



دانشکده مهندسی کامپیوتر  
و فناوری اطلاعات

بسمه تعالی

# دانشگاه صنعتی امیرکبیر (پلی تکنیک تهران) دانشکده مهندسی کامپیوتر و فناوری اطلاعات



دانشکده مهندسی کامپیوتر و فناوری اطلاعات  
دانشگاه صنعتی امیرکبیر

۱۳۹۷ - ۱۳۹۶



دانشگاه صنعتی امیرکبیر  
( پلی تکنیک تهران )

## درس شبکه های کامپیوتری، نیم سال اول سال تحصیلی ۹۷-۹۸ تمرین برنامه نویسی

توجه: پاسخ تمرین ها باید به صورت دستنویس تحویل داده شود.

نام و نام خانوادگی:

شماره دانشجویی:

نمره:

توجه: برای صرفه جویی در کاغذ تکالیف را یا دو رو پرینت بگیرید و یا از کاغذهای باطله یک رو سفید استفاده کنید.

### مقدمه

در این تمرین 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<sup>۱</sup> را بر روی سرویس دهنده<sup>۲</sup> انجام داده و نتایج را باز می گرداند. نرم افزار دوم یک سیستم توزیع فایل p2p است.

### فوانین

۱. پیاده سازی باید فقط با زبان جاوا انجام شود.

۲. ارسال و دریافت بسته ها تماماً با استفاده از Socket ها در جاوا صورت می گیرد و قسمت های مربوط به ساخت، گوش دادن، ارسال و دریافت می بایست در کد شما بوده و قابل رویت باشند.

۳. تمرین به صورت گروه های حداکثر دو نفره می تواند انجام شود. (گروه تک نفره نمره اضافی ندارد)

### 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 می کنیم 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 می کند:

Calculation Request:

\$ operator \$ op1 \$ op2 \$

<sup>1</sup> client

<sup>2</sup> server



# درس شبکه های کامپیوتری، نیم سال اول تحصیلی ۹۷-۹۸

## تمرین برنامه نویسی



صفحه: 2 از 3

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:

\$ calculation time \$ result \$

که در آن calculation time زمانی است که در سمت سرویس دهنده برای انجام محاسبه سپری شده است. دقت کنید که 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 شما می بایست هر دوی آن ها را پیاده سازی نمایید.

## سیستم توزیع فایل P2P

این نرم افزارها که شاید یکی از معروف ترین آن 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 منشأ 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 پخش<sup>3</sup> 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 پخش برای همه در شبکه ارسال می کند. اگر یکی از گره ها 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 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

<sup>3</sup> Broadcast



# درس شبکه های کامپیوتری، نیم سال اول تحصیلی ۹۷-۹۸

## تمرین برنامه نویسی



صفحه: 3 از 3

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 استفاده نمایند یا از یک پورت مشخص، از آنجایی 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 را به صورت همزمان 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 آفست استفاده نماییم، داریم: (آفست نشان می دهد این بسته چندمین بسته از مجموعه بسته های ارسالی برای فایل مورد تقاضا است. به طور مثال اگر آفست برابر با ۱۰ باشد نشان می دهد این بسته دهمین بسته از فایل ارسالی را در بر گرفته 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 می شود، هر برنامه متقاضی تنها یک تقاضا می دهد و تا زمانی که پاسخ آن را دریافت نکرده است تقاضای دیگری ارسال نمی کند.

موفق باشید