Laboratory Manual

For

LINUX Operating System & Programming

(CT 115)

B.Tech (IT)

SEM I

June 2010

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

Table of Contents

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

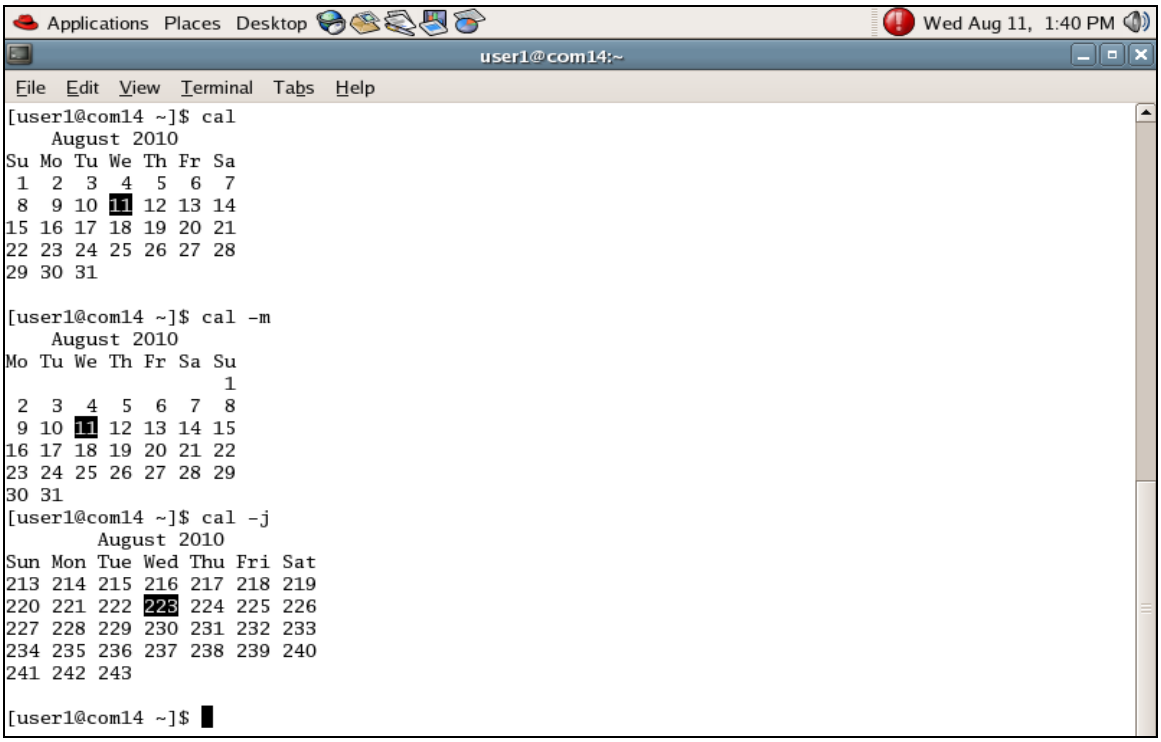
5) MKDIR and RM:

At the terminal write the following:

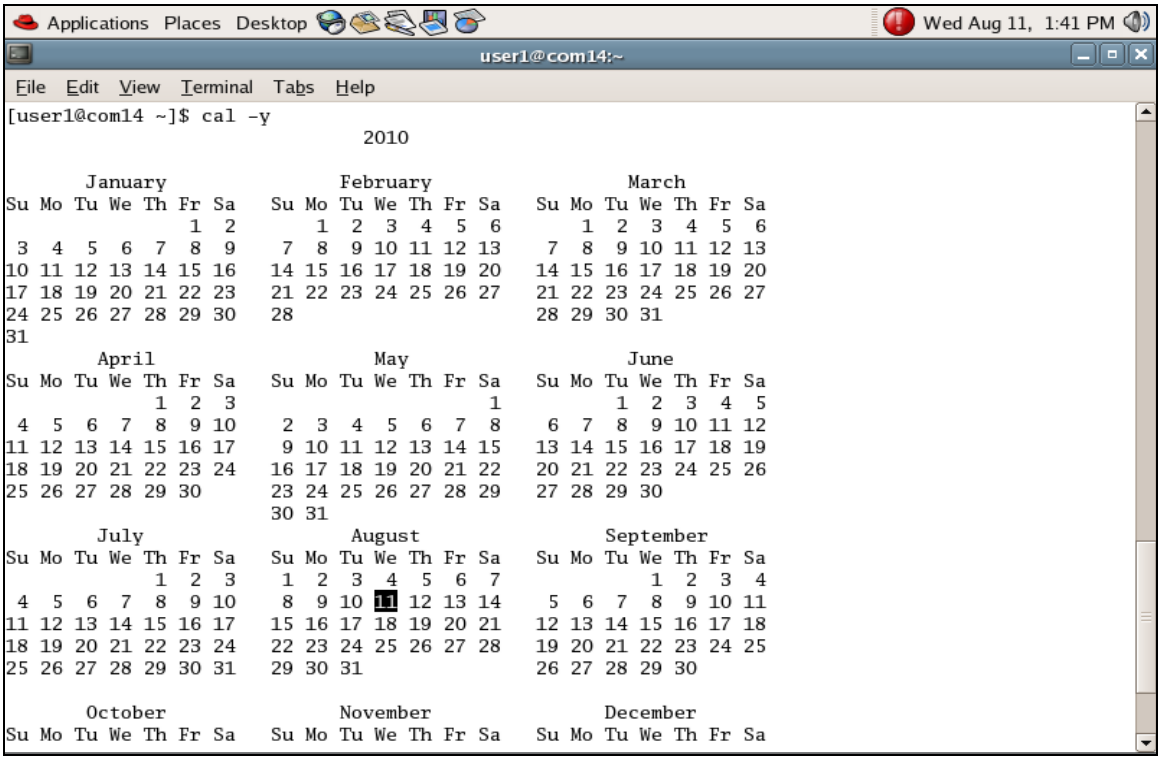
- [user1@com]$ cd Desktop/
- [user1@com]$ ls
- [user1@com]$ cd newfiles/
- [user1@com]$ ls
- [user1@com]$ mkdir newfile1
- [user1@com]$ ls
- [user1@com]$ rm Sum_Of_Digits.txt
- [user1@com]$ ls

3.3.2 Compilation /Running and Debugging the Solution:

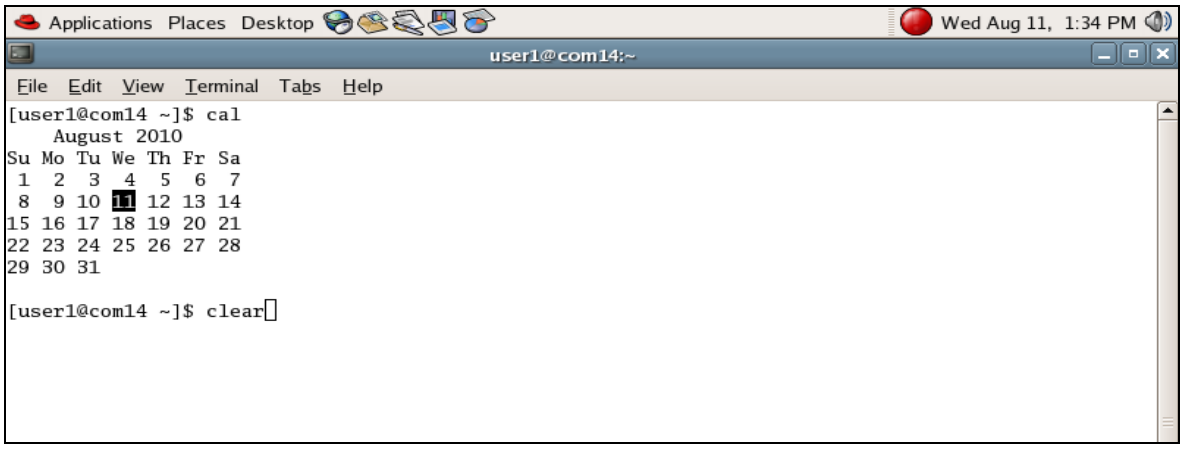
- The code written above will display the following output.
- For the first command CAL the output is like this:

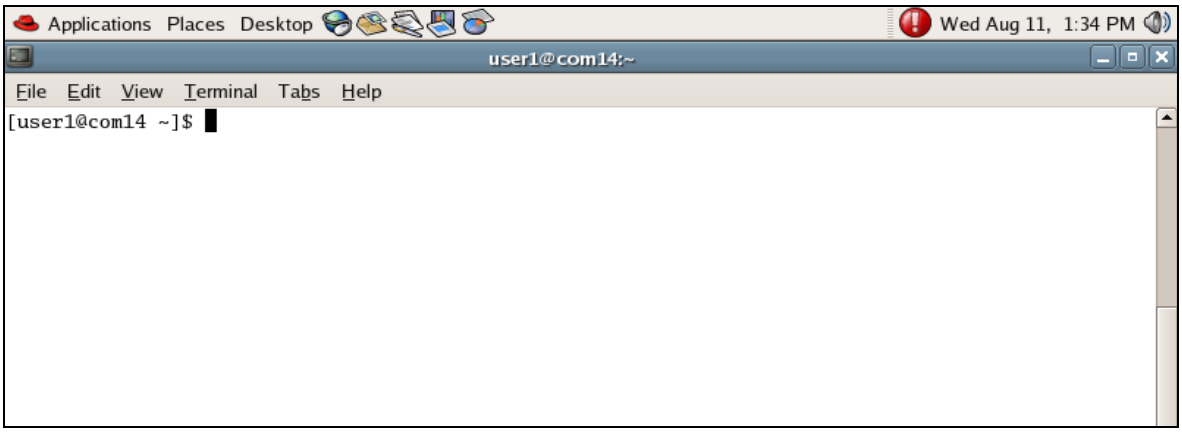


- The cal command with the option y will display the following output.

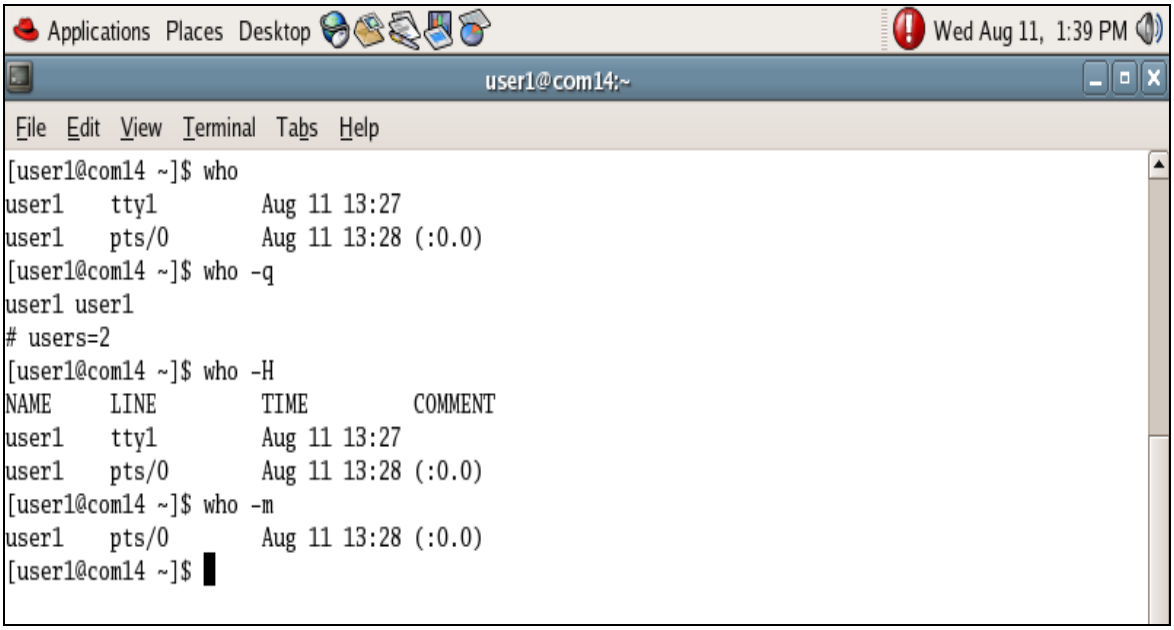


- For the second command CLEAR :

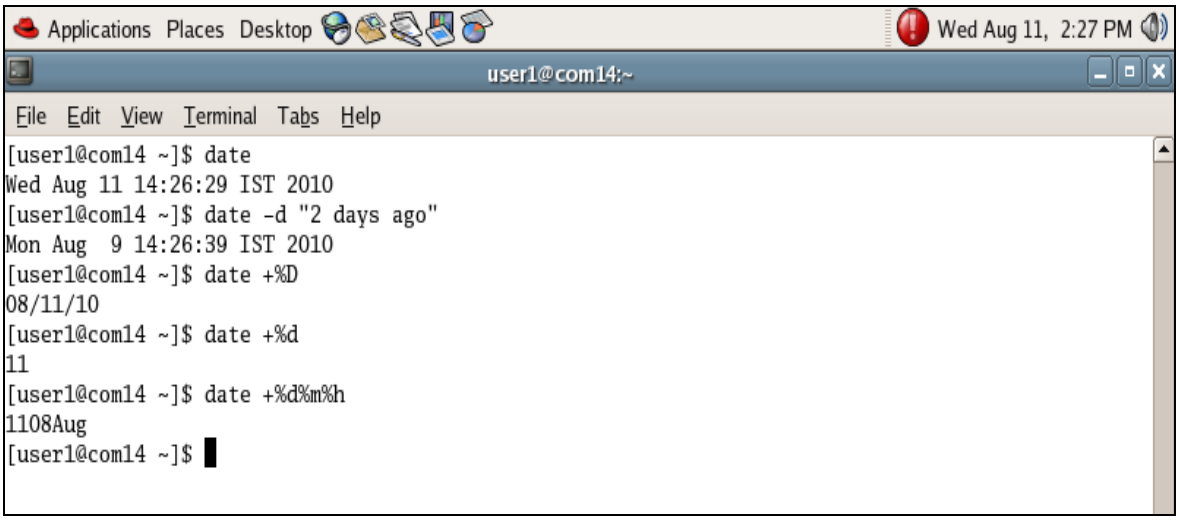




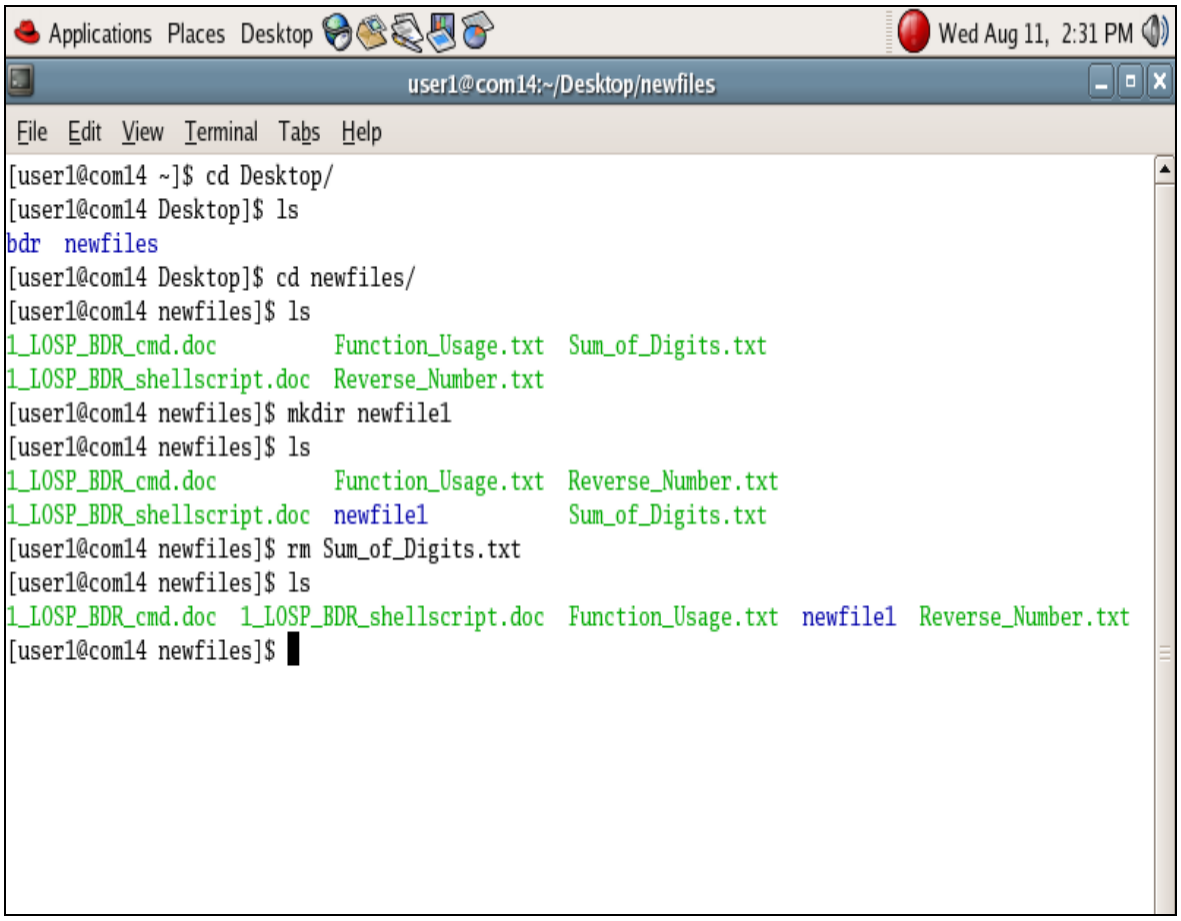
- For the third command WHO :



- For the command DATE:



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



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.

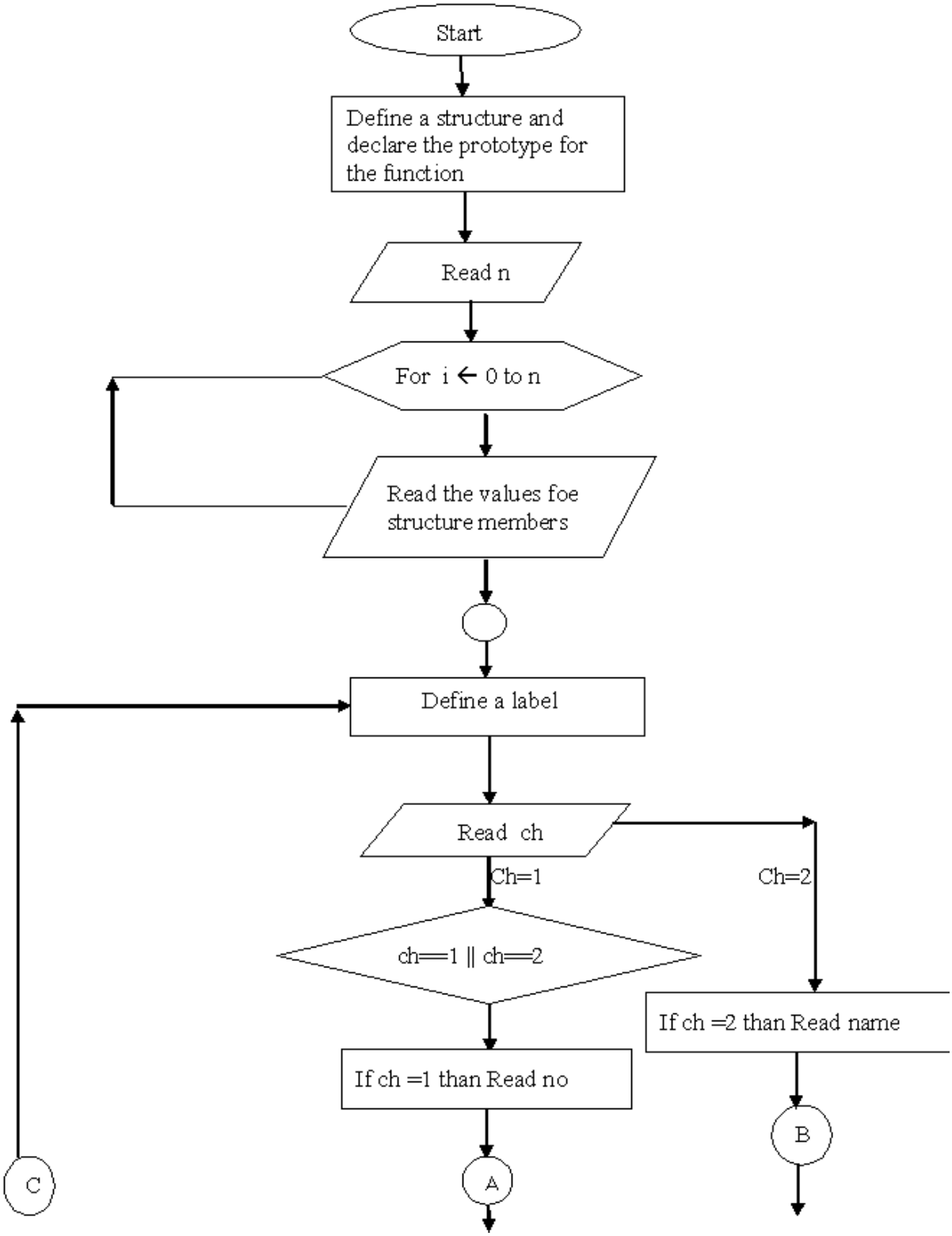
3.1 Analyzing the Problem:

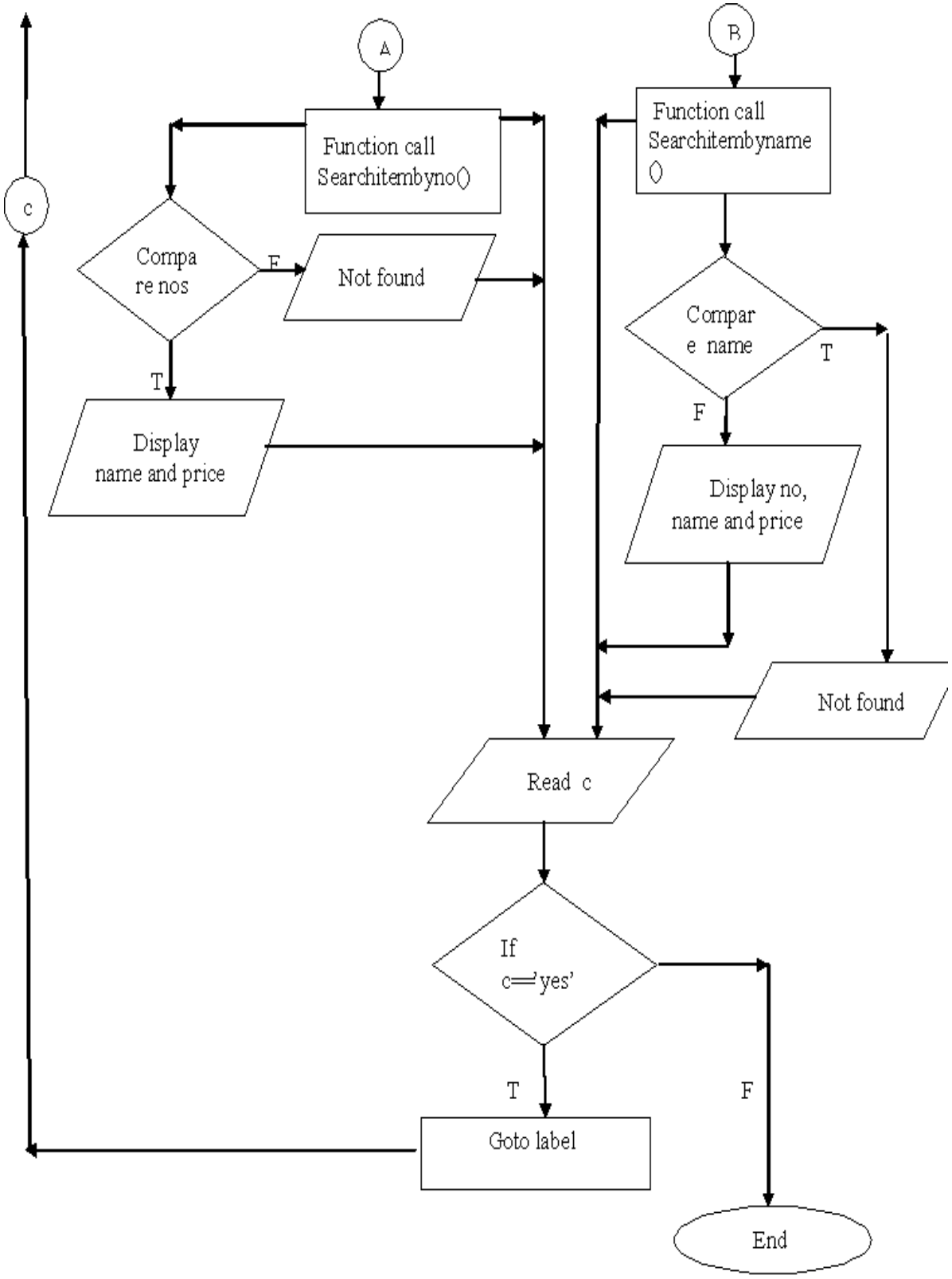
- Firs create a file named as “stru.c”.
- After that includes the standard input/output files.
- Now define the structure “item” and its members.
- Create the functions which are necessary for the program.
- Now create the main function and take the information.
- After that quit from the terminal using :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
Read n
For i ← 0 to n
Read the values foe structure members
Define a label
Read ch
Ch=1
Ch=2
ch==1 || ch==2
If ch =2 than Read name
If ch =1 than Read no
B
A
C


A
B
c
Function call
Searchitembyno()
Function call
Searchitembyname
()
Compa
re nos
F
Not found
T
Display
name and price
Compar
e name
T
F
Display no,
name and price
Not found
Read c
If
c=='yes'
T
F
Goto label
End

3.3 Implementing the Solution:

3.3.1 Writing Source Code:

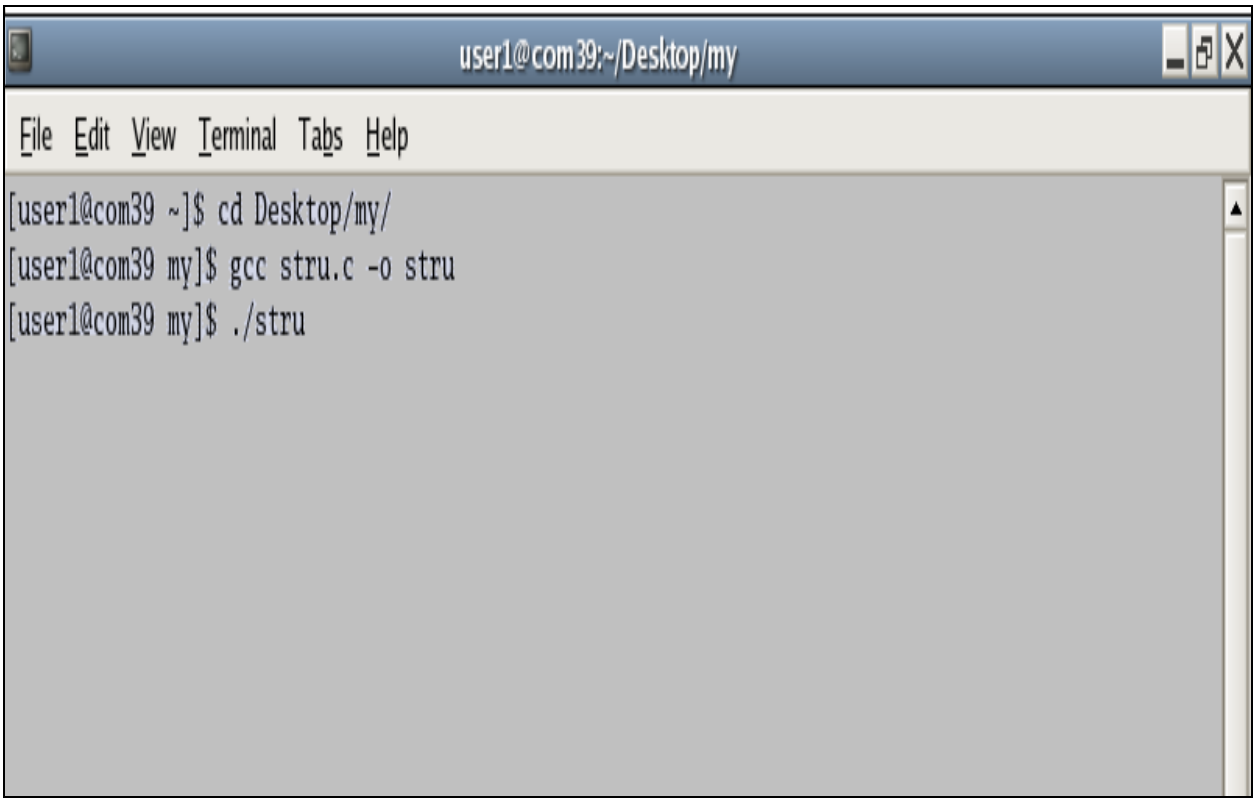
```
#include<stdio.h>

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 ");
   }
}
```

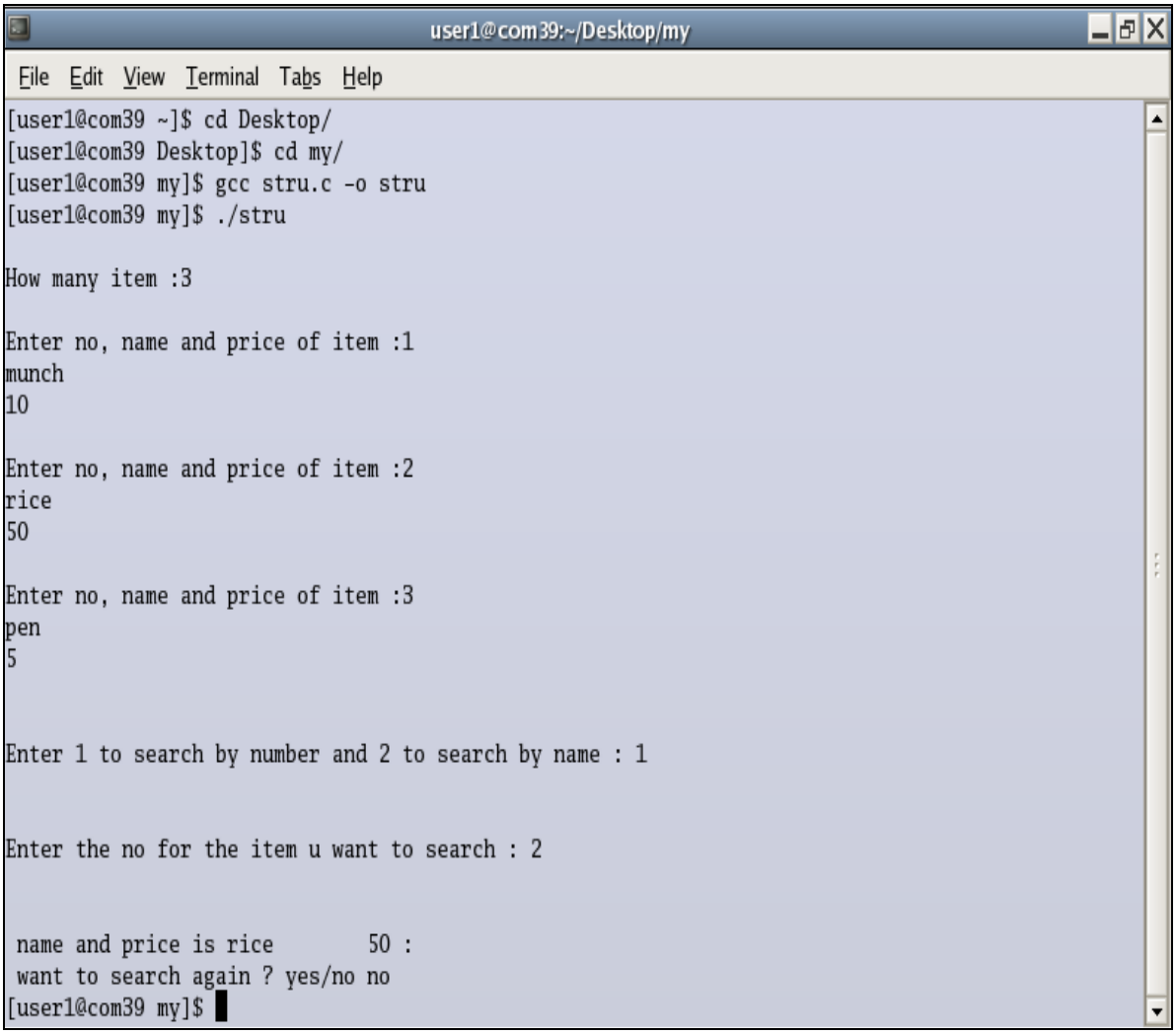
3.3.2 Compilation /Running and Debugging the Solution:

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



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

$ ./stru



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:

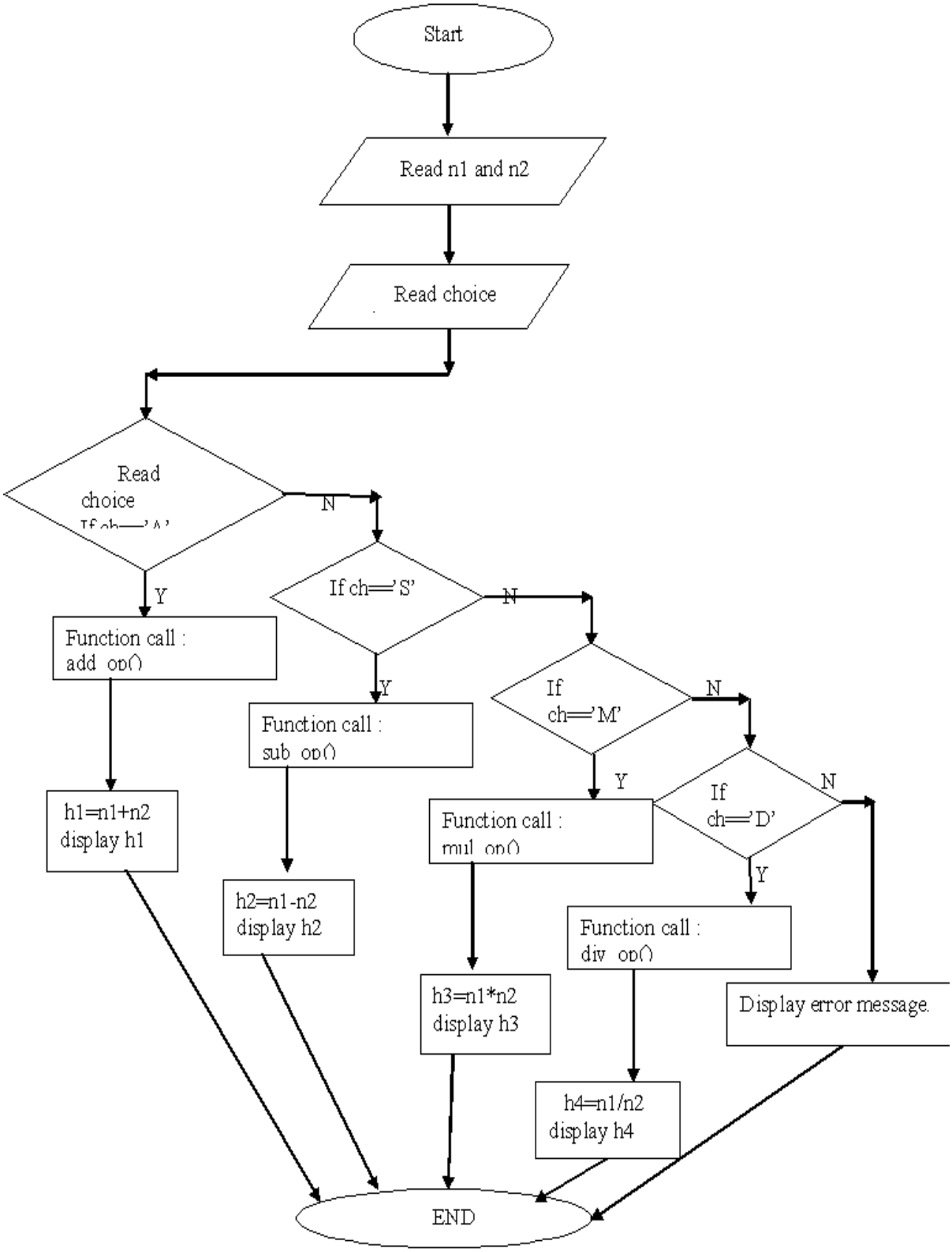
3.1 Analyzing the Problem:

- 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:


Start
Read n1 and n2
Read choice
Read choice
If ch=='A'
N
Y
Function call : add op()
h1=n1+n2 display h1
If ch=='S'
N
Y
Function call : sub op()
h2=n1-n2 display h2
If ch=='M'
N
Y
Function call : mul op()
h3=n1*n2 display h3
If ch=='D'
N
Y
Function call : div op()
h4=n1/n2 display h4
Display error message.
END

3.3 Implementing the Solution:

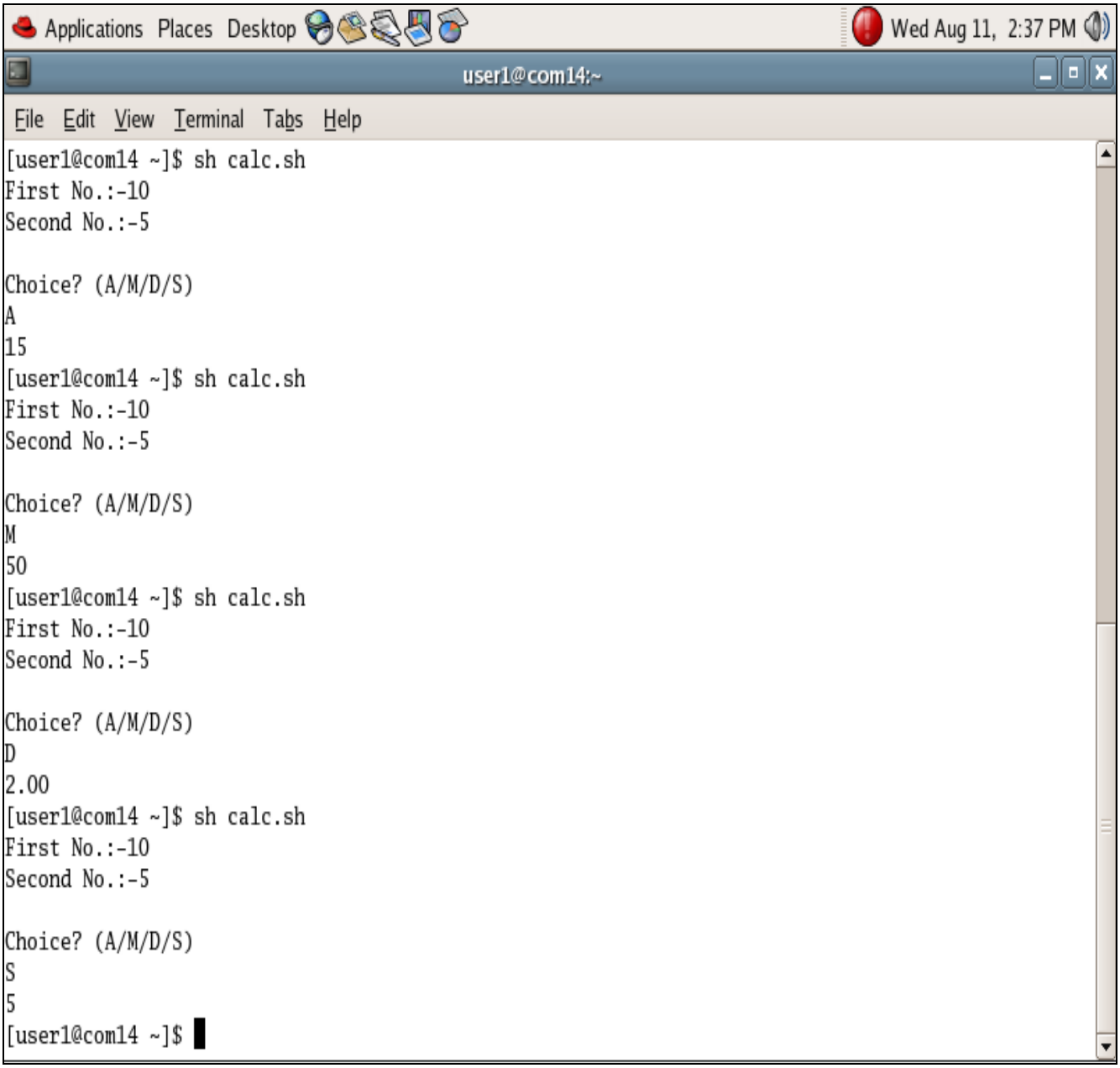
3.3.1 Writing Source Code:

```
#!/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:



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. File System of Unix
    a. Ordinary Files
    b. Directory Files
    c. Device Files
5. Characteristics of Unix System
    a. Multi-User
    b. Multi-Tasking

EXPERIMENT-1

Aim: (A) Overview of the UNIX Operating System.

- Explain that what UNIX is. What is the difference between UNIX and Linux.
- What is the history of UNIX as well as Linux.

Tools: Linux operating system, terminal.

Procedure:

- Explain the history of the UNIX operating system. As well as what is Linux. Difference between UNIX and 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.
4. Explain the use of > and <.

EXPERIMENT-2

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

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. With the use of cp, mkdir, rm, cat and rmdir, make a tree structure.
2. Explain the file command.
3. Explain the wc command.
4. Explain the difference between mv and cp.
5. Explain absolute and relative paths.

EXPERIMENT-3

Aim: Explain the following commands: cd, mkdir, rmdir, rm, cp, file, 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-4

- Answer the following questions.

1. Explain the bc command and the following
    a. ibase
    b. obase
    c. scale
    d. length()
2. Explain all the options of ls command.

EXPERIMENT-4

Aim: Explain the following commands: Ps, more, less, alias, ln.

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-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.

EXPERIMENT-5

Aim: Explain the following commands: Ps Cmp, comm, diff, od.

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 difference between chgrp and chown in detail.
2. Explain read, write and execute permission for a file and a directory.

EXPERIMENT-6

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-7

- Answer the following questions.

1. Explain the three modes of vi editor in detail.
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
3. Find everything in your home directory modified in last 24 hours

EXPERIMENT-7

Aim: Explain the following commands: unmask, find, and touch. Also explain VI editor in detail.

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-8

- Answer the following questions.

1. Explain the applications of wild-card characters in detail.

EXPERIMENT-8

Aim: (A) Explain what wild-card characters are.
(B) Explain the following wild-card characters:
  - *
  - ?
  - Character Set [,]

(C) Explain the use of Redirection
(D) Explain the use of Escaping and Quoting

Tools: Linux operating system, terminal.

Procedure:

- Explain about the wild card characters. And also use these characters.
- Explain about the redirection and also use it.
- Explain Escaping and Quoting with proper examples.

TUTORIAL-9

- Answer the following questions.

1. Solve the following questions using grep filter
    a. What would be the output of grep a b c?
    b. What is the meaning of grep <include> filename?
    c. How would you match a file name starting with two carats?
    d. Count the number of lines having 'printf' in a file and store it in a variable.
    e. How would you list only the soft-links from the ls command?

EXPERIMENT-9

Aim: Explain the following filters: head, tail, cut, paste, sort, uniq, tr.

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-10

- Answer the following questions.

1. Solve the following questions using sed filter
    a. How would you insert a line above a particular line?
    b. How would you replace a particular word?
    c. How would you insert START above the first line and END below the last line?
    d. How would you replace Director with Executive Director?
2. What do you mean by ch \ {m,n\ }?

EXPERIMENT-10

Aim: Explain the following advance filter: grep, sed.

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-11

- Answer the following questions.

1. Explain the use of expr command in detail.
2. Explain the control loops like:
    a. if
    b. if-then-else
    c. if-then-elif-then-else
    d. case structure

EXPERIMENT-11

Aim: (A) Design a Shell Script where there are three hard links of a file named calendar.sh, date.sh and list.sh. The script should behave differently depending on the name of the script through which it is run.
(B) To 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-12

- 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-12

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-13

- Answer the following questions.

1. Design a Shell Script for user-interactive calculator using functions.

EXPERIMENT-13

Aim: (A) Explain the usage of Functions in Shell Scripts.
(B) Design a Shell Script for user-interactive calculator using functions.

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-14

- Answer the following questions.

1. Which are the different types of loops, explain it with examples.

EXPERIMENT-14

Aim: Using loops, create the following patterns:

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

Tools: Linux operating system, terminal.

Procedure:

- Write a script to design the given pattern.

References

Reference Books

- 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
- http://sunsite.dk/linux-newbie
- http://www.ssc.com/ssc/bash/%20bash.pdf
- http://www.pixelbeat.org/