سیستم عامل
پروژه ۱: فراخوانی سیستمی
زمان تحویل: ۲۹ مهر

در این پروژه قرار است شما یک بازی آنلاین "کشتی جنگی (battleship) " را با استفاده از فراخوانی‌های سیستمی به زبان C و استفاده از socket programming پیاده‌سازی کنید. برای آشنایی کلی با این بازی ساده می‌توانید به لینک زیر مراجعه کنید:

www.battleshiponline.org

در پیاده‌سازی این سیستم یک سرور مرکزی داریم که وظیفه پیدا کردن و جفت کردن حریف‌ها را با یکدیگر دارد اما می‌خواهیم سیستم بازی آنلاین خود را به‌گونه‌ای طراحی کنیم که در صورت در دسترس نبودن سرور نیز کاربران بتوانند با یافتن حریف بازی جدیدی شروع کنند.

شرح بازی:

تا به اینجا احتمالا از طریق لینکی که در اختیار شما قرار دادهشد با نحوه انجام بازی اصلی آشنا شده‌اید. در اینجا قصد داریم تا به هدف کم شدن بار برنامه‌نویسی و تمرکز بیشتر روی جنبه‌های سیستم عاملی پروژه بازی را کمی ساده‌تر کنیم. برای شروع بازی لازم است که هر یک از بازیکنان نقشه زمین خود که شامل محل کشتی‌های جنگیشان است را انتخاب کنند. این زمین به صورت یک مربع خانه‌بندی شده با ابعاد ۱۰x۱۰ است که محل قرار گیری کشتی‌ها در این ماتریس از طریقی فایلی که در کنار برنامه قرار دارد به کلاینت داده می‌شود. در این فایل خانه‌هایی که در آن‌ها کشتی قرار گرفته با ۱ و خانه‌های خالی با عدد صفر نشان داده شده‌اند.(نمونه فایل ورودی برای نقشه اولیه بازی در کنار صورت پروژه قرار گرفته‌است.)

پس از مشخص شدن نقشه‌ها و اعلام آمادگی طرفین بازی شروع می‌شود. در اینجا برای سادگی فرض می‌کنیم نوبت در شروع بازی همواره با فردی است که با او تماس گرفته شده‌است. در هر مرحله از بازی فردی که نوبت را در اختیار دارد با اعلام یک مختصات به طرف مقابل در واقع نقطه‌ای در زمین حریف که قصد شلیک کردن به آن را دارد مشخص می‌کند. حال با توجه به نتیجه شلیک که توسط حریف اعلام می‌شود تصمیم‌گیری در مورد حرکت بعدی اتفاق می‌افتد: اگر گلوله به کشتی برخورد کرده بود نوبت مجددا در اختیار بازیکن قبلی قرار می‌گیرد و در غیر این صورت نوبت به بازیکن مقابل داده می‌شود. بازی تا به آنجا ادامه پیدا می‌کند که یک بازیکن موفق شود تمام خانه‌های حریف که در آن کشتی قرار دارد را هدف بگیرد و به این ترتیب برنده بازی شود.

انتخاب حریف در شرایط وجود سرور:

ابتدا سناریویی که سرور در شبکه موجود است را بررسی می‌کنیم. در این حالت سرور هر یک ثانیه از طریق Signal پیام Heartbeat را روی پورت X می‌فرستد و محتوای این پیغام پورت و آدرس IP ای است که سرور روی آن listen می‌کند. هر یک از کلاینت‌ها پس از اطلاع از وجود و صحت کارکرد سرور از طریق پیام Heartbeat لازم است تا اطلاعات خود که شامل یک نام کاربری و پورت و آدرس IPای که روی آن listen می‌کنند را برای سرور بفرستند و به این طریق سرور از وجود کاربر و درخواست او برای یک حریف مطلع میشود و در این زمان کاربر باید منتظر بماند تا یک حریف برای او پیدا شود. در این مرحله سرور پس از دریافت یک درخواست دیگر برای شروع بازی دو کاربر قبلی را به عنوان جفت در نظر گرفته و اطلاعات مورد نیاز برای اتصال آن دو را در اختیار کاربر دوم قرار می‌دهد. پس از برقراری اتصال مستقیم بین این دو کاربر بازی آغاز می‌شود و از این پس ارتباط بین دو کاربر مطابق قوانینی است که پیش‌تر در بخش شرح بازی توضیح داده‌شد.

انتخاب حریف در شرایط نبود سرور:

همان طور که گفته شد می‌خواهیم سیستم خود را به گونه‌ای طراحی کنیم که در شرایطی که سرور ما قادر به فعالیت نبود و یا در شبکه حضور نداشت هم کابران ما قادر به یافتن حریف و شروع بازی باشند. برای این کار یک کاربر پس از ورود به شبکه اگر پیغام مبتنی بر موجود بودن سرور در شبکه روی پورت X را دریافت نکرد باید با استفاده از Network Broadcast روی پورت Y درخواست خود برای یافتن حریف و شروع یک بازی جدید را به اطلاع سایر کاربران موجود در شبکه برساند. در اینجا پیاده‌سازی مکانیزم صحیح برای یافتن حریف و یا انتخاب یکی از آن‌ها در صورت حضور چند کاربر پیشین برعهده شما است و هر روش صحیح و قابل توجیه برای پیاده‌سازی قابل قبول است. در پیاده‌سازی این بخش دقت کنید تا پیاده‌سازی را به نحوی انجام دهید که بقیه کاربرانی که اعلام آمادگی کرده‌اند اما به عنوان حریف انتخاب نشده‌اند مجددا به جستجو برای حریف جدید ادامه دهند.

توجه کنید که نیازی نیست تا شما حالتی را پیاده‌سازی کنید که تعدادی از کاربران برای حریف به سرور درخواست داده‌اند اما هنوز برای آن‌ها پاسخی نیامده‌است و در این زمان سرور از دسترس خارج می‌شود. فرض کنید در هنگام خروج سرور از شبکه به تمامی درخواست‌ها تا قبل از این لحظه پاسخ داده‌شده است.

انتخاب یک حریف مشخص:

در این بخش قصد داریم که به سیستم مان یک قابلیت جدید اضافه کنیم. این قابلیت جدید قرار است به ما این امکان را بدهد که به عنوان یک بازیکن با داشتن نام کاربری یک بازیکن دیگر از او برای بازی کردن دعوت کنیم. حال این سناریو را در وجود سرور بررسی می‌کنیم.

در حالتی که سرور در شبکه موجود باشد در صورت وجود این نوع درخواست از سوی یک کاربر در صورت موجود بودن فرد مورد درخواست در شبکه (آنلاین بودن این فرد) اطلاعات این فرد به بازیکن درخواست کننده داده می شود تا برای شروع بازی به او متصل شود و در غیر این صورت درخواست این فرد را ذخیره می‌کند و هرگاه حریف مورد نظر آنلاین شد در مورد

درخواستی که برای او ایجاد شده بود به او اطلاع رسانی می‌شود. توجه کنید که درخواست هر فرد تا زمان آنلاین بودن او در شبکه معتبر است و بعد از آن هیچ اعتباری ندارد. پیاده‌سازی مکانیزمی که بااستفاده از آن بتواند در مورد آنلاین بودن یا نبودن یک کاربر تصمیم بگیرد بر عهده شما است و هر روش صحیحی قابل قبول است. برای مثال می‌توانید در میان دستورهایتان یک دستور login که به سرور فرستاده می‌شود نیز پیاده‌سازی کنید.

بخش‌های امتیازی:

۱.انتخاب یک حریف مشخص در شرایط نبود سرور:

در حالتی که سرور در شبکه موجود نیست کاربر مجددا باید از طریق Network Broadcast جستجویش برای حریف مشخص را اعلام کند و درصورتی که کاربر مورد نظر او در شبکه موجود بود از طرف این کاربر با او تماس حاصل می‌شود و در غیر این صورت این درخواست معتبر نمی‌باشد.

۲.سناریوی بازگشت سرور به شبکه و پیاده‌سازی دستور status روی سرور:

در این بخش می‌خواهیم تا سرور آماری از نتیجه بازی‌ها را درخود ذخیره کند. به این منظور در پایان هر بازی نتیجه آن بازی باید به نحوی به سرور مرکزی اعلام شود. توجه کنید که در صورتی که سرور در شروع یک بازی در شبکه حضور نداشت اما تا پیش از آفلاین شدن حداقل یکی از طرفین بازی به شبکه برگشت کاربران وظیفه دارند تا نتیجه بازی‌هایی که در نبود سرور اتفاق افتاده است را به اطلاع او برسانند. حال با وارد کردن دستور status روی سرور باید بتوانیم آمار نتایج را ببینیم. فرمت نمایش این نتایج به عهده خودتان است اما در آن باید به نام کاربران بازی و برنده آن اشاره کنید. توجه کنید در این بخش باید شرایطی که در آن سرور در شبکه نیست و سپس به شبکه اضافه می‌شود را به طور کامل پیاده‌سازی کنید یعنی در این بخش باید سناریویی که در آن فردی در نبود سرور درخواست بازی داده است اما حریفی برای او یافت نشده است و سپس سرور وارد شبکه می‌شود را نیز پیاده کنید. در این حالت باید مکانیزمی طراحی کنید که با استفاده از آن درخواست این کاربر مجددا برای سرور فرستاده شود.

نکات مهم :

۱. در کد کلاینت و سرور به کمک فراخوان سیستمی select، تمام I/O ها به شکل Asynchronous انجام شوند و هیچ بخشی از کدتان blocking نباشد.

۲. کلاینت و سرورتان باید این‌گونه اجرا شوند:

./server - - server-broadcast-port X - - client-broadcast-port Y
./client - - server-broadcast-port X - - client-broadcast-port Y

تعیین فرمت پیاده‌سازی سایر دستورها بر عهده خودتان است.

خلاصه‌ای از پروژه و سناریوهایی که باید پیاده‌سازی کنید:

-کلاینت‌ها روی پورت Y اطلاعات و پیام‌هایشان را برای سایر کلاینت‌ها می‌فرستند و روی پورت M منتظر درخواست پیام هستند.(دقت کنید که پورت M برای کلاینت‌های مختلف متفاوت باشد تا امکان تست کردن برنامه وجود داشته باشد.)

سرور روی پورت X پیام Heartbeat را می‌فرستد و روی پورت N که در پیام Hearbeat اعلام می‌کند منتظر درخواست برای حریف و یا اعلام نتیجه از طرف کلاینت‌ها است.

سناریوهایی که باید پیاده کنید:

۱. سرور درون شبکه وجود دارد و یک کلاینت درخواست حریف و بازی جدید می‌کند.

۲.سرور در شبکه و وجود دارد و یک کلاینت درخواست بازی با یک کاربر مشخص می‌کند.

۳. سرور درون شبکه وجود ندارد و یک کلاینت درخواست حریف و بازی جدید می‌کند.

۴.سرور در شبکه و وجود ندارد و یک کلاینت درخواست بازی با یک کاربر مشخص می‌کند.

۵.سرور در شبکه وجود ندارد و بعد به شبکه اضافه می‌شود.

توجه کنید که اعلام نتایج بازی در هر یک از شرایط بالا باید درنظر گرفته شود و همچنین اگر یکی از طرفین در طول بازی آفلاین شد طرف دیگر باید ناتمام ماندن بازی را به اطلاع سرور برساند.

این سناریوها بخش‌های مربوط به بخش امتیازی را هم شامل می‌شوند.(سناریو ۴ و ۵ و بخش اعلام نتایج)

نکات پایانی:

۱.در این پروژه باید به زبان C کد بزنید و کدهایتان باید با gcc قابل کامپایل کردن باشد.

۲.نکاتی که در جلسه توجیهی یا فروم درس مطرح می‌شوند بخشی از صورت پروژه هستند لذا به شما توصیه می‌شود که حتما در جلسه توجیهی شرکت کنید.

۳.حتما log ها مورد نظر را که شامل قطع یا وصل شدن کلاینت‌ها و یا سرور و سایر درخواست‌ها و نتیجه بازی در هر حرکت است را چاپ کنید. در هنگام تحویل چاپ این logها بخشی از نمره شما را تشکیل می‌دهد.

۴. پیاده‌سازی شما باید توسط فراخوانیهای سیستمی مانند write،read، open، create و ... انجام شود و استفاده از توابع کتابخانه‌ای حتی کتابخانه استاندارد مانند fprintf و fopen مجاز نیست.(معیار اینکه یک تابع فراخوانی سیستمی است یا خیر این است که بتوانید نام این تابع را در لیست فراخوانیهای سیستمی در بخش دوم لینوکس به آدرس http:// linux.die.net/man پیدا کنید.)

۵.توابع کتابخانهای که با فراخوانیهای سیستمی قابل پیادهسازی نیستند مانند strcat،atoi و ... مجاز هستند.

۶.نها تابعهایی که از system call استفاده میکنند و نیازی به پیادهسازی آنها نیست free و malloc و realloc هستند.

۷.برای آشنایی با Socket Programming میتوانید به صفحات زیر مراجعه کنید:

https://beej.us/guide/bgnet/html/single/bgnet.html#clientserver

https://beej.us/guide/bgnet/html/single/bgnet.html#broadcast