



# باسمه تعالی

## سیستم های عامل

### پروژه ی اول

تاریخ تحویل: ۲۷ مهر



در این پروژه قرار است شما یک سیستم اشتراک گذاری فایل<sup>۱</sup> را با استفاده از socket programming و فراخوانی های سیستمی به زبان C پیاده سازی کنید.

هدف از انجام این پروژه آشنایی با فراخوانی های سیستمی زبان C و یادگیری مبانی socket programming می باشد.

### شرح کلی سیستم اشتراک گذاری فایل:

هدف کلی این سیستم، ایجاد قابلیت به اشتراک گذاشتن فایل برای کلاینت های سیستم می باشد. کلاینت ها باید بتوانند فایل های خود را (برای سرور یا کلاینت های دیگر) ارسال کنند و فایل های درخواستی شان را در صورت وجود دریافت کنند.

فایل های مورد نظر از سرور دریافت می شود و در صورت نبود سرور در شبکه، فایل ها مستقیماً بین کلاینت ها تبادل می شود. در واقع می خواهیم سیستم خود را به گونه ای طراحی کنیم که در صورت در دسترس نبودن سرور نیز کاربران بتوانند با یکدیگر فایل به اشتراک بگذارند.

توجه شود که برای اطلاع از وجود و صحت کارکرد سرور، باید از روش Heartbeat استفاده شود. بدین صورت که سرور هر یک ثانیه از طریق Signal یک پیام Heartbeat به نشانه ی زنده بودن روی پورت X<sup>۲</sup> برادکست می کند. محتوای این پیام شماره پورت Y است که سرور روی آن listen می کند و منتظر اتصال کلاینت هاست. بدین صورت هر یک از کلاینت ها از وجود سرور مطلع می شوند.

### در شرایط وجود سرور:

در این سناریو، سرور روی پورت مشخصی (پورت Y) listen می کند و منتظر اتصال کلاینت هاست. هر کلاینت پس از اتصال به سرور مرکزی، امکان آپلود یا دانلود فایل های مورد نظر خود را با وارد کردن دستورات متناظر دارد.

برای آپلود فایل، کلاینت باید از دستور upload استفاده و نام فایل را مشخص کند. پس از آپلود یک فایل، این فایل از این پس در فایل های سرور وجود دارد و کلاینت های دیگر می توانند با مشخص کردن اسم فایل، آن را از سرور درخواست و دانلود کنند.

<sup>۱</sup> File sharing

<sup>۲</sup> کلاینت ها پورت X را از طریق ورودی برنامه در اختیار دارند

برای دانلود نیز به همین صورت نام فایل مورد نظر در کنار دستور download باید مشخص شود. سرور به دنبال فایل با این نام می گردد و در صورت وجود، آن را برای کلاینت ارسال می کند. در صورتی که فایلی با این نام یافت نشود، باید اطلاع رسانی مناسب به سمت کلاینت انجام شود تا جست و جو برای فایل از طریق سناریوی دوم (نظیر به نظیر) صورت گیرد.

#### در شرایط نبود سرور (نظیر به نظیر):

همانطور که گفته شد می خواهیم سیستم خود را به گونه ای طرح کنیم که در شرایطی که سرور ما قادر به فعالیت نبود و یا در شبکه حضور نداشت هم کاربران قادر به اشتراک گذاشتن فایل باشند. برای این کار کاربر A در صورت در دسترس نبودن سرور، نام فایل درخواستی خود را با استفاده از network broadcast روی پورت Y می فرستد و به این صورت کاربران دیگر که در شبکه حضور دارند از درخواست این کاربر مطلع می شوند. در این شرایط اگر کاربری این فایل را در اختیار دارد، با اتصال مستقیم به کاربر A، فایل را برایش می فرستد و کاربر A برادکست کردن نام فایل درخواستی را متوقف می کند. در غیر این صورت کاربر A می تواند با یک فاصله ی زمانی مناسب دوباره درخواست خود را برادکست کند چرا که ممکن است کاربر جدیدی به سیستم اضافه شود و یا فایل به تازگی روی یکی از نظیر ها قرار گرفته باشد.

در اینجا پیاده سازی مکانیزم صحیح اتصال مستقیم دو کاربر و انتخاب یک کاربر به عنوان فرستنده از میان کاربر های دارای فایل بر عهده شماست. هر روش صحیحی جهت پیاده سازی این سناریو، قابل قبول است. دقت کنید پیاده سازی را به نحوی انجام دهید که بقیه ی کاربرانی که فایل را در اختیار دارند اقدام به فرستادن چندباره ی فایل به کاربر A نکنند و فایل درخواستی تنها یک بار توسط یک کاربر فرستاده شود.

همانطور که گفته شد در صورتی که سرور در دسترس نباشد فایل از این طریق دریافت می شود. اما توجه شود در سناریویی که سرور در دسترس است اما فایل درخواستی روی سرور قرار ندارد نیز همین روند تکرار خواهد شد و به روش نظیر به نظیر فایل دریافت می شود. چرا که ممکن است فایل های نظیرها هنوز روی سرور آپلود نشده باشند.

#### نکات مهم:

- فرض کنید برنامه کلاینت هر کاربر منتظر اتمام اجرای هر دستور می ماند و نیازی به پیاده سازی قابلیت ارسال و دریافت همزمان چند فایل توسط یک کاربر نیست.
- 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 localhost (127.0.0.1) 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 heartbeat_port_X
```

```
./client heartbeat_port_X broadcast_port_Y client_port_M
```

## نکات پایانی:

- در این پروژه باید به زبان C کد بزنید و کدهایتان باید با gcc قابل کامپایل کردن باشد.
- توجه شود که پروژه های درس تک نفره اند و پروژه باید به صورت انفرادی انجام شود.
- نکاتی که در جلسه توجیهی و یا فروم درس مطرح می شود بخشی از صورت پروژه هستند لذا به شما توصیه می شود که حتما در جلسه توجیهی شرکت کنید.
- حتما log مورد نظر که شامل قطع و یا وصل شدن کلاینت و سرور یا سایر درخواست هاست 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 استاندارد مانند fopenf و fprintf مجاز نیست. (توابعی که فراخوانی سیستمی محسوب می شوند را می توانید در لیست فراخوانیهای 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 سیستم کال 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>