پروژه اول درس

دانشکده مهندسی برق و کامپیوتر | **سیستم عامل، بهار ۱۴۰۱**

استاد: | **مهلت تحویل:** | **طراحان:**

دکتر مهدی کارگهی | **یکشنبه ۱۴ فروردین** | پارسا کوت‌زری - علی زارع

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 بازی دوز (X-O) تحت خط فرمان[2] همراه با تماشاچی با استفاده از socket programming و فراخوانی‌های سیستمی زبان c پیاده‌سازی کنیم.

برای یک نمونه اجرا و آشنایی با قوانین این بازی می‌توانید از لینک زیر استفاده کنید:

https://playtictactoe.org/

---

[1] process
[2] Command line

نحوه اجرای برنامه:

در این پروژه یک سرور مرکزی داریم که وظیفه ساخت اتاق بازی برای بازیکنان و تماشاچیان را بر عهده دارد. این سرور همواره روی پورت مشخصی گوش می‌کند تا کلاینت‌ها به آن متصل شوند. افراد می‌توانند به عنوان کلاینت به سرور وصل شوند و به او اعلام کنند که قصد بازی کردن یا تماشای بازی‌های در حال اجرا را دارد. توجه کنید که سرور یک پردازه[3] و هر کلاینت یک پردازه جدا است.

بازی دوز دونفره است. به محض اینکه دو نفر به سرور اعلام آمادگی برای بازی کردند، سرور یک پورت جدید به آن دو نفر اعلام می‌کند تا آن‌ها روی آن پورت بتوانند با هم دیگر ارتباط داشته باشند و بازی کنند. ارتباط هر کلاینت با سرور از نوع TCP است و پس از شروع بازی، ارتباط کلاینت‌ها با هم از نوع UDP و broadcast خواهد بود.

بازیکنان به نوبت حرکت خود را انجام می‌دهند تا بازی تمام شود. در پایان، نتیجه بازی و وضعیت نهایی صفحه توسط یکی از بازیکنان به سرور اعلام می‌شود و برنامه کلاینت‌ها به پایان می‌رسد.

سرور یک فایل برای جمع‌آوری نتایج بازی‌ها دارد و وقتی نتیجه یک بازی را دریافت کرد نتیجه بازی را (که شامل وضعیت نهایی صفحه بازی است) به پایان این فایل اضافه می‌کند.

اگر یک کلاینت به سرور وصل شود و بگوید قصد تماشای بازی‌ها را دارد، سرور به او لیستی از پورت بازی‌های در حال اجرا ارسال می‌کند. کلاینت یکی از آن‌هارا انتخاب می‌کند. حالا می‌تواند روی آن پورت به پیام‌های برادکست بازیکنان گوش دهد و بازی را دنبال کند. توجه کنید که صفحه بازی باید به درستی برای تماشاچیان نشان داده شود.

[3] process

تایمر:

هر بازیکن برای حرکت خود یک دقیقه زمان دارد.اگر یک دقیقه بگذرد و بازیکن حرکتی نکند نوبتش رد می‌شود. 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:

- شکل ورودی برای حرکات و شیوه نشان دادن صفحه بازی در ترمینال به دلخواه خودتان است و هر روش معقولی مورد قبول است.
- همزمان چندین بازی مختلف می‌تواند در جریان 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.

pkootzari@gmail.com

alizare1@ut.ac.ir