



## باسمه تعالی سیستم‌های عامل پروژه‌ی اول درس تاریخ تحویل



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 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 سیستم عامل است که این ارتباط را ممکن می‌کند.

### شرح پروژه:

در این پروژه قرار است سیستم «مزایده بر روی تعدادی کالا» را با استفاده از socket programming و فراخوانی‌های سیستمی زبان C پیاده‌سازی کنید.

### نحوه‌ی انجام مزایده:

یک سرور داریم که می‌خواهد تعدادی کالا را به مزایده بگذارد. افراد در قالب گروه‌های 5 نفره در مزایده‌ی یک کالا شرکت می‌کنند و فرد با بالاترین پیشنهاد قیمت، آن مزایده را برنده می‌شود. سپس سرور آن کالا را از لیست کالاهای حذف می‌کند.

### شرح کلی مزایده:

در این مزایده یک سرور مرکزی داریم که همواره روی پورت مشخصی (پورت X) گوش می‌کند و منتظر اتصال کلاینت‌ها (خریداران) است. سرور و هر کلاینت، یک پردازنده هستند. هر کاربر پس از اتصال به سرور، لیست کالاهای موجود را مشاهده می‌کند (دقت کنید که گروه مزایده مربوط به کالاهای موجود در لیست هنوز تکمیل نشده و مزایده مربوط به آن‌ها آغاز نشده است). سپس شماره کالای مورد نظر را، برای شرکت در مزایده مربوط به آن، به سرور اعلام می‌کند. سرور وظیفه‌ی تشکیل گروه و اعلام برنده مزایده را دارد. به ترتیب ورود کاربران، سرور به هر نفر یک آیدی اختصاص می‌دهد و پس از تکمیل گروه مزایده، سرور یک پورت broadcast به آن گروه اختصاص می‌دهد و با تعیین شروع کننده‌ی مزایده، مزایده را آغاز می‌کند. همچنین پس از اعلام نتیجه‌ی مزایده، سرور کالای فروش رفته را از لیست کالاهای موجود حذف می‌کند.

ارتباط بین سرور و هر کاربر از نوع TCP، و پس از شروع مزایده، ارتباط بین کاربران یک مزایده از نوع UDP خواهد بود.

پس از شروع مزایده، خریداران مزایده را آغاز میکنند و به نوبت قیمت پیشنهادی خود را روی پورت مشخص شده از سمت سرور در ابتدای مزایده، برای بقیه‌ی خریداران می‌فرستند. هر خریدار 10 ثانیه وقت دارد تا قیمت پیشنهادی خود را اعلام کند. اگر این قیمت از قیمتی که تا الان تعیین شده کمتر بود، قیمت کالا تغییری نمیکند و همچنین اگر پس از اتمام این مهلت هیچ پیشنهادی ندهد، نوبت به نفر بعدی میرسد. در پایان اگر در یک دور کامل، تمام افراد قیمت پایین تری پیشنهاد دادند یا نوبتشان تمام شد، برنده پایان مزایده را به سرور اعلام میکند. سپس سرور آن مزایده را میبندد و آن کالا را از لیست خارج میکند.

### تایمر:

برای اندازه‌گیری زمان نوبت هر خریدار، شما باید از signal های 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 فراخوان های 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 آدرس های IP را 127.0.0.1 (localhost) در نظر بگیرید.
- با قرار دادن stdin در لیستی که به select می دهید، می توانید 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 بنویسید و کدهایتان باید با gcc قابل کامپایل کردن باشند.
- برای تحویل پروژه می توانید از یکی از دو روش زیر استفاده کنید:
  - o تمامی نتایج را در یک فایل فشرده شده با عنوان OS-CA1-[#SID](#).zip در محل باگذاری در سایت درس آپلود کنید.
  - o رای تحویل پروژه ابتدا یک مخزن خصوصی در سایت GitLab ایجاد نموده و سپس پروژه خود را در آن Push کنید. سپس اکانت TA\_OS\_UT را با دسترسی Maintainer به مخزن خود اضافه نمایید. کافی است در محل بارگذاری در سایت درس، آدرس مخزن، شناسه آخرین Commit و گزارش پروژه را بارگذاری نمایید.
- حتما log مورد نظر که شامل قطع و یا وصل شدن کلاینت و سرور یا سایر درخواست ها است را چاپ نمایید. در هنگام تحویل این log ها بخشی از نمره شما را تشکیل می دهند.
- پیاده سازی شما باید توسط فراخوانی های سیستمی مانند create، open، read، write و ... انجام شود و استفاده از توابع کتابخانه ای حتی کتابخانه ی استاندارد مانند fopenf و fprintf مجاز نیست. (توابعی که فراخوانی سیستمی محسوب می شوند را می توانید در لیست فراخوانی های سیستمی در بخش دوم لینوکس به آدرس <https://linux.die.net/man> پیدا کنید.)
- توابع کتابخانه ای که با فراخوانی های سیستمی قابل پیاده سازی نیستند مانند atoi، strcat و ... مجاز هستند.
- تنها توابعی که از فراخوانی سیستمی استفاده می کنند و نیازی به پیاده سازی آن ها نیست، free و malloc و realloc هستند.
- برای آشنایی با socket programming می توانید 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>