//Olubukola Akintoroye

#include <stdio.h>

#include <stdlib.h>

#include <openssl/rc4.h>

#include <sys/stat.h>

#include <fcntl.h>

#include <unistd.h>

#include <string.h>

#include "standalonerc42.h"

//#include <rc4\_skey.c>

//#include <rc4\_enc.c>

//Note to self: its bad practice to #include a source file :(

//#include <rc4\_locl.h>

#define H\_KEY "yufytugjhbgytcrtxc"

int encryp(int fd, int fd2){

RC4\_KEY key;

off\_t filesize; //will hold the size of the file

filesize = lseek(fd,0,SEEK\_END); //records the size of file in bytes

if(filesize < 0){ //checks for error

perror("Cannot provide offset");

exit(0);

}

lseek(fd,0,SEEK\_SET);// set byte counter back to first byte

unsigned char \*inbuff = (unsigned char\*)malloc(filesize); //input buffer

unsigned char \*outbuff = (unsigned char\*)malloc(filesize); //output buffer

if(read(fd,inbuff,filesize) == -1){ //checks if file is readable

perror("cannot read file");

exit(0);

}

//encryption / decryption

RC4\_set\_key(&key,sizeof(H\_KEY),(const unsigned char \*)H\_KEY);

RC4(&key,filesize,inbuff,outbuff);

//write encryption to new file

lseek(fd,0,SEEK\_SET);// set byte counter back to first byte

if(write(fd2,outbuff,filesize) == -1){

perror("cannot write to file");

exit(0);

}

free(inbuff);

free(outbuff); //free memory that was allocated for outbuff

return 0;

};

int main(int argc, char\* argv[]){

int input;

int output;

//check for the number of arguments

if(argc != 3) //8

{

perror("Incorrect number of arguments!");

// exit(0);

}

//open both files

input = open(argv[1],O\_RDONLY,0644);

if(input == -1){

perror("error opening the input file");

}

output = open(argv[2],O\_WRONLY|O\_APPEND|O\_CREAT,0644);

if(output == -1){

perror("error opening the output file");

}

//pass both files to encryption file

encryp(input,output);

//close both files

close(input);

close(output);

return 0;

};