

به نام خدا

حمیدرضا دهباشی قربانعلی گروه ۳

پروژه مبانی برنامه نویسی (چت اپلیکیشن)

ابتدا نحوه اجرای برنامه را توضیح می دهیم سپس به شرح توابع و یک سری از متغیرهای گلوبال می پردازیم. با اجرا شدن برنامه، ابتدا پیام خوش آمدگویی به نمایش در می آید و سپس وارد منوی ورود می شویم که حامل ۴ گزینه است:

(۱) Log in: این گزینه جهت ورود به برنامه است و از ما نام کاربری و رمز عبور می خواهد.

(۲) Sign up: این گزینه جهت عضو شدن در چت اپلیکیشن است و از ما نام کاربری و رمز عبور می خواهد.

(۳) Search user: این گزینه جهت جستجو برای وجود یا عدم وجود کاربری مشخص در سیستم است.

(۴) Bye: برنامه تمام می شود.

پس از Log in وارد منوی کاربری می شویم که ۳ گزینه دارد:

(۱) Join channel: این گزینه جهت عضو شدن در کانال است و از ما نام آن کانال را می خواهد و اگر کانال خصوصی باشد، رمز عبور هم نیاز است.

(۲) Create channel: این گزینه جهت ساختن کانال است و از ما نوع آن (خصوصی یا عمومی) و نام آن را می خواهد.

(۳) Log out: این گزینه جهت خروج کاربر است. و وارد منوی قبلی می شود.

پس از عضو شدن یا ساختن کانال منوی کانال به اجرا در می آید که حاوی ۵ گزینه است:

(۱) Send message: این گزینه پیامی در گروه می فرستد.

۲) Refresh: این گزینه پیام های دیده نشده توسط کاربر را به نمایش در می آورد.

۳) Channel members: این گزینه اعضای آنلاین کانال را منتشر می کند.

۴) Search message: این گزینه به دنبال کلمه ای در میان پیام ها می گردد و آن ها را برمی گرداند.

۵) Leave: این گزینه جهت خارج شدن از کانال فعلی است. و وارد منوی قبلی می شود.

و حال به شرح توابع می پردازیم.

:Client

void EnterProgram(); //----->

int Menu(); //-----> Menus

int Chatting_Menu(); //----->

این ۳ تابع، توابعی هستند که ساختار منوها را شکل می دهند.

.....
int User_Pass(); //-----> Getting username and password

این تابع جهت دریافت نام کاربری و رمز عبور است.

.....
char* PrepareToSend(char order[]); //-----> This function create the messege that should be sent to server

در این تابع ورودی رشته ایست که با توجه به آن (مثلا Register) تابع پیامی را که قرار است به سرور فرستاده شود برمی گرداند. یعنی اگر Register باشد به قسمتی که مربوط به آن است می رود سپس اقدامات را انجام می دهد.

.....
int Members(char*); //-----> It is for showing members of channel

int Refresh(char*); //-----> It is for showing messeges of channel since last seen

در این توابع رشته ای به صورت جیسون دریافت می شود و از آن رشته قسمت content را به نمایش می - گذاریم.

.....
int Response(char*); //-----> This function sends messege to server and receive its answer

این تابع مسئول فرستادن پیام به سرور و دریافت آن است.

int Join_Check(char*); //-----> This function is used when user wants to join channel and it checks if there is problem or not

int Create_Check(char*); //-----> This function is used when user wants to create channel and it checks if there is problem or not

int Enter_Check(char*); //-----> This function is used when user wants to log in and it checks if there is problem or not

int Register_Check(char*); //-----> This function is used when user wants to register and it checks if there is problem or not

int SU_Check(char*); //-----> This function is used when user wants to search for a user and it checks if there is problem or not

int SM_Check(char*); //-----> This function is used when user wants to serach for a word in messeges and it checks if there is problem or not

در این توابع بررسی می شود که آیا **type** پیام دریافتی **error** است یا خیر. و اگر بود خطا و اگر نبود به ادامه برنامه می پردازد.

char* Tabdil(char order[1000] ,int n); // This function moves on its input([order] here) and return the result (example-----> inputs:"hassan",3, it returns "san")

int find_char(char s[1000],char ch); // This function finds the first place which character(ch) exists in the string(s)

char* GetO(char* object,char item[]); // It works like cJSON_GetObjectItem

char* GetA(char array[],int i); // It works like cJSON_GetArrayItem used for messeges

int GetAS(char []); // It works like cJSON_GetArraySize used for messeges

```
char* GetAM(char array[1000],int i); // It works like cJSON_GetArrayItem used for members
```

```
int GetASM(char []); // It works like cJSON_GetArraySize used for members
```

این توابع برای کار با جیسون هستند.

به دنبال کاراکترهای متمایز کننده مانند "{", "}" در **object** می‌گردد و با استفاده از این تغییرات لازم را بوجود می‌آورند. در اولی **item** را پیدا می‌کند و مقدار جلوی آن را برمی‌گرداند. و در دومی به اندازه **i** جلو می‌رود و آن را برمی‌گرداند. در سومی سایز یک آرایه را با استفاده از حساسیت نسبت به "{", "}" پیدا می‌کند. و چهارمی و پنجمی هم مانند دومی و سومی هستند با حساسیت‌ها متفاوت.

```
.....
```

```
int Daryaft1(); // Makes the Enter menu
```

```
int Daryaft2(); // Makes the after Enter menu
```

```
int Daryaft3(); // Makes the channel menu
```

```
int Daryaft4(); // Makes the public-private choose menu
```

این توابع ظاهر منوها را می‌سازند، به این ترتیب که ابتدا صفحه را پاک می‌کند، سپس چیزی را که باید نمایش می‌دهد.

```
.....
```

```
void delay(int); // It stops program for some time
```

```
.....
```

```
char User[1000],Pass[1000],AuthToken[1000], Ch_Pass[1000];
```

همانطور که از نام‌های این متغیرها برمی‌آید به ترتیب از چپ به راست جهت نگهداری نام کاربری، رمز عبور، توکن، رمز کانال هستند.

:Server

```
int PreProcess();  
int PrepareToGet();  
int GetOrder(int,int);  
int Process(int,int);  
int AnsToClient(int,int);
```

این توابع به ارتباط میان سرور و کلاینت مربوطند.

در preprocess صرفاً توابع دیگر صدا زده می‌شوند. و در preparetoget آماده سازی جهت دریافت پیام از کلاینت صورت می‌گیرد (bind,listen,...) و getorder پیام را دریافت می‌کند. و سپس وارد process می‌شود که با توجه به پیام تغییرات لازم روی آن صورت می‌گیرد. و پس از تمامی تغییرات رشته گلوبال answer را در تابع anstoclient می‌فرستد برای کلاینت. و ورودی های این توابع هم سوکت های کلاینت و سرور است.

```
.....  
int Regist();  
int Login();  
int Members();  
int Send();  
int Logout();  
int Leave();  
int Create();  
int Create_P();  
int Join();  
int Refresh();  
int Channel_Pass();
```

این توابع یکسری اعمال انجام می‌دهند تا **answer** را بسازند.

تابع **regist** بررسی می‌کند آیا فایلی به نام **user** وجود دارد یا نه و طبق آن عمل می‌کند ارور یا آن را می‌سازد. و تابع **login** هم بر همین اساس چک کردن فایل عمل می‌کند و اگر بود آرایه **useronline** را تغییر می‌دهد و به آن این **user** را اضافه می‌کند. و البته به آن هم توکنی خاص می‌دهد و در آرایه **autoken** می‌ریزد. تابع **logout** هم دقیقا عکس آن را انجام می‌دهد و از تمامی آن آرایه‌ها **user** را پاک می‌کند.

تابع **create** و **create_p** برای ساخت کانال است، اولی عمومی و دومی اختصاصی و اگر کانال در فایل موجود نبود، آن را می‌سازد. و **join** هم **user** پیشنهاد دهنده را به کانال متصل می‌کند و اگر کانال خصوصی بود پیامی را آماده می‌کند که پاسخ آن به تابع **channel_pass** می‌رود جهت چک کردن رمز وارد شده.

تابع **send** هم صرفا پیامی را به پیام‌های موجود در کانال اضافه می‌کند، با استفاده از رابطه میان توکن و آرایه **autoken** و اندیس مورد نظر کانال مشخص را باز کرده، آرایه **messeges** را باز می‌کند و به آن اضافه می‌کند.

تابع **refresh** و **members** جهت خواندن آرایه‌های **messeges** و **members** هر کانال است. با این تفاوت که در آرایه **refresh**، آرایه‌ای داریم با عنوان **refc** که تعداد پیام‌هایی که هر کاربر دیده را نشان می‌دهد و طبق آن می‌خواند.

در آخر هم تابع **leave** کاربر مشخصی را از مانل مشخصی حذف می‌کند و پیام حذف آن را هم اضافه می‌کند و در فایل می‌نویسد.

.....
`int ANS(char[],char[]);`

`int ANS2(char[],char[]);`

این دو تابع جهت ساختن **answer** است به این ترتیب که ورودی‌های **content** و **type** است با این تفاوت که **ans2** ورودی دومش به صورت جیسون است. ولی اولی هر دو را رشته می‌گیرد.

.....
`int Token();`

`int tok_eq(char[]);`

```
int tok_check();  
int tok_em(int);  
int us_check(char[]);  
int andis(char[]);
```

این تابع ها مربوط به **token** و **user** و ارتباطات میان آن هاست.

Token رشته ای رندوم میسازد و به **autoken** اضافه می کند و کاربر را هم همینطور.

Tok_eq بررسی می کند که آیا توکن در آرایه **autoken** موجود است و اگر آری **user** را مقداردهی می کند.

Tok_check بررسی می کند که آیا توکن ایجادشده تکراری است یا نه.

Tok_em این تابع **autoken** را می کاهد. به اینصورت که اندیس **i** ام که ورودی گرفتیم را حذف می کند.

Us_check() مانند **tok_check** است برای **user** ها.

Andis رشته ای به عنوان کاربر کنونی می گیرد و اندیس آن را در **user_online** برمیگرداند.

.....

```
int find_char(char s[10000],char ch);  
char* Tabdil2(char ord[10000] ,int n);
```

این دو تابع در کلاینت توضیح داده شده اند.

```
int Tabdil(int);
```

مانند **tabdil2** است فقط آرایه گلوبال **order** را تغییر می دهد.

.....

```
int GetASM(char array[10000]);  
char* GetAM(char array[10000],int i);  
char* GetA(char array[10000],int i);  
int GetAS(char array[10000]);
```



```
char* GetO(char* object,char item[10000]);
```

توابع بالا در کلاینت توضیح داده شده اند.

```
char* createS(char* str);
```

```
char* createA();
```

```
char* createO();
```

این توابع تغییرات مختصری ایجاد می کنند.

اولی دبل کوتیشن می گذارد. دومی [] و سومی {}

```
int AddIA(char* array, char* item);
```

```
int AddIO(char* object, char* item, char* value);
```

این دو دستور **item** هایی را به آرایه و آبجکت جیسون اضافه می کند. باز هم با استفاده از حساسیت نسبت به کاراکتر هایی خاص.

```
int DelIA(char[],int);
```

```
int DelIAM(char[],int);
```

این دو دستور مورد i ام را حذف میکنند از آرایه جیسون.

```
.....  
int Srch_M();
```

```
int Srch_U();
```

این دو دستور جهت جستجو در کانال و یا فایل است برای لغتی یا کاربری.

برای **srch_M** تک تک پیام های کانال مورد نظر را چک می کند و با استفاده از توابع پایین لغت را چک می کند.

در **srch_U** میبیند که آیا فایل با آن نام وجود دارد یا نه.

```
.....  
int EX(char*);
```

این تابع برای چک کردن این است که آیا کاربر های موجود در یک کانال آنلاین هستند یا خیر. و ورودی آن یک کاربر است و آن را با **useronline** بررسی می کند.

.....

```
int Poi(char*,char*);
```

این تابع لغتی را در عبارتی جستجو می کند اگر موجود بود ۱ و در غیر اینصورت ۰ را برمی گرداند.

.....

```
char canal[10000][10000]; //clients are joined
```

```
char kanal[10000][10000]; //clients want to join
```

این دو آرایه کانال های هر کاربر را مشخص می کند. اولی کانال هایی را که قطعی است و دومی کانال هایی که قطعی نیست و خصوصی هم هستند.

.....

```
char user_online[10000][10000];
```

```
char user[10000];
```

user کاربری است که باید با آن کار کنیم و **user_online** تمامی کاربران آنلاین را در خود دارد.

.....

```
char order[10000];
```

```
char Answer[10000];
```

این دو پیام دریافتی از کلاینت و ارسالی به آن را در خود ذخیره می کند.

.....

```
char autoken[10000][10000];
```

```
int tokCount=0;
```

```
int usCount=0;
```

autoken توکن های کاربران آنلاین را دارد و به وسیله آن کاربر را شناسایی می کند. و **tokcount** و **uscount** تعداد کاربران آنلاین و توکن های حال حاضر است.

.....
int refc[10000]={0};

این آرایه نشان می دهد که هر کاربر چه تعداد از پیام های کانال خود را مشاهده کرده است.

.....
char rub[10000];

char rub2[10000];

char rub3[10000];

char rub4[10000];

char rub5[10000];

char rub6[10000];

این ها آرایه هایی هستند برای استفاده در توابع