MULTIPLE FILE HANDLER

Presented by :

Somya Srivastava

Kunal Sharma

Software Requirements :

Operating System : Ubuntu

Compiler : GCC

Interface : Terminal Based Application

Hardware Requirements :

RAM : minimum 512 MB

OS type: 32\64 bits

Backup Media :

PEN DRIVE, COMPACT DISC

Operational feasibility :

The person having the working knowledge of the computer can easily

operate the system; he or she does not require any sort of training.

Economic feasibility

Softwares used in this project are freely available , only the hardware's cost is taken into account.

The following are the benefits of the project :

Cost reduction

This system would reduce the cost in paper and paper related

tasks. This would also reduce cost of operating the system

Manually.

Improve service-level benefits

The proposed system improves the system performance because the older system is based on manual processing while proposed system is based on computer processing.

Time saving benefits

It saves a lot of time and provides same result in very less amount of time.

Management feasibility

This system would be feasible because the organization has to maintain a lot of records.

Problem Definition

This project has been designed to make the organization of any type automated.

---To provide the organization with the facility to gather the services available.

---To make the information flow easier.

---To keep the confidential data safe .

Modules Implemented :

This project combines 4 modules :

Module 1. CRUD on Multiple Files

Module 2. Encryption-Decryption on Multiple Files

Module 3. Sending-Recieving of file through Socket Programming

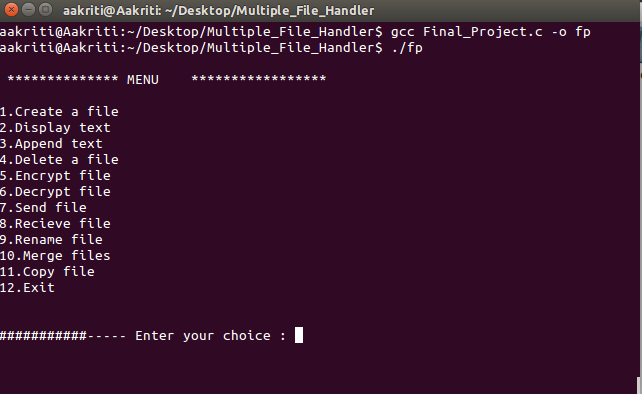
Module 4. Other functions to be performed on Multiple Files(Copying,Merging,Renaming).

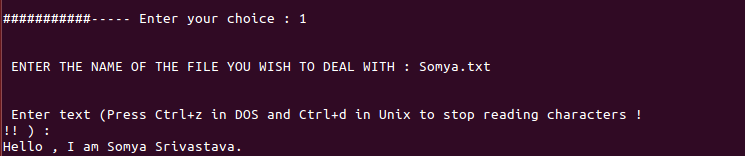
1. CRUD on Multiple Files

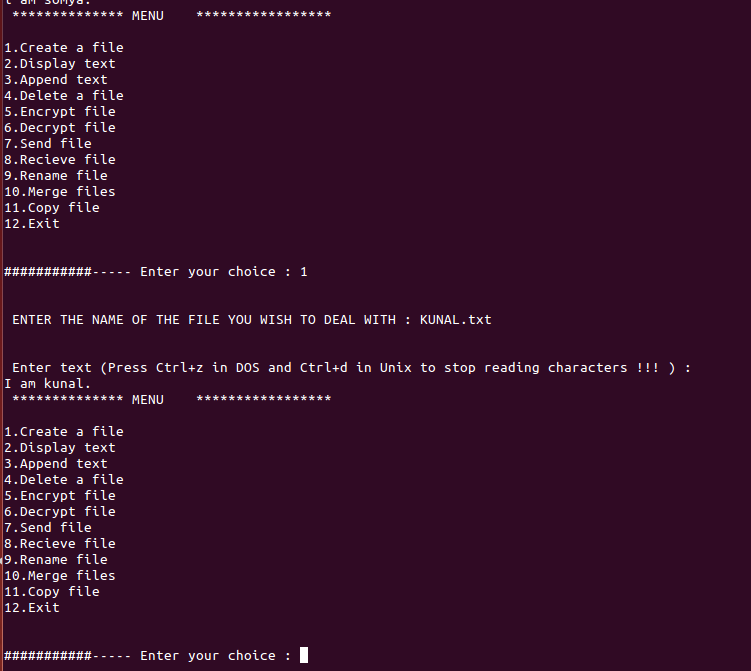
1.1. CREATE :

Multiple files can be created using this module.

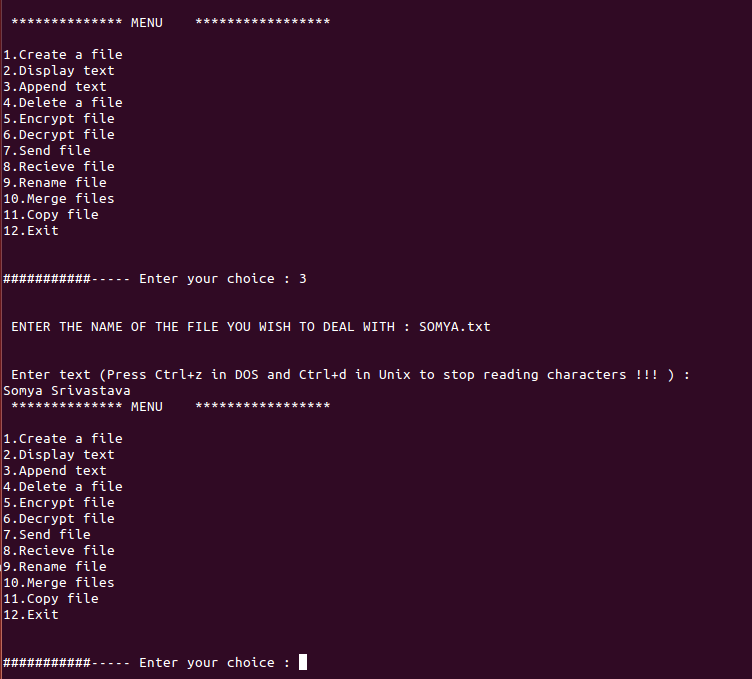
The user is asked to enter the filename and then the file is created.

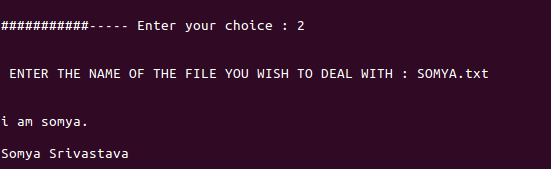




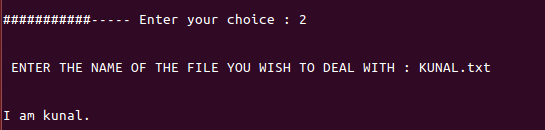


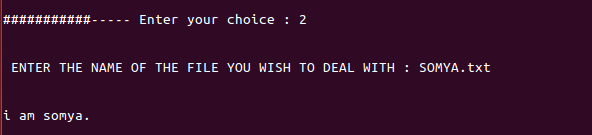
1.2. APPEND



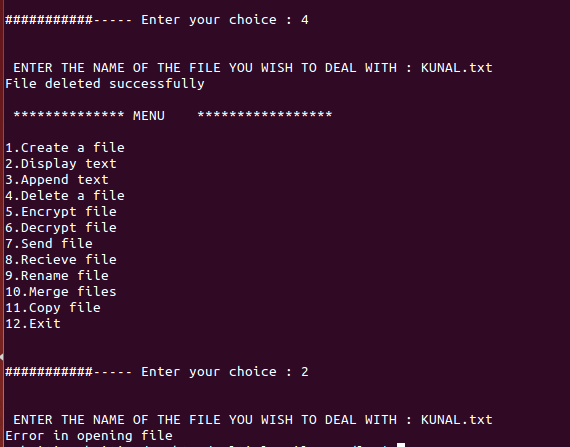


1.3. DISPLAY





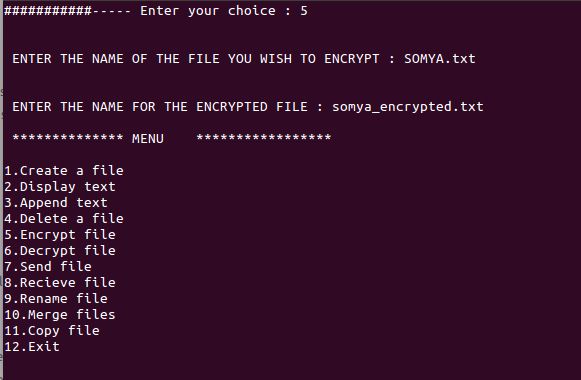
1.4. REMOVE

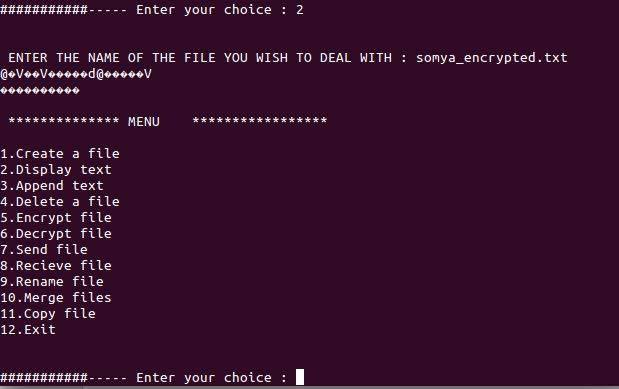


2. Encryption-Decryption on Multiple Files

2.1.ENCRYPTION

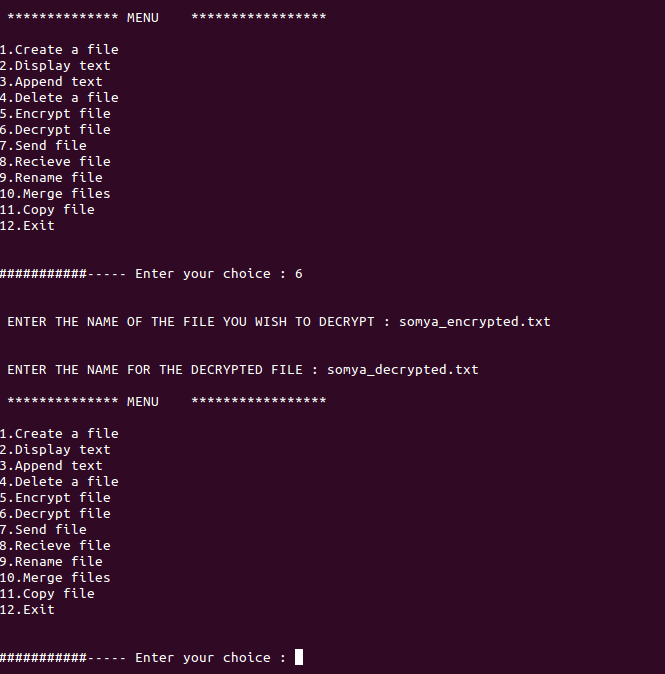
Multiple files can be encrypted on the system. The user can remove the original file , keeping the encrypted file on the system thus ensuring the security of the confedential data.

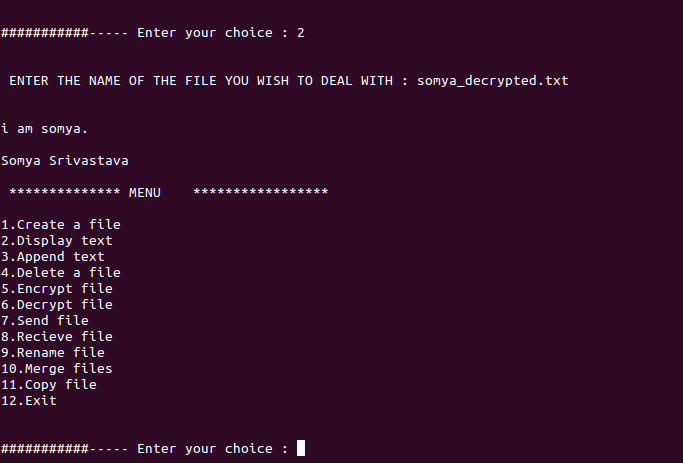




2.2DECRYPTION

For using the data , one can always decrypt the encrypted file .

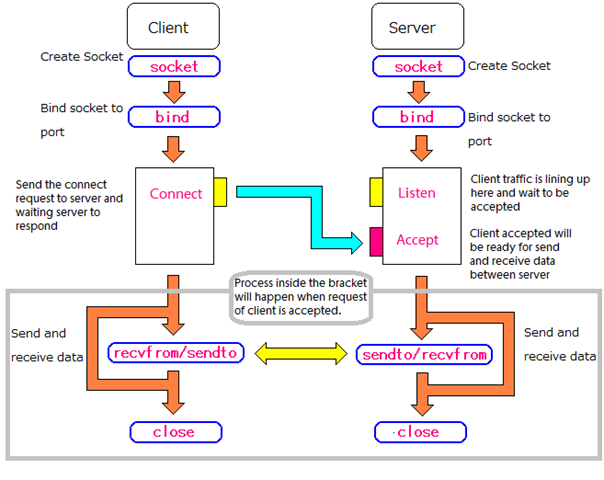




3. Sending-Recieving of file through Socket Programming

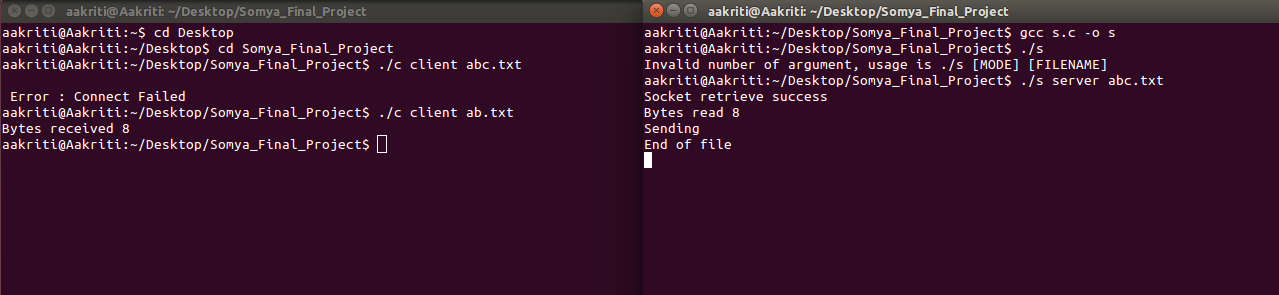
Multiple files can be send and recieved .

This modules utitlizes the client-server architecture .



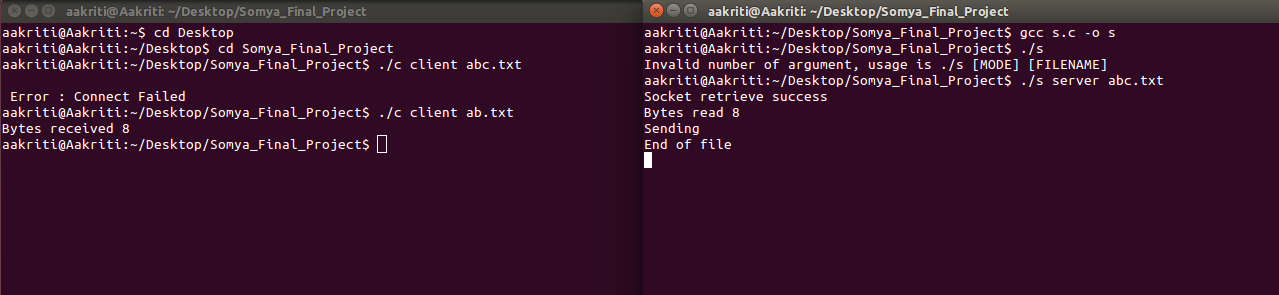
3.1.RECIEVING THE FILE

The client is activated when the user has to RECIEVE the data .



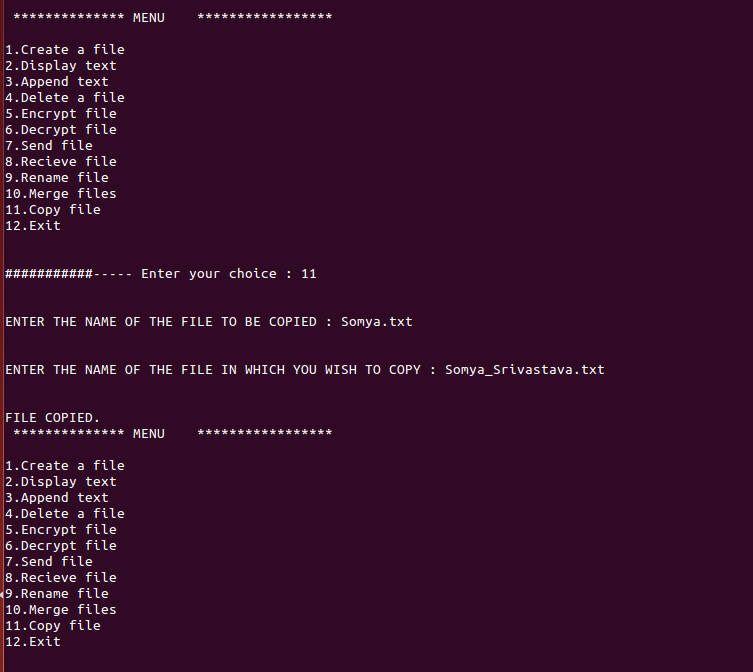
3.2.SENDING THE FILE

The server is activated when the user has to SEND the data.

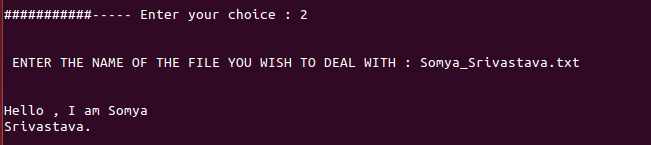


4. Other functions to be performed on Multiple Files

4.1.COPYING ONE FILE TO ANOTHER

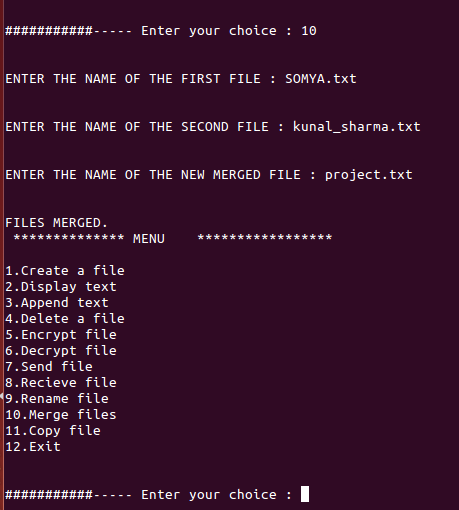


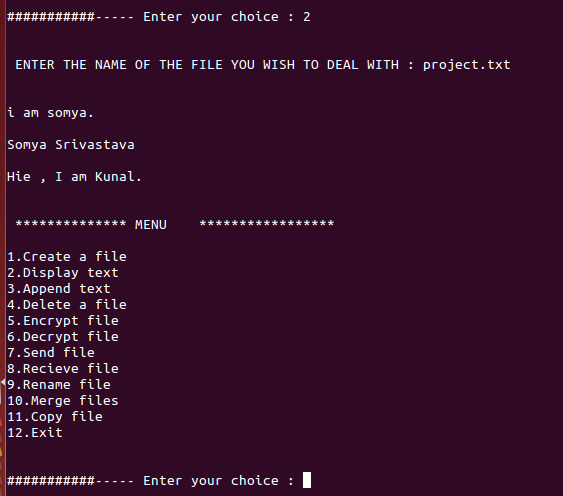
The user can copy any one file to another file.



4.2.MERGING TWO FILES INTO THIRD FILE

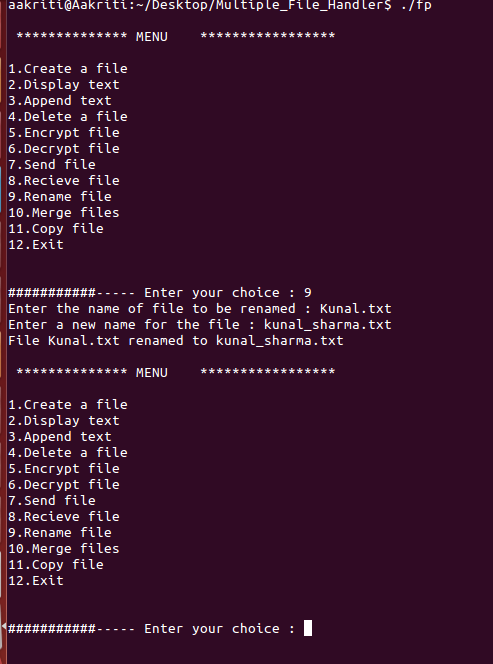
The user can merge any two files into third file.



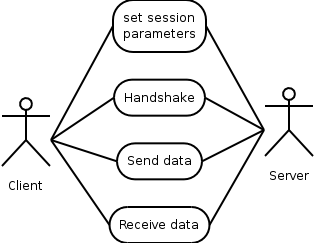


4.3.RENAMING ANY FILE

The user can rename any file.



USE CASE DIAGRAM FOR CLIENT-SERVER



Limitations

1.The code for sending and recieving file is quite unrealible due various reasons.

2.Only point to point half duplex communication is possible through socket programming.

Further Extensions

This project can be used as a general framework for any type of management system.Only the structures have to be fitted into the project.

For example

Employee management system

//GLOBAL DECLARATION OF STRUCTURE

struct date

{

int year,month,day;

};

struct person

{

int recno;

char name[20];

date dob; //using date structure as datatypeint salary;

};