



مقدمه

در این پروژه قصد داریم شما را با پیاده سازی دو نرم افزار آشنا کنیم. اولین نرم افزار یک ماشین حساب توزیع شده است که محاسبات سرویس گیرنده را بر روی سرویس دهنده انجام داده و نتایج را باز می گرداند. نرم افزار دوم یک سیستم توزیع فایل P2P است.

بخش اول: ماشین حساب توزیع شده

این نرم افزار می تواند محاسبات سرویس گیرنده را بر روی سرویس دهنده انجام دهد. این نرم افزار در مواقعی کاربرد دارد که حجم محاسبات بالا بوده و سیستم سرویس گیرنده قادر به انجام آن ها نیست و یا در مواقعی که نرم افزار فقط قابلیت نصب بر روی یک سیستم را داشته و مجموعه ای از سرویس گیرندگان می خواهند از آن سرویس بگیرند. به عنوان مثال نرم افزارهایی مانند متلب این ویژگی را دارند .

برای سادگی فرض میکنیم این ماشین حساب عملیات های ساده را انجام می دهد. مدل ارتباط این نرم افزار به صورت سرویس گیرنده-سرویس دهنده است. این نرم افزار از پروتکل ارتباطی زیر استفاده میکند :

Calculation Request Format:

\$ operator \$ op1 \$ op2 \$

که در آن operator عملیاتی می باشد که قرار است در سمت سرویس دهنده انجام شود. همانطور که بیان شد این نرم افزار از عملیات های ساده ریاضی پشتیبانی می کند پس مقدار این فیلد یکی از حالت های زیر است :

- Add
- Subtract
- Divide
- Multiply
- Sin
- Cos
- Tan
- Cot

فیلدهای op1 و op2 عملوندهای عملگر مشخص شده هستند. دقت کنید که با توجه به عملگر انتخاب شده ممکن است op2 موجود نباشد .

Calculation Response Format:

\$ calculation \$time \$result

که در آن calculation time زمانی است که در سمت سرویس دهنده برای انجام محاسبه سپری شده است. دقت کنید که این زمان شامل زمانی که برای پارس پیام و .. صرف شده نیست. فیلد result حاصل عملیات خواسته شده بر روی عملوندهای داده شده است .

دقت داشته باشید که پیاده سازی این قسمت در واقع شامل دو نرم افزار سرویس دهنده و سرویس گیرنده است و می بایست هر دوی آن ها را پیاده سازی کنید.

بخش دوم: سیستم توزیع فایل P2P

این نرم افزارها که شاید یکی از معروف ترین ان ها نرم افزار Torrent باشد، فایل ها را بین کاربران توزیع می کنند و با پیروی نکردن از مدل سرویس دهنده و سرویس گیرنده دیگر فایل ها تنها از تعدادی نقطه مشخص پخش نمی شوند. با توجه به این موضوع احتمال رهگیری منشا فایل کمتر می شود و از آن برای توزیع فایل به صورت غیرقانونی نیز استفاده می گردد .

برای سادگی شما تنها قسمت توزیع فایل را پیاده سازی خواهید کرد به این ترتیب که یک نرم افزار پیاده سازی می شود که توانایی توزیع کردن یک فایل یا دریافت آن را دارد، در حالت توزیع کردن با دریافت تقاضا فایل را با استفاده از پروتکل UDP برای متقاضی ارسال می کند و در حالت دریافت فایل، می تواند برای آن فایل تقاضای همه پخشی تولید کند. برای مثال :

```
p2p -receive hello.txt
```

با دستور برنامه ی شما در حالت دریافت فایل اجرا شده و تقاضا فایل hello.txt به صورت همه پخشی برای همه، در شبکه ارسال میکند. اگر یکی از گره ها به صورت مثال با دستور زیر اجرا شده باشد :

```
p2p -server -filepaths c:\FileStore
```

با دریافت تقاضای فایل hello.txt پاسخ را که فایل موردنظر است به صورت مستقیم برای متقاضی ارسال می کند. دقت کنید متقاضی می بایست بر روی همان آدرس آی پی و پورتی که با آن تقاضا را ارسال کرده است در حال گوش دادن باشد. شما می بایست برای زمانی که نرم افزار شما در حال توزیع فایل قرار دارد یک پورت مشخص برای آن در نظر بگیرید که انتخاب این پورت به عهده ی شما خواهد بود. تقاضا دهنده ها می توانند از پورت های تصادفی یا یک پورت مشخص استفاده نمایند. از انجایی که سرویس دهنده از روی تقاضای ارسالی پاسخ می دهد بین این دو حالت تفاوتی وجود نداشته و انتخاب آن برعهده شما خواهد بود.

دقت داشته باشید برای این سیستم شما تنها یک نرم افزار طراحی می کنید. این نرم افزار هر دو حالت دریافت و ارسال فایل را پشتیبانی می کند و می تواند در حالتی که کاربر از او می خواهد اجرا شود. دقت کنید که نیازی نیست هر دوی این حالت ها به صورت هم زمان در حال اجرا باشند .

پروتکل این نرم افزار به شکل زیر می باشد:

File Request:

filename

که در آن فیلد filename اسم فایلی است که برای آن تقاضا می دهیم.

از آنجایی که بسته های UDP محدودیت دارند، شما نیاز دارید که یک اندازه ی مشخص برای بسته ها را در نظر گرفته و بسته های خود را با آن اندازه ارسال کنید، در صورتی که اندازه ی فایل از اندازه ی بسته ی شما بزرگتر باشد می بایست آن را قطعه قطعه و ارسال نمایید، در نهایت در سمت متقاضی می بایست آن را به صورت یکپارچه در قالب یک فایل ذخیره کنید. با توجه به این توضیحات برای مثال اگر بسته های خود را ۱۲۸ بیتی در نظر بگیریم و از بایت اول آن برای نمایش شماره بسته استفاده نماییم، داریم: (شماره بسته نشان میدهد این بسته چندمین بسته از مجموعه بسته های ارسالی برای فایل مورد تقاضا است. به طور مثال اگر شماره بسته برابر با ۱۰ باشد نشان میدهد این بسته دهمین بسته از فایل ارسالی را در برگرفته است).

File Response:

offset[0]

data[1-127]

از آنجایی که این برنامه در شبکه های محلی استفاده می گردد احتمال وقوع خطا کم بوده و برای سادگی نیازی نیست که پروتکل UDP را برای آن قابل اطمینان نمایید .

توجه داشته باشید برای سادگی فرض میشود، هر برنامه متقاضی تنها یک تقاضا می دهد و تا زمانی که پاسخ آن را دریافت نکرده است تقاضای دیگری ارسال نمی کند .

نکات

۱. پروژه به صورت گروه های دوفره انجام می شود.
۲. در این پروژه فقط مجاز هستید از کتابخانه های socket برای ارتباط شبکه استفاده کنید.
۳. مشخص شدن ثقلب به منزله نمره صفر پروژه درسی می باشد.
۴. فایل های برنامه نویسی شده به همراه یک گزارش ۱ صفحه ای را در سامانه LMS پلود کنید.