



## پروژه اول درس



سیستم عامل، پاییز 1400

دانشکده مهندسی برق و کامپیوتر

طراحان:

مهلت تحویل:

استاد:

علی زارع - سارا توکلی

یکشنبه ۲ آبان

دکتر مهدی کارگهی

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<sup>1</sup> 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 سیستم پرسش و پاسخ برای رشته‌های مختلف درسی (مشابه stackoverflow) با استفاده از socket programming و فراخوانی‌های سیستمی زبان C پیاده‌سازی کنیم.

---

<sup>1</sup> process

## نحوه اجرای برنامه:

در این پروژه یک سرور مرکزی داریم که برای تعدادی رشته تحصیلی (کامپیوتر-برق-عمران-مکانیک) اتاق‌هایی برای پرسیدن سوالات می‌تواند تشکیل می‌دهد. این سرور همواره روی پورت مشخصی گوش می‌کند تا کلاینت‌ها به آن متصل شوند. افراد می‌توانند به عنوان کلاینت به سرور وصل شوند و شماره (یا نام) رشته‌ای که می‌خواهند وارد آن شوند و سوال بپرسند را به سرور بدهند. توجه کنید که سرور یک پردازش و هر کلاینت یک پردازش جدا است.

ظرفیت هر اتاق ۳ نفر است. به محض اینکه تعداد افراد یک اتاق به حد نصاب برسد سرور یک پورت جدید به افراد آن اتاق اعلام می‌کند تا آن‌ها روی آن پورت بتوانند پیام‌ها را برادکست کنند. ارتباط هر کلاینت با سرور از نوع TCP است و پس از پر شدن یک اتاق، ارتباط کلاینت‌ها با هم از نوع UDP و broadcast خواهد بود.

کلاینت‌ها به نوبت می‌توانند سوال بپرسند و این سوال به همه افراد موجود در اتاق برادکست می‌شود و بقیه کاربرها می‌توانند به نوبت به این سوال جواب بدهند. در انتها بعد از این که همه به سوال جواب دادند فردی که سوال پرسیده بهترین جواب را به همه افراد اتاق اعلام می‌کند و سپس پیامی شامل سوال خود و جواب‌های بقیه را به سرور ارسال می‌کند.

سرور یک فایل برای جمع‌آوری همه سوال‌ها و جواب‌های آن دارد و رشته پیام دریافت‌شده توسط کسی که سوال پرسیده را در انتهای این فایل اضافه می‌کند.

بعد از اینکه هر نفر سوال خود را پرسید و یک دور سوال پرسیدن به پایان رسید کار آن اتاق به پایان می‌رسد و برنامه کلاینت به انتها می‌رسد.

## تایمر:

هر کلاینت برای پاسخ‌دادن به یک سوال یک دقیقه زمان دارد. اگر جواب را نداند می‌تواند واژه pass را برادکست کند تا نوبت به نفر بعدی برسد. اگر یک دقیقه بگذرد و کلاینت چیزی نگوید نوبتش رد می‌شود.

برای پیاده‌سازی تایمر باید از سیگنال SIGALRM و سیستم کال 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:

- همزمان چندین اتاق مربوط به یک رشته می‌تواند وجود داشته باشد. برای مثال ۳ نفر برای رشته کامپیوتر به سرور درخواست می‌دهند و از سرور پورت دریافت می‌کنند. حین این که افراد قبلی در حال پرسش پاسخ هستند، افراد جدیدی می‌توانند به سرور وصل شوند و درخواست رشته کامپیوتر کنند و وقتی حد نصاب رسیدند اتاق جدیدی تشکیل می‌شود.
- کسی که سوال پرسیده هنگام ارسال جواب‌ها به سرور، بهترین جواب‌را به نحو دلخواه (مثلا با یک علامت \*) مشخص کند.
- سرور تمام سوالات مربوط به یک رشته را در یک فایل می‌ریزد و به ازای هر اتاق یک فایل جدید نمی‌سازد.
- 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.

[saratvk1377@gmail.com](mailto:saratvk1377@gmail.com)

[alizarel@ut.ac.ir](mailto:alizarel@ut.ac.ir)