TEXT EDITOR IN C WITH SOCKET PROGRAMMING

A

PROJECT REPORT

submitted in partial fulfillment of the requirements for the award of the degree of

BACHELOR OF TECHNOLOGY

in

COMPUTER SCIENCE Specialization in

Open Source & Open Standards

By:

Name Roll No

Shivam Prakash R100216080

Shaurya Gairola R100216078

Shivank Srivastava R100216082

Sneheel Shivam R100216089

Under the guidance of

TRIPTI MISRA

Assistant Professor Department of Systemics



Department of Cybernetics

School of Computer Science

UNIVERSITY OF PETROLEUM AND ENERGY STUDIES Dehradun-248007

December 2018



CANDIDATES DECLARATION

We hereby certify that the project work entitled **Text Editor in C with Socket Programming** in partial fulfillment of the requirements for the award of the Degree of Bachelor of Technology in Computer Science And Engineering with Specialization in Open Source & Open Standards and submitted to the Department of Cybernetics at School of Computer Science, University of Petroleum And Energy Studies, Dehradun, is an authentic record of our work carried out during a period from September, 2018 to December, 2018 under the supervision of **Tripti Misra, Assistant Professor Department of Systemics**.

The matter presented in this project has not been submitted by us for the award of any other degree of this or any other University.

(Name of Student(s)) Roll No.

Shivam Prakash R100216080

Shaurya Gairola R100216078

Shivank Srivastava R100216082

Sneheel Shivam R100216089

This is to certify that the above statement made by the candidate is correct to the best of my knowledge.

(Date: 17 December 2018) Tripti Misra

Project Guide

Dr. Monit Kapoor

Head

Department of Cybernetics

School of Computer Science

University of Petroleum and Energy Studies

Dehradun - 248007 (Uttarakhand)

ACKNOWLEDGEMENT

We wish to express our deep gratitude to our guide **Tripti Misra Ma’am**, for all the advice, encouragement and constant support she has given us throughout our project work. This work would not have been possible without her support and valuable suggestions.

We sincerely thank to our Head of the Department, **Dr. Monit Kapoor**, for his great support in doing our project name at SoCS.

We are also grateful to **Dr. Manish Prateek** Professor and Dean SCS and **Dr.**

**Kamal Bansal** Dean CoES, UPES for giving us the necessary facilities to carry out our project work successfully.

We would like to thank all our friends for their help and constructive criticism during our project work. Finally we have no words to express our sincere gratitude to our parents who have shown us this world and for every support they have given us.

Name Shivam Prakash Shaurya Gairola Shivank Srivastava Sneheel Shivam

Roll No. R100216080 R100216078 R100216082 R100216089

ABSTRACT

A text editor is a simple but a paramount facility required in every field, especially an organization like a University. Therefore, we intend to bring in a Basic Text Editor with some additional features scooped up. Encrypt/Decrypt feature is complimented in the Text Editor using algorithms. And it's not just a TEXT EDITOR; any document written could be transferred among various devices through an in-built file sharing application. The Project involves the use of Language C coupled with Socket Programming for File Sharing assistance. Thus, ultimately bringing to the table a Text Editor in C along with Socket Programming. The current scenario of software industry & educational field is related to software programming languages which have a huge collection, writing and transferring the files over the network. Therefore, an open source Text Editor integrated with file sharing can be exploited by a huge audience.

TABLE OF CONTENTS

Contents

1 Introduction 7

2 Literature Review 7

3 Problem Statement 7

4 Objective 7

5 Design Methodology 7

6 Implementation 7

6.1 Pseudocode . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 8

6.2 Output Screen . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 9

7 Schedule 11

7.1 Flow Chart. . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 11

7.2 Data Flow Diagram. . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 12

8 Result Analysis 13

9 Conclusion & Future Scope 13

A APPENDIX I PROJECT CODE 15

LIST OF FIGURES

List of Figures

1 Text Editor . . . . . . . . . . . . . . . . . . . . . . . . . . .. . . . . 9

2 Attributes of the Text Editor . . . . . . . . . . . . . . . . . . . . . . . 9

3 File Sharing . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 10

4 Encryption . . . . . . . . . . . . . . . . . . . . . . . . . . .. . . . . 10

5 Encrypted File. . . . . . . .. . . . . . . . . . . . . . . . . . . . . . .. . 10

6 Pert Chart . . . . . . . . . . . . . . . . . . . . . . . . . . .. . . . . . 11

7 Flow Chart . . . . . . . . . . . . . . . . . . . . . . . . . . .. . . . . 11

8 DFD 0 . . . . . . . . . . . . . . . . . . . . . . . . . . .. . . . . . . 12

9 DFD 1 . . . . . . . . . . . . . . . . . . . . . . . . . . .. . . . . . . 12

1 Introduction

To begin with, the Text Editor we’ve made has all the fundamental features like cut, copy, paste etc. Our text editor has various other attributes besides being a text editor. The Basic text Editor like that of Linux doesn’t have spell Check facilities. But our text editor helps the user with spell check as well. Moreover, the user can share files among two or more networks via an in-built file sharing option. This is done with the help of Socket Programming. Thus, the text editor allows easier communication between users over a single network. Security in File Sharing has also been kept in Context. Any data that is shared via our Text Editor will be encrypted and could only be decrypted on the receiver side.

2 Literature Review

Most word processors can read and write files in plain text format, allowing them to open files saved from text editors. Saving these files from a word processor requires ensuring the file is written in plain text format, and that any text encoding won't obscure the file for its intended use. However, very few text editors are there that support encrypted data and documents sharing as well. This paper describes an approach to the idea of identifying the most important facts regarding Text editor with basic attributes and the integrated File Sharing options in it. Thus, it can be utilized in any of the institutional field or commercial field

3 Problem Statement

Text Editors too had all the basic functionalities of a text editor, but they lacked a secured file sharing attribute with encrypted data.

4 Objective

The final goal of this project is to make a user-friendly text editor with fundamental features of cut, copy, paste etc. complemented with a file sharing option under socket programming and encryption-decryption techniques.

5 Design Methodology

a) Specifying the software and various components of the architecture

-GUI for text editor-socket programming

b) Specifying the bindings between the tasks and the resources, some manually and some by the design tools (open source libraries)

c) Specifying the port interconnections between the resources.

d) Analysis: extracting the data required for analysis and doing the analysis

6 Implementation

A flow chart for design and algorithms to Implements bindings between the tasks and the resources some manually and some by the design tools (open source libraries) with testing and review being done on weekly basis.

6.1 Pseudocode

The text editor is implemented such that it fulfills the all functionality of default linux text editor as well as we can share the text file from server to client and vice-versa with the help of socket programing.

Pseudocode for implementing texteditor and socket programing ----->

Some headerfiles are used in code------>

#include<gtk/gtk.h>

#include<stdio.h>

#include <stdlib.h>

#include <errno.h>

#include <string.h>

#include <unistd.h>

#include <sys/types.h>

#include <sys/socket.h>

#include <netinet/in.h>

#include <netdb.h>

#include<ctype.h>

Step1.To create a top-level window of texteditor we have used gtk\_window\_new() [2] function.

Step2.To create verticle and horizontal container for widgets we have used gtk\_vbox\_new() and gtk\_hbox\_new() [2].

Step3.To create keyboard shortcuts we have used accel group and [ ] GtkItemFactoryEntry[3] for creating menu which contains File, Edit , Search, Server and Client icons.

Step4. For selecting file with the use of file path we've used [ ]gtk\_file\_selection\_new()[4] for creating file choosing dialog

Step5.File should be saved in [ ]utf-8 format[5]

Step4. For socket programing and file transfering we've created some user define functions

void servert(void) (connecting server to network)

void client(void)(connecting client to network)

void recieves(void)(Receive file at the server end)

6.2 Output Screen

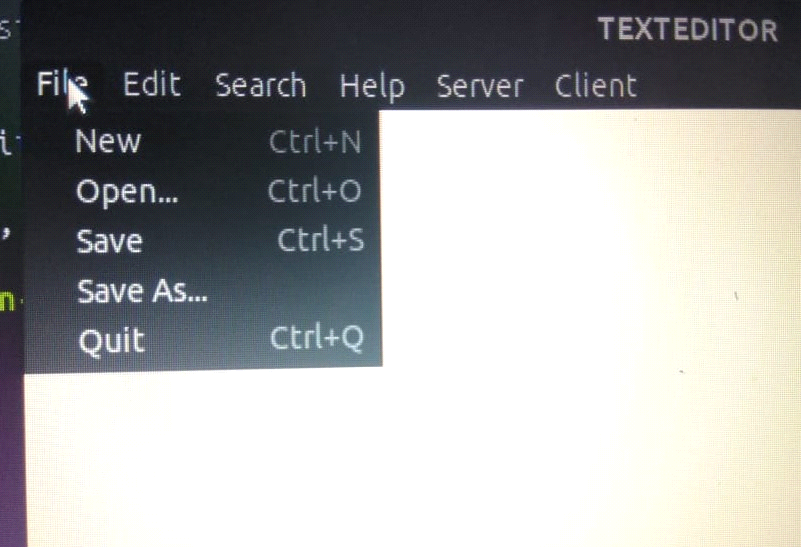


Fig1.Text Editor

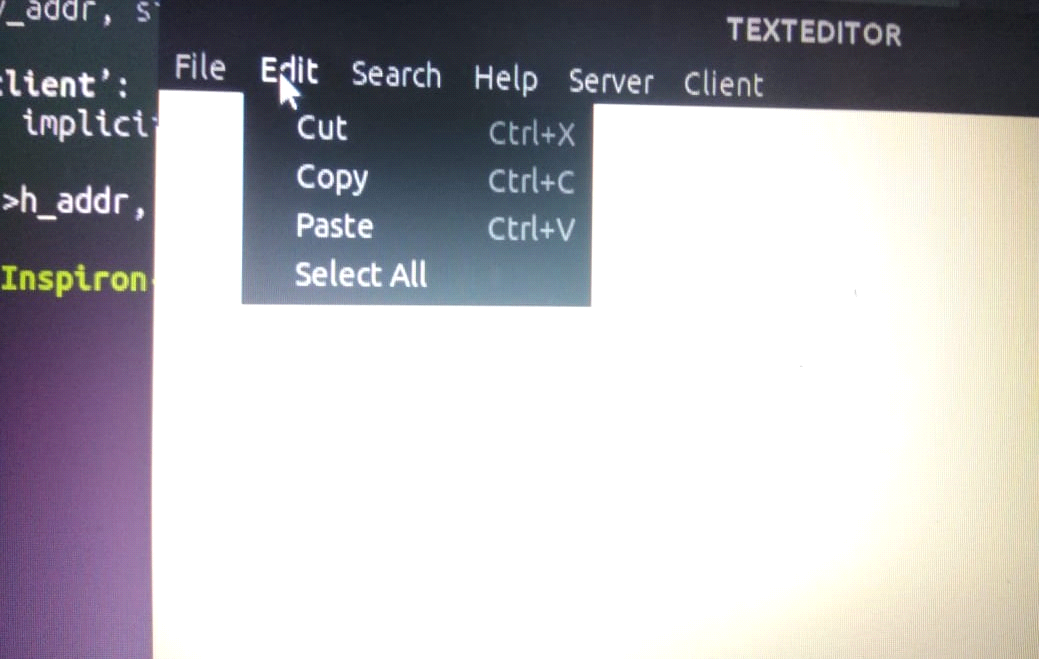


Fig2.Attributes of the Text Editor

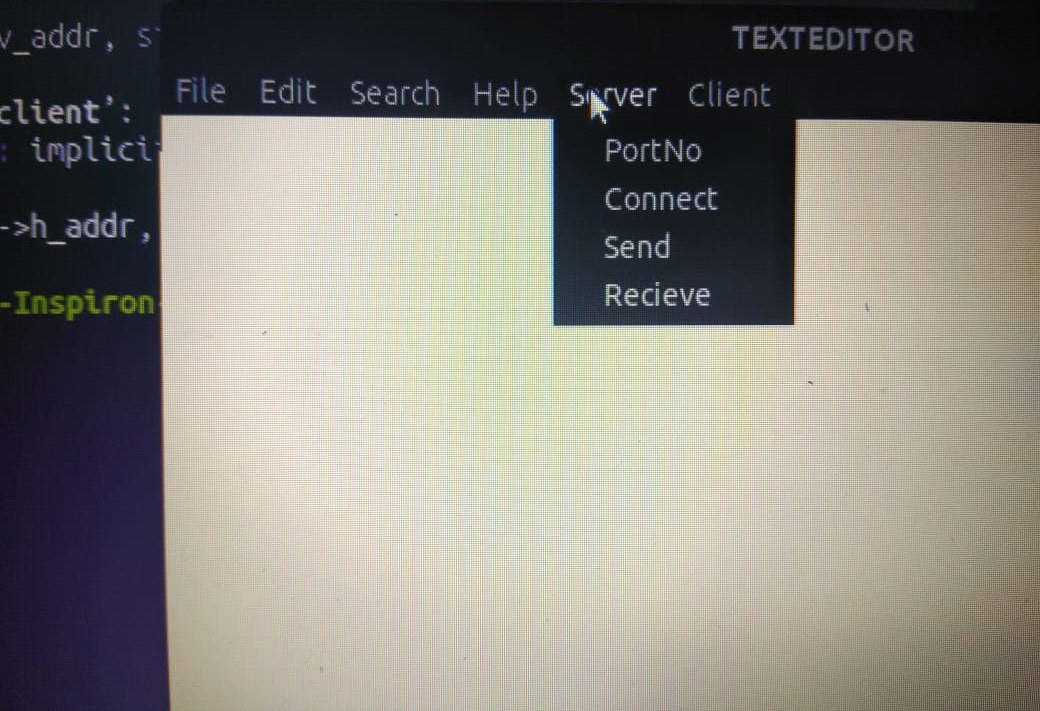


Fig3.File Sharing



Fig4.Encryption

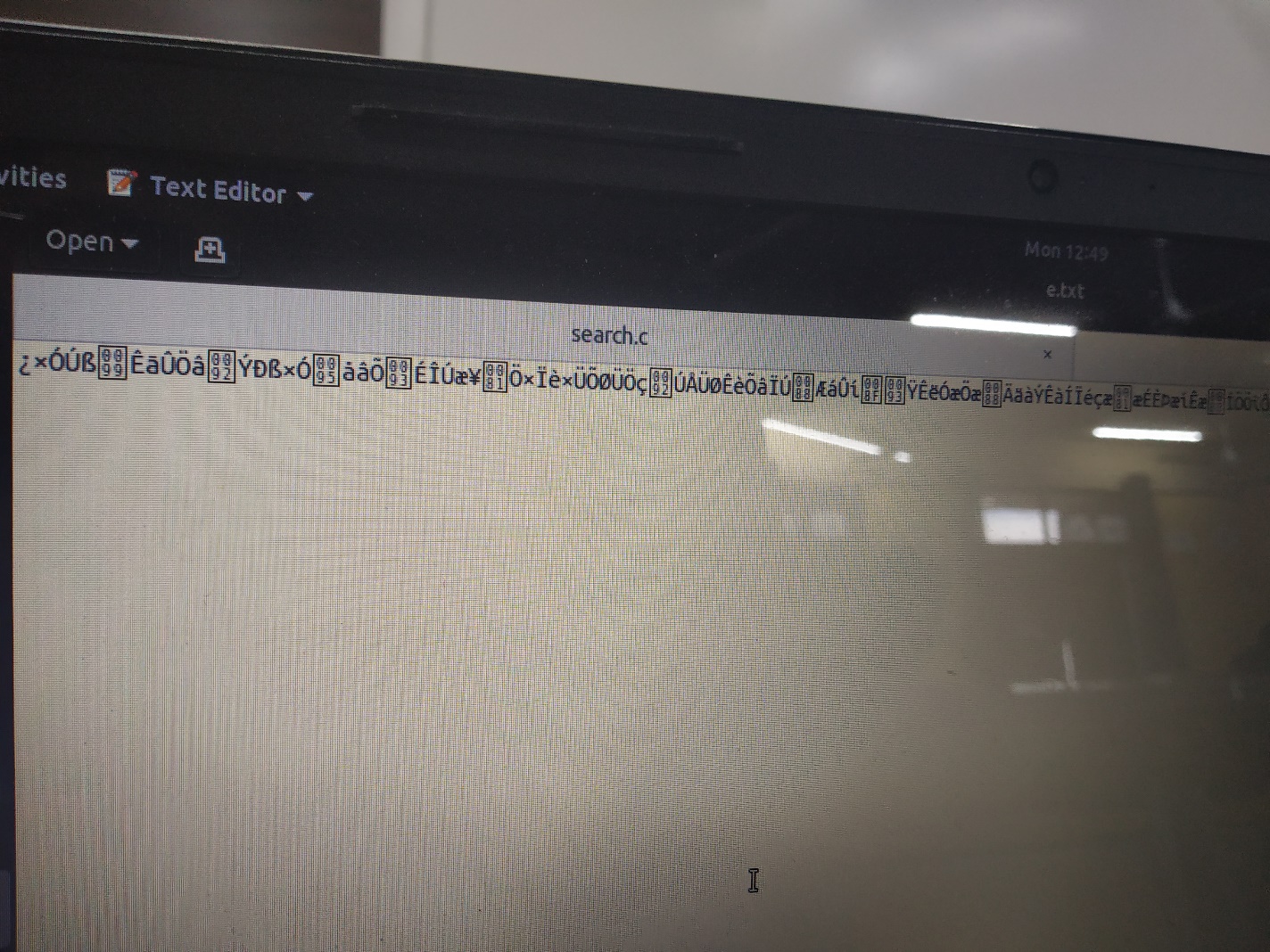


Fig5.Encrypted File

7 Schedule

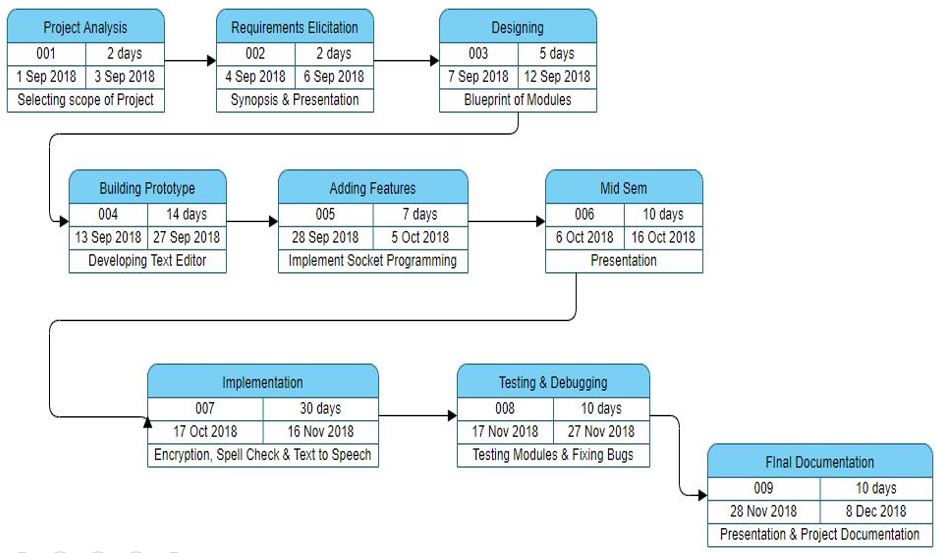


Fig6.Pert Chart

7.1 Flow-Chart

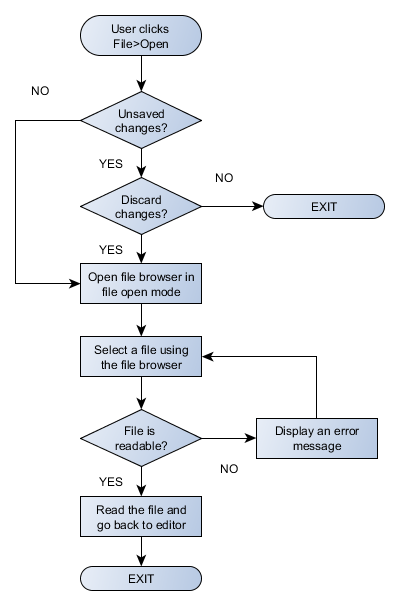


Fig7.Flow Chart

7.2 Data Flow Diagram

Level 0:

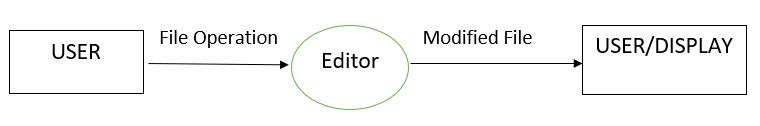


Fig8.DFD 0

Level 1:

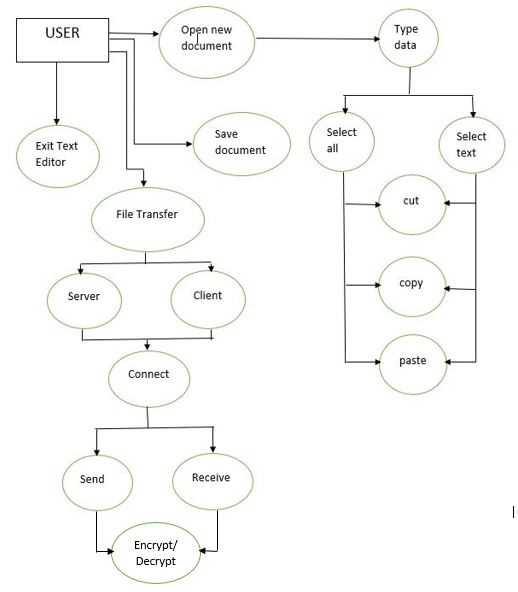


Fig9.DFD 1

8 Result Analysis

The Text Editor built could Edit Text, Copy and Paste the written text. The written text could be saved as a document. Once a text file is saved, it could be shared within a network with the help of the network ids of the receiver. The shared file is encrypted with a key to make the file secure and to avoid data breach.

9 Conclusions and Future Scope

It is an Open Source software thus welcoming the suggestions and the contributions of it users. Thus, several of its attributes could be polished to make better software that could be used in an OS as inbuilt software as well. Spell suggestion could be integrated with spell check as well that would help the users even more. File sharing in a text editor is rare combination. Therefore, the software could be of a great commercial usage in future.

References

[1] Team "GTK+ Documentation" , Gtk.org .[Online]. Avaliable:https://www.gtk.org/documentation.php .[Accessed October.10,2018].

[2] J Bodnar "GTK+ programming tutorial" , Zetcode.com .[Online]. Avaliable:http://zetcode.com/gui/gtk2/ .[Accessed October.10,2018].

[3] "GtkItemFactory: GTK+ 2 Reference Manual" , Developer.gnome.org. [Online].Available:https://developer.gnome.org/gtk2/stable/GtkItemFactory.html#GtkItemFactoryEntry.[Accessed October.10,2018].

[4] "GtkFileSelection: GTK+ 2 Reference Manual" , Developer.gnome.org.[Online]. Available:https://developer.gnome.org/gtk2/stable/GtkFileSelection.html#gtk-file-selection-new.[Accessed October.10,2018].

[5] "UTF-8 Encoding" , Fileformat.info .[Online]. Avaliable:https://www.fileformat.info/info/unicode/utf8.htm.[Accessed October.10,2018].

[6] "Socket" , Pubs.opengroup.org .[Online].

Avaliable:http://pubs.opengroup.org/onlinepubs/009695399/functions/socket.html.

A APPENDIX I PROJECT CODE

File Transfer Server-Client

void servert(void)

{

socklen\_t clilen;

struct sockaddr\_in serv\_addr, cli\_addr;

sockfds = socket(AF\_INET, SOCK\_STREAM, 0);

if (sockfds < 0)

error("ERROR opening socket");

bzero((char \*) &serv\_addr, sizeof(serv\_addr));

serv\_addr.sin\_family = AF\_INET;

serv\_addr.sin\_addr.s\_addr = INADDR\_ANY;

serv\_addr.sin\_port = htons(portno);

if (bind(sockfds, (struct sockaddr \*) &serv\_addr,

sizeof(serv\_addr)) < 0)

error("ERROR on binding");

listen(sockfds,5);

clilen = sizeof(cli\_addr);

sockfds = accept(sockfds, (struct sockaddr \*) &cli\_addr, &clilen);

if(sockfds < 0)

error("ERROR on accept");

}

void client(void)

{

struct sockaddr\_in serv\_addr;

sockfdc = socket(AF\_INET, SOCK\_STREAM, 0);

if (sockfdc < 0)

error("ERROR opening socket");

if (server == NULL) {

fprintf(stderr,"ERROR, no such host\n");

exit(0);

}

bzero((char \*) &serv\_addr, sizeof(serv\_addr));

serv\_addr.sin\_family = AF\_INET;

bcopy((char \*)server->h\_addr,

(char \*)&serv\_addr.sin\_addr.s\_addr,

server->h\_length);

serv\_addr.sin\_port = htons(portno);

if (connect(sockfdc,(struct sockaddr \*) &serv\_addr,sizeof(serv\_addr)) < 0)

error("ERROR connecting");

}

void sends(void)

{

char \*fname;

char buffer[1024];

bzero(buffer,1024);

FILE \*f;

int ch =0 ;

int lines = 0;

char c;

GtkWidget \*dialog = gtk\_file\_selection\_new("Share File...");

int resp = gtk\_dialog\_run(GTK\_DIALOG(dialog));

if(resp == GTK\_RESPONSE\_OK)

{

fname = g\_strdup(gtk\_file\_selection\_get\_filename(GTK\_FILE\_SELECTION(dialog)));

gtk\_widget\_destroy(dialog);

}

else

{

gtk\_widget\_destroy(dialog);

}

f=fopen(fname,"r");

while((c=getc(f))!=EOF)

{

if(c == '\n')

lines++;

}

bzero(buffer,1024);

write(sockfds, &lines, sizeof(int));

rewind(f);

while(ch != lines+1)

{

fscanf(f ,"%[^\n]%\*c", buffer);

write(sockfds,buffer,1024);

bzero(buffer,1024);

ch++;

}

fclose(f);

close(sockfds);

printf("The file was sent successfully");

}

void recievec(void)

{

char buffer[1024];

int lines = 0;

char \*fname;

FILE \*fp;

int ch = 0;

GtkWidget \*dialog = gtk\_file\_selection\_new("File Name...");

int resp = gtk\_dialog\_run(GTK\_DIALOG(dialog));

if(resp == GTK\_RESPONSE\_OK)

{

fname = g\_strdup(gtk\_file\_selection\_get\_filename(GTK\_FILE\_SELECTION(dialog)));

gtk\_widget\_destroy(dialog);

}

else

{

gtk\_widget\_destroy(dialog);

}

fp = fopen(fname,"a");

read(sockfdc, &lines, sizeof(int));

bzero(buffer,1024);

while(ch != lines+1)

{

read(sockfdc , buffer , 1024);

fprintf(fp , "%s\n" , buffer);

bzero(buffer,1024);

ch++;

}

fclose(fp);

close(sockfdc);

printf("The file was received successfully\n");

}

Encryption Decryption

void ec\_clicked ()

{

char \*p;

p = (gchar \*)gtk\_entry\_get\_text((GtkEntry \*)key\_entry);

key = (char \*)p;

keys = (char \*)p;

printf("%s", key);

printf("%s", keys);

char \*fname;

int a;

FILE \*f1,\*f2;

char c;

printf("%s", key);

GtkWidget \*dialog = gtk\_file\_selection\_new("Share File...");

int resp = gtk\_dialog\_run(GTK\_DIALOG(dialog));

if(resp == GTK\_RESPONSE\_OK)

{

fname = g\_strdup(gtk\_file\_selection\_get\_filename(GTK\_FILE\_SELECTION(dialog)));

gtk\_widget\_destroy(dialog);

}

else

{

gtk\_widget\_destroy(dialog);

}

f1 = fopen(fname, "r");

f2 = fopen("e.txt", "w");

c = fgetc(f1);

printf("%s", key);

while (c != EOF)

{

if(\*key == '\0')

key=keys;

a = \*key;

c = c + a;

fputc(c, f2);

key++;

c = fgetc(f1);

}

fclose(f1);

fclose(f2);

}

void encryption(void){

GtkWidget \*win;

GtkWidget \*vbox;

GtkWidget \*hbox;

GtkWidget \*key\_button;

win = gtk\_window\_new (GTK\_WINDOW\_TOPLEVEL);

gtk\_window\_set\_title(GTK\_WINDOW(win), "KEY");

vbox = gtk\_vbox\_new (FALSE, 2);

gtk\_container\_add (GTK\_CONTAINER (win), vbox);

hbox = gtk\_hbox\_new (FALSE, 2);

gtk\_box\_pack\_start (GTK\_BOX (vbox), hbox, FALSE, FALSE, 0);

key\_entry = gtk\_entry\_new();

gtk\_box\_pack\_start (GTK\_BOX (hbox),key\_entry, TRUE, TRUE, 0);

key\_button = gtk\_button\_new\_with\_label ("keyno");

gtk\_box\_pack\_start (GTK\_BOX (hbox), key\_button, FALSE, FALSE, 0);

g\_signal\_connect(G\_OBJECT (key\_button), "clicked",G\_CALLBACK (ec\_clicked), NULL);

gtk\_widget\_show\_all(win);

}

void dc\_clicked ()

{

char \*p;

p = (gchar \*)gtk\_entry\_get\_text((GtkEntry \*)key\_entry);

key = (char \*)p;

keys = (char \*)p;

printf("%s", key);

printf("%s", keys);

char \*fname;

int a;

FILE \*f1,\*f2;

char c;

printf("%s", key);

GtkWidget \*dialog = gtk\_file\_selection\_new("Share File...");

int resp = gtk\_dialog\_run(GTK\_DIALOG(dialog));

if(resp == GTK\_RESPONSE\_OK)

{

fname = g\_strdup(gtk\_file\_selection\_get\_filename(GTK\_FILE\_SELECTION(dialog)));

gtk\_widget\_destroy(dialog);

}

else

{

gtk\_widget\_destroy(dialog);

}

f1 = fopen(fname, "r");

f2 = fopen("d.txt", "w");

c = fgetc(f1);

printf("%s", key);

while (c != EOF)

{

if(\*key == '\0')

key=keys;

a = \*key;

c = c - a;

fputc(c, f2);

key++;

c = fgetc(f1);

}

fclose(f1);

fclose(f2);

}

void decryption(void){

GtkWidget \*win;

GtkWidget \*vbox;

GtkWidget \*hbox;

GtkWidget \*key\_button;

win = gtk\_window\_new (GTK\_WINDOW\_TOPLEVEL);

gtk\_window\_set\_title(GTK\_WINDOW(win), "KEY");

vbox = gtk\_vbox\_new (FALSE, 2);

gtk\_container\_add (GTK\_CONTAINER (win), vbox);

hbox = gtk\_hbox\_new (FALSE, 2);

gtk\_box\_pack\_start (GTK\_BOX (vbox), hbox, FALSE, FALSE, 0);

key\_entry = gtk\_entry\_new();

gtk\_box\_pack\_start (GTK\_BOX (hbox),key\_entry, TRUE, TRUE, 0);

key\_button = gtk\_button\_new\_with\_label ("keyno");

gtk\_box\_pack\_start (GTK\_BOX (hbox), key\_button, FALSE, FALSE, 0);

g\_signal\_connect(G\_OBJECT (key\_button), "clicked",G\_CALLBACK (dc\_clicked), NULL);

gtk\_widget\_show\_all(win);

}

File Saving

void load\_file(char \*fname)

{

FILE \*f;

GtkTextIter p;

char fbuf[CHAR\_BUF];

size\_t l;

if(fname == NULL)

{

GtkWidget \*dialog = gtk\_file\_selection\_new("Open File...");

int resp = gtk\_dialog\_run(GTK\_DIALOG(dialog));

if(resp == GTK\_RESPONSE\_OK)

{

fname = g\_strdup(gtk\_file\_selection\_get\_filename(GTK\_FILE\_SELECTION(dialog)));

gtk\_widget\_destroy(dialog);

}

else

{

gtk\_widget\_destroy(dialog);

return;

}

}

f = fopen(fname, "r");

if(fname != filename)

{

g\_free(filename);

filename = fname;

gtk\_window\_set\_title(GTK\_WINDOW(window), filename);

}

gtk\_text\_buffer\_get\_end\_iter(buf, &p);

while((l = fread(fbuf, 1, sizeof(fbuf), f)) > 0)

{

gsize br, bw;

gchar \*text;

text = g\_locale\_to\_utf8(fbuf, l, &br, &bw, NULL);

gtk\_text\_buffer\_insert(buf, &p, text, bw);

g\_free(text);

}

}

gboolean save\_file(char \*fname)

{

FILE \*f;

int ok = TRUE;

if(fname == NULL)

{

GtkWidget \*dialog = gtk\_file\_selection\_new("Save File As...");

int resp = gtk\_dialog\_run(GTK\_DIALOG(dialog));

if(resp == GTK\_RESPONSE\_OK){

fname = g\_strdup(gtk\_file\_selection\_get\_filename(GTK\_FILE\_SELECTION(dialog)));

gtk\_widget\_destroy(dialog);

}

else

{

gtk\_widget\_destroy(dialog);

return FALSE;

}

}

f = fopen(fname, "w");

GtkTextIter start, end, p;

gtk\_text\_buffer\_get\_bounds(GTK\_TEXT\_BUFFER(buf), &start, &end);

p = start;

while(!gtk\_text\_iter\_equal(&start, &end)) {

gchar \*buf, \*fbuf;

gsize br, bw;

gtk\_text\_iter\_forward\_chars(&p, CHAR\_BUF);

buf = gtk\_text\_iter\_get\_slice(&start, &p);

fbuf = g\_locale\_from\_utf8(buf, -1, &br, &bw, NULL);

g\_free(buf);

fwrite(fbuf, bw, 1, f);

g\_free(fbuf);

start = p;

}

if(ok)

{

gtk\_text\_buffer\_set\_modified(buf, FALSE);

if(fname != filename) {

g\_free(filename);

filename = fname;

gtk\_window\_set\_title(GTK\_WINDOW(window), filename);

}

}

return ok;

}