پروژه اول شبکه های کامپیوتری

پیاده سازی پیام رسان ساده

محمد توکلی

9731014

این پروژه درمورد socket programming است که با استفاده از مدل client-server پیاده سازی شده است.میدانیم در این مدل server به تنهایی به تمامی درخواست client ها پاسخ میدهد.

میدانیم در این معماری ارتباطات از نوع tight coupling است یعنی client ها برای برقراری ارتباط با سرور باید اطلاعات زیر را داشته باشند :

1) Service name

2) Service interface

3) Server address

4) Service port

در این پروژه که هدف پیاده سازی یک پیام رسان است client ها باید بتوانند باهم ارتباط برقرار کنند و پیام ها و فایل های خودرا بهم بفرستند.

برای این منظور ابتدا درخواست های خودرا به server اراسال میکنند و سرور پس از دریافت request های کلاینت ارتباط او را با client های دیگر برقرار میکند.

یعنی هرگاه دو کلاینت بخواهند به یکدیگر پیام بفرستند باید این کاررا از طریق server انجام دهند.

یعنی پیام های خودرا به سرور ارسال میکنند و سرور پس از دریافت پیام ها آنهارا به client مورد نظر ارسال میکند.

توضیح کلاس ها و توابع پیاده سازی شده:

server : این کلاس همانطور که ذکر شد وظیفه پاسخ دادن به درخواست های client هارا برعهده دارد و اینکاررا به صورت multi-threading انجام میدهد یعنی همزمان میتوانیم به درخواست های چندین کاربر پاسخ بدهیم.

در این کلاس برای اینکه multi-threading پیاده سازی شود برای هر client یک ترد پیاده سازی میشود تا به طور جداگانه بتواند به درخواست های آن client پاسخ بدهد.

همچنین یک ترد وظیفه برقراری ارتباط با client هایی که به سرور وصل میشوند را دارد.

```
accept_incoming_connections()
```

این متد در داخل یک ترد قرار دارد و وظیفه ی برقراری ارتباط با client هایی که میخواند به لیست کاربران اضافه شوند را دارد و هر کابری که اضافه شود آنرا به لیست کاربران اضافه میکند و درون خود یک ترد را به وجود می آورد که به تنهایی وظیفه پاسخگویی به درخواست های آن کاربر را دارد و آدرس و شماره پورت کاربر را در list دخیره میکنیم.

```
handle_client
```

این متد که درون یک ترد قرار دارد وظیفه پاسخگویی به درخواست های یک کاربر خاص را برعهده دارد یعنی برای هرکاربری که به سرور وصل میشود این متد همیشه آماده پاسخگویی به درخواست های آن کاربر است.

در این متد ابتدا نام کاربر را دریافت می کنیم و چک میکنیم که نام کاربر وجود نداشته باشد و هرکاربری باید نام کابری یکتا خودرا داشته باشد.

سپس این کاربر را در database ذخیره میکنیم سپس در یک حلقه به درخواست های این کاربر پاسخ میدهیم و پیام های او را دریافت میکنیم.

اگر کاربر #quit را ارسال کند ارتباط او با سرور تمام میشود و از database حذف میشود.

اگر #list را وارد کند میتواند لیست کاربران آنلاین را مشاهده کند.

اگر #client را وارد کند سپس باید نام کاربری که میخواهد با او ارتباط برقرار کند را وارد کند اگر کاربر آنلاین باشد ارتباط برقرار میشود و اگر آفلاین باشد باید نام دیگری را وارد کند.

و اگر #change را وارد کند میتواند نام کاربری خودرا تغییر دهد.

```
broadcast(msg, prefix=""):
```

این متد یک پیام را به تمامی client ها ارسال میکند

```
message_to_client(msg, spec, prefix=""):
```

این متد پیام را به یک کاربر خاص که به او داده میشود ارسال میکند.

```
add_to_database(name, ip, port):
```

این متد یک کاربر را به دیتا بیس اضافه میکند.

```
get_client(name):
```

این متد آدرس و شماره پورت یک کاربر را بر اساس نام او برمیگرداند

```
delete_from_database(name):
```

این متد یک کاربر را از دیتابیس حذف میکند.

```
receive_message():
```

سرور میتواند لیست کاربران آنلاین را مشاهده کند

Client : این کلاس وظیفه مدیریت امور مربوط به کلاینت ها را برعهده دارد و پیام های کاربر را به سرور ارسال میکند و پیام های مربوط به خودرا از سرور دریافت میکند.

در این کلاس یک ترد برای دریافت پیام های کاربر صدا زده میشود که هرگاه سرور پیامی را به این کلاینت ارسال کرد آنرا دریافت کند و آنرا به کاربر نشان دهد.

```
receive():
```

این متد وظیفه دریافت پیام هایی که به کاربر ارسال میشود را برعهده دارد.

اگر file# را دریافت کند منتظر دریافت فایل از سمت سرور که کاربر دیگری آنرا ارسال کرده است میماند و چون اندازه فایل هراندازه میتواند باشد ما باید دریک حلقه آن فایل را دریافت کنیم زیرا بافری که مسئول تبادل پیام ها است حجک مشخصی دارد(1kb)

و بیشتر از آن نمیتواند دریافت کند پس باید آنرا در یک حلقه قرار دهیم.

```
send(event=None):
```

این متد که در یک حلقه قرار دارد پیام کاربر را ارسال میکند.با استفاده از console پیام را از کاربر دریافت میکند و آنرا به سمت سرور ارسال میکند.

اگر کاربر file# را وارد کند یعنی قصد ارسال فایل را دارد.سپس آدرس فایلی را که قصد ارسال آن را دارد وارد میکند و به فایل به قسمت هایی که حجم آن ها به اندازه حجم بافر است تقسیم میشوند و ارسال میشوند.

Db : این کلاس وظیفه مدیریت مربوط به دیتابیس را برعهده دارد.در این پروژه یک table ساخته شده است که کاربرهای آنلاین را درون خود ذخیره میکند و برای هر کاربر اطلاعاتی نظیر اسم , آدرس و شماره پورت رو ذخیره میکند.

```
create_db():
```

این متد دیتا بیس را با مشخصات ذکر شده به وجود می آورد.

```
create_table():
```

این متد تیبل user را به وجود می آورد

```
insert_user(name, ip, port):
```

این متد کاربر را به دیتابیس اضافه میکند

```
select():
```

این متد تمام کاربران آنلاین را باز میگرداند.

```
check_exists(name):
```

این متد چک میکند کاربری با اسم داده شده در دیتابیس وجود دارد یا خیر.

```
delete_user(name):
```

یک کابر را با از دیتابیس به وسیله اسم آن حذف میکند.

```
get_online_users():
```

اسم کاربران آنلاین را باز میگرداند.

```
change_name(oldName, newName):
```

اسم یک کاربر را عوض میکند.

شرکت کننده های سیستم و FSM : در این پروژه سه سیستم client , server و db تعریف شده است که ارتباطات آنها در شکل نشان داده شده است.