



به نام خدا  
تمرین کامپیوتری ۱  
دانشکده‌ی مهندسی برق و کامپیوتر دانشگاه تهران  
شبکه‌های کامپیوتری، نیم‌سال دوم سال تحصیلی ۹۷-۹۸  
زمان تحویل: ۱۹ خرداد ۱۳۹۸



## مقدمه

در این پروژه، هدف آشنایی با مفاهیم Socket Programming در زبان‌های برنامه‌نویسی Python یا C++ و ماشین‌های مجازی و برنامه‌نویسی در سیستم عامل‌های Unix-based است و نحوه ارتباط بین چندین دستگاه مجازی که همه بر روی یک Local Host ولی با Port‌های مختلف هستند را فرا بگیرید و پیکربندی یک شبکه کوچک با استفاده از Socket Programming را بشناسید.

## قسمت اول: Design Your Chatroom

در این قسمت یک Chatroom ساده را پیاده‌سازی می‌کنید. این قسمت شامل یک برنامه سرور است و تعدادی Client. تعداد کلاینت‌ها می‌تواند هر عددی باشد. برنامه سرور ابتدا اجرا می‌شود. سپس برنامه کلاینت به اجرا در می‌آید. هر کلاینت یک ارتباط TCP با سرور برقرار می‌کند و ارتباط تا انتهای برنامه باز می‌ماند.

- در صورتی که از پایتون استفاده می‌کنید، برنامه Server با صورت دستور زیر اجرا خواهد شد

```
python server.py -sp 9090
```

که -sp مشخص‌کننده این است که عددی که در ادامه می‌آید (9090)، Port ای است که Server بر روی آن منتظر دریافت Connection و بعد از آن پیام است.

- برنامه Client با دستور زیر اجرا خواهد شد

```
python client.py -sip server-ip -sp 9090
```

که باید در آرگومان‌های ورودی برنامه ip و Port مربوط به Server را بدهید.

زمانی که یکی از کلاینت‌ها پیامی ارسال می‌کند، پیام به سرور ارسال شده و این پیام به سایر Client‌ها Broadcast می‌شود. پیامی که بر روی صفحه کاربران نمایش داده می‌شود به صورت روبرو است

<From Client\_ip, Client\_port>: Message!

```
Rouzbah at Rouzbah's MacBook  
Communication/CA1 using  
Python server.py -sp 9090  
Server Initialized on port 9090
```

```
Rouzbah at Rouzbah's MacBook Pro in ~/Deskt  
Communication/CA1 using  
Python client.py -sip localhost -sp 9090  
Connected to server. Start sending messages  
> Hello World!  
<From ('127.0.0.1', 55513)> Socket Programmin  
g!  
>
```

```
Rouzbah at Rouzbah's MacBook Pro in ~/Deskt  
work*/Network_Communication/CA1 using  
Python client.py -sip localhost -sp 9090  
Connected to server. Start sending messages  
<From ('127.0.0.1', 60790)> Hello World!  
> Socket Programming!  
>
```

در تصویر ۳ نکته ذکر شده را مشاهده می کنید.

## قسمت دوم: Design Your FTP!

در این قسمت از شما خواسته می شود که شبکه ای همچون FTP دانشگاه را پیاده سازی کنید. سرور دارای تعدادی فایل است و به درخواست کلاینت فایل ها را برای آن ارسال می کند. از سمت کلاینت سه نوع پیغام به سرور ارسال می شود شامل:

- **List**: سرور لیست فایل های خود را بر می گرداند.
- **Retrieve**: با ارسال نام فایل، کلاینت درخواست ارسال آن فایل را از سرور می دهد.
- **Delete**: با ارسال این دستور به همراه نام فایل، فایل مورد نظر حذف می شود.

این دستورات از سمت کلاینت برای سرور ارسال می شود و سرور پاسخ آن را ارسال می کند. Server در صورت وجود نداشتن فایل پیغام **File Not Found!** را برای Client ارسال می کند و اگر فایل مورد نظر را در اختیار داشت، فایل را به قطعه های 1000 بایتی تقسیم کرده و به ترتیب برای Client ارسال خواهد کرد.

لازم به ذکر است که فرمت راه اندازی Server و Client به صورت مورد ۱ و ۲ سوال اول است.

در صورت داشتن سوال می توانید با دستیار آموزشی درس تماس بگیرید (rbostandoust@gmail.com).

موفق باشید

r.bostandoust@gmail.com