

۱. کلاینت

در تابع اصلی (main) یک رشته به طول MAX (ماکرو = ۱۰۲۴) در حافظه heap اختصاص داده می‌شود و در ادامه این رشته بین توابع پاس داده می‌شود و تغییرات لازم صورت می‌گیرد.

`void connect_disconnect (char* str)` 🚀

این تابع یک رشته دریافت می‌کند. پس از ساخت سوکت برای اتصال به سرور، رشته دریافتی برای سرور ارسال می‌شود و پاسخ سرور در همان رشته قرار می‌گیرد (که به علت malloc کردن در اختیار سایر توابع است). و سپس سوکت را قطع می‌کند.

`void user_menu (char* str)` 🚀

این تابع همان منوی کاربری است و ورودی آن رشته malloc شده در تابع main است. پس از نشان دادن گزینه‌های موجود یک عدد از کاربر می‌گیرد و با توجه به آن عملیات‌های مختلف را انجام می‌دهد.

اگر کاربر قصد ایجاد اکانت داشته باشد؛ پس از انتخاب گزینه ۱، نام و رمز خود را وارد می‌کند. اگر عملیات موفقیت‌آمیز باشد؛ کاربر ثبت‌نام می‌شود و به منوی کاربری باز می‌گردد تا گزینه ۲ را انتخاب کند. اگر هم مشکلی وجود داشته باشد؛ ارور برای کاربر فرستاده می‌شود و کاربر به منوی کاربری باز می‌گردد.

گزینه ۲ برای ورود به اکانت است که باز هم با وارد کردن نام و رمز انجام می‌شود. اگر عملیات موفقیت‌آمیز باشد؛ کاربر وارد می‌شود و به منوی اصلی می‌رود. اگر هم مشکلی وجود داشته باشد؛ ارور برای کاربر فرستاده می‌شود و کاربر به منوی اصلی باز می‌گردد. همچنین گزینه ۳ برای خروج از برنامه است.

`void main_menu (char* str, char* token)` 🚀

این تابع منوی اصلی است و ورودی آن رشته malloc شده در تابع main و توکن ارسال شده توسط سرور است. در اینجا نیز کاربر عددی را وارد می‌کند و عملیاتی انجام می‌شود.

اگر کاربر قصد ایجاد کانال داشته باشد؛ پس از انتخاب گزینه ۱، نام کانال را وارد می‌کند. اگر عملیات موفق باشد کاربر وارد کانال ایجاد شده می‌شود و منوی کانال را مشاهده می‌کند. اگر هم مشکلی ایجاد شود، ارور برای کاربر فرستاده می‌شود و کاربر به منوی اصلی برمی‌گردد.

گزینه ۲ برای جوین شدن در یک کانال ایجاد شده است. اگر عملیات موفقیت‌آمیز باشد؛ کاربر وارد کانال می‌شود و منوی کانال را مشاهده می‌کند. اگر هم مشکلی وجود داشته باشد؛ ارور برای کاربر فرستاده می‌شود و کاربر به منوی اصلی بازمی‌گردد.

گزینه ۳ برای خروج کاربر از منوی اصلی است؛ اگر عملیات موفقیت‌آمیز باشد؛ کاربر خارج می‌شود و به منوی کاربری می‌رود. اگر هم مشکلی وجود داشته باشد؛ ارور برای کاربر فرستاده می‌شود و کاربر به منوی اصلی بازمی‌گردد.

`void channel_menu (char* str, char* token)` 🚦

این تابع منوی کانال است و ورودی آن رشته malloc شده در تابع main و توکن ارسال شده توسط سرور است. در اینجا نیز کاربر عددی را وارد می‌کند و عملیاتی انجام می‌شود.

گزینه ۱ برای ارسال پیام است. اگر عملیات موفقیت‌آمیز باشد؛ پیام ثبت می‌شود و کاربر به منوی کانال برمی‌گردد؛ در غیر این صورت پیام ارور نشان داده می‌شود و کاربر به منوی کانال برمی‌گردد.

گزینه ۲ برای دیدن تغییرات ایجاد شده در کانال و پیام‌های ارسالی است. اگر عملیات موفقیت‌آمیز باشد؛ کاربر پیام‌ها را مشاهده می‌کند و به منوی کانال برمی‌گردد؛ در غیر این صورت پیام ارور نشان داده می‌شود و کاربر به منوی کانال برمی‌گردد.

گزینه ۳ برای مشاهده اعضای کانال است. اگر عملیات موفقیت‌آمیز باشد؛ کاربر اعضا را مشاهده می‌کند و به منوی کانال برمی‌گردد؛ در غیر این صورت پیام ارور نشان داده می‌شود و کاربر به منوی کانال برمی‌گردد.

گزینه ۴ برای بیرون آمدن از کانال است. اگر عملیات موفقیت‌آمیز باشد؛ کاربر به منوی اصلی برمی‌گردد؛ در غیر این صورت پیام ارور نشان داده می‌شود و کاربر به منوی کانال برمی‌گردد.

در همه‌ی منوهای بالا اگر ورودی غیر مجاز باشد، کاربر ارور را مشاهده و به همان منو بازمی‌گردد.

۲. سرور

در تابع main یک حلقه بی‌نهایت قرار دارد که همواره به کلاینت متصل می‌شود؛ درخواست او را می‌گیرد و پاسخ مناسب را می‌دهد.

```
void connect_client ();
```

این تابع برای ساخت سوکت و اتصال سرور به کلاینت می‌باشد. همچنین در این تابع چند پوشه و فایل تکست نیز ساخته می‌شود که حاوی اطلاعات کاربران و سایر اطلاعات مورد نیاز سرور است. (دیتابیس) این تابع پس از متصل کردن کلاینت تابع answering را صدا می‌زند تا پاسخ کلاینت داده شود.

```
void answering (int server_socket, FILE *users, FILE *channels, FILE *AuthToken, FILE *channel_member)
```

این تابع شامل همه‌ی پاسخ‌های احتمالی سرور می‌باشد و بنا به نوع درخواست به آن‌ها پاسخ می‌دهد.

ورودی این تابع سوکت و فایل‌های ساخته شده در تابع connect_client است.

پس از دریافت درخواست کلاینت یکی از موارد زیر رخ می‌دهد :

➤ برای درخواست register، اگر نام کاربر در فایل users وجود نداشته باشد؛ کاربر ثبت‌نام می‌شود و اطلاعات او در فایل نوشته می‌شود. در غیر این صورت پیام خطا به کلاینت فرستاده می‌شود.

➤ برای درخواست login، اگر نام کاربر وجود نداشته باشد و یا پسورد او اشتباه باشد؛ ارور مربوطه ارسال می‌شود و در صورت موفقیت آمیز بودن همه‌ی موارد یک توکن ساخته می‌شود و در فایل authtoken به همراه نام کاربر نوشته می‌شود.

➤ برای درخواست create channel، اگر توکن و نام کانال مشکلی داشته باشند، ارور مربوطه ارسال می‌شود. در غیر این صورت فایلی به نام کاربر ساخته می‌شود که در آن نام کانال نوشته می‌شود. همچنین فایلی به نام کانال ساخته می‌شود که در آن پیام‌ها نوشته می‌شود. در نهایت در فایل channels نام کانال نوشته می‌شود.

➤ برای درخواست join channel، اگر توکن و نام کانال مشکلی داشته باشند، ارور مربوطه ارسال می‌شود. در غیر این صورت فایلی به نام کاربر ساخته می‌شود که در آن نام کانال نوشته می‌شود. همچنین در فایلی به نام کانال پیام جوین شدن نوشته می‌شود.

- برای درخواست `send`، اگر توکن مشکل داشته باشد، ارور مربوطه ارسال می‌شود. در غیر این صورت آن پیام در فایل کانال نوشته می‌شود و پیام موفقیت به کلاینت فرستاده می‌شود.
- برای درخواست `refresh`، اگر توکن مشکل داشته باشد، ارور مربوطه ارسال می‌شود. در غیر این صورت پیام‌های درون فایل کانال خوانده و برای کلاینت فرستاده می‌شوند.
- برای درخواست `channel member`، اگر توکن مشکل داشته باشد، ارور مربوطه ارسال می‌شود. در غیر این صورت اسامی اعضا از فایل `channel member` استخراج و برای کلاینت فرستاده می‌شود.
- برای درخواست `leave`، اگر توکن مشکل داشته باشد، ارور مربوطه ارسال می‌شود. در غیر این صورت پیام خروج کاربر در فایل کانال نوشته می‌شود. همچنین در فایل `channel member` تغییرات لازم اعمال می‌شود.
- برای درخواست `logout`، اگر توکن مشکل داشته باشد، ارور مربوطه ارسال می‌شود. در غیر این صورت توکن کاربر از فایل `authtoken` پاک می‌شود و کاربر خارج می‌شود.

همچنین چند تابع کمکی دیگر در سرور استفاده شد که به شرح زیر است :

`int maximum (int a, int b)` 🌈

این تابع با گرفتن دو عدد صحیح، عدد بزرگتر را بازمیگرداند!

`int search_file (FILE *fptr, char *wanted)` 🌈

این تابع در یک فایل به دنبال یک رشته می‌گردد و در صورت وجود یک و در غیر این صورت صفر بازمی‌گرداند.

`int search_file_gets (FILE *fptr, char *wanted)` 🌈

این تابع مانند تابع قبل است با این تفاوت که به جای `fgets`، `fscanf` می‌کند.