پروژه اول درس

دانشکده مهندسی برق و کامپیوتر | **سیستم عامل، بهار ۱۴۰۲**

استاد:	**مهلت تحویل:**	**طراحان:**
دکتر مهدی کارگهی	**یکشنبه ۲۱ اسفند**	**محمدطاها فخاریان - امیرمهدی جودی**

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[1] 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 سیستم‌عامل است که این ارتباط را ممکن می‌کند.

شرح پروژه:

در این پروژه می‌خواهیم یک سامانه پرسش و پاسخ درسی بین دانشجویان و دستیاران آموزشی، تحت خط فرمان[2] با استفاده از socket programming و فراخوانی‌های سیستمی زبان c پیاده‌سازی کنیم.

---

[1] process
[2] Command line

نحوه اجرای برنامه:

در این پروژه یک سرور مرکزی داریم که وظیفه برقراری ارتباط بین دانشجویان و دستیاران آموزشی را دارد. این سرور همواره روی پورت مشخصی گوش می‌کند تا کلاینت‌ها (شامل دانشجویان و دستیاران آموزشی) به آن متصل شوند. دانشجویان می‌توانند به عنوان کلاینت به سرور وصل شوند و به او اعلام کنند که قصد پرسیدن سوال از دستیاران آموزشی را دارند یا به سایر جلسات پرسش و پاسخ دسترسی داشته باشند و دستیاران آموزشی می‌توانند لیست پرسش‌ها را از سرور درخواست کرده و با انتخاب خود، به پرسش‌ها جواب دهند. توجه کنید که سرور یک پردازه[3] و هر کلاینت یک پردازه جدا است.

هر کلاینت پس از اتصال به سرور باید مشخص کند که دانشجو است یا دستیار آموزشی. دانشجو پس از ورود به سیستم، سوال خود را وارد می‌کند و برای سرور می فرستد و منتظر می ماند تا یک دستیار آموزشی پرسش او را پاسخ دهد. دستیار آموزشی پس از ورود به سیستم، درخواست لیست پرسش‌ها را می دهد. پس از انتخاب یک پرسش، سرور یک پورت جدید به آن دو نفر اعلام می‌کند تا آن‌ها روی آن پورت بتوانند با هم دیگر ارتباط داشته باشند. ارتباط هر کلاینت با سرور از نوع TCP است و پس از شروع جلسه، ارتباط کلاینت‌ها با هم از نوع UDP و broadcast خواهد بود.

هر پرسش سه وضعیت دارد: در انتظار پاسخ، درحال پاسخ، پاسخ داده شده.

- پرسش‌های در انتظار پاسخ توسط هیچ دستیار آموزشی انتخاب نشده‌اند.
- پرسش‌های در حال پاسخ در یک جلسه بین دانشجو و دستیار آموزشی در حال بحث می‌باشند.
- پرسش‌های پاسخ داده شده توسط یک دستیار آموزشی قبلاً جواب داده شده‌اند.

[3] process

هر دستیار آموزشی می‌تواند به لیست پرسش‌ها دسترسی داشته باشد. این لیست شامل پرسش های در انتظار پاسخ است و پرسش‌های دیگر برای دستیار آموزشی نمایش داده نمی‌شود. پس از اتمام جلسه، دانشجو وضعیت نهائی جلسه شامل پاسخ صحیح را برای سرور می‌فرستد.

سرور یک فایل برای جمع‌آوری پاسخ پرسش‌ها دارد و وقتی نتیجه یک جلسه را دریافت کرد، پرسش و پاسخ صحیح آن جلسه را به پایان این فایل اضافه می‌کند.

اگر یک دانشجو به سرور وصل شود و بگوید قصد شرکت در جلسه‌ای را دارد، سرور یک لیست از پورت جلسات در حال برگزاری (پرسش های در حال پاسخ) را برای او ارسال می کند. دانشجو یک پورت را انتخاب کرده و به پیام های برادکست دانشجو و دستیار آموزشی گوش می‌دهد و جلسه را دنبال کند. توجه کنید که پیام های دستیار و دانشجو از این لحظه باید برای کلاینت جدید نمایش داده شود.

تایمر:

در هر جلسه دستیار برای پاسخ دادن یک دقیقه زمان دارد. اگر یک دقیقه بگذرد و دستیار پاسخی ندهد، جلسه تمام شده و پرسش مجددا به حالت در انتظار پاسخ در می آید.

برای پیاده‌سازی تایمر باید از سیگنال SIGALRM و سیستم‌کال alarm استفاده کنید.

هم‌زمانی سیستم:

در کل طول اجرای برنامه، سرور باید بتواند به طور هم‌زمان به چندین کلاینت و درخواست‌های آن‌ها رسیدگی کند، ولی برخی از سیستم‌کال‌ها حالت blocking دارند و اجرای برنامه آن‌جا متوقف می‌شود. برای حل این مشکل از سیستم‌کال select استفاده می‌کنیم. این سیستم‌کال می‌تواند ارتباطات و I/O ها را بدون بلاک کردن مدیریت کند.

در این پروژه هم باید به کمک سیستم‌کال select، تمام I/O ها باید بدون اینکه روند اجرای برنامه بلاک شود انجام شوند.

نکات مهم:

- 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.
- میتوانید فرض کنید پرسش های مطرح شده یکتا هستند. بنابراین نیازی به چک کردن این موارد نیست.
- همزمان چندین جلسه مختلف می‌تواند در جریان باشد.
- هر دستیار آموزشی یا دانشجو فقط در یک جلسه می تواند باشد.
- 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 <server_port>

نکات پایانی:

- در این پروژه کدهایتان باید به زبان c باشد و با gcc قابل کامپایل شدن باشد.
- توجه کنید که پروژه‌های درس تک نفره‌اند.
- در حین اجرای برنامه logهای مناسبی مانند وصل شدن کلاینت یا درخواست‌ها چاپ کنید تا روند اجرای برنامه مشخص باشد. این logها هنگام تحویل بخشی از عملکرد کد را نشان می‌دهند.

- پیاده‌سازی شما باید به کمک سیستم‌کال‌ها مانند read, write, open و ... باشد و استفاده از توابع کتابخانه‌ای مانند fopen مجاز نیست. توابعی که سیستم‌کال هستند در https://linux.die.net/man/2 قابل مشاهده‌اند.
- توابع کتابخانه‌ای که توسط سیستم‌کال‌ها قابل پیاده‌سازی نیستند مانند atoi, strcat, strcpy, sprintf و ... مجاز هستند.
- برای آشنایی با برنامه‌نویسی سوکت می‌توانید از منابع زیر و ویدیوهایی که در سایت درس قرار داده شده استفاده کنید.

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 می‌دهید 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 می‌توانید از طریق فروم درس در ایلرن و یا ایمیل به دستیاران آموزشی پروژه سوال خود را بپرسید.

taha.fakharian@gmail.com

a.m.joudi1799@ut.ac.ir