Laboratory Manual

For

Element of Linux OS and C Programming-I

(CT 116)

B.Tech (IT)

SEM I

June 2016

Faculty of Technology
Dharmsinh Desai University
Nadiad.
www.ddu.ac.in

Table of Contents

EXPERIMENT-7

TUTORIAL-8

EXPERIMENT-8

TUTORIAL-9

EXPERIMENT-9

TUTORIAL-10

EXPERIMENT-10

TUTORIAL-11

EXPERIMENT-11

TUTORIAL-12

EXPERIMENT-12

Sample Experiment

1 AIM: (A) Explain the following commands:

- clear
- cal
- who
- date
- mkdir
- rm

2 TOOLS/APPARATUS: Linux operating system.

3 STANDARD PROCEDURES:

3.1 Analyzing the Problem:

Start the Linux and enter the user name and password. Now write startx and after that open the terminal.

At the terminal try the different commands and see the output.

3.2 Designing the Solution:

At the terminal first perform the command CAL without and with the different options available for it.

Like $ cal and then enter. The calendar will be displayed at the terminal.

$ cal –m and then enter. In the calendar Monday will be displayed as the first day of the week.

Same way perform the other commands like CLEAR, WHO, DATE, MKDIR, RM.

3.3 Implementing the Solution:

3.3.1 Writing Source

Code: 1) CAL:

At the terminal write the following:

```
[user1@com]$ cal
[user1@com]$  cal  -m
[user1@com]$   cal   -j
[user1@com]$ cal –y
```

2) CLEAR:

At the terminal write the following:

```
[user1@com]$ clear
```

3) WHO:

At the terminal write the following:

```
[user1@com]$ who
[user1@com]$   who   -q
[user1@com]$ who -H
```

```
[user1@com]$ who –m
```

4) DATE:

At the terminal write the following:

```
[user1@com]$ date
     [user1@com]$ date –d “2 days ago”
     [user1@com]$ date +%D
[user1@com]$ date +%d
[user1@com]$ date +%d%m%h
```

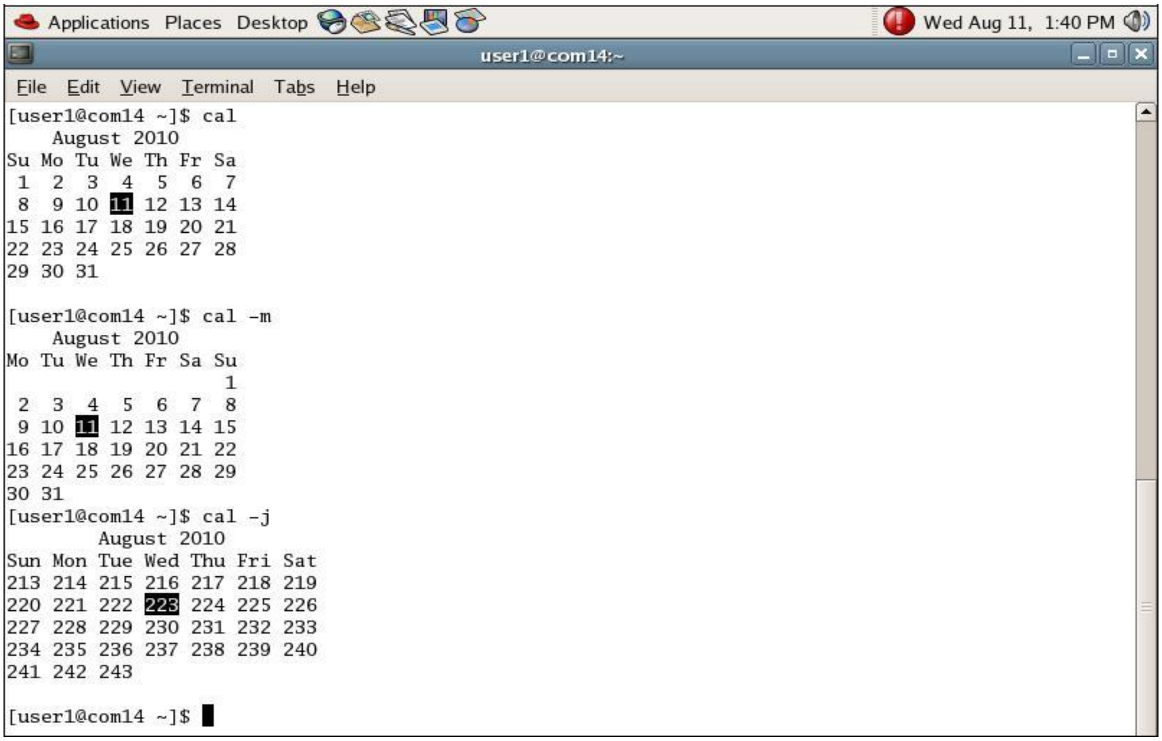
5) MKDIR and RM:

At the terminal write the following:

```
     [user1@com]$   cd   Desktop/
     [user1@com]$ ls
     [user1@com]$ cd newfiles/
     [user1@com]$              mkdir
     newfile1 [user1@com]$ ls
     [user1@com]$   rm   Sum_Of_Digits.txt
     [user1@com]$ ls
```

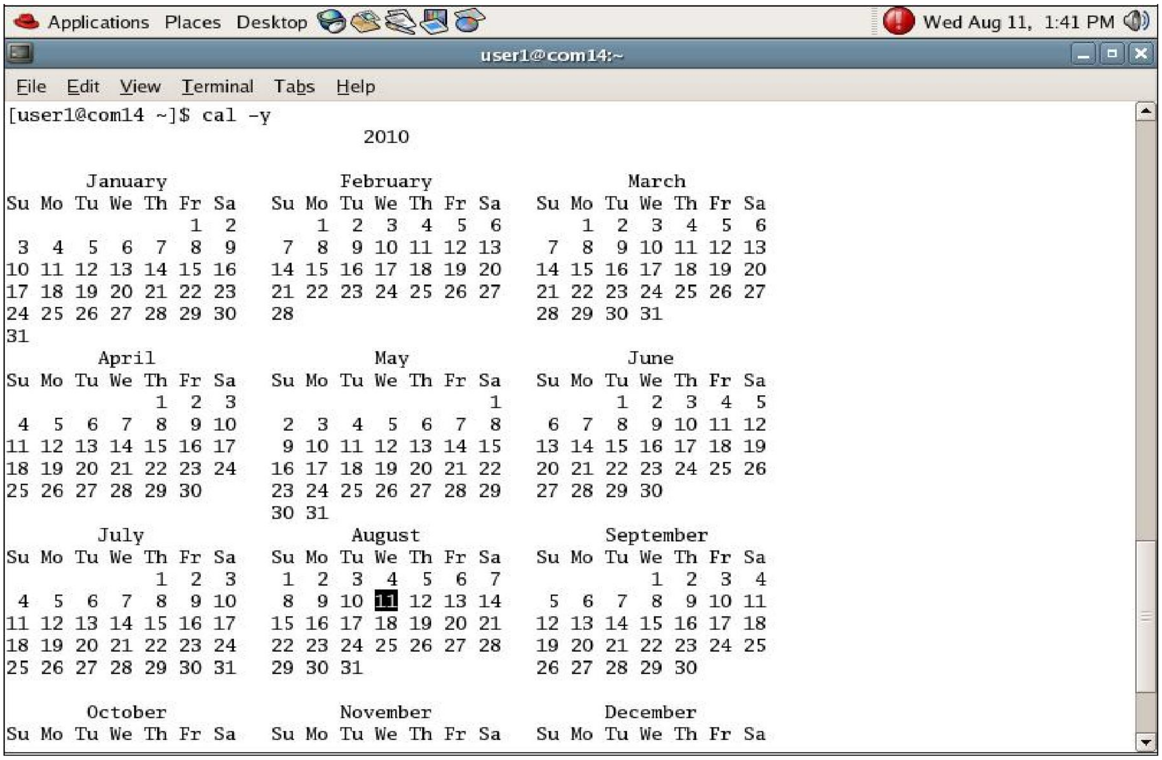
3.3.2 Compilation /Running and Debugging the Solution:

The code written above will display the following output. For the first command CAL the output is like this:


Applications Places Desktop
Wed Aug 11, 1:40 PM
user1@com14:~
File Edit View Terminal Tabs Help

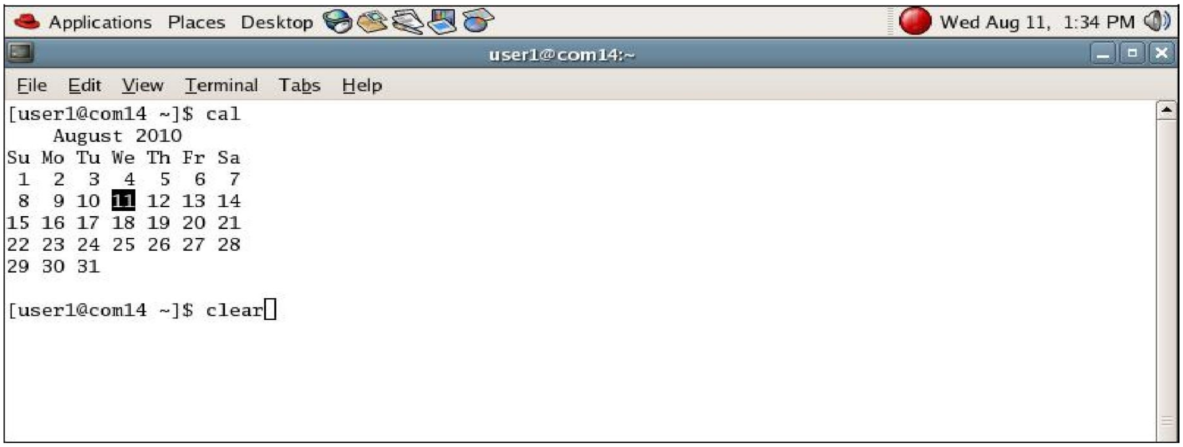
```
[user1@com14 ~]$ cal
    August 2010
Su Mo Tu We Th Fr Sa
 1  2  3  4  5  6  7
 8  9 10 11 12 13 14
15 16 17 18 19 20 21
22 23 24 25 26 27 28
29 30 31

[user1@com14 ~]$ cal -m
    August 2010
Mo Tu We Th Fr Sa Su
                   1
 2  3  4  5  6  7  8
 9 10 11 12 13 14 15
16 17 18 19 20 21 22
23 24 25 26 27 28 29
30 31
[user1@com14 ~]$ cal -j
        August 2010
Sun Mon Tue Wed Thu Fri Sat
213 214 215 216 217 218 219
220 221 222 223 224 225 226
227 228 229 230 231 232 233
234 235 236 237 238 239 240
241 242 243

[user1@com14 ~]$ 
```

The cal command with the option y will display the following output.


Applications Places Desktop
Wed Aug 11, 1:41 PM
user1@com14:~
File Edit View Terminal Tabs Help

```
[user1@com14 ~]$ cal -y
                               2010

       January               February                 March
Su Mo Tu We Th Fr Sa   Su Mo Tu We Th Fr Sa   Su Mo Tu We Th Fr Sa
                1  2       1  2  3  4  5  6       1  2  3  4  5  6
 3  4  5  6  7  8  9    7  8  9 10 11 12 13    7  8  9 10 11 12 13
10 11 12 13 14 15 16   14 15 16 17 18 19 20   14 15 16 17 18 19 20
17 18 19 20 21 22 23   21 22 23 24 25 26 27   21 22 23 24 25 26 27
24 25 26 27 28 29 30   28                     28 29 30 31
31
        April                   May                   June
Su Mo Tu We Th Fr Sa   Su Mo Tu We Th Fr Sa   Su Mo Tu We Th Fr Sa
             1  2  3                      1          1  2  3  4  5
 4  5  6  7  8  9 10    2  3  4  5  6  7  8    6  7  8  9 10 11 12
11 12 13 14 15 16 17    9 10 11 12 13 14 15   13 14 15 16 17 18 19
18 19 20 21 22 23 24   16 17 18 19 20 21 22   20 21 22 23 24 25 26
25 26 27 28 29 30      23 24 25 26 27 28 29   27 28 29 30
                       30 31
        July                  August                September
Su Mo Tu We Th Fr Sa   Su Mo Tu We Th Fr Sa   Su Mo Tu We Th Fr Sa
             1  2  3    1  2  3  4  5  6  7             1  2  3  4
 4  5  6  7  8  9 10    8  9 10 11 12 13 14    5  6  7  8  9 10 11
11 12 13 14 15 16 17   15 16 17 18 19 20 21   12 13 14 15 16 17 18
18 19 20 21 22 23 24   22 23 24 25 26 27 28   19 20 21 22 23 24 25
25 26 27 28 29 30 31   29 30 31               26 27 28 29 30

       October               November               December
Su Mo Tu We Th Fr Sa   Su Mo Tu We Th Fr Sa   Su Mo Tu We Th Fr Sa
```

For the second command CLEAR :

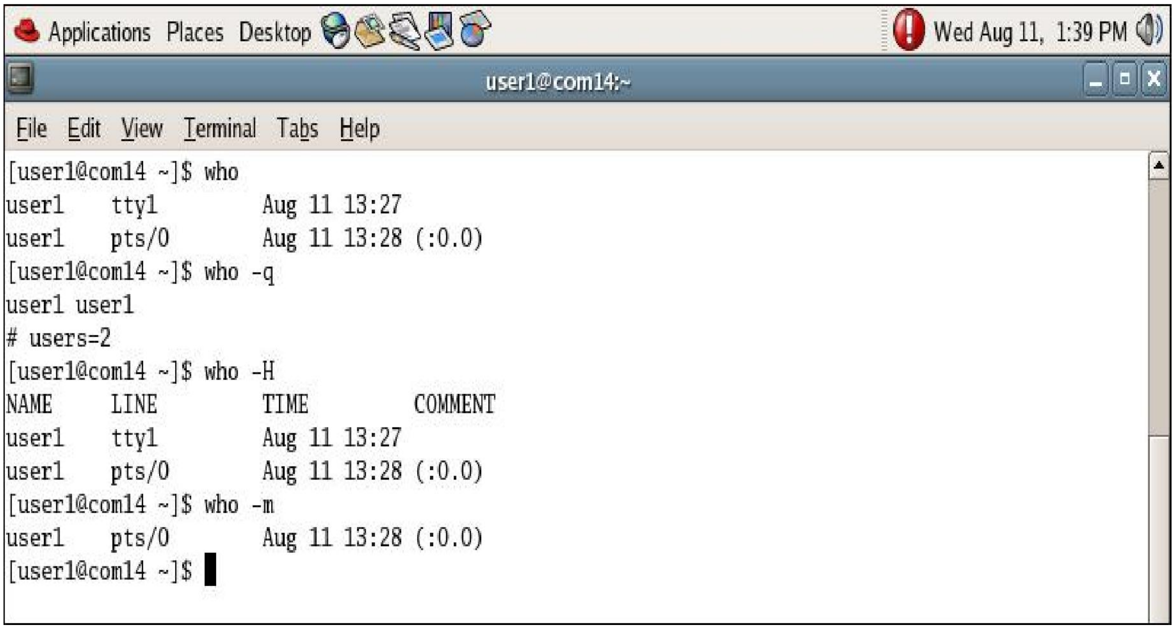

Applications Places Desktop
Wed Aug 11, 1:34 PM
user1@com14:~
File Edit View Terminal Tabs Help

```
[user1@com14 ~]$ cal
    August 2010
Su Mo Tu We Th Fr Sa
 1  2  3  4  5  6  7
 8  9 10 11 12 13 14
15 16 17 18 19 20 21
22 23 24 25 26 27 28
29 30 31

[user1@com14 ~]$ clear
```

Applications Places Desktop
Wed Aug 11, 1:34 PM
user1@com14:~
File Edit View Terminal Tabs Help

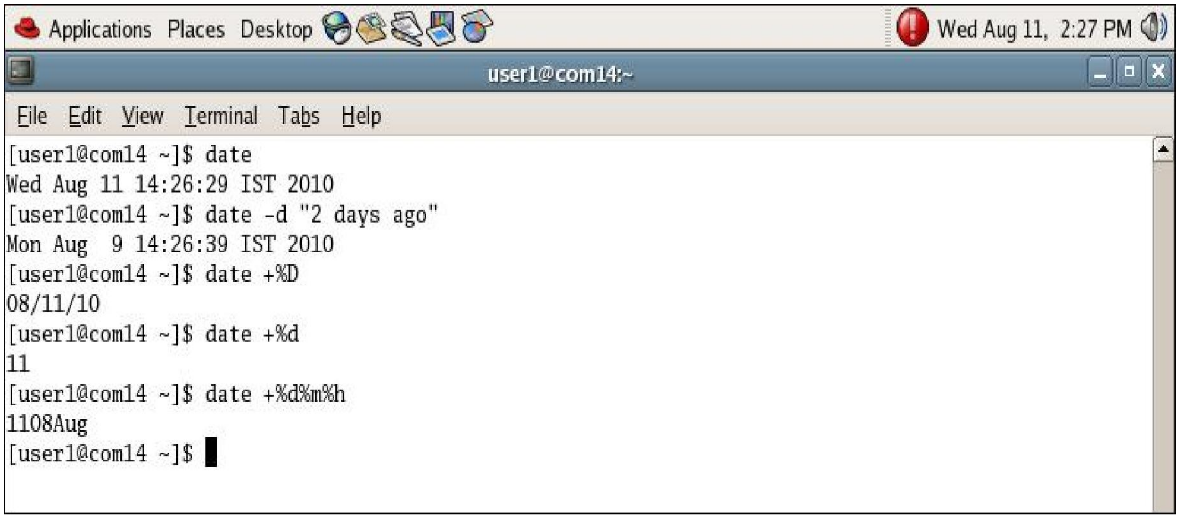
```
[user1@com14 ~]$
```

For the third command WHO :


Applications Places Desktop
Wed Aug 11, 1:39 PM
user1@com14:~
File Edit View Terminal Tabs Help

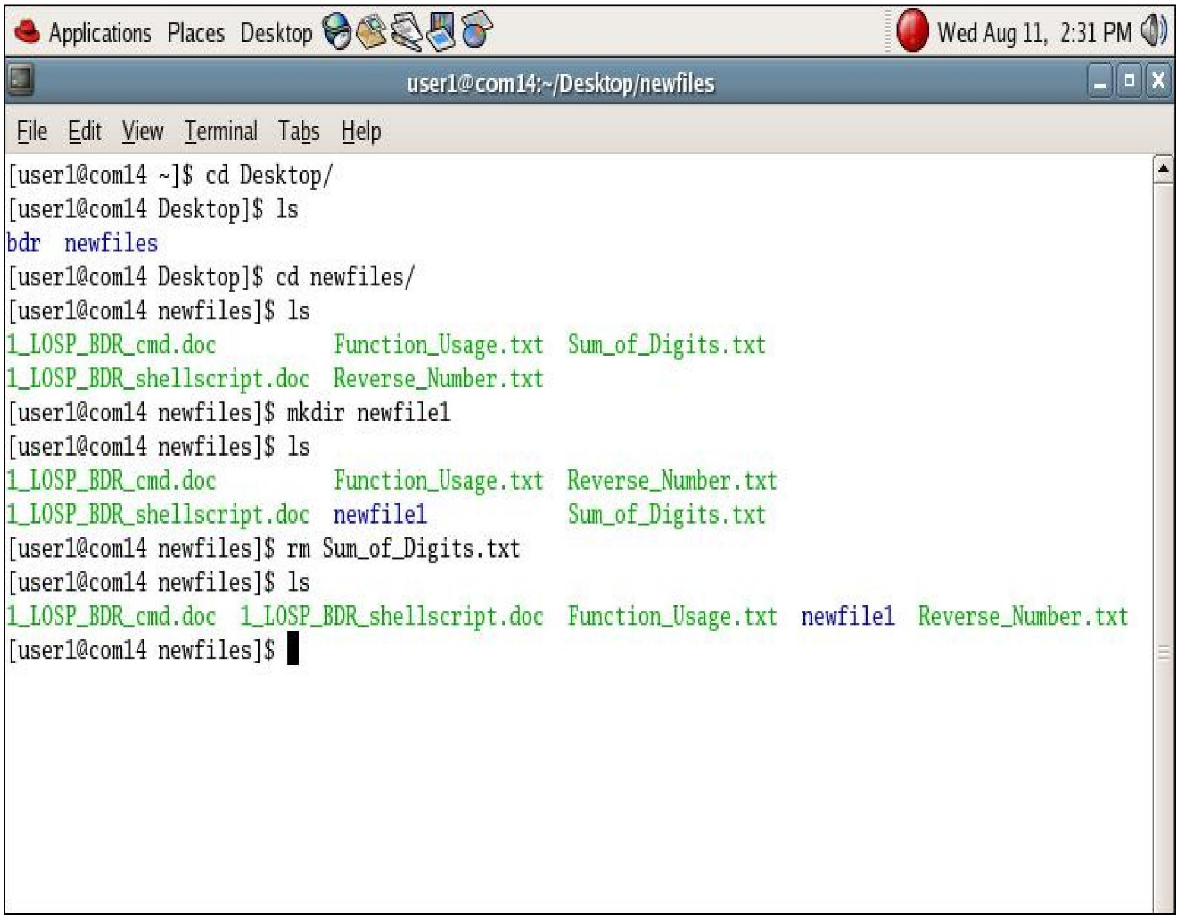
```
[user1@com14 ~]$ who
user1    tty1         Aug 11 13:27
user1    pts/0        Aug 11 13:28 (:0.0)
[user1@com14 ~]$ who -q
user1 user1
# users=2
[user1@com14 ~]$ who -H
NAME     LINE         TIME         COMMENT
user1    tty1         Aug 11 13:27
user1    pts/0        Aug 11 13:28 (:0.0)
[user1@com14 ~]$ who -m
user1    pts/0        Aug 11 13:28 (:0.0)
[user1@com14 ~]$
```

For the command DATE:


Applications Places Desktop
Wed Aug 11, 2:27 PM
user1@com14:~
File Edit View Terminal Tabs Help

```
[user1@com14 ~]$ date
Wed Aug 11 14:26:29 IST 2010
[user1@com14 ~]$ date -d "2 days ago"
Mon Aug  9 14:26:39 IST 2010
[user1@com14 ~]$ date +%D
08/11/10
[user1@com14 ~]$ date +%d
11
[user1@com14 ~]$ date +%d%m%h
1108Aug
[user1@com14 ~]$ 
```

For the commands MKDIR and RMDIR the output will be like this:


Applications Places Desktop
Wed Aug 11, 2:31 PM
user1@com14:~/Desktop/newfiles
File Edit View Terminal Tabs Help

```
[user1@com14 ~]$ cd Desktop/
[user1@com14 Desktop]$ ls
bdr  newfiles
[user1@com14 Desktop]$ cd newfiles/
[user1@com14 newfiles]$ ls
1_LOSP_BDR_cmd.doc          Function_Usage.txt  Sum_of_Digits.txt
1_LOSP_BDR_shellscript.doc  Reverse_Number.txt
[user1@com14 newfiles]$ mkdir newfile1
[user1@com14 newfiles]$ ls
1_LOSP_BDR_cmd.doc          Function_Usage.txt  Reverse_Number.txt
1_LOSP_BDR_shellscript.doc  newfile1            Sum_of_Digits.txt
[user1@com14 newfiles]$ rm Sum_of_Digits.txt
[user1@com14 newfiles]$ ls
1_LOSP_BDR_cmd.doc  1_LOSP_BDR_shellscript.doc  Function_Usage.txt  newfile1  Reverse_Number.txt
[user1@com14 newfiles]$ 
```

3.4 Testing the Solution:

All the commands will display the output based on it and the options given to that command.

If we are giving a command and the option to that command then that option must be of that command only otherwise will display the error.

4 Conclusions :

Using this we can run different command and see the output.

1 AIM: (B) Write a C program in Linux. Define a structure for items. The members are item number, item name, item price. Take all the details for at least 5 items. Using function search for the particular item by its name or by its number.

2 TOOLS/APPARATUS: Linux OS, VI Editor.

3 STANDARD PROCEDURES:
COMMON PROCEDURE:

Step 1: start Linux in your computer and login in it and enter startx.
Step 2: Create a folder with your Id Number or Name Followed by RollNo.
Step 3: now go to your folder from the terminal and after that open the VI editor with the desired program name with extension C.
Step 4: now write your program and quit back to the terminal.

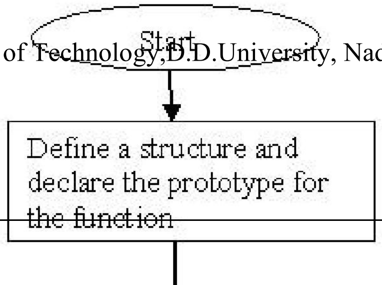
3.1 Analyzing the Problem:

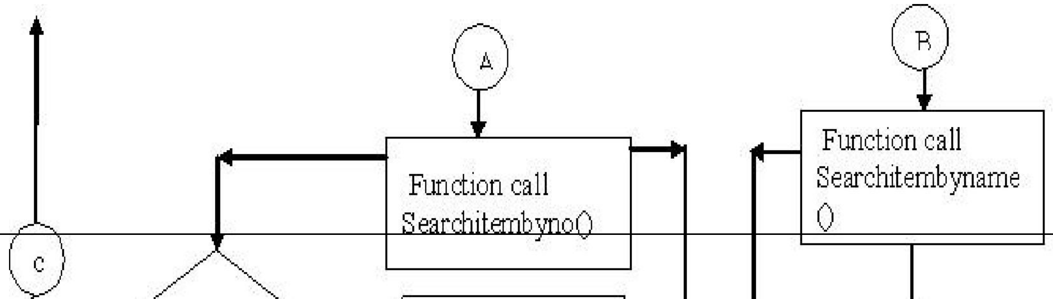
Firs create a file named as “stru.c”.
After that includes the standard input/output files.
Now define the structure “item” and its members.
Create the functions which are necessary for the program.
Now create the main function and take the information.
After that quit from the terminal using :q!.

3.2 Designing the Solution:

Create a c file name as “stru.c”.
Define the structure named as “item”. Also define its members that are item number, item name, item price.
Make a function to search a record by an item number that is “searchitembyno()” or by item name that is “searchitembyname()”.
Now in the main function declare a variable of the structure item “it[5]”. It should be an array because we want all the details for at least 5 items.
Now take the all the details by the user using scanf() function.
Now enter the choice by which you want to search. And call the function for desire output. And display the details using printf() function.

Flowchart:


Start
Define a structure and declare the prototype for the function


A
B
C
Function call Searchitembyno()
Function call Searchitembyname ()

3.3 Implementing the Solution:

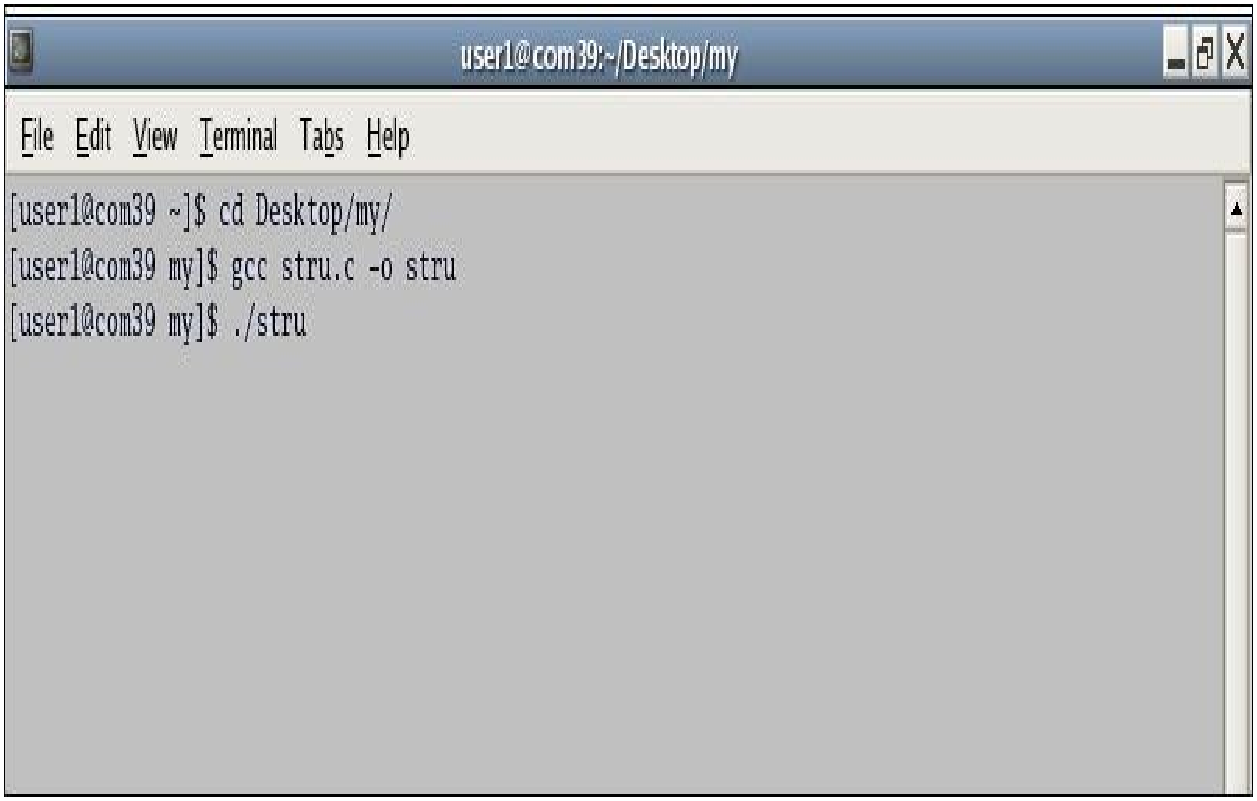
3.3.1 Writing Source Code:

```
#include<stdio.h>

struct item                    //defining the structure name item
{
int i_no;
char name[20];
int price;
};

void searchitembyno(struct item i1[ ] ,int,int); //function prototype to search by item number

void searchitembyname(struct item i1[ ],char [ ],int); //function prototype to search by
item name

void main( )
{
  struct item it[10],t;
  char str[10],c[20];
  int no,i,j,a,ch;
  clrscr();
  printf("\nHow many item :");
  scanf("%d",&n); // take n number of items
  for(i=0;i<n;i++)
   {
      printf("\nEnter no, name and price of item :");
      scanf("\n%d %s %d",&it[i].i_no,it[i].name,&it[i].price);
 }

label:
// now for searching there are two options
printf("\n\nEnter 1 to search by number and 2 to search by name :
"); scanf("%d",&ch);

switch(ch)
{
case 1: printf("\n\nEnter the no for the item u want to search : ");
          scanf("\n%d",&a);
          printf("\n");
          searchitembyno(it,a,n); //this is a function call to search by
          number break;
case 2: printf("\n\nEnter the name of the item u want to search : ");
          scanf("\n%s",str);
          printf("\n");
```

```
            searchitembyname(it,str,n); //this is a function call to search by
            name break;
default:  break;
}

printf("\n want to search again ? y/n "); //to search again Enter ‘y’ or to stop searching Enter
‘n’.

scanf(“%c”,&c);
if((strcmp(c==”yes”))==0)
{
 goto label ;
}
  }
//function definition ro search by number
void searchitembyno(struct item i1[],int a,int n)
 {
     int i,flag=0;
     for(i=0;i<n;i++)
     {
         if(i1[i].i_no==a)   //comparing nos
          {
            flag=1;
            printf("\n name and price is %s %d :",i1[i].name,i1[i].price);
            break;
          }
     }

    if(flag==0)
     {
       printf("\nitem not found ");
     }
}

//function definition ro search by name
void searchitembyname(struct item i1[],char s[],int n)
  {
     int i,flag=0;
     for(i=0;i<n;i++)
      {
          if((strcmp(i1[i].name,s))==0)  //comparing string
           {
             flag=1;
             printf("\n no name and price is %d %s %d :",i1[i].i_no,i1[i].name,i1[i].price);
             break;
            }
      }

     if(flag==0)
      {
         printf("\nitem not found ");
    }
}
```

3.3.2 Compilation /Running and Debugging the Solution:

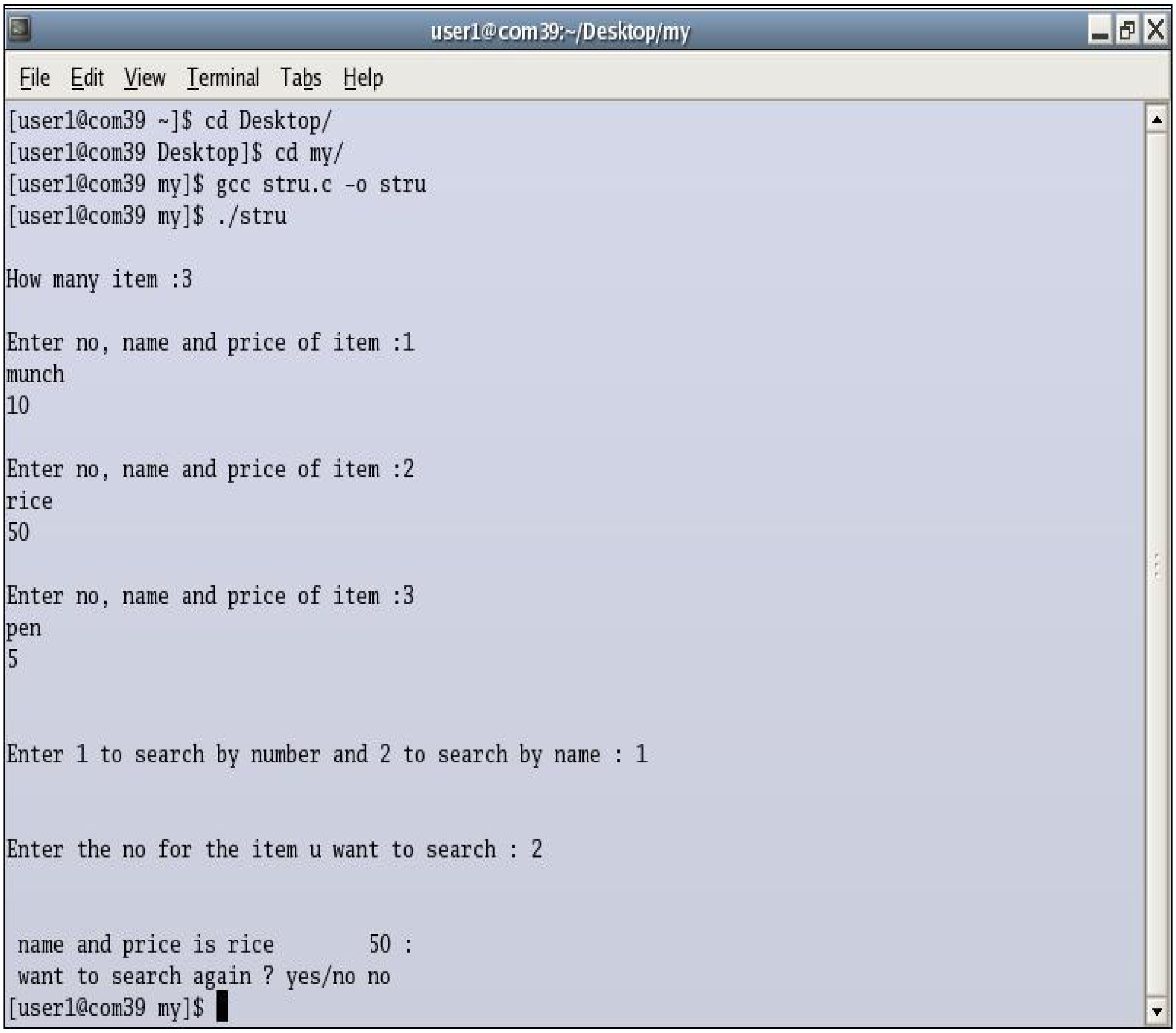
To compile the program at the terminal you have to write like: $ gcc stru.c –o stru



```
[user1@com39 ~]$ cd Desktop/my/
[user1@com39 my]$ gcc stru.c -o stru
[user1@com39 my]$ ./stru
```

If Successful Compilation is done then Run the Code Using following:

$ ./stru



```
[user1@com39 ~]$ cd Desktop/
[user1@com39 Desktop]$ cd my/
[user1@com39 my]$ gcc stru.c -o stru
[user1@com39 my]$ ./stru

How many item :3

Enter no, name and price of item :1
munch
10

Enter no, name and price of item :2
rice
50

Enter no, name and price of item :3
pen
5


Enter 1 to search by number and 2 to search by name : 1


Enter the no for the item u want to search : 2


 name and price is rice        50 :
 want to search again ? yes/no no
[user1@com39 my]$
```

3.4 Testing the Solution:

User must have entered all the details with respected to its data type.

In search by name or number if that record is found than it will display the desire output. Otherwise it will display that item not found.

If we enter the data which is not match to its data type than it will give unpredictable output.

4 Conclusions :

Hence we can compile and execute the C program in Linux.

1 AIM: (C) Write Script, using function and case statement to perform basic math operation as follows + Addition, - Subtraction, x Multiplication, / Division.

2 TOOLS/APPARATUS: Linux operating system.

3 STANDARD PROCEDURES:

3.1 Analyzing the Problem:

Start the Linux gives the user name and password. Write startx and after that open the terminal.

For writing the script open the VI editor with the proper script name.

After that to execute the script write the proper script name with sh option.

3.2 Designing the Solution:

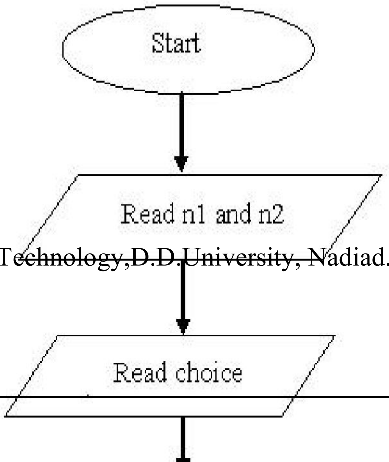
Open the VI editor with the script name "calc.sh".

In that write four function for Addition add_op(), Subtraction sub_op(), Division div_op() and Multiplication mul_op().

To Read the two numbers n1 and n2 write the code for it. To perform the particular operation read the choice for it.

After that execute the program. Based on choice call that particular function. It will produce the desired output.

Flowchart:



3.3 Implementing the Solution:

3.3.1 Writing Source Code:

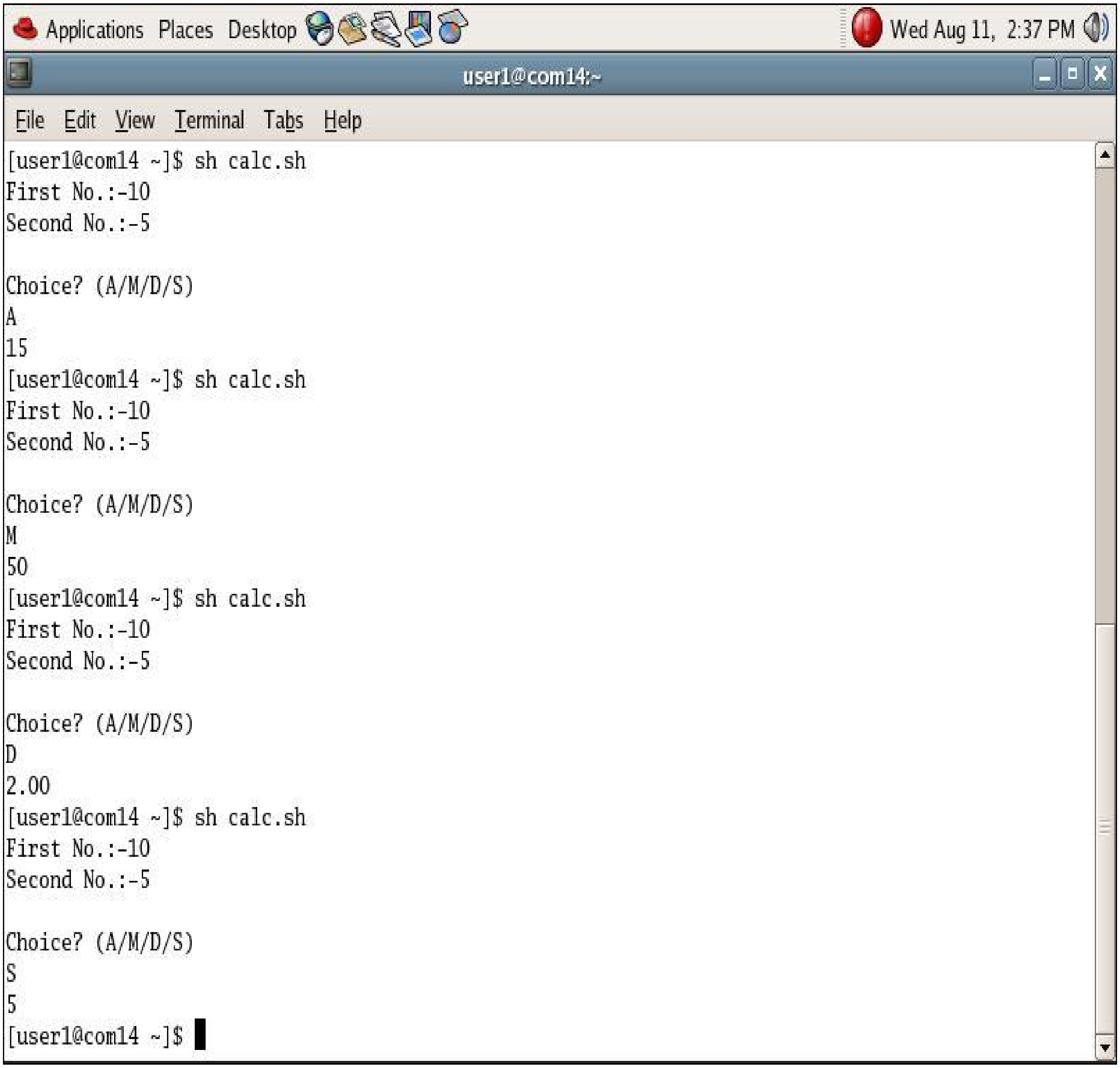
```
#!/bin/bash
add_op()
{
bc<<H1
scale=2
$n1+$n2
H1 return
}
sub_op()
{
bc<<H2
scale=2
$n1-$n2
H2
return
}
mul_op()
{
bc<<H3
scale=2
$n1*$n2
H3
return
}
div_op()
{
bc<<H4
scale=2
$n1/$n2
H4
return
}
flag=1
while [ $flag -eq 1 ]
do
        echo -e "First No.:-\c"; read n1 echo -
        e "Second No.:-\c"; read n2 echo -e
        "\nChoice? (A/M/D/S)";
 read choice
        case $choice in
        A)  add_op  ;  flag=0  ;;
        M)  mul_op  ;  flag=0  ;;
        D) div_op ; flag=0 ;;
        S) sub_op ; flag=0 ;;
        *) echo "Sorry Incorrect Entry, Please Try Again."
        ;; esac
done
```

3.3.2 Compilation /Running and Debugging the Solution:

To run this script, write at the terminal like this:

```
$ sh calc.sh
```

It will display the following output:


Applications Places Desktop
Wed Aug 11, 2:37 PM
user1@com14:~
File Edit View Terminal Tabs Help

```
[user1@com14 ~]$ sh calc.sh
First No.:-10
Second No.:-5

Choice? (A/M/D/S)
A
15
[user1@com14 ~]$ sh calc.sh
First No.:-10
Second No.:-5

Choice? (A/M/D/S)
M
50
[user1@com14 ~]$ sh calc.sh
First No.:-10
Second No.:-5

Choice? (A/M/D/S)
D
2.00
[user1@com14 ~]$ sh calc.sh
First No.:-10
Second No.:-5

Choice? (A/M/D/S)
S
5
[user1@com14 ~]$
```

3.4 Testing the Solution:

We have to give desired values.

If we give character instead of integers than it will produce unpredictable output.

Same way for the choice. If we enter another character instead of the given choices than it will be error.

4 Conclusions :

Hence this script will run. It takes the values and can do the desire task.

Required Software/ Software Tool:
-Linux Operating System.

Common procedure:

Step 1**:** Analyze the problem statement

Step 2: Perform the command with each and every option available for it. And if the script is there than design the script and run it.

Step 3: Compile code using gcc compiler for Linux, which will create a.out executable file.

Step 4: Test the program using sample input and write down output.

TUTORIAL-1

□□Answer the following questions.

1. Introduction about the Operating System.
2. Difference between Unix and Windows.
3. Explain Kernel and Shell
4. Characteristics of Unix System
    a. Multi-User
    b. Multi-Tasking

EXPERIMENT-1

Aim: (A) Overview of the UNIX Operating System and overview of basic commands.

Tools: Linux operating system, terminal.

Procedure:

Explain the history of the UNIX operating system. As well as what is Linux.

TUTORIAL-2

□□Answer the following questions.

1. Explain the difference between tput and clear.
2. Explain the pwd command.
3. Explain the date command with all the options.
1. Explain the bc command and the following
    a. ibase
    b. obase
    c. scale
    d. length()

EXPERIMENT-2

Aim: Explain the following commands: clear, cal, who, date, tput, exit, pwd, bc, wc, cat.

Tools: Linux operating system, terminal.

Procedure:

Explain all the commands. Also run all these commands with options available for it. And see the output.

Write the description about all these commands.

TUTORIAL- 3

□□Answer the following questions.

1. Explain all the options of ls command.
2. Explain the file command.
3. Explain the difference between mv and cp.
4. Explain absolute and relative paths.

EXPERIMENT-3

Aim: Explain the following commands: cd, mkdir, rmdir, rm, cp, cmp, comm, diff

Tools: Linux operating system, terminal.

Procedure:
Explain all the commands. Also run all these commands with options available for it. And see the output.

Write the description about all these commands.

TUTORIAL-4

□□Answer the following questions.

1. Explain difference between chgrp and chown in detail.
2. Explain read, write and execute permission for a file and a directory.
3. Explain the three modes of vi editor in detail.

EXPERIMENT-4

Aim: Explain the file permissions. Also explain the following commands: chmod, chown, chgrp.

Tools: Linux operating system, terminal.

Procedure:

Explain the permission for the file and also the commands available for it. Run all these commands with options available for it. And see the output.

Write the description about all these commands.

TUTORIAL-5

□□Answer the following questions.

1. Explain i-node and i-number.
2. Explain touch command in detail.
3. What do you mean by link? How do you add or delete a link, explain with an example.
4. Explain kill command .

EXPERIMENT-5

Aim: 1. Explain the following commands: ps, kill, umask, more, less, alias, ln.

2. Solve the following problems with the use of find command:

a. Change permission of all files and directory
b. Find all files which contain *
c. Find all file whose name is not ending with .c
d. Find either directory starting with d or files starting with f
e. Find all the hard-links of a file.
f. Find all files which are modified later than a year
g. Find all files of the owner user1 in the directory d1
h. Find all the files having four soft links
i. Find all files modified after f1
j. Find all files not modified after f1, having owner as root and not ending with
.c

Tools: Linux operating system, terminal.

Procedure:

Explain all the commands. Also run all these commands with options available for it. And see the output.

Write the description about all these commands.

TUTORIAL-6

Answer the following questions.

1. Explain the use of expr command in detail.
2. Explain the use of test command.
3. Explain the case structure of shell script with example.
4. Explain the control loops like
   a. If
   b. If-then-else
   c. If-then-elif-then-else

EXPERIMENT-6

Aim: (A) Design a shell script to find average of numbers.
(B) Find which number is greater amongst the three entered numbers.
(C) Design a shell script by which only the word "DDU" is displayed from the lines in any file.

Tools: Linux operating system, terminal.

Procedure:

Explain the shell script. Also design the given script.

Make a shell script which finds the greatest number among the three numbered entered by the user.

Make a file in which some contents are there. Now make a script which finds that the word "DDU" and display it.

TUTORIAL-7

□□Answer the following questions.

1. Explain the use of loops in a shell script.
2. What are command-line arguments?
3. Explain the difference between while loop and until loop.
4. Explain the following:
    a. for loop
    b. while loop
    c. until loop

EXPERIMENT-7

Aim: (A) Design a shell script which would display the summation of the digits of the given number.
(B) Design a shell script to reverse a given number.
(C) Design a shell script for a simple calculator.

Tools: Linux operating system, terminal.

Procedure:

Make script in which user has to enter one number. Now design a script in which you have to add the digits of the number.
Make script in which number entered by the user will reverse.
Design a calculator in which the basic operation like addition, multiplication, division and subtraction is done. Here the value for each and every operation should be entered by the user.

TUTORIAL-8

□□Answer the following questions.

1. Explain basename,set and shift command.
2. Explain HERE document in short

EXPERIMENT-8

Aim:

(A) Design a shell script to count no. of ordinary files and directories in your system

(B) Create Following Pattern

```
    *                              1
   * *                            22
  * * *                          333
 * * * *                        4444
* * * * *                      55555
```

Tools: Linux operating system, terminal.

Procedure:

What do you mean by function explain it briefly. Design a shell script using function. Design a script using function for a calculator.

TUTORIAL-9

Answer the following questions.

1. Program & Programming Language
2. Compiler/Interpreter/Linker/Debugger
3. Basic Structure of C Program.

EXPERIMENT-9

Aim: (A) Overview of Turbo C++ IDE.
(B) Overview of GCC.
(C) First Program: Display Message - “Hello World” on Standard

Tools: Turbo C.

Procedure:
□ Explain what the turbo C++ IDE is.
□ Give the overview of GCC compiler.
□ Make a C program to display the message “Hell World” in Linux and Borland C.

TUTORIAL-10

Answer the following questions.

1. What is a comment? Why is it necessary?
2. Data types and use of *sizeof* operator
3. Classify briefly all operators
4. Difference between = and = =
5. What is the difference between prefix and postfix operators (i++, ++i)

EXPERIMENT-10

Aim:
(A) Use of \n, \t and escape sequences.
(B) W.A.P to convert the temperature unit from Fahrenheit to Celsius using the formula C = (F – 32) / 1.8.
(C) Assume that any month is of 30 days. Now you are given total days. Find out the exact number of Years, Months & Days.
(D) You are given time in total seconds. Convert it into Hour: Min: Seconds format.

Tools: Turbo C.

Procedure:

- Explain the use of \n, \t and escape sequences.
- Make a C program in which converts the temperature taken by user in Fahrenheit to Celsius. And for conversion use the formula which is given.
- Make a program in which the total numbers of days are given and you have to convert that in perfect no of days, year and month.
- Same as the above program but here instead of days the seconds are given and you need to convert it into hour, min and seconds format.

TUTORIAL-11

Answer the following questions.

1. Explain the conditional operator (exp1 ? exp2 : exp3).
2. Explain the different storage classes (auto, static, extern, register).
3. What is initialization? Why is it important?
4. What are enumeration variables?

EXPERIMENT-11

Aim: (A) W.A.P to determine whether input number is ODD or EVEN. Display appropriate
message.
(B) W.A.P that will display Grade of student according to his/her marks using if else ladder.
(Write same program using switch case also)

Marks > 80 then Grade = A
Marks between 61 & 80 then Grade = B
Marks between 51 & 60 then Grade = C
Marks between 41 & 50 then Grade = D
Marks between 35 & 40 then Grade = E
Marks < 35 then Grade = F

(C) W.A.P that computes and prints the Factorial of a given number.
(D) W.A.P that computes and prints the Fibonacci series.

Tools: Turbo C.
Procedure:

□ Enter the number. And make the coding that if the entered number is odd than display the message the “No is odd”. Else display that “No is even”.
□ Make a C code for grading a student based on the marks. You need to use the if else condition. And also make the same using switch case condition.
□Write a program to make the factorial of a given number. In that user enter a number and you have to find the factorial of a given number. (e.g. 5! =5 * 4 * 3 * 2 * 1)
□ Write a program to generate the Fibonacci series for n terms. Where the value for n is given by the user.

TUTORIAL-12

Answer the following questions.

1. Why the loop is required?
2. Explain Forward Jump.
3. Explain Backward Jump.
4. Explain Counter-Controlled and Sentinel-Controlled Loops.

EXPERIMENT-12

Aim: (A) W.A.P to print the following

```
      *                                    1
     * *                                  1 2
     * * *                                1 2 3
     * * * *                                    1 2 3 4
     * * * * *                            1 2 3 4 5

     *                                         1
    * *                                       22
   * * *                                     333
  * * * *                                   4444
 * * * * *                                 55555
```

(B) W.A.P to count Blanks, Tabs and Newlines using *while and getchar.*
(C) Simple calculator using *do while.*
(D) W.A.P to check whether the input number is prime or not.
(E) W.A.P. to display even numbers between 2 to 20 without using the modulo operation.

Tools: Turbo C.
Procedure:
□Using loops generate all the patterns.
□ Write a C code which will count the no of blanks, tabs and newlines in a program, using while and getchar. Here the string is taken by getchar function which is used to get a character from the user.
□ Make a program in which no is entered by user and u need to check that no is prime or not. Divide the no by 2 and at the end if u get 1 than that no is prime else not. Write a C code which displays the even numbers between 2 to 20.

References

Reference Books

Unix Concepts and Applications , 4th Ed. Sumitabha Das
Linux Shell Script Programming by Todd Meadors
Unix Shell Programming by Stephen Kochan (Revised ed)
Linux and Unix shell Programming by David Tansley
Sams Teach yourself shell programming in 24 hours, 2nd ed by Sriranga Veeraraghavan

Programming in ANSI C by Balagurusamy, 2nd Ed., TMH
http://www.oopweb.com/CPP/Download/crc.zip

http://www.freebookcentre.net/programming-books-download/C-Programming- Tutorial.html