OS Project Code

#include<stdio.h>

#include<stdlib.h>

#include<sys/types.h>

#include<sys/stat.h>

#include<fcntl.h>

#include<string.h>

#include<unistd.h>

#include<time.h>

void clear\_str();

void clear\_input();

void create\_file();

void open\_txt\_file();

void open\_file();

void close\_file();

void delete\_file();

void write\_file();

void read\_file();

void append\_file();

void Read\_seek\_file();

void Write\_seek\_file();

void rename\_file();

void select\_file();

void pause1();

void get\_file\_attribute(struct stat stats);

void set\_file\_attribute();

char FILENAME[20];

int n\_line= 0;

int fd;

int main()

{

    int x;

    clear\_str(FILENAME);

    while(1)

    {

    system("clear");

    printf("\t\t\tVirtual File System\n\n");

    printf("\t\t\tCurrent File:%s\n\n",FILENAME);

    printf("0:Select a file\n");

    printf("1:Create a file\n");

    printf("2:Open a txt file\n");

    printf("3:Delete a file\n");

    printf("4:Read a file\n");

    printf("5:Write a file\n");

    printf("6:Append a file\n");

    printf("7:Seek to read a file\n");

    printf("8:Seek to write a file\n");

    printf("9:Get file attribute\n");

    printf("10:Set file attribute\n");

    printf("11:Rename a file\n");

    printf("12:Exit\n\n\n");

    printf("Please select your option:");

    scanf("%d",&x);

    if(x==0)

    {

    select\_file();

    pause1();

    }

    else if(x==1)

    {

    create\_file();

    pause1();

    }

    else if(x==2)

    {

    open\_txt\_file();

    pause1();

    }

    else if(x==3)

    {

    delete\_file();

    pause1();

    }

    else if(x==4)

    {

    read\_file();

    pause1();

    }

    else if(x==5)

    {

    write\_file();

    pause1();

    }

    else if(x==6)

    {

    append\_file();

    pause1();

    }

    else if(x==7)

    {

    Read\_seek\_file();

    pause1();

    }

    else if(x==8)

    {

    Write\_seek\_file();

    pause1();

    }

    else if(x==9)

    {

    struct stat stats;

    if (stat(FILENAME, &stats) == 0)

    {

      get\_file\_attribute(stats);

    }

    else

    {

      printf("Unable to get file properties.\n");

      printf("Please check whether '%s' file exists.\n",FILENAME);

    }

    pause1();

    }

    else if(x==10)

    {

    set\_file\_attribute();

    pause1();

    }

    else if(x==11)

    {

    rename\_file();

    pause1();

    }

    else if(x==12)

    {

    exit(1);

    pause1();

    }

    }

}

void select\_file()

{

    system("ls");

    printf("Enter the file name you want to select:");

    scanf("%s",FILENAME);

    char file\_buffer[1024];

    char \*s;

    s=file\_buffer;

    clear\_str(s);

    fd=open(FILENAME,O\_RDONLY,S\_IRUSR);

    if(fd!=-1)

    {

    read(fd,s,sizeof(file\_buffer));

    int count=0;

    int i=0;

    while(s[i]!='\0')

    {

    if(s[i]=='\n')

    {

    count++;

    }

    i++;

    }

    n\_line=count;

    }

    else

    {

    printf("Error Selecting the file\n");

    clear\_str(FILENAME);

    }

    close(fd);

}

void pause1()

{

    int ch;

    printf("\n\n\nPress Enter to Continue:");

    fflush(stdin);

    clear\_input();

    ch=getchar();

}

void create\_file()

{

    int fd;

    char str[20];

    char \*s=str;

    printf("Enter the name of file you want to create:");

    clear\_str(s);

    scanf("%s",str);

    fd=creat(s,S\_IREAD|S\_IWRITE|S\_IEXEC);

    if(fd==-1)

    {

    perror("Error Creating file\n");

    }

    else

    {

    printf("%s FILE created successfully!\n",s);

    }

}

void clear\_input()

{

    while(getchar()!='\n');

}

void clear\_str(char \*ch)

{

    int i=0;

    while(ch[i]!='\0')

    {

    ch[i]='\0';

    i++;

    }

}

void write\_file()

{

    int fd;

    char buffer[28];

    char \*s;

    s=buffer;

    clear\_str(s);

    clear\_input();

    printf("Enter the data:");

    scanf("%[^\n]%\*c",buffer);

    strcat(s,"\n");

    fflush(stdin);

    fd=open(FILENAME,O\_WRONLY|O\_TRUNC,S\_IWUSR);

    if(fd!=-1)

    {

    write(fd,buffer,strlen(s));

    n\_line=1;

    printf("\n\nData Written Successfully!\n");

    }

    else

    {

    printf("Error opening file\n");

    }

    close(fd);

}

void read\_file()

{

    int fd;

    char buffer[28\*n\_line];

    char \*s;

    s=buffer;

    clear\_str(s);

    fd=open(FILENAME,O\_RDONLY,S\_IRUSR);

    if(fd!=-1)

    {

    read(fd,s,28\*n\_line);

    printf("\n\nData Read Successfully!\n");

    }

    else

    {

    printf("Error opening file\n");

    }

    int i=0;

    int count=0;

    while(buffer[i]!='\0')

    {

    int flg=0;

    if(count==n\_line)

    {

    flg=1;

    }

    if(flg==1)

    {

    break;

    }

    if(buffer[i]=='\n')

    {

    count++;

    }

    printf("%c",buffer[i]);

    i++;

    }

    close(fd);

}

void append\_file()

{

    int fd;

    char buffer[28];

    char \*s;

    s=buffer;

    clear\_str(s);

    printf("Enter the data:");

    clear\_input();

    scanf("%[^\n]%\*c",buffer);

    strcat(s,"\n");

    fflush(stdin);

    fd=open(FILENAME,O\_WRONLY|O\_APPEND,S\_IWUSR);

    if(fd!=-1)

    {

    write(fd,buffer,strlen(s));

    printf("\n\nData Appended Successfully!\n");

n\_line++;

    }

    else

    {

    printf("Error opening file\n");

    }

    close(fd);

}

void rename\_file()

{

    char new[20];

    char \*s=new;

    printf("Enter new file name:");

    scanf("%s",new);

    rename(FILENAME,s);

    clear\_str(FILENAME);

    strcpy(FILENAME,s);

    printf("\n\nFIlE RENAMED Successfully!\n");

}

void Read\_seek\_file()

{

    int n;

    printf("Enter the line number, you want to read:");

    scanf("%d",&n);

    if(n>n\_line)

    {

    perror("Line Number greater than number of lines!\n");

    }

    else

    {

    printf("\n\nSEEKING to read LINE NUMBER %d\n",n);

    int fd;

    fd=open(FILENAME,O\_RDONLY,S\_IRUSR);

    char buffer[28\*n];

    char \*s;

    s=buffer;

    clear\_str(s);

    if(fd!=-1)

    {

    read(fd,s,28\*n);

    }

    else

    {

    printf("Error opening file\n");

    }

    int count=1;

    int count2=0;

    int i=0;

    while(buffer[i]!='\0')

    {

    int flg=0;

    if(count==n)

    {

    int temp=i;      while(buffer[temp]!='\n')

    {

    count2++;

    temp++;

    }

    flg=1;

    }

    if(flg==1)

    {

    break;

    }

    if(buffer[i]=='\n')

    {

    count++;

    }

    i++;

    }

    char buffer2[28];

    lseek(fd,i,SEEK\_SET);

    read(fd,buffer2,count2+1);

    write(1,buffer2,count2+1);

    close(fd);

    }

}

void Write\_seek\_file()

{

    int n;

    printf("Enter the line number, you want to write:");

    scanf("%d",&n);

    if(n>n\_line)

    {

    perror("Line Number greater than number of lines!\n");

    }

    else

    {

    printf("\n\nSEEKING to write LINE NUMBER %d\n",n);

    int fd;

    fd=open(FILENAME,O\_RDWR,S\_IRUSR|S\_IWUSR|S\_IXUSR);

    char buffer[28\*n];

    char \*s;

    s=buffer;

    clear\_str(s);

    if(fd!=-1)

    {

    read(fd,s,28\*n);

    }

    else

    {

    printf("Error opening file\n");

    }

    int count=1;

    int count2=0;

    int i=0;

    while(buffer[i]!='\0')

    {

    int flg=0;

    if(count==n)

    {

    flg=1;

    }

    if(flg==1)

    {

    break;

    }

    if(buffer[i]=='\n')

    {

    count++;

    }

    i++;

    }

    char buffer2[28];

    char \*s2=buffer2;

    clear\_str(s2);

    lseek(fd,i,SEEK\_SET);

    clear\_input();

    printf("Enter the data:");

    scanf("%[^\n]%\*c",buffer2);

    strcat(s2,"\n");

    write(fd,buffer2,strlen(s2));

    fflush(stdin);

    close(fd);

    }

}

void get\_file\_attribute(struct stat stats)

{

struct tm dt;

printf("\nFile access: ");

if (stats.st\_mode & S\_IRUSR)

    {

    printf("read ");

    }

if (stats.st\_mode & S\_IWUSR)

    {

    printf("write ");

    }

if (stats.st\_mode & S\_IXUSR)

    {

    printf("execute");

    }

printf("\nFile size: %ld", stats.st\_size);

dt = \*(gmtime(&stats.st\_ctime));

printf("\nCreated on: %d-%d-%d %d:%d:%d", dt.tm\_mday, dt.tm\_mon, dt.tm\_year + 1900,

                                           dt.tm\_hour, dt.tm\_min, dt.tm\_sec);

dt = \*(gmtime(&stats.st\_mtime));

printf("\nModified on: %d-%d-%d %d:%d:%d", dt.tm\_mday, dt.tm\_mon, dt.tm\_year + 1900,

                                           dt.tm\_hour, dt.tm\_min, dt.tm\_sec);

}

void set\_file\_attribute()

{

    int x;

    printf("Select the access you want to give to the file= %s (2=read 4=write 6=read&write 7=read&write&execute):",FILENAME);

    scanf("%d",&x);

    if(!chmod(FILENAME,S\_IRUSR) && x==2)

    {

    printf("Access given successfully\n");

    }

    else if(!chmod(FILENAME,S\_IWUSR) && x==4)

    {

    printf("Access given successfully\n");

    }

    else if(!chmod(FILENAME,S\_IRUSR|S\_IWUSR) && x==6)

    {

    printf("Access given successfully\n");

    }

    else if(!chmod(FILENAME,S\_IRUSR|S\_IWUSR|S\_IXUSR) && x==7)

    {

    printf("Access given successfully\n");

    }

    else

    {

    printf("Access not given\n");

    }

}

void open\_txt\_file()

{

    if(!fork())

    execlp("gedit", "gedit", FILENAME, NULL);

}

void delete\_file()

{

    int del=remove(FILENAME);

    if(del!=1)

    {

    printf("%s Deleted Successfully!\n",FILENAME);

    clear\_str(FILENAME);

    }

    else

    {

    printf("FIle Not Deleted \n");

    }

}