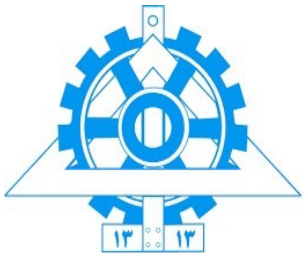

باسمه تعالی

سیستم‌های عامل

پروژه‌ی اول درس

مهلت تحویل: 15 فروردین 1400

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 پردازه 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 لایه های پایین‌تر سیستم عامل است که این ارتباط را ممکن می‌کند.

شرح پروژه:

در این پروژه قرار است یک سیستم «بازار کار آنلاین» را با استفاده از socket programming و فراخوانی‌های سیستمی زبان C پیاده‌سازی کنید.

نحوه‌ی انجام پروژه:

یک سرور داریم که می‌خواهد تعدادی پروژه را برای انجام دادن به کابران متخصص بسپارد. کاربران در قالب گروه‌های ۵ نفره برای هر پروژه‌ی موجود شرکت می‌کنند. سپس برای انجام آن پروژه پیشنهاد قیمت می‌دهند و کاربر با کمترین پشنهاد قیمت، آن پروژه را برنده شود. در نهایت سرور آن پروژه را ا ز لیست پروژه‌ها حذف می‌کند.

شرح کلی پروژه:

در این پروژه یک سرور مرکزی داریم که همواره روی پورت مشخصی (پورت X) گوش می‌کند و منتظر اتصال کلاینت‌ها (کاربران متخصص) است. سرور و هر کلاینت، یک پردازه هستند. هر کاربر پس از اتصال به سرور، لیست پروژه‌های موجود را مشاهده می‌کند (دقت کنید که گروه مربوط به پروژه‌های موجود در لیست هنوز تکمیل نشده و کاربران می‌توانند به آن بپیوندند). سپس برای شرکت در گروه مربوط به هر پروژه، شماره آن را به سرور اعلام می‌کند.

سرور وظیفه‌ی تشکیل گروه و اعلام برنده پروژه را دارد. برای تشکیل گروه، در ابتدا به ترتیب ورود کاربران به هر کاربر یک آیدی (برای تعیین نوبت) اختصاص می‌دهد و پس از تکمیل گروه، یک پورت broadcast به آن گروه اختصاص می‌دهد و فعالیت آن گروه را آغاز می‌کند. همچنین پس از اعلام نتیجه‌ی گروه هر پروژه، سرور آن را از لیست پروژه‌های موجود حذف می‌کند.

ارتباط بین سرور و هر کاربر از نوع TCP، و پس از شروع فعالیت گروه، ارتباط بین کاربران آن گروه از نوع UDP خواهد بود. پس از شروع فعالیت گروه، کاربران به نوبت قیمت پیشنهادی خود را روی پورت مشخص شده از سمت سرور در ابتدای تشکیل گروه، برای بقیه‌ی کاربران آن گروه می‌فرستند. هر کاربر 10 ثانیه وقت دارد تا قیمت پیشنهادی خود را اعلام کند. اگر این قیمت از قیمتی که تا الان تعیین شده بشتر باشد و یا پس از اتمام این مهلت هیچ پیشنهادی ندهد، نوبت به نفر بعدی می‌رسد.

در پایان اگر در یک دور کامل، تمام کاربران قیمت بالاتری از کمترین قیمت پیشنهادی، پیشنهاد دهند یا نوبنشان تمام شود، کاربر برنده شماره پروژه را به سرور اعلام می‌کند. سپس سرور آن گروه را می‌بندد و آن پروژه را از لیست خارج می‌کند.

تایمر:

برای اندازه‌گیری زمان نوبت هر کاربر، شما باید از signalهای unix و به طور دقیق‌تر، از سیگنال SIGALRM استفاده کنید.

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 فراخوانی‌های سیستمی blocking هستند، برای رفع این مشکل از فراخوان سیستمی select استفاده می‌کنیم. این فراخوان مسئول مانیتور کردن ارتباطات همزمان است و باعث می‌شود 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 آدرس‌های IP را localhost (127.0.0.1) در نظر بگیرید.
- با قرار دادن stdin در لیستی که به select می‌دهید، می‌توانید 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 بزنید و کدهایتان باید با gcc قابل کامپایل کردن باشند.
- انجام این پروژه به صورت **انفرادی** می‌باشد.
- برای تحویل پروژه می‌توانید یکی از دو روش زیر استفاده کنید:

۱. تمامی نتایج را در یک فایل فشرده شده با عنوان OS-CA1-<#SID>.zip در محل بارگذاری در سایت درس آپلود کنید.

۲. ابتدا یک مخزن خصوصی در سایت GitLab ایجاد نموده و سپس پروژه خود را در آن Push کنید. سپس اکانت UT_OS_TA را با دسترسی Maintainer به مخزن خود اضافه نمایید. کافی است در محل بارگذاری در سایت درس، آدرس مخزن و شناسه آخرین Commit را بارگذاری نمایید.

- حتما log مورد نظر که شامل قطع و یا وصل شدن کلاینت و سرور و 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 write، read، open، create و ... انجام شود و استفاده از توابع کتابخانه‌ای حتی کتابخانه‌های 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.)
- استفاده از توابع کتابخانه‌ای که با فراخوانی‌های سیستمی قابل پیاده‌سازی نیستند مانند atoi، strcat و ... مجاز است.
- 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/single/bgnet.html#clientserver
https://beej.us/guide/bgnet/html/single/bgnet.html#broadcast