that do not contain vowels.

8.2 RESOURCE/REQUIREMENTS

Linux operating system ,vi-editor, shell-interpreter

8.3 PROGRAM LOGIC

1. Initialize total=0 in begin part
2. Using if command check each line to count the number of lines in a file that do not contain vowels in body part.
3. Display total lines to output stream that do not contain vowels in end part

8.4 DESCRIPTION / PROCEDURE

```
BEGIN{ print Displaying number of lines in a file that do not contain vowels
      total=0}
{if($0!~/[aeiouAEIOU]/)
total=total + 1}
END{print "The total lines in a file that do not contain vowels:",total}
```

INPUT:

awk prog8.awk lp1.sh

Displaying number of lines in a file that do not contain vowels

OUTPUT:

The total lines in a file that do not contain vowels:1

8.5 LAB ASSIGNMENT

1. Write a awk script to display between lines eg.2 to 8 lines to output stream.
2. Write a awk script to find sum of total salary of all employees of given file.

8.6 PRE-LAB QUESTIONS

1. Define awk? What are the features of awk script?
2. What is the syntax for awk script?

8.7 POST-LAB QUESTIONS

1. Illustrate difference between awk and sed.
2. Write a awk command to search given pattern in file if found display to output entire line.

EXPERIMENT 9

9.1 OBJECTIVE

Write an awk script to find the number of characters, words and lines in a file.

9.2 RESOURCE/REQUIREMENTS

Linux operating system ,vi-editor, shell-interpreter

9.3 PROGRAM LOGIC

Write awk script to find the number of characters, words and lines in a file

9.4 DESCRIPTION / PROCEDURE

```
BEGIN{ print Displaying number of characters, words and lines in a
file} {word=words + NF}
{len = length($0)}
{charcount=charcount + len}
END{print The total number of characters, words and lines in a file is:
print("Words:\t",words)
print("Lines:\t",NR)
print("Chars:\t",len) }
```

INPUT:

awk prog9.awk lp5.sh

OUTPUT:

The total number of characters, words and lines in a file
is: Words:12
Lines:3
Chars:39

9.5 LAB ASSIGNMENT

1. Write a awk script to convert lower case characters to upper case of given file.

9.6 PRE-LAB QUESTIONS

1. How input file is given to awk script?
2. What is the use of next, getline and exit control actions in awk

9.7 POST-LAB QUESTIONS

1. Explain about associative arrays in awk.

EXPERIMENT 10

10.1 OBJECTIVE

Write a C program that makes a copy of a file using Systems calls

10.2 RESOURCE/REQUIREMENTS

Linux operating system, vi –editor, shell interpreter

10.3 PROGRAM LOGIC

1. Read source filename, destination filename
2. Open given source filename
3. Read the content from file and write to destination file
4. Repeat step 3 till end of file reaches
5. close open files

10.4 DESCRIPTION / PROCEDURE

```
#include<stdio.h>
#include<unistd.h>
#include<sys/types.h>
#include<string.h>
void main() {
    char src[10], dest[10], buff;
    int fd,fd1;

    fd1=open("dest",O_WRONLY|O_CREAT|O_TRUNC|S_IRUSR|S_IWUSR);
    while(read(fd,&buff,1));
    wirte(fd1,&buff,1);
    printf("The copy of a file is succeeded");
    close(fd);
    close(fd1);
}
```

INPUT:

```
cc prog10.c
./a.out
entr the source file name:
file1
enter the destination file
name: file2
```

OUTPUT:

The copy of a file is successes

10.5 LAB ASSIGNMENT

1. Write a c-program to count number lines in a file.

10.6 PRE-LAB QUESTIONS

1. What is meant by file descriptor and user file descriptor starts from which number

10.7 POST-LAB QUESTIONS

1. What are the file descriptors values of keyword, monitor, error.
2. What is the use of lseek() function .

EXPERIMENT 11

11.A.1 OBJECTIVE

Write a C Program to Implement the Unix command cat using system calls.

11.A.2 RESOURCE/REQUIREMENTS

Linux operating system, vi –editor, shell interpreter

11. A.3 PROGRAM LOGIC

1. Open file which is input given by command line arguments
2. Read content from opened file
3. Display content to output stream
4. Repeat step 2 and step 3 till end of file reach.

11. A.4 DESCRIPTION / PROCEDURE

```
#include<fcntl.h>
#include<sys/stat.h> #define
BUFSIZE 1
int main(int argc, char **argv)
{
    int fd1; int n; char buf;
    fd1=open(argv[1],O_RDONLY); printf("Displaying
content of file\n"); while((n=read(fd1,&buf,1))>0)
    {
        printf("%c",buf); /* or
        write(1,&buf,1); */
    }
    return (0);
}
```

INPUT:

cc prog1 1a.c unit1

OUTPUT:

Displays content of file
1. .

11. 5 LAB ASSIGNMENT

Write a c-program to count number words in a file

11. 6 PRE-LAB QUESTIONS

1. What is the difference between open() and fopen()?

11. 7 POST-LAB QUESTIONS

1. What is the difference between read(), write() and scanf, printf respectively .

EXPERIMENT 12

12.1 OBJECTIVE

Write a C program that takes one or more file or directory names as command line input and reports the following information on the file.

1. file type
2. number of links
3. read, write and execute permissions
4. time of last access

(Note: use /fstat system calls)

12.2 RESOURCE/REQUIREMENTS

Linux operating system, vi –editor, c-compiler

12.3 PROGRAM LOGIC

1. Open a file using fopen() function
2. Read a file and display a file properties to output stream.

12.4 DESCRIPTION / PROCEDURE

```
#include<stdio.h>
main()
{
FILE *stream;
int buffer_character;
stream=fopen("test","r");
if(stream==(FILE*)0)
{
fprintf(stderr,"Error opening file(printed to standard error)\n");
fclose(stream);
exit(1);
}}
if(fclose(stream)==EOF)
{
fprintf(stderr,"Error closing stream.(printed to standard
error)\n); exit(1);
}
return();
}
```

12.5 LAB ASSIGNMENT

1. Write a C program to count number of words, lines and characters using system calls

12.6 PRE-LAB QUESTIONS

1. What is the difference between stat(), fstat() and lstat() functions

12.7 POST-LAB QUESTIONS

List properties of files and different types of files in Linux

EXPERIMENT 13

13.1 OBJECTIVE

Write a C program to emulate the Unix ls -l command.

13.2 RESOURCE/REQUIREMENTS

Linux operating system, vi -editor, c-compiler

13.3 PROGRAM LOGIC

1. Declare and initialize required objects.
2. Read the directory name form the user.
3. Open the directory using opendir() system call and report error if the directory is not available.
4. Read the entry available in the directory.
5. Display the directory entry ie., name of the file or sub directory.
6. Repeat the step 6 and 7 until all the entries were read.

13.4 DESCRIPTION / PROCEDURE

```
#include <stddef.h> #include
<stdio.h> #include
<sys/types.h> #include
<dirent.h> int main (void) {
DIR *dp;
struct dirent *ep; dp =
opendir ("."); if (dp !=
NULL) {
while (ep = readdir (dp))
printf("%s\n", ep->d_name);
closedir (dp);
}
else
perror ("Couldn't open the directory"); return
0;
}
INPUT: cc 13.c
./a.out
OUTPUT: 2 3 a.out 1 4 ..... 13.c
```

13.5 LAB ASSIGNMENT

Write a C-program to simulate `nl` command.

13.6 PRE-LAB QUESTIONS

1. What is the purpose of O_CREAT and O_SYNC.

13.6 POST-LAB QUESTIONS

1. What is the use of -> operator?