



باسمه تعالی سیستم‌های عامل پروژه‌ی اول درس



هدف از انجام این پروژه آشنایی با فراخوانی‌های سیستمی زبان C و یادگیری مبانی socket programming است.

سوکت چیست؟

سوکت یک مکانیزم برای برقراری ارتباط بین دو پردازنده روی یک یا چند ماشین است. در این ارتباط دو طرفه، سوکت مثل یک پایانه است که ما اطلاعات را به آن می‌فرستیم یا از آن دریافت می‌کنیم. در واقع سوکت نوعی abstraction برای لایه‌های پایین‌تر سیستم عامل است که این ارتباط را ممکن می‌کند.

شرح پروژه:

در این پروژه قرار است بازی «نقطه‌بازی» را با استفاده از socket programming و فراخوانی‌های سیستمی زبان C پیاده‌سازی کنید.

نحوه‌ی انجام بازی:

یک نقشه حاوی تعدادی نقطه داریم. هر شخص در نوبت خودش، اجازه دارد که دو نقطه را به یکدیگر وصل کند. فردی که ضلع چهارم یک مربع را قرار می‌دهد، آن مربع را به نام خودش می‌زند و یک امتیاز می‌گیرد. در انتها کسی که تعداد بیشتری مربع را به خود اختصاص داده باشد، برنده‌ی بازی اعلام می‌شود.

شرح کلی بازی:

در این بازی یک سرور مرکزی داریم که همواره روی پورت مشخصی (پورت X) گوش می‌کند و منتظر اتصال کلاینت‌ها (بازیکن‌ها) است. سرور و هر کلاینت، یک پردازنده هستند. هر کاربر پس از بالا آمدن برنامه‌اش به سرور متصل می‌شود و می‌تواند بین گروه‌های دو، سه و یا چهار نفره انتخاب و به سرور اعلام کند. سرور وظیفه‌ی تشکیل گروه و شروع بازی به محض تکمیل ظرفیت آن گروه را

دارد. ارتباط بین سرور و هر بازیکن از نوع TCP، و پس از شروع بازی، ارتباط بازیکن‌ها با یکدیگر از نوع UDP (broadcast) خواهد بود.

مدیریت هر گروه بازی توسط سرور از طریق اختصاص و اعلام شماره‌ی پورت broadcast مخصوص آن بازی به بازیکن‌های آن صورت می‌گیرد. پس از تکمیل نفرات، کاربران در گروه‌های انتخابی خود وارد بازی می‌شوند و به نوبت صفحه‌ی بازی را تغییر می‌دهند و نتیجه را روی پورت مشخص شده از سمت سرور در اول بازی برای بقیه‌ی اعضای گروه می‌فرستند.

نقشه‌ی بازی:

متناسب با تعداد بازیکنان هر گروه، اندازه‌ی نقشه‌ی بازی متفاوت خواهد بود. نقشه‌ی گروه‌های دو نفره شامل ۹ نقطه (۳*۳)، گروه‌های سه نفره شامل ۱۶ نقطه (۴*۴) و گروه‌های چهار نفره شامل ۲۵ نقطه (۵*۵) است.

نوبت‌دهی:

در ابتدا همه‌ی بازیکنان نقشه‌ی بازی را دارند. به ترتیب نوبت، هر فرد خط مورد نظر خود را به بقیه اعلام می‌کند و نفر بعدی بازی را ادامه می‌دهد. نوبت‌دهی بازیکنان را به روش دلخواه پیاده‌سازی کنید.

امتیازدهی:

هر کسی که ضلع چهارم یک مربع را تکمیل کرد، یک امتیاز برایش ثبت می‌شود. در انتهای بازی، فرد با بیشترین امتیاز به عنوان برنده اعلام می‌شود.

تایمر:

در حین بازی، زمان انجام حرکت هر شخص محدود است. بدین صورت که هر بازیکن ۲۰ ثانیه فرصت دارد که حرکت بعدی خود را انجام دهد. اگر این زمان تمام شد و بازیکن خط مورد نظر خود را مشخص نکرد، نوبت به نفر بعدی او می‌رسد. برای اندازه‌گیری زمان حرکت هر بازیکن شما باید از signalهای unix و به طور دقیق‌تر، از سیگنال SIGALRM استفاده کنید.

امتیازی:

فردی که ضلع چهارم مربع را قرار می‌دهد به عنوان جایزه، اجازه‌ی گذاشتن خطی دیگر را داشته باشد. (دوباره نوبت خودش باشد که بازی را ادامه دهد).

همزمانی سیستم:

در کل طول برنامه (در کد کلاینت و سرور)، تمام سیستم باید به صورت همزمان در حال اجرا باشد تا سرور بتواند همزمان به چند کلاینت رسیدگی کند. با توجه به این که تعدادی از فراخوان های سیستمی blocking هستند، برای رفع این مشکل از فراخوان سیستمی select استفاده می کنیم. این فراخوان مسئول مانیتور کردن ارتباطات همزمان است و باعث می شود که تمام I/O ها به شکل Asynchronous انجام شوند و هیچ بخشی از کد blocking نباشد.

نکات مهم:

- در کد کلاینت و سرور به کمک فراخوان سیستمی select، تمام I/O ها باید به شکل Asynchronous انجام شوند و هیچ بخشی از کدتان نباید blocking باشد.
- تمامی آدرس های IP را (127.0.0.1) localhost در نظر بگیرید.
- با قرار دادن stdin در لیستی که به select می دهید، می توانید از کنسول بدون بلاک شدن برنامه، ورودی بخوانید.
- کلاینت و سرورتان باید اینگونه اجرا شوند:

./server port_X

./client port_X

نکات پایانی:

- در این پروژه باید به زبان C کد بنویسید و کدهایتان باید با gcc قابل کامپایل کردن باشند.
- برای تحویل پروژه می توانید از یکی از دو روش زیر استفاده کنید:
 - تمامی نتایج را در یک فایل فشرده شده با عنوان OS-CA1-<#SID>.zip در محل باگذاری در سایت درس آپلود کنید.
 - با توجه به مشکلات پیش آمده برای استفاده از GitLab توسط ip های ایران، امکان استفاده از این سایت برای آپلود پروژه وجود ندارد. اما می توانید از سایت bitbucket به جای آن استفاده کنید و مانند GitLab پروژه خود را پیش ببرید. همچنین باید اکانت os_ta_fall2020@yahoo.com را به عنوان یکی از اعضا به مخزن خود اضافه کنید. در انتها کافی است در محل بارگذاری در سایت درس، آدرس مخزن و شناسه آخرین commit خود را آپلود کنید.
- توجه شود که پروژه های درس تک نفره اند و پروژه باید برای دانشجویان مهندسی کامپیوتر به صورت انفرادی و برای دانشجویان غیر مهندسی کامپیوتر در قالب گروه سه نفره انجام شود.
- نکاتی که در جلسه توجیهی و یا فروم درس مطرح می شود بخشی از صورت پروژه هستند لذا به شما توصیه می شود که حتما در جلسه توجیهی شرکت کنید.

- حتما log مورد نظر که شامل قطع و یا وصل شدن کلاینت و سرور یا سایر درخواست‌ها است را چاپ نمایید. در هنگام تحویل این log ها بخشی از نمره شما را تشکیل می‌دهند.
- پیاده‌سازی شما باید توسط فراخوانی‌های سیستمی مانند create, open, read, write و ... انجام شود و استفاده از توابع کتابخانه‌ای حتی کتابخانه‌ی استاندارد مانند fopenf و fprintf مجاز نیست. (توابعی که فراخوانی سیستمی محسوب می‌شوند را می‌توانید در لیست فراخوانی‌های سیستمی در بخش دوم لینوکس به آدرس <https://linux.die.net/man/> پیدا کنید).
- توابع کتابخانه‌ای که با فراخوانی‌های سیستمی قابل پیاده‌سازی نیستند مانند strcat, atoi و ... مجاز هستند.
- تنها توابعی که از فراخوانی سیستمی استفاده می‌کنند و نیازی به پیاده‌سازی آن‌ها نیست، free و malloc و realloc هستند.
- برای آشنایی با socket programming می‌توانید به صفحات زیر مراجعه کنید:

<https://beej.us/guide/bgnet/html/#client-server-background>

<https://beej.us/guide/bgnet/html/#system-calls-or-bust>

<https://beej.us/guide/bgnet/html/#broadcast-packetshello-world>

aylin.jamali@ut.ac.ir

saratvk1377@gmail.com

mahtaa.rafiiee@gmail.com