



پروژه اول درس



سیستم های عامل - پاییز ۱۴۰۳

مهلت تحویل:

دانشکده مهندسی برق و کامپیوتر

طراحان تمرین:

استاد: دکتر مهدی کارگهی

سینا طبسی، سروش صادقیان

در این پروژه می‌خواهیم با فراخوانی سیستمی زبان C و یادگیری مبانی socket programming آشنا شویم.

سوکت چیست؟

سوکت یک مکانیزم برای برقراری ارتباط بین دو پردازش^۱ روی یک یا چند ماشین است. در این ارتباط دوطرفه، سوکت مثل یک پایانه می باشد که ما اطلاعات را به آن فرستاده یا از آن دریافت می نماییم. در حقیقت سوکت نوعی abstraction برای لایه‌های پایین تر سیستم عامل بوده که این ارتباط را ممکن می کند.

شرح پروژه

در این پروژه قصد داریم یک سامانه آموزش ساده تحت خط فرمان^۲ و با استفاده از socket programming و فراخوانی های سیستمی زبان C پیاده سازی کنیم.

^۱ Process

^۲ Command line

نحوه اجرای برنامه

در این پروژه می‌خواهیم سناریو انتقال درخواست‌های آموزشی میان کاربر (کلاينت) و سامانه آموزش (سرور) را پیاده‌سازی نماییم. روند کلی برنامه به این صورت می‌باشد که کاربران به سرور متصل شده و درخواست‌های آموزشی خود را از سامانه آموزش دریافت کرده و یا به آن ارسال می‌کنند. همچنین در صورتی که سامانه آموزش دسترس نباشد، کاربران باید با ناظران برای دریافت و یا ارسال درخواست‌ها ارتباط برقرار نمایند. در ادامه پروژه وظایف کلاينت و سرور به صورت مفصل توضیح داده شده است.

سامانه آموزش

سامانه آموزش همواره روی پورت مشخص A گوش می‌کند تا کاربران به آن متصل شوند. به طور کلی سامانه آموزش دارای دو وضعیت می‌باشد:

(1) آماده خدمت رسانی

(2) در حال تعمیر

در حالتی که سامانه آموزش در وضعیت "آماده خدمت رسانی" باشد، کاربران می‌توانند برای دریافت و ارسال درخواست‌هایشان با سامانه آموزش ارتباط برقرار کنند. سامانه آموزش نیز باید درخواست‌های دریافت شده از کاربران را به صورت فایل txt ذخیره نماید. توجه شود که این نوع ارتباط کاربر و سامانه آموزش از نوع TCP می‌باشد. در حالتی که سامانه آموزش در وضعیت "در حال تعمیر" باشد، کاربران موظف هستند که برای دریافت و ارسال درخواست‌های آموزشی با ناظران ارتباط برقرار نمایند. توجه شود که ارتباط میان کاربران و ناظران از نوع UDP و Broadcast می‌باشد. در خصوص

اطلاع یافتن از وضعیت سامانه آموزش، باید از روش Heartbeat استفاده شود. بدین صورت که سامانه آموزش هر یک ثانیه بر روی پورت B از طریق Signal یک پیام Heartbeat فرستاده تا به کاربران اطلاع بدهد در وضعیت "آماده خدمت رسانی" می باشد. توجه شود که ارتباط در روش Heartbeat به صورت Broadcast می باشد.

کاربر

کاربر پس از انتخاب نقش خود برای اطلاع پیدا کردن از وضعیت سامانه آموزش، باید پیام های ارسال شده بر روی پورت B را بررسی نمایند. در صورتی که وضعیت سامانه آموزش "آماده خدمت رسانی" باشد، کاربر با سامانه آموزش ارتباط TCP برقرار کرده و دو گزینه زیر برای او به نمایش در خواهد آمد:

(1) ارسال درخواست های جدید

(2) دریافت درخواست های موجود

با انتخاب مورد اول، کاربر درخواست آموزشی جدید خود را در سامانه آموزش ارسال می کند و سامانه آموزش موظف می باشد که درخواست آموزشی جدید را به صورت فایل txt ذخیره کرده و همچنین موضوع آن درخواست آموزشی را به لیست درخواست های موجود خود اضافه نماید. با انتخاب گزینه دوم، کاربر از میان درخواست های موجود در سامانه آموزش یک درخواست آموزشی را انتخاب می نماید. درخواست آموزشی انتخاب شده باید به صورت فایل txt همراه با مشخصات درخواست دهنده در سمت کلاینت ذخیره شود.

در صورتی که وضعیت سامانه "در حال تعمیر" باشد و کاربر به دنبال دریافت درخواست آموزشی مشخصی باشد، کاربر مورد نظر باید بر روی پورت A موضوع

درخواست آموزشی را به صورت Network Broadcast ارسال نماید تا ناظران با دریافت درخواست و بررسی آن و در صورت موجود بودن موضوع درخواست آموزشی مورد نظر در میان درخواست‌های ناظر، موضوع آن را در یک ارتباط مستقیم برای کاربر ارسال نماید. باید به این نکته توجه بشود که ممکن است ناظران در زمان‌های مختلف به Network Broadcast متصل شوند. در نتیجه لازم است که درخواست آموزشی کاربر با فاصله زمانی معینی بر روی پورت A تکرار شود تا ناظرانی که به تازگی بر روی پورت A در حال دریافت پیام‌ها می‌باشند، بتوانند درخواست کاربر را مشاهده نمایند. پس از آنکه کاربر درخواست آموزشی مورد نظر را از یکی از ناظران دریافت نمود، درخواست آموزشی به صورت Network Broadcast بر روی پورت A خاتمه می‌یابد.

ناظر

ناظر پس از انتخاب نقش خود همانند کاربر باید از وضعیت سامانه آموزش اطلاع پیدا کند. در صورتی که وضعیت سرور "آماده خدمت رسانی" باشد، ناظر با سامانه آموزش ارتباط TCP برقرار کرده و پس از آن گزینه "دریافت درخواست‌های موجود" به اون نمایش داده می‌شود. پس از انتخاب این گزینه، لیست درخواست‌های موجود در سامانه آموزش برای ناظر نمایش داده می‌شود و ناظر می‌تواند با انتخاب این درخواست‌ها آن را به لیست درخواست‌های موجود خود اضافه نماید. در صورتی که وضعیت سرور "در حال تعمیر" باشد، ناظر موظف است پورت A یا همان Network Broadcast را بررسی نماید تا به درخواست کاربران پاسخ دهد.

درخواست آموزشی

هر درخواست آموزشی شامل موضوع، نام و نام خانوادگی فرستنده، شماره دانشجویی، تعداد واحدهای گذرانده و تعداد واحدهای اخذ شده در این ترم می باشد. اطلاعات خواسته شده باید توسط کاربر فراهم شده و برای سامانه آموزش ارسال نماید. همچنین در صورتی که کاربر از سامانه آموزش موضوع درخواست آموزشی را دریافت کرد، لازم است که اطلاعات دیگر خواسته شده را فراهم کرده و آن درخواست را ذخیره نماید.

همزمانی سیستم

در کل طول اجرای برنامه، سرور باید بتواند به طور همزمان به چندین کلاینت و درخواست‌های آن‌ها رسیدگی کند، ولی برخی از سیستم‌ها حالت blocking دارند و اجرای برنامه آن‌جا متوقف می‌شود. برای حل این مشکل از سیستم‌ها select استفاده می‌کنیم. این سیستم‌ها می‌تواند ارتباطات و I/O ها را بدون بلاک کردن مدیریت کند. در این پروژه هم باید به کمک سیستم‌ها select، تمام I/O باید بدون اینکه روند اجرای برنامه بلاک شود انجام شوند.

نکات تکمیلی

- فرمت و محتوای نمایش داده شده برای هریک از انواع پیام‌ها به عهده خودتان است و هر نوع نمایش معقولی که محتوای پیام‌های رد و بدل شده به درستی نمایش دهد قابل قبول است.

- ارتباط هر کلاینت با سرور از نوع TCP بوده و ارتباط کلاینت‌ها با هم از نوع UDP و broadcast خواهد بود. برای آشنایی با ارتباطات TCP و UDP می‌توانید از لینک‌های قرار گرفته در قسمت "نکات پایانی" استفاده کنید.
- تضمین می‌شود که تنها یک کاربر در حال فرستادن درخواست Network Broadcast بر روی پورت A در یک زمان مشخص باشد. بنابراین نیازی به رسیدگی به سناریوهای دیگر نمی‌باشد.
- تضمین می‌شود که تنها یک ناظر در حال فرستادن فایل مورد نظر برای کاربر می‌باشد. بنابراین نیازی به رسیدگی به سناریوهای دیگر نمی‌باشد.
- می‌توانید فرض کنید موضوع درخواست‌های آموزشی یکتا هستند. بنابراین نیازی به چک کردن این موارد نمی‌باشد.
- توجه داشته باشید که درخواست‌های آموزشی به صورت فایل در ارتباط‌های TCP و UDP انتقال پیدا **نمی‌کند**. بنابراین لازم به پیاده‌سازی سیستم دریافت و بارگذاری فایل در سطح سوکت نمی‌باشد.
- تمامی آدرس‌های IP را localhost یا همان 127.0.0.1 در نظر بگیرید.
- با قرار دادن stdin در لیستی که به select می‌دهید می‌توانید بدون بلاک شدن از کنسول ورودی بخوانید.
- کلاینت و سرور باید به این شکل اجرا شوند:

`./server <server_port>`

`./client heartbeat_port_B broadcast_port_A client_port_M`

نکات پایانی

- در این پروژه کدهایتان باید به زبان C و با gcc قابل کامپایل شدن باشد.

- توجه داشته باشید که پروژه های درس تک نفره اند.
 - در حین اجرای برنامه log های مناسبی مانند وصل شدن کلاینت یا درخواست ها چاپ کنید تا روند اجرای برنامه مشخص باشد. این log ها هنگام تحویل بخشی از عملکرد کد را نشان می دهند.
 - پیاده سازی شما باید به کمک سیستم کال ها مانند read, write, open و ... باشد و استفاده از توابع کتابخانه ای مانند fopen مجاز نمی باشد. توابعی که سیستم کال هستند در <https://linux.die.net/man/2> قابل مشاهده می باشند.
 - توابع کتابخانه ای که توسط سیستم کال ها قابل پیاده سازی نیستند مانند atoi, strcat, strcpy, sprintf و ... مجاز می باشند.
 - برای آشنایی با برنامه نویسی سوکت می توانید از منابع زیر و ویدیوهایی که در سایت درس قرار داده شده استفاده کنید.
- <https://www.geeksforgeeks.org/tcp-server-client-implementation-in-c/>
- <https://www.educative.io/answers/how-to-implement-udp-sockets-in-c>
- <https://beej.us/guide/bgnet/html/#client-server-background>
- <https://beej.us/guide/bgnet/html/#system-calls-or-bust>
- <http://beej.us/guide/bgnet/html/#broadcast-packetshello-world>
- فایل نهایی که تحویل می دهید باید شامل موارد زیر باشد:
 - فایل کد سرور
 - فایل کد کلاینت
 - Makefile (در صورت وجود)

این فایل ها در قالب یک فایل فشرده zip با نام OS_CA1_<SID>.zip در صفحه درس آپلود کنید.

- در صورتی که سوالی داشتید می توانید از طریق فروم درس در ایلرن و یا ایمیل به دستیاران آموزشی پروژه سوال خود را بپرسید.

➤ sinatabassi80@gmail.com

➤ sadeqiansoroosh@gmail.com