

نرم افزار چت ناشناس

با توجه به p2p بودن این نرم افزار، باید یک کد واحد را در همه ی سیستم ها اجرا کنیم بنابراین برای مشخص کردن عملکرد هر یک از این نرم افزارها میبایست در ابتدا مد کاری آن را دریافت کرده و با توجه به آن ادامه ی خدمت رسانی ممکن خواهد بود. پس با اجرای برنامه و مشاهده ی Enter mode: باید یکی از دو واژه ی broadcast یا listen را بنویسیم. حال اگر در مد برادکست باشیم، باید به طور مداوم پیام Hello را به صورت همه پخش در شبکه ارسال کنیم. که این کار را هر ۲ ثانیه یکبار تا زمانی که peer دیگری برای برقراری ارتباط پیدا شود، تکرار می کنیم. توجه شود که ممکن است چند listener داشته باشیم و سپس یک broadcaster وارد شود. در این صورت broadcaster تنها با یکی از این listener ها ارتباط برقرار کرده و وارد مرحله ی چت با او می شود.

پس از این که دو peer ارتباط از نوع UDP را به صورت بالا و به طور کامل و صحیح به اتمام رساندند، برای شروع چت ارتباط دیگری از نوع TCP بینشان برقرار می شود. با تقاضای شروع چت از سمت chat client و تایید آن توسط chat server، انتقال پیام برای هر یک از طرفین ممکن می شود. برای همزمانی دریافت و ارسال peer ها، برای هر کدام ۲ ترد در نظر گرفته شده که یکی پیام های دیگری را دریافت و دیگری به طور موازی در حال اجرا بوده و پیام ها را به سمت دیگر ارسال می کند.

پیام هایی که یک peer قرار است ارسال کند، به سادگی تایپ شده و در سمت دیگر پس از دریافت، در مقابل "C:" یا "S:" چاپ می شوند که به این معناست که این پیام از سمت کلاینت یا سرور دریافت شده (مقصود سمت دیگر چت است و صرفاً برای خوانایی بیشتر می باشد).

همچنین هر یک از طرفین می تواند به این چت خاتمه دهد و این کار را با تایپ عبارت "bye" یا "Bye" می تواند انجام دهد. به این صورت، هر کدام از peer ها که این جمله را تایپ کند، بلافاصله سوکت خود را close کرده و سمت دیگر با دیدن عبارت "bye" متوجه بسته شدن سمت دیگر میشود و اگر چیز دیگری برای ارسال تایپ کند، بلافاصله پیغامی دریافت می کند که مبنی بر بسته شدن سوکت سمت دیگر چت است و سوکت این سمت نیز بسته می شود.

توجه شود که سایر جزئیات پیاده سازی مثل انتخاب پورت تصادفی برای listener و ... همگی مطابق صورت پروژه انجام شده و دیگر در این جا شرح داده نشده.

بررسی چند سناریو:

- اگر در ابتدا دو listener و سپس یک broadcaster داشته باشیم، broadcaster تنها با یکی از این listener ها ارتباط برقرار می کند. که عملکرد هر یک از این سه peer در شکل های زیر مشاهده می شود:

```
Enter mode:
listen
Hello received from a broadcaster...
I am ready on port 34624
Chat initiation request received!
Let's chat...
|
```

```
Enter mode:
listen
Hello received from a broadcaster...
I am ready on port 5157
|
```

```
Enter mode:
broadcast
Hello to all listeners!
Connection accepted with port 34624
Chat initiation request sent...
|
```

- اگر در ابتدا تنها یک broadcaster داشته باشیم، تا زمانی که یک listener برای برقراری ارتباط وارد شبکه شود، عمل broadcasting هر دو ثانیه یک بار تکرار می شود:

```
Enter mode:
broadcast
Hello to all listeners!
Hello to all listeners!
Hello to all listeners!
Connection accepted with port 16697
Chat initiation request sent...
|
```

```
Enter mode:
listen
Hello received from a broadcaster...
I am ready on port 16697
Chat initiation request received!
Let's chat...
|
```

- حال سناریوی چت را برای دو peer مشاهده می کنیم:

```
Enter mode:
broadcast
Hello to all listeners!
Connection accepted with port 40304
Chat initiation request sent...
hello
S : hi
S : how are you?
thank you
bye
Connection closed by client.

Process finished with exit code 0
|
```

```
Enter mode:
listen
Hello received from a broadcaster...
I am ready on port 40304
Chat initiation request received!
Let's chat...
C : hello
hi
how are you?
C : thank you
C : bye
bye
Connection closed by client.

Process finished with exit code 0
|
```