

مقدمه

در این پروژه قصد داریم شما را با پیاده‌سازی دو نرم‌افزار آشنا کنیم. اولین نرم‌افزار یک ماشین حساب توزیع شده است که محاسبات سرویس‌گیرنده^۱ را بر روی سرویس‌دهنده^۲ انجام داده و نتایج را باز می‌گرداند. نرم‌افزار دوم یک سیستم توزیع فایل p2p است.

ماشین حساب توزیع شده

این نرم‌افزار می‌تواند محاسبات سرویس‌گیرنده را بر روی سرویس‌دهنده انجام دهد. این نرم‌افزار در مواقعی کاربرد دارد که حجم محاسبات بالا بوده و سیستم سرویس‌گیرنده قادر به انجام آن‌ها نیست و یا در مواقعی که نرم‌افزار فقط قابلیت نصب بر روی یک سیستم را داشته و مجموعه‌ای از سرویس‌گیرندگان می‌خواهند از آن سرویس بگیرند. به عنوان مثال نرم‌افزارهایی مانند متلب این ویژگی را دارند.

برای سادگی فرض می‌کنیم این ماشین حساب عملیات‌های ساده را انجام می‌دهد. مدل ارتباط این نرم‌افزار به صورت سرویس‌گیرنده-سرویس‌دهنده است. این نرم‌افزار از پروتکل ارتباطی زیر استفاده می‌کند:

Calculation Request:

\$ operator \$ op1 \$ op2 \$

که در آن operator عملیاتی می‌باشد که قرار است در سمت سرویس‌دهنده انجام شود. همانطور که بیان شد این نرم‌افزار از عملیات‌های ساده ریاضی پشتیبانی می‌کند پس مقدار این فیلد یکی از حالت‌های 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.

¹ client

² server

Calculation Response:

\$ calculation time \$ result \$

که در آن calculation time زمانی است که در سمت سرویس‌دهنده 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 پیاده‌سازی این قسمت در واقع شامل دو نرم‌افزار سرویس‌دهنده و سرویس‌گیرنده است و شما می‌بایست هر دوی آن‌ها را پیاده‌سازی نمایید.

سیستم توزیع فایل P2P

این نرم‌افزارها که شاید یکی از معروف‌ترین آن‌ها نرم‌افزار Torrent باشد، فایل‌ها را بین کاربران توزیع می‌کنند و با پیروی نکردن از مدل سرویس‌دهنده و سرویس‌گیرنده دیگر فایل‌ها تنها از تعدادی نقطه مشخص پخش نمی‌شوند. با توجه به این موضوع احتمال رهگیری منشأ فایل کمتر می‌شود و از آن برای توزیع فایل به صورت غیرقانونی نیز استفاده می‌گردد.

برای سادگی شما تنها قسمت توزیع فایل را پیاده‌سازی خواهید کرد به این ترتیب که یک نرم‌افزار پیاده‌سازی می‌شود 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 می‌کند و در حالت دریافت فایل، می‌تواند برای آن فایل تقاضای همه‌پخش^۳ تولید کند. برای مثال:

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

با دستور برنامه‌ی شما در حالت دریافت فایل اجرا شده و تقاضا فایل hello.txt به صورت همه‌پخش برای همه در شبکه ارسال می‌کند. اگر یکی از گره‌ها به صورت مثال با دستور زیر اجرا شده باشد:

```
p2p -serve -name hello.txt -path C:\Downloads\hello.txt
```

با دریافت تقاضای فایل hello.txt پاسخ را که فایل موردنظر است به صورت مستقیم برای متقاضی ارسال می‌کند. دقت کنید متقاضی می‌بایست بر روی همان آدرس آی‌پی و پورتی که با آن تقاضا را ارسال کرده است در حال گوش دادن باشد. شما می‌بایست برای زمانی که نرم‌افزار شما در حال توزیع فایل قرار دارد یک پورت مشخص برای آن در نظر بگیرید که انتخاب این پورت به عهده‌ی شما خواهد بود. تقاضا دهنده‌ها می‌توانند از پورت‌های تصادفی استفاده نمایند یا از یک پورت مشخص، از آنجایی که سرویس‌دهنده از روی تقاضای ارسالی پاسخ می‌دهد بین این دو حالت تفاوتی وجود نداشته و انتخاب آن برعهده شما خواهد بود.

دقت داشته باشید برای این سیستم شما تنها یک نرم‌افزار طراحی می‌کنید. این نرم‌افزار هر دو حالت دریافت و ارسال فایل را پشتیبانی می‌کند و می‌تواند در حالتی که کاربر از او می‌خواهد اجرا شود. دقت کنید که نیازی نیست هر دوی این حالت‌ها را به صورت همزمان در حال اجرا باشند.

پروتکل این نرم‌افزار به شکل زیر می‌باشد:

File Request:

filename

که در آن فیلد filename اسم فایلی است که برای آن تقاضا می‌دهیم.

از آنجایی که بسته‌های UDP محدودیت دارند، شما نیاز دارید که یک اندازه‌ی مشخص برای بسته‌ها را در نظر گرفته و بسته‌های خود را با آن اندازه ارسال کنید، در صورتی که اندازه‌ی فایل از اندازه‌ی بسته‌ی شما بزرگتر باشد می‌بایست آن را قطعه قطعه و ارسال نمایید، در نهایت در سمت متقاضی می‌بایست آن را به صورت یکپارچه در قالب یک فایل ذخیره کنید. با توجه به این توضیحات برای مثال اگر بسته‌های خود را ۱۲۸ بیتی در نظر بگیریم و از بایت اول آن برای

³ Broadcast

نمایش آفست استفاده نماییم، داریم: (آفست نشان می‌دهد این بسته چندمین بسته از مجموعه بسته‌های ارسالی برای فایل مورد تقاضا است. به طور مثال اگر آفست 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 شبکه‌های محلی استفاده می‌گردد احتمال وقوع خطا کم بوده و برای سادگی نیازی نیست که پروتکل UDP را برای آن قابل اطمینان نماییم.

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

قوانین

۱. پروژه به صورت گروهی انجام می‌شود.

۲. در این پروژه فقط مجاز هستید از کتابخانه‌های socket برای ارتباط شبکه استفاده کنید.

۳. مشخص شدن تقلب به منزله نمره صفر پروژه درسی می‌باشد. (تشخیص تقلب با استفاده از ابزارهای تشخیص همانندی کد صورت می‌گیرد).

۴. فایل‌های برنامه نویسی شده به همراه یک گزارش ۱ صفحه‌ای را در سامانه درس افزار آپلود کنید.

محمد رستمی

موفق باشید.