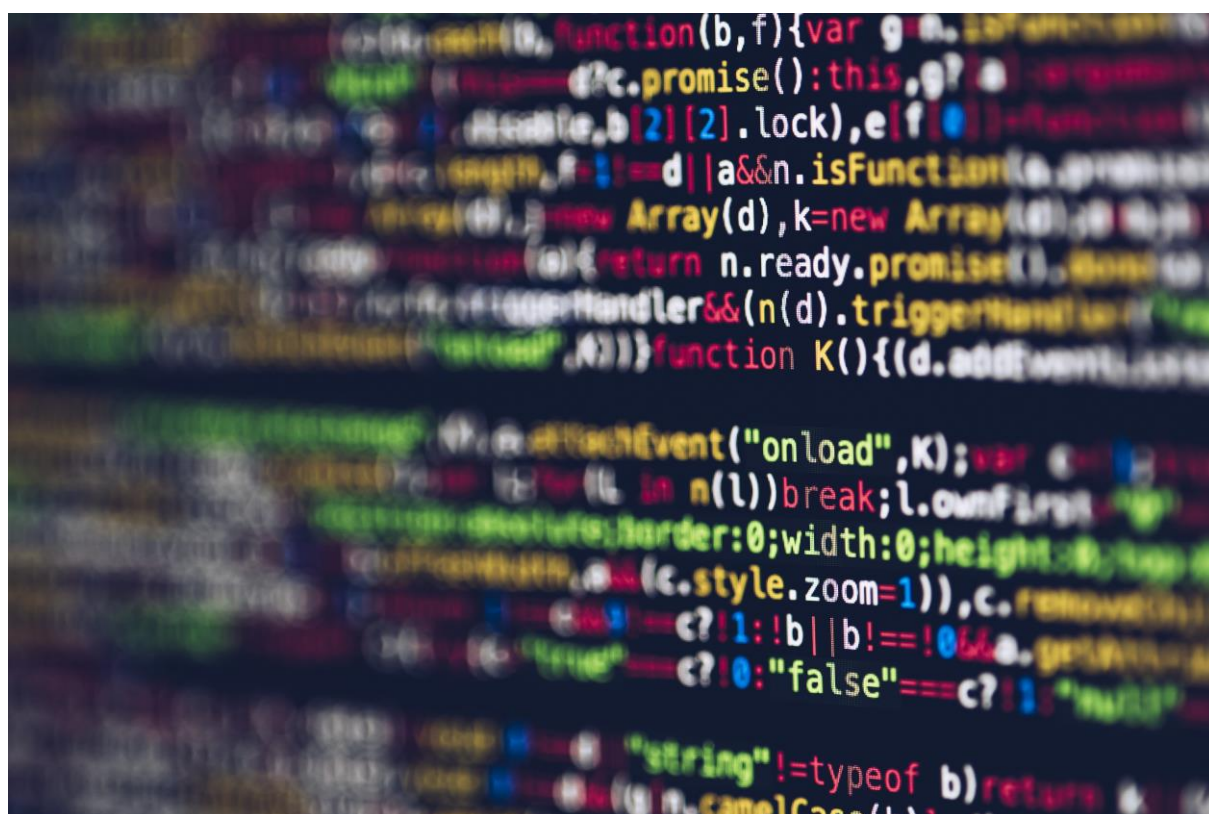


مستند سازی پروژه مبانی برنامه نویسی

سید یاسین موسوی

98110351

گروه 2 ، استاد فکوری



فهرست :

- توابع و روند client
 - کتابخانه ها
 - توابع connect,socket,send,recv
 - توابع:
 - acc
 - Chat
 - Remove_firstANDlast_char
 - Mystringstrtor
 - Mystringstrtor2
 - Part_number
 - lsequal
 - My_json_getobject
 - My_object_getItem
 - Arrayquot_number
 - Arraybruket_number
 - ArrayObjectQuote
 - ArrayObjectBrucket
 - ArrayIndexQuote
 - ArrayIndexBruket

- توابع و روند server
 - کتابخانه ها
 - توابع
 - توابع مشترک با client
 - تابع json_get_string

- Struct detail

توابع و روند client

کتابخانه ها

کتابخانه های روبرو برای نوشتن برنامه استفاده شده:

```
#include <stdio.h>
#include <time.h>
#include <stdbool.h>
#include <stdlib.h>
#include <string.h>
#include <conio.h>
#include <dir.h>
#include <process.h>
#include <winsock2.h>
```

توابع connect,socket,send,recv

به طور کلی از تابع های متنوع در قسمت client استفاده نشده و صرفا بوسیله یک تابع main و سپس تابع chat (که متعاقبا توضیح داده خواهد شد) و یک حلقه while بینهایت در chat برنامه نوشته شده است. به اینصورت که پس از ساخت سوکت برنامه تابع چت را صدا میزند و با یک حلقه بینهایت در آن دستور ها انجام میشوند.

تابع socket : برای ساخت سوکت به منظور اتصال به سرور استفاده میشود که به صورت زیر در کد برنامه نوشته شده:

```
server_socket = socket(AF_INET, SOCK_STREAM, 0);
```

تابع connect : برای اتصال سوکت ساخته شده به سوکت سرور به کار میرود:

```
connect(server_socket, (SA*)&servaddr, sizeof(servaddr));
```

تابع send : برای فرستادن پیام (char buffer) به سرور (server_socket) استفاده میشود:

```
send(server_socket, buffer, sizeof(buffer), 0);
```

تابع recv : برای دریافت پاسخ (char buffer) از سرور (server_socket) استفاده میشود:

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

تابع chat

تابع چت دارای یک ورودی سرور (server_socket) است و بدون خروجی است و صرفا دستوراتی را اجرا میکند:

```
void chat(int server_socket)
```

تابع چت شامل چند حلقه بینهایت تو در تو است که در هر دستور آن یک بار از دستورات connect,recv,send,socket استفاده میشود.

در حلقه اول دستورات منو ورود (login menu) ، در حلقه دوم دستورات منو کانال ها (channel menu) و در منو سوم دستورات send menu آورده شده که هر کدام از این حلقه ها در حلقه ماقبل خود قرار دارند. حال به نوبت به تعریف توابع مورد استفاده در برنامه میپردازیم:

- **Acc** : تابع برای accept کردن کلاینت توسط سرور:

```
void acc() {  
  
    int len = sizeof(client);  
    client_socket = accept(server_socket, (SA *)&client, &len);  
    if (client_socket < 0)  
    {  
        printf("Server acceptance failed...\n");  
        exit(0);  
    }  
    else  
        printf("+Server accepted the client..\n");  
}
```

- **Remove_firstANDlast_char** : یک رشته را ورودی میگیرد و کاراکتر اول و آخر آن را حذف میکند. خروجی ندارد.

```
void remove firstANDlast char(char s[]){
```

- **Mystringstrtor** : دو رشته ورودی میگیرد و برای جداسازی قطعات یک رشته به فرمت جیسون بکار میرود، به اینصورت که تا یافتن یک کاما و نقل قول (quote and comma) ادامه میدهد و رشته ماقبل آنها را در ورودی دوم خود ذخیره میکند و این رشته را از ابتدای ورودی اول حذف میکند.

```
void mystringStrtor(char main[], char s[])
```

- **Mystringstr2** : مشابه قبلی ولی در ابتدا براکت ها را از رشته مورد نظر (main) پاک میکند:

```
void mystringStrtore2(char main[], char s[])
```

- **Part_number** : تعداد « " , » ها را در تنها ورودی خود می شمارد و بصورت عدد خروجی میدهد:

```
int part number (char main[])
```

- **lseaqual** : برای بررسی برابری دو رشته ، یک رشته با کوتیشن و یک رشته بدون کوتیشن بکار میرود. دو ورودی دارد و خروجی آن یک عدد صحیح است: خروجی یک به معنای برابری و خروجی صفر به معنای نابرابری دو رشته است:

int isequal(char main[], char s[])

-
-
- **My_json_getobject** : یک رشته جیسون با دو مولفه میگیرد و با استفاده از وزودی که مولفه اول را جدا میدهد، مولفه دوم را از رشته کلی جدا میکند.

void my json getobject(char main[], char first[], char second[])

- **My_object_getItem** : یک جیسون بدون آرایه میگیرد و با ساتفاده از مولفه اول مولفه دوم را برمیگرداند.

void my objcet getItem(char main[], char first[], char second[])

- **Arrayquot_number** :سایز یک آرایه جیسون را بر اساس تعداد کوتیشن ها میدهد.

int ArrayQuote number(char main[])

- **Arraybruket_number** :سایز یک آرایه جیسون را بر اساس تعدا براکت ها میدهد.

int ArrayBruket number(char main[])

- **ArrayObjectQuote** : با استفاده از این تابع عنصر مورد نظر خود را در یک آرایه با استفاده از رتبه آن عنصر میان بقیه عناصر(رتبه از نظر جایگاهی) بدست میآوریم

void ArrayObjectQuote (char main[], char s[], int x)

- **ArrayObjectBrucket** : مشابه تابع بالا از نوع براکت.

void ArrayObjectBracket (char main[], char s[], int x)

- **ArrayIndexQuote** : جایگاه عضو بعدی و قبلی یک عضو آرایه را در نظر میگیریم و سپس از تابع بالایی(ArrayObjectQuote-Bruket) استفاده میکنیم.

void ArrayIndexQuote(char main[], int x, int * first, int * second)

- **etArrayIndexBruk** : مشابه بالا.

void ArrayIndexBracket(char main[], int x, int * first, int * second)

توابع و روند Server

کتابخانه ها:

```
#include <stdio.h>
#include <time.h>
#include <stdbool.h>
#include <stdlib.h>
#include <string.h>
#include <conio.h>
#include <dir.h>
#include <process.h>
#include <winsock2.h>
```

تابع json_get_string : یک جیسون مشابه جیسون های channel تحویل میگیرد و محتویات content آن را برمیگرداند.

```
void json_get_string(char main[], char s[]){
```

Struct details

استراکت اصلی سرور است که اطلاعاتی نظیر , username,channel name , token درون آن ذخیره میشود:

```
struct detail{
    char user[100],token[100],channel[100],channel_content[1000];
};
struct detail online_people[1000];
```