**به نام خدا**

گزارش پروژه مبانی برنامه سازی ترم پاییز 98-99 نام دانشجو: سهیل مهدیزاده

**مقدمه:** این پروژه به طور کلی مهارت دانشجویان را در کار با رشته و استرینگ می سنجد و بخش عمده ای از این پروژه را ارسال و دریافت همین رشته ها تشکیل می دهد.در ادامه نحوه کار کرد سرور و کلاینت و بعضی از توابع مهم آن را بررسی می کنیم.

**کلاینت:**

تابع Create\_Socket(): از آن جایی که سرور ما Stateless است یعنی پس از دریافت هر پیغام سوکت ارتباطی قطع می شود، این تابع کلیدی برای ساختن سوکت و خروجی دادن آن نیاز است.(خروجی این تابع int می باشد که همان سوکت ماست.) نحوه کارکرد آن هم به کتابخانه winsock2.h بر میگردد که با توابع socket() و Connect() به سرور وصل می شود و سوکت می سازد. توجه کنید که این کتابخانه استراکتی نیز در خود دارد که برای set کردن شماره پورت و IP به آن نیاز داریم.

از اینجا به یک سری توابع مشابه داریم که با توابع send و recv رکوعست را فرستاده و جواب را دریافت می کنند. برای مثال تابع زیر را در نظر بگیرید:

void Creating\_Channel(char User\_Auth\_Code[] ,char name[]){

char Creating\_Channel\_Request[2\*MAX];

int server\_socket = Creating\_Socket();

sprintf(Creating\_Channel\_Request,"create channel %s, %s\n",name,User\_Auth\_Code);

send(server\_socket, Creating\_Channel\_Request, sizeof(Creating\_Channel\_Request), 0);

memset(Creating\_Channel\_Request, 0,sizeof(Creating\_Channel\_Request));

recv(server\_socket, Creating\_Channel\_Request, sizeof(Creating\_Channel\_Request), 0);

cJSON \*My\_Answer = cJSON\_Parse(Creating\_Channel\_Request);

cJSON \*type = cJSON\_GetObjectItem(My\_Answer, "type");

if(strcmp("Error", type -> valuestring) == 0){

printf("Cannot Create the Channel try again...\n");

return;

}

else{

printf("Channel successfully created!\n");

Conversation\_Menu(User\_Auth\_Code);

return;

}

}

همانطور که مشاهده می کنید ورودی تابع نام کانال و توکن کاربر است.توکن کاربر در تابع Login() از سرور دریافت میشود و از آن به بعد چون در همه رکوعست ها به آن نیاز داریم، به عنوان ورودی آن را دریافت می کنیم. سپس توسط متود Sprintf() رشته رکوعست را می سازیم و به سرور می فرستیم. پاسخ سرور به صورت cJSON میباشد که برای پارس کردن آن به توابع cJSON\_Parse و cJSON\_GetObjectItem نیاز داریم. تابع اول رشته ی دریافتی را malloc کرده و در استراکتی به صورت رشته ذخیره میکند. چون JSON به صورت {”<type”:“Error”,“content”:“<error\_message“} است، این تابع ورودی اش یک تایپ است و خروجی اش محتوایات آن. مثلا در تابع بالا ما محتویات type و content را استخراج می کنیم و اگر type ، Error نباشد کاربر چنل را میسازد.

**سرور:**

در سرور دقیقا برعکس کاری که در کلاینت انجام میشود، عملی می شود. یعنی دریافت رکوعست و فرستادن پاسخ. برای مثال تابع Login() را در نظر بگیرید:

void Checking\_the\_Login\_Request(char Username[],char Password[]){

//Making the Path String!

char Directory[MAX];

sprintf(Directory,"./Users/%s.user.txt",Username);

FILE \*fptr;

fptr = fopen(Directory,"r");

if(fptr == NULL){

char Answer[MAX];

Creating\_cJSON\_String(Answer,"type","content","Error","No Such Username!");

Connect\_and\_Send(Answer,MAX);

fclose(fptr);

return;

}

else{

char String[MAX];

fscanf(fptr,"%[^EOF]",String);

//Parsing the cJSON!

cJSON \*cptr;

cptr = cJSON\_Parse(String);

cJSON \*username;

username = cJSON\_GetObjectItem(cptr, "username");

cJSON \*password = cJSON\_GetObjectItem(cptr, "password");

if(strcmp(Password,password -> valuestring)==0 && strcmp(Username, username -> valuestring)==0){

char string[32];

Auth\_Generator(string, 32);

strcpy(Users[User\_Counter].Authenticator\_Code,string);

char Answer[MAX];

Creating\_cJSON\_String(Answer,"type","content","AuthToken",Users[User\_Counter].Authenticator\_Code);

strcpy(Users[User\_Counter].name, Username);

Users[User\_Counter].Channel\_Flag = 0;

Users[User\_Counter].Online\_User\_Flag = 1;

User\_Counter++;

Connect\_and\_Send(Answer,MAX);

}

else{

char Answer[MAX];

Creating\_cJSON\_String(Answer,"type","content","Error","Wrong Password Or Username");

Connect\_and\_Send(Answer,MAX);

}

}

}

ابتدا Directory دیتا بیس کاربر ها ساخته میشود. سپس فایل .txt کاربر که نام آن بصورت <User\_Name>.user.txt است باز میشود. اگر null برگردانده شود بعنی کاربر وجود ندارد پس JSON خطا ساخته میشود و تابع تمام میشود. در غیر این صورت می رویم برای چک کردن UserName و Password. ورودی تابع Username و Password ورودی کاربر است. این دو با دیتا بیس ساخته شده چک می شوند جواب به صورت JSON برگردانده می شود. این JSON با توابع cJSON\_CreateObject() و cJSON\_AddItemToObject() ساخته میشود.