

/\*

Group: B

Name: Brennan Schlittler

Email: brennan.schlittler@okstate.edu

Date: 10/10/22

Description:

Connects a client to the server (if the server is active) and processes input between the two

compile: gcc client.c -o client

execute: ./client

Tested on csx2

\*/

#include <arpa/inet.h>

#include <netinet/in.h>

#include <stdio.h>

#include <stdlib.h>

#include <string.h>

#include <sys/socket.h>

#include <sys/types.h>

#include <unistd.h>

#include <errno.h>

#define MAX 200

int main()

{

system("clear");

//Initialize variables

int clientSock;

```

struct sockaddr_in serverAddress;

char buffer[1024];

int port = 5150;

clientSock = socket(AF_INET, SOCK_STREAM, 0);

socklen_t addrSize;

//Open socket
if(clientSock < 0)
{
    perror("[-] Socket error");
    exit(1);
}

printf("[+] TCP server socket created \n");


//Configure memory address
memset(buffer, '\0', sizeof(buffer));


memset(&serverAddress, '\0', sizeof(serverAddress));

serverAddress.sin_family = AF_INET;

serverAddress.sin_port = htons(port);

serverAddress.sin_addr.s_addr = inet_addr("127.0.0.1");


connect(clientSock, (struct sockaddr*)&serverAddress, sizeof(serverAddress));

printf("Connected to the server \n");


//previously use #define MAX 200, in the #include section of the file
char no_of[50];

char data[MAX], option[MAX], opt[100][100];

int n,no_of_opts;

bzero(buffer, sizeof(buffer));

```

```

recv(clientSock, buffer, sizeof(buffer), 0);    //Reading the Options sent from client

printf("(Please enter corresponding integer value)\nPlease select an option:");

char * token = strtok(buffer, ",");    //Tokenizing the String based on the delimiter ","
int j=0;
while(token != NULL)
{
    printf("\n%d. %s",j+1, token); //Displaying the Options
    j++;
    token = strtok(NULL, ",");
}

char input[20],in[20],im[20];

printf(">> "); //Taking the User input for the option selected
scanf("%s", input);
strcpy(im,input);
send(clientSock, input, strlen(input), 0);

int impl = atoi(im);

recv(clientSock, buffer, 1024, 0);

system("clear");
token = strtok(buffer, ",");    //Tokenizing the String based on the delimiter ","
j = 0;
printf("Please select a column:");
while(token != NULL)
{

```

```
printf("\n%d. %s",j+1, token); //Displaying the Options  
j++;  
token = strtok(NULL, ",");  
}
```

```
input[0] = '\0';  
printf("\nPlease enter one of the options:\n>>"); //Taking the User input for the option selected  
scanf("%s", input);  
strcpy(in,input);  
send(clientSock, input, strlen(input), 0);  
int inp = atoi(in);  
int flag = 0;
```

```
if(impl == 1){  
    if(inp == 1){  
        flag =2;  
    }else if(inp == 2){  
        flag = 3;  
    }else{  
        flag = 5;  
    }  
}else{  
    if(inp == 1){  
        flag =3;  
    }else if(inp == 2){  
        flag = 6;  
    } else{  
        flag = 7;  
    }  
}
```

```
}
```

```
//wait for server to send ready message.
```

```
recv(clientSock, buffer, sizeof(buffer), 0);
```

```
char process_buff[4000];
```

```
while(1){
```

```
    //Write user options
```

```
    bzero(buffer, sizeof(buffer));
```

```
    input[0] = '\0';
```

```
    printf("Please select option:\n");
```

```
    printf("1. Display the records\n");
```

```
    printf("2. Save the records\n");
```

```
    printf("3. Display the summary\n");
```

```
    printf("4. Exit\n");
```

```
    printf(">> ");
```

```
    scanf("%s", input);
```

```
    printf("\n");
```

```
    //Send input to server
```

```
    send(clientSock, input, strlen(input), 0);
```

```
    //break if option 4
```

```
    if(atoi(input) == 4)
```

```
        break;
```

```
    //Wait for server response
```

```
    memset(process_buff, 0 , sizeof(process_buff));
```

```

bzero(process_buff, sizeof(process_buff));

recv(clientSock, process_buff, 4000, 0);

printf("%s\n", process_buff);


//Go into option 1 if chosen
    //input choice
char record[100];
printf("Please select a record: \n");
printf(">> ");
scanf(" %[^\\n]%"c", record);


    //send choice to server
send(clientSock, record, strlen(record), 0);


    //recieve summary and display it
char arr[200][200];
if( recv(clientSock, arr, sizeof(sizeof(char) * 200) * 200, 0) < 0){
    return 1;
}


    const char s[4] = ",";
    if(atoi(input) == 1){
        for(int i=0;i<200;i++){
            if(strlen(arr[i]) !=0){
                char* token = strtok(arr[i], s);

                int j=1;

                //printf("===> %d \n",flag);

                while(token != 0){
                    if(j== flag){

```

```

        token = strtok(0,s);
    }

    printf(" %s",token);
    token = strtok(0,s);
    j++;

}

printf("\n");
printf("\n");

}

}

}else if(atoi(input) == 2){
    FILE *fptr = fopen("out.txt","wb");
    for(int i=0;i<200;i++){
        if(strlen(arr[i]) !=0){
            char* token = strtok(arr[i], s);
            int j=1;
            //printf("==> %d \n",flag);
            while(token != 0){
                if(j== flag){
                    token = strtok(0,s);
                }
                fprintf(fptr, " %s", token);
                token = strtok(0,s);
                j++;
            }
            fprintf(fptr, "\n");

```

```
fprintf(fp, "\n");
```

```
}
```

```
}
```

```
}
```

```
}
```

```
printf("Exiting Server. Goodbye!\n");
```

```
return 0;
```

```
}
```