به نام خدا

محمد سیفائی ۹۶۲۴۳۰۳۳

حمیدرضا رنجبرپور ۹۶۲۴۳۰۳۰

روند کلی برنامه

برنامه در ابتدا دو عدد برای پورت UDP (broadcast_port) و (my_chat_port) چت انتخاب می‌کند و سپس دو نخ[1] ساخته و هر دو را همزمان اجرا می‌کند. در نخ اول (udp_broadcast) یک سوکت UDP روی پورت تصادفی همه‌پخشی[2] باز شده و در یک پورت ثابت شروع به درخواست همه‌پخشی شامل پورت چت تصادفی خود می‌کند و در سوکت دوم (udp_listen) یک سوکت UDP دیگر باز شده و بر روی همان پورت ثابتی که سوکت اول درخواست خود را ارسال می‌کرد شروع به گوش کردن می‌کند.

در صورت دریافت شدن یک پیام همه‌پخشی سوکت منتظر دریافت پیام از سوکت مقابل می‌ماند تا پورت چت آن را بگیرد و سپس روی پورتی که در درخواستش آن را ارسال کرده بود سرور چت را راه اندازی می‌کند (listen) و دوباره خبر موفقیت راه اندازه را به طرف مقابل از طریق همان کانال UDP ارسال می‌کند و از این لحظه به بعد نقش سرور را در چت بازی می‌کند.

از آن طرف در طرف مقابل با دریافت اولین درخواست UDP توسط سوکت منتظر[3] درخواست‌های همه‌پخشی متوقف می‌شود و سوکت منتظر به درخواست‌دهنده، پورت چت خود را ارسال کرده و آدرس و پورت درخواست‌دهنده را ذخیره می‌کند (other_addr)، منتظر گرفتن خبر موفقیت راه‌اندازی سرور می‌ماند و پس از دریافت آن به سرور چت متصل می‌شود (connect). برنامه دریافت‌کننده درخواست از این به بعد کلاینت چت است.

سرور در انتظار اتصال کلاینت می‌ماند و پس از اتصال هر برنامه روندی مشابه را طی می‌کنند (handle_chat). روند چت در ابتدا دو نخ دیگر را اجرا می‌کند، یکی برای دریافت ورودی پیام‌های کاربر از طریق کیبورد و ارسال آن، دیگری برای دریافت پیام‌های طرف مقابل.

پیام‌ها از طریق دو رنگ مجزا از هم جدا شده‌اند یکی برای کاربر و دیگری برای طرف مقابل. (پیام‌های سیستم شامل خطا و اعلان‌های دیگر به رنگ قرمز نشان داده می‌شوند)

در صورت وارد شدن کلمه EXIT توسط کاربر و یا در صورت اختلال یا بروز خطا در هر یک از این مراحل فرایند، سوکت از طرف بسته می‌شود (close_tcp_connection و `is_connected`) و هر دو برنامه همه‌پخشی و انتظاذ در UDP را از سر می‌گیرند.

[1] Thread
[2] Broadcast
[3] Listener

پروتکل

پروتکل به دو بخش همه‌پخشی (udp.py) و چت (chat.py) تقسیم می‌شود که هر دو در فرمت UTF-8 تبدیل می‌شوند.

در بخش همه‌پخشی:

درخواست به این فرمت ارسال می‌شود:

```
"I will listen on {server-chat-port}"
```

پاسخ درخواست:

```
"I am ready on Port {client-chat-port}"
```

اعلام راه‌اندازی سرور چت:

```
"I am listening at {server-chat-port}"
```

در بخش چت ارسال پیام‌ها به این صورت است:

```
"{sender-name}:{message}"
```

که نام فرستنده در ابتدا از شروع برنامه از طریق کیبورد گرفته شده است.