Assignment 2

CAP 448

Corse title: Linux and shell scripting

Roll No. RD2112A103
NAME - JAYSHRI LAL PANDIT
Reg NO. 12111670
Section :- D2112

Assignement Set :- set B

(1.) What is the importance of a text editors ? Discuss in detail the different text editors used in Red Hat Linux. Also differentiate CLI editors and GUI editors.

Ans:- The text editors are one of the most frequently used applications on Linux systems. Linux text editors can be used for editing text files, writing codes, updating user instruction files, and more. A Linux system supports multiple text editors. There are two types of text editors in Linux, which are given below :-

(i) Commond line text editors
(ii) GUI text editors

The different text editors used in Red Hat Linux discussed in details given below :-

- **vi / VIM Editor :-**

vim editor is one of the most used and powerful command-line based editor of the Linux system. It is a user-friendly

editor. It is also called programmer's editor because most programmer prefer vi editor. It has enhanced functionalities of old vi editor. By default, it is supported by most Linux distors and provides the same enviroment for the all the Linux distors. Vi editor has some special features such as vi modes and syntax highliting that makes it powerful than other text editors. Generally, it has two modes :-

(i) Command Mode :- The command mode allows us to Perform actions on files. By default, it Starts in command mode. In this mode, all types of words are considered as commands. We can execute command in this mode.

(ii) Insert Mode :- The insert mode allows to insert text on files. To switch from command mode to insert mode, press the Esc to exit from active and 'i' key.

- **Gedit Editor :-**

Gedit editor is the default editor for the GNOME desktop environment.

When we open a file, it will open with the Gedit editor. It provides straightforward functionalities like any basic text editor. It is a lightweight editor with a straight forward user interface. It was publicly released in the year 2000 with a GNOME desktop enviroment. It is developed using the C Programming language and supports all font family.

Some key features of the gedit text editor are as following:—

- It provides sntax highlighting.
- It supports internationalized text.
- It supports several programming language.

- **Nano editor**

Nano is a straight forward editor. It is designed for both beginners and advanced users. It has many customization features. Some advanced features of a nano text editor are as following:—

(i) It supports syntax highlighting
(ii) It has undo and redo options
(iii) It has paper support to read

from standard ~~op~~ input.

- ~~Sum~~ Sublime Text :-

The sublime text editor is also one of the most popular IDE-based text editor. It is used as a development enviroment tool more than a text editor. It has several features to support many programming and ~~mark~~ Mark-up languages. Further, it supports numerous pluging to make it more than text editor.

Some key features of a sublime text editor are as following :-

(i) It has an excellent command palette.
(ii) It is a python-based ~~plu~~ plugin API.
(iii) It supports ~~parrel~~ parallel editing of code.
(iv) It provides project-specific preferences.

- ~~VS Code~~ VSCode
- GNU emacs
- Atom editor
- Brackets editor
- Pico editor
- Bluefish
- Kate/Kwrite

etc.

Differentiation of CLI editors and GUI editors are given below:-

CLI	GUI
(1) It stands for command line interface	(1) It stands for Graphical user interface.
(2) It is very fast in execution.	(2) It is slower in execution.
(3) user should have good knowledge of commands.	(3) A beginner user can easily handle it.
(4) It does not require more memory.	(4) It requires more memory as it contains a lot of graphical components.
(5) It is not possible to change the appearance.	(5) It has a customize options to change the appearance.
(6) It has a interface for the user to type the command lines to perform the tasks.	(6) It has a interface that allows users to intract with graphical icons and visual indicators.

(2.) Write down the steps to write and execute a C program to find Fibonacci series in shell.

Ans:- The steps to write and execute a C program to find fibonacci series in shell is given below:-

Step 1: Mount CD 2.

Step 2: Open new terminal and install gcc package by using below command

```
# rpm -ivh /mnt/cdrom/RedHat/RPMS/gcc
-3.2.2-5.i38c.rpm
```

Step 3: Now go to new terminal and create a file b fibonacciserie.c by using gedit command. When we press enter it ask "would you like to create it?" click on yes.

Step 4: Now write the fibonacci series program and save it

```
#include <stdio.h>
int main()
{
```

```
int n1=0, n2=1, n3, i, number;
printf(" Enter the number of terms:");
scanf(" %d ", &number);
printf(" \n %d %d ", n1, n2);

for (i=2; i<number; ++i)
{
   n3 = n1 + n2;
   printf(" %d ", n3);
   n1 = n2;
   n2 = n3;
}
return 0;

}
```

Step 5:- Now compile this program by using below command:-

~~# gcc -o~~

gcc -o fibonacciserie fibonacciserie.c

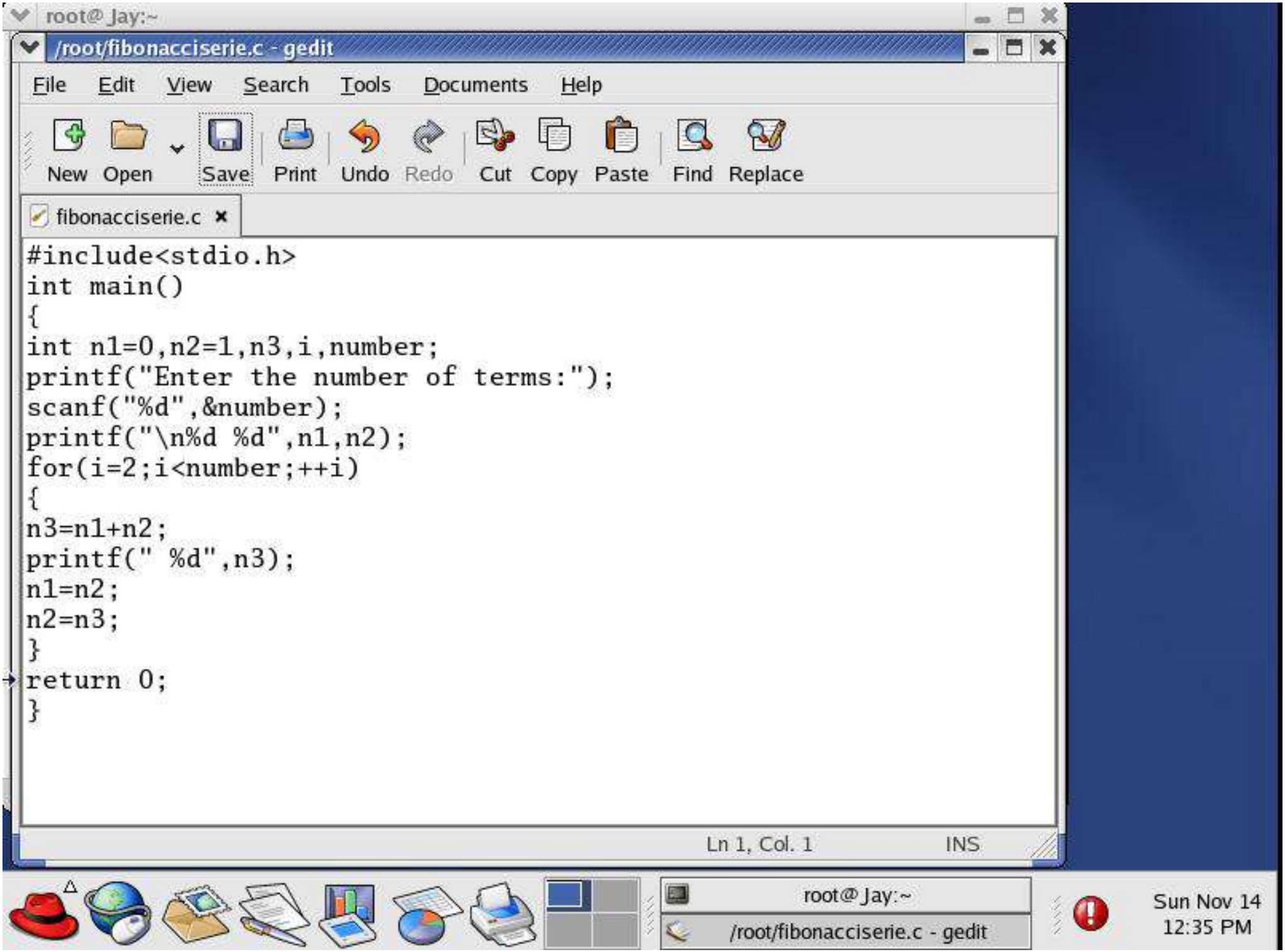
Step 6: To execute this program type the below command:-

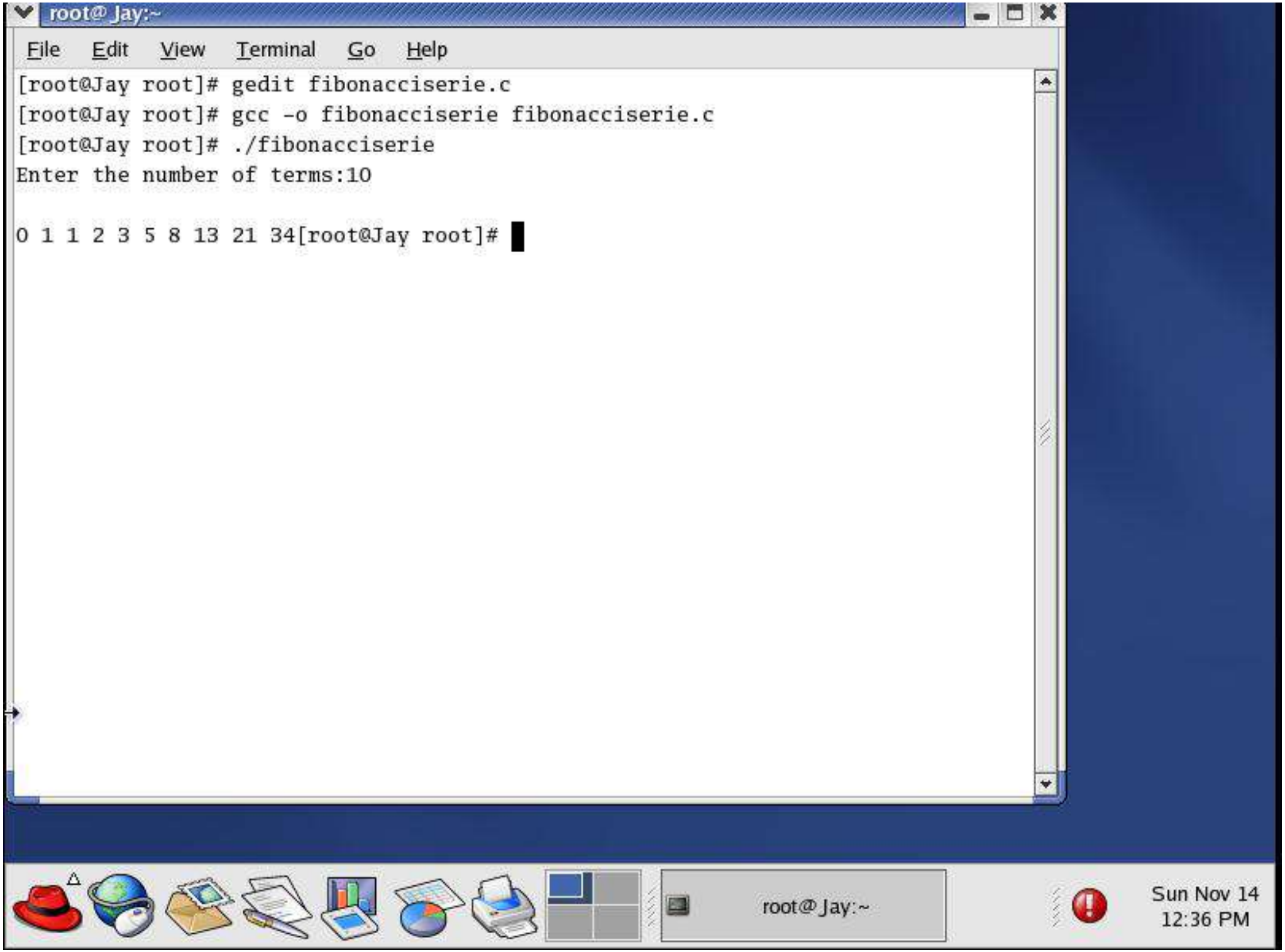
./fibonacciserie

Output

Enter the number of terms: 10 (we enter)

0, 1 1 2 3 5 8 13 21 34


root@ Jay:~
/root/fibonacciserie.c - gedit
File Edit View Search Tools Documents Help
New Open Save Print Undo Redo Cut Copy Paste Find Replace
fibonacciserie.c
#include<stdio.h>
int main()
{
int n1=0,n2=1,n3,i,number;
printf("Enter the number of terms:");
scanf("%d",&number);
printf("\n%d %d",n1,n2);
for(i=2;i<number;++i)
{
n3=n1+n2;
printf(" %d",n3);
n1=n2;
n2=n3;
}
return 0;
}
Ln 1, Col. 1
INS
root@ Jay:~
/root/fibonacciserie.c - gedit
Sun Nov 14
12:35 PM


root@ Jay:~
File Edit View Terminal Go Help
[root@Jay root]# gedit fibonacciserie.c
[root@Jay root]# gcc -o fibonacciserie fibonacciserie.c
[root@Jay root]# ./fibonacciserie
Enter the number of terms:10
0 1 1 2 3 5 8 13 21 34[root@Jay root]#
root@ Jay:~
Sun Nov 14
12:36 PM

(3) What is the need of file permissions? Discuss. Take examples and also explain in detail the different types of file permissions along with user groups.

Ans:→ The need of file permissions because it help support user privacy by protecting access to the following restricted data, such as system State and a user's contact information.

if a user has read permission, he or she can read but not modify the file. A user needs curite permissions to modify the file. To run the file as a compiled program, the user needs execute Permissions. Basically there are two ~~types~~ ways of file permission :-

(i) Permission Types
(ii) Permission Groups

(i) Every file has three basic permission type :-

- Read :- The read permission refer to a user's capability to read the contents of the file.
- write :- The write permissions refers

to a user's capability to write or modify a file.

- execute :- The execute permission affects a user's capability to execute a file.

(ii) Every file have three sets of people for whom we may specify permissions.

- owner :- The owner permissions apply only the owner of the file. they not impact the actions of other users.
- group :- The Group permissions apply only the group that has been assigned to the file. They not affect the actions of other users.
- all user :- The all users permissions apply to all other users on the system.

Note:- you can view the permissions of a file with ls -l command

If you wanted to add or remove

Permissions to the user, use the command "chmod" with "+" or "-" along with the ~~r (read~~ "u" for users, "g" for group, and "o" for others user.

~~for example~~

If wanted to add or remove permissions to the user, use the command "chmod" with a "+" or "-" along with the r (read), w (write), x (execute) attribute followed by the name of the file or directory.

for example, chmod +rwx "name of the file"

chmod -rwx "name of the dire"

e.g chmod +rwx Jay.txt

chmod -rwx Jay.txt

The commond is similar to the previous used. but this you add a "g" for group or "o" for users.

chmod g+w Jay.txt

chmod g-wx Jay.txt

The end