



مقدمه:

هدف از انجام این تمرین آشنایی با فراخوانی‌های سیستمی و به طور خاص برنامه‌نویسی سوکت^۳ بر بستر IPv4^۴ می‌باشد. در حالت کلی به منظور برقراری ارتباط بر این بستر دو روش معمول وجود دارد: (۱) اتصال محور^۵ (۲) بدون (نیاز به) اتصال^۶. از طرفی برای نحوه ارسال اطلاعات نیز روش‌های متفاوتی وجود دارد که دو روشی که در این تمرین بررسی خواهند شد عبارتند از Unicast و Broadcast. در حالت عادی، امکان استفاده از Broadcast فقط در یک ارتباط بدون اتصال فراهم می‌باشد، این در حالی است که امکان استفاده از Unicast، هم در ارتباط اتصال محور و هم در ارتباط بدون اتصال فراهم است.

شرح تمرین:

در دنیای رقابت‌های بی‌پایان ce، بازی بزرگ و جذابی به نام "سنگ، کاغذ، قیچی" در اتاق‌هایی برگزار می‌شود. این بازی در ظاهر ساده است، اما تنها کسانی که دارای زیرکی و سرعت عمل بالایی هستند می‌توانند برنده شوند. در این بازی، دو بازیکن از تیم‌های مختلف می‌توانند در یک اتاق رو در روی یکدیگر قرار بگیرند.

^۱ belmi@ut.ac.ir

^۲ sobhan13495782sobhan@gmail.com

^۳ Socket Programming

^۴ Internet Protocol Version 4

^۵ Connection-Oriented

^۶ Connectionless

سنگ، کاغذ یا قیچی؟! انتخابی است که در این اتاق‌ها صورت می‌گیرد. این انتخاب باید در سکوت و پنهان از چشم حریف انجام شود. در هر نبرد، بازیکنان تنها ۱۰ ثانیه برای انتخاب دارند. اگر یک بازیکن در این مدت تصمیم نگیرد، بازی را باخته است، اما اگر هیچ‌کدام تصمیم نگیرند، نبرد بی‌نتیجه و مساوی اعلام می‌شود.

پس اتمام نبرد، اتاق، برنده نهایی را به همگان اعلام می‌کند.

نقش‌ها:

در این بازی نقش‌های متفاوتی وجود دارد. هر یک از این نقش‌ها ویژگی‌ها و وظایف منحصر به فرد خود را دارند و سرنوشت بازی را تحت تاثیر قرار می‌دهند. در ادامه به توضیح هرکدام از نقش‌ها خواهیم پرداخت.

۱. بازیکن عادی:

این بازیکنان پس از ورود به اتاق، با گرفتن لیست اتاق‌های موجود از سمت مسئول برگزاری، در بازی "سنگ، کاغذ، قیچی" شرکت می‌کنند. آن‌ها پس از دریافت منوی بازی از اتاق (سنگ، کاغذ، قیچی)، انتخابشان را در ۱۰ ثانیه انجام داده و منتظر اعلام نتیجه می‌مانند. این بازیکنان پس از اتمام نبرد امکان بازی مجدد را دارند.

۲. مسئول برگزاری :

مسئول برگزاری با توجه به شرایط، اقدامات زیر را انجام خواهد داد:

1. گرفتن نام بازیکن به هنگام ورود به بازی.
2. اعلام اتاق‌های موجود به بازیکنان خارج از اتاق.

اتاق:

همان‌طور که پیش‌تر اشاره شد، اتاق‌ها محل بازی بازیکنان است.

- در هر اتاق تنها امکان بازی کردن دو نفر وجود دارد.
- به محض حاضر شدن هر دو نفر، بازی شروع می‌شود.
- اتاق، منوی انتخاب بازی را برای بازیکنان می‌فرستد (سنگ، کاغذ یا قیچی).
- بازیکنان، انتخابشان را برای سرور اتاق می‌فرستند (توجه کنید که انتخاب بازیکنان می‌بایست از بازیکن دیگر مخفی باشد).
- در صورت انتخاب نکردن هر کدام از بازیکنان در مدت ۱۰ ثانیه پس از ارسال منو توسط اتاق، به منزله شکست بازیکن تلقی خواهد شد. ولی در صورتی که هر دو نفر در این بازه زمانی انتخابی انجام ندهند، بازی مساوی شده و اتاق فردی را به عنوان برنده اعلام نمی‌کند.
- اعلام نتیجه بازی و مشخص کردن برنده، بازنده و یا مساوی شدن دو بازیکن، به همراه نام آن‌ها، از سمت اتاق.

بازی:

نحوه بازی:

ابتدا فردی که قصد عضویت در بازی را دارد به مسئول برگزاری مراجعه می‌کند و به سوال مسئول برگزاری در خصوص هویت (نام) خود پاسخ می‌دهد. سپس مسئول برگزاری لیستی از اتاق‌های موجود را برای بازیکن ارسال می‌کند. در نهایت بازیکن در اتاق قرار می‌گیرد (به اتاق وصل می‌شود). در این فاصله، ورودی این بازیکن، تنها برای اتاق فرستاده خواهد شد. در صورت وجود فرد دیگر در همان اتاق، بازی به شکلی که پیش‌تر توضیح داده شد انجام می‌گیرد و در این فاصله، اتاق از لیست اتاق‌های موجود، تا لحظه اتمام بازی، حذف می‌شود. در نظر داشته باشید اگر بازیکنی قصد ورود به اتاقی را داشت که موجود نیست (به دلیل در دست داشتن لیست قدیمی)، از ورود او با پیغام مناسب جلوگیری شود.

اتمام بازی

مسئول برگزاری می‌تواند در هر زمان با تایپ دنبالهٔ `end_game` در ورودی استاندارد خود⁷ به بازی پایان دهد. پس از اعلام اتمام بازی توسط مسئول برگزاری، نتیجه نهایی بازی شامل تعداد دست‌هایی که هر بازیکن برده است برای تمامی افراد، `broadcast` خواهد شد. همچنین، در صورت مغایرت ورودی با دنباله، بازی ادامه پیدا می‌کند.

راهنمایی‌ها:

1. هر کدام از موجودیت‌های بازی از منظر پیاده‌سازی، نقش خود را دارند (سرور یا کلاینت)، پیشنهاد می‌شود ابتدا در ارتباط با این موضوع فکر کنید و سپس در صورت نیاز از راهنمایی‌های زیر استفاده کنید:

a. مسئول برگزاری⁸

b. بازیکنان معمولی⁹

c. اتاق¹⁰

2. از آنجایی که شماره پورت‌ها تا عدد ۱۰۲۴ رزرو شده برای استفاده‌های سیستمی هستند، برای راه‌اندازی اتاق‌ها، از شماره پورت‌های خالی بیشتر از این عدد استفاده کنید.

3. برای حل موضوع ارتباطات همزمان می‌توانید از `select` و یا `poll` به دلخواه استفاده کنید.

4. فراخوانی سیستمی `socket` یک توصیف‌کننده فایل¹¹ که یک عدد است را بر می‌گرداند، این عدد تنها منحصر به پردازش¹² است که آن را فراخوانی کرده و در پردازش دیگر معتبر نخواهد بود.

⁷ `stdin`

⁸ Main Server

⁹ Client

¹⁰ Subserver

¹¹ File Descriptor

¹² Process

فرض‌های پروژه:

بجز مواردی که در پروژه خواسته شده‌اند، مانند استفاده از broadcast در اعلان‌های عمومی یا استفاده نکردن از broadcast (اعلام انتخاب‌های بازیکنان در اتاق‌ها)، می‌توانید برای دیگر مواردی که در پروژه با آن برخورد می‌کنید، فرض دلخواه در عین حال منطقی خودتان را داشته باشید. علاوه بر این، به موارد زیر در پیاده‌سازی پروژه توجه داشته باشید:

- برای اعلان‌های عمومی مانند اعلام نتیجه هر بازی توسط اتاق و یا نتیجه کلی توسط مسئول برگزاری از broadcast استفاده کنید. (اجباری)
- برای انتقال انتخاب‌های بازیکنان در اتاق‌ها حق استفاده از broadcast را ندارید. (اجباری)
- برای سرور اصلی (مسئول برگزاری) و سرورهای کمکی (اتاق‌ها) تنها حق استفاده از یک برنامه و امکان یک اجرا (یک پردازش) وجود دارد. همچنین برای کلاینت‌ها حق استفاده از یک برنامه و امکان چند اجرا وجود دارد. (اجباری)
- می‌توانید فرض کنید آدرس آپی و شماره پورت مربوط به اعلان‌های عمومی ذکر شده در صورت پروژه، به صورت hardcode در برنامه سرور و کلاینت وجود دارد. (اختیاری)

ورودی:

به منظور راه‌اندازی مسئول برگزاری از دستور زیر استفاده کنید:

```
$ server.out {IP} {Port} {#Rooms }
```

که در آن:

- IP: آدرس آپی مسئول برگزاری است.
- Port: شماره پورته که مسئول برگزاری روی آن قرار می‌گیرد.
- #Rooms: تعداد اتاق‌هایی که در بازی قابل استفاده خواهند بود.

به منظور راه‌اندازی بازیکنان معمولی از دستور زیر استفاده کنید:

\$ client.out {IP} {Port}

که در آن:

- IP: آدرس آیپی مسئول برگزاری است.
- Port: شماره پورته که مسئول برگزاری روی آن قرار می‌گیرد.

طراحی:

پیش از شروع تمرین، ابتدا شمای طراحی خود شامل: موجودیت‌های پروژه، اتصالات میان آن‌ها و نوع اتصالات را مشخص کنید (broadcast یا unicast و همچنین Connectionless یا Connection-Oriented) و در کنار فایل‌های پروژه خود قرار دهید. سپس به پیاده‌سازی آن بپردازید. شمای طراحی بخشی از نمره شما را به خود اختصاص می‌دهد.

نکات و نحوه تحویل

- کدهای شما می‌بایست به زبان C++/C و در سیستم عامل **Linux** کامپایل شوند.
- توجه داشته باشید که در تمامی مراحل از فراخوانی‌های سیستمی موجود (مانند read, write, open) بجای توابع آماده (مانند scanf, printf, fopen) استفاده نمایید. در صورت استفاده از توابع آماده در هر بخش، نمره آن بخش، لحاظ نمی‌شود.
- استفاده از داده‌ساختارها (مانند vector, string, unordered_map) و امکانات زبان C++ و همچنین توابع انجام عملیات رشته‌ای (مانند sprintf, strtol, atoi) بلامانع است.
- تمامی فایل‌های خود از قبیل کد، Makefile و شمای طراحی خود را در پوشه‌ای تحت عنوان OS-CA1-StudentNumber که در آن Student Number، شماره دانشجویی شما خواهد بود قرار داده، آن را zip کنید و در سامانه بارگذاری کنید.

- این تمرین صرفاً برای یادگیری شما طرح شده است. در صورت محرز شدن تقلب در تمرین، مطابق با قوانین درس برخورد خواهد شد.
- در صورت داشتن سوال می‌توانید از طریق فروم درس یا شرکت در جلسات رفع اشکال سوالات خود را مطرح کنید.
- مواردی که در جلسه توجیهی، فروم درس و گروه اسکایپی درس توضیح داده می‌شوند جزئی از پروژه خواهند بود.

موفق باشید!