



دانشکده‌ی مهندسی کامپیوتر

شبکه‌های کامپیوتری

مدرس: مهدی جعفری

## Socket Programming (TCP)

گردآورندگان: محمد مهدی به نصر - کمیل یحیی زاده

شنبه ۴ آذر ۱۴۰۲

در این سوال می‌توانید از هر زبان برنامه نویسی استفاده کنید. دقت کنید که برنامه شما باید به صورت multi-thread یا async باشد که بتواند همزمان به چندین client سرویس دهد.  
قسمت‌های مشخص شده با علامت \* سوالات و یا مطالب کاربردی هستند که جواب دادن و خواندن آنان اختیاری است اما نمره اضافی ندارد.

## ۱ HTTP Proxy

در این تمرین می‌خواهیم که یک HTTP Proxy را پیاده‌سازی کنیم. تو دنیای حال حاضر بیشتر تمرکز و وظیفه HTTP Proxy حفاظت از حریم خصوصی، بهینه‌سازی ترافیک و مدیریت دسترسی به محتوا است.

### ۱.۱ مقدمه

در این تمرین می‌خواهیم که یک HTTP Proxy ساده را پیاده‌سازی کنیم. پراکسی ما در دو حالت plain text و connect کار می‌کند که به کمک این دو، می‌توان هم درخواست‌های HTTP و هم درخواست‌های HTTPS را پراکسی کرد. همچنین همان‌طور که از اسم HTTP Proxy پیدا است، پکت‌های ارتباطی آن، با پروتکل HTTP کار می‌کنند.

### ۲.۱ حالت plain text

همان‌طور که گفته شد اولین پکت درخواست HTTP Proxy از جنس یک پکت HTTP است. به مثال زیر نگاه کنید:

```
GET http://www.google.com/204 HTTP/1.1
Host: www.google.com
User-Agent: curl/8.0.1
Accept: */*
```

همان‌طور که می‌بینید، این دقیقاً یک پکت HTTP است که صرفاً به جای request target آن، آدرس مقصدی که بسته باید به آن فرستاده شود، داده می‌شود. در مثال زیر، بسته باید به آدرس http://www.google.com/204 فرستاده شود. باید بتوانید با دستور زیر، یک نمونه درخواست plain text به یک سرور پراکسی بر روی 127.0.0.1:12345 بزنید.

```
curl -v -x "http://127.0.0.1:12345" "www.google.com/204"
```

با ارسال درخواست بالا باید این بسته به سایت www.google.com از طرف پراکسی سرور ارسال شود:

```
GET /204 HTTP/1.1
Host: www.google.com
User-Agent: curl/8.0.1
Accept: */*
```

همان‌طور که می‌بینید، تنها در خط اول و request target تفاوت بین دو بسته وجود دارد به نحوی که آدرس دامنه و پروتکل آن از request target حذف می‌شود. سپس باقی ارتباط بدون دست خوردن توسط پراکسی به دو طرف فروارده می‌شود. بدین معنا که اگر کلاینت پکتی برای پراکسی فرستاد (در ادامه‌ی استریم) باید بدون دست خوردن برای گوگل فرستاده شود و برعکس.

## ۳.۱ حالت connect

در صورتی که HTTP request method شما برابر connect بود، باید کار کمی متفاوت‌تری را انجام دهید. این نوع درخواست معمولاً در درخواست‌های پراکسی کردن دیتای HTTPS دیده می‌شود. به عنوان مثال، اگر دستور زیر را اجرا کنیم:

```
curl -v -x "http://127.0.0.1:12345" "https://www.google.com/204"
```

این درخواست به پراکسی http ارسال می‌شود:

```
CONNECT www.google.com:443 HTTP/1.1
Host: www.google.com:443
User-Agent: curl/8.0.1
Proxy-Connection: Keep-Alive
```

دقت کنید که در اینجا اطلاعات مربوط به آدرس دقیق صفحه (در قسمت قبلی مثلاً 204) در درخواست وجود ندارد چرا که این درخواست برای تونل ایجاد کردن به عنوان پراکسی برای اتصال امن به سایت مورد نظر می‌باشد. حالا چگونه باید با این نوع درخواست "connect" برخورد کنیم؟ وقتی این درخواست وارد می‌شود، ابتدا باید به سرور www.google.com:443 وصل شویم (که در خط اول درخواست آمده) و سپس دقیقاً این پیام و دو خط جدید<sup>۱</sup> را به کلاینت ارسال کنیم:

```
HTTP/1.1 200 Connection established
```

پس از این مرحله، تمام پیام‌های بعدی باید به صورت دو طرفه توسط پراکسی منتقل شوند. پیشنهاد می‌شود برای درک بهتر مسئله، که یک سرور پراکسی ساده مانند "gost" را فعال کنید و با استفاده از نرم‌افزار "Wireshark" پکت‌های آن را بررسی کنید.

## ۴.۱ \*Caching

ما در این بخش توقع داریم سرور پروکسی دارای این قابلیت باشد که بتوان imageهایی که مشاهده میکند را در فایل یا دیتابیس ذخیره کند و درخواست‌های مجدد آن عکس از طریق تمام کلاینت‌ها انجام میشود از فایل یا دیتابیس HTTP proxy خوانده شود.

## ۲ تحویل دادنی‌ها

فرمت فایل‌هایی که تحویل می‌دهید باید به صورت زیر باشد:

۱. فایل README که نام، نام‌خانوادگی، شماره دانشجویی و توضیحات کد است.
۲. کدهای خود را می‌توانید به هر زبانی می‌خواهید بنویسید ولی پیشنهاد ما پایتون است.
۳. نباید از Package و Library که برای ایجاد HTTP Proxy استفاده می‌شوند استفاده شوند، در این زمینه می‌توانید از طریق کوئرا از TAها سوال بپرسید.
۴. یک فایل Makefile که پروژه‌ی شما رو بیلد می‌کند. در صورتی که از dependency خاصی استفاده می‌کنید حتماً آن را در Makefile خود نصب کنید.

## ۳ تست و ارزیابی

فرمت فایل‌هایی که تحویل می‌دهید باید به صورت زیر باشد:

۱. تست صحت: با توجه به تایپ HTTP Proxy رکویست‌هایی کلاینت‌ها از طریق HTTP Proxy می‌فرستند باید فقط ip HTTP Proxy قابل مشاهده باشد و ایپی‌های کلاینت‌ها قابل رویت نباشد
۲. تست هم‌رندی: برنامه باید بتواند همزمان چندین وبسایت را باز کند.

موفق باشید