

بر نام خداوند، تشنه‌ی مهربان



دانشکده‌ی مهندسی کامپیوتر

زمستان ۱۳۹۳

تمرین برنامه‌نویسی صفرم* شبکه‌های رایانه‌ای

دانشگاه صنعتی شریف

مدرس: مهدی خرازی

۱. هدف‌ها

- آشنایی با سامانه‌های عامل UNIX،
- آشنایی با UNIX Socket API،
- نوشتن Makefile،
- آشنایی با پروتکل SOCKS،
- ساختن پوشه، نام‌گذاری و فشرده‌سازی و ارسال صحیح تمرین.

۲. مقدمه

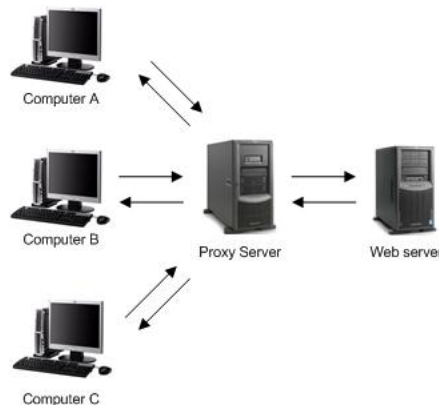
یک معماری مناسب برای اتصال چند کاربر روی شبکه، معماری کارخواه-کارگزار است. در این تمرین از شما خواسته شده یک proxy بر مبنای این معماری پیاده‌سازی کنید. به این منظور، کارخواه‌ها باید طبق یک پروتکل خاص با کارگزار صحبت کنند و کارگزار هم باید طبق یک پروتکل مشخص به درخواست‌ها پاسخ دهد.

۳. روش کار

کارگزار proxy نقش واسطه بین کارخواه و یک منبع خارجی را ایفا می‌کند. برای شکل‌گیری ارتباط، کارخواه یک درخواست حاوی آدرس یک منبع خارجی را به کارگزار ارسال می‌کند. سپس کارگزار در نقش یک کارخواه با منبع

* با سپاس از بهنام مومنی، علی فتاح‌المنان، عرفان عبدی، سعید محلوچی‌فر، هادی ذوالفقاری و مهران خلدی

خارجی ارتباط برقرار می‌کند. با برقراری این ارتباط، اطلاعات لازم از منبع خارجی دریافت می‌شود و در اختیار کارخواه قرار می‌گیرد. به این ترتیب کارخواه از دید منبع خارجی پنهان می‌ماند.



شکل ۱: نقشه‌ی شبکه‌ی شامل سه کارخواه، یک منبع خارجی و یک کارگزار proxy

در این تمرین قصد داریم یک کارگزار SOCKS 4A پیاده‌سازی کنیم. SOCKS پروتکلی است که هدف آن ایجاد یک proxy در لایه‌ی ۵ یا لایه‌ی جلسه^۱ از مدل ۷ لایه‌ی OSI است. به این ترتیب proxy مورد نظر به لایه‌ی کاربرد روی آن وابسته نخواهد بود. برای نمونه، به هنگام استفاده از یک HTTP proxy، محدود به ارسال و دریافت بسته‌های HTTP هستیم در حالی که در هنگام استفاده از یک کارگزار SOCKS این محدودیت وجود ندارد.

۴. پروتکل SOCKS 4

عملکرد این پروتکل بر اساس فرستادن پیام بین کارخواه و کارگزار است. کارخواه یک پیام درخواست به کارگزار SOCKS ارسال می‌کند و کارگزار با توجه به وضعیت، به درخواست پاسخ می‌دهد. ساختار پیام‌ها در پروتکل SOCKS همانند شکل ۲ است.

1	2	3	4	5	6	7	8	...	N
VN	CD	DST PORT	DST IP				ID	Null	

شکل ۲: ساختار بسته‌ها در SOCKS 4

در این پیام، بایت اول نشان‌گر نسخه‌ی مورد استفاده است که برای SOCKS 4 برابر 0x04 خواهد بود. بایت بعدی کد دستور است و مقدار آن بر اساس نوع درخواست یا پاسخ تعیین می‌شود که در ادامه شرح داده می‌شود.

^۱Session Layer

بایت سوم و چهارم نشان دهنده‌ی شماره‌ی پورت منبع خارجی است که کارخواه قصد ارتباط با آن را دارد. بایت پنجم تا هشتم به آدرس IP منبع خارجی اختصاص دارد. بعد از آدرس IP نام کاربری قرار می‌گیرد که طول آن دلخواه است و کاربرد آن شناسایی کاربران است. در این تمرین نام کاربری در بسته‌های درخواست وجود دارد ولی کارگزار SOCKS از آن استفاده‌ای نمی‌کند. در نهایت یک بایت پوچ^۲ قرار می‌گیرد که نشان‌دهنده‌ی انتهای پیام است.

۱.۴. SOCKS 4A

این پروتکل کارکردی مشابه پروتکل SOCKS 4 دارد. تنها تغییر آن نسبت به SOCKS 4 این است که در این پروتکل کارخواه می‌تواند به جای آدرس IP، نام منبع خارجی را ارسال کند. به این ترتیب وظیفه‌ی واکاوی این نام بر عهده‌ی کارگزار خواهد بود. برای این کار، در صورتی که کارخواه از آدرس IP منبع خارجی مطلع نباشد، ۴ بایت مربوط به آن را به صورت 0.0.0.X که X یک بایت دلخواه و مخالف صفر است، قرار می‌دهد و بعد از ارسال درخواست، یک رشته‌ی پوچ پایانی^۳ دیگر را که حاوی نام دامنه است ارسال می‌کند.

۵. برنامه‌ی کارگزار

برنامه‌ی کارگزار برای اجرا نیاز به ورودی زیر دارد:

```
$/server.out [port]
```

در اینجا port نشان‌دهنده‌ی پورتی است که کارگزار روی آن برای اتصال کاربران گوش می‌کند.

اتصال موفق هر کارخواه با نمایش پیغام زیر مشخص می‌شود:

```
A client from [IP]:[port] was connected
```

در این پیغام IP و port به ترتیب نشان دهنده‌ی آدرس IP و شماره‌ی پورتی است که کارخواه با استفاده از آن درخواست را ارسال کرده است. ممکن است در هر لحظه چندین کارخواه متصل باشند. بنابراین، کارگزار باید بتواند به چندین درخواست همزمان خدمت دهد.

پس از اتصال کارخواه، کارگزار منتظر دریافت درخواست از کارخواه می‌ماند. در پیام درخواست‌شده، نسخه‌ی مورد استفاده باید برابر 0x04 باشد. همچنین کد دستور باید برابر 0x01 باشد که نشان دهنده‌ی دستور اتصال است. در غیر این صورت بسته نادیده گرفته می‌شود. درخواست ممکن است حاوی آدرس IP یا نام دامنه‌ی منبع خارجی باشد. اگر آدرس IP در درخواست نخست به شکل 0.0.0.X باشد، کارگزار منتظر دریافت یک پیام پوچ پایانی دیگر که شامل نام دامنه است (پس از نام کاربری) می‌ماند. هنگام دریافت یک درخواست، کارگزار پیغام زیر را نمایش می‌دهد:

```
The client from [IP]:[port] requested for [IP/name]:[port]
```

^۲Null

^۳Null terminated

در صورتی که درخواست حاوی آدرس IP باشد، باید آن را نمایش دهد و در غیر این صورت نام دامنه را نشان دهد.

پس از دریافت درخواست، کارگزار در صورت نیاز، آدرس IP را واکاوی می‌کند و یک اتصال به منبع خارجی را آغاز می‌کند. در صورتی که این اتصال با موفقیت شروع شود، پیام زیر را نمایش می‌دهد:

```
Connection established with [IP]:[port]
```

در این پیام، IP و port به ترتیب نشان‌دهنده‌ی آدرس IP و شماره‌ی پورت منبع خارجی هستند. در صورتی که اتصال موفقیت‌آمیز نبود باید پیام زیر نمایش داده شود:

```
Could not connect to [IP/name]:[port]
```

عدم توانایی کارگزار در واکاوی نام دامنه نیز ناموفق بودن ارتباط محسوب می‌شود. در صورتی که کارگزار در واکاوی نام دامنه به مشکل برخورد باید در پیام بالا از نام استفاده کند و در غیر این صورت آدرس IP را نمایش دهد. پس از نمایش پیغام فوق، کارگزار باید یک پیام حاوی وضعیت اتصال را به کارخواه ارسال کند. ساختار این پیام که در شکل ۳ نشان داده شده است، همانند پیام درخواست است. در بایت اول (که در بسته‌ی درخواست نشان‌دهنده‌ی نسخه‌ی پاسخ بود) مقدار 0x00 قرار می‌گیرد. در بایت بعدی که نشان‌دهنده‌ی وضعیت اتصال است، یکی از دو مقدار 0x5A یا 0x5B قرار می‌گیرد. در صورتی که اتصال موفقیت‌آمیز بود، مقدار 0x5A و در غیر این صورت مقدار 0x5B در بایت دوم قرار می‌گیرد. شش بایت بعدی در پیام پاسخ باید همگی برابر صفر باشند^۴. توجه کنید که این پیام شامل ۸ بایت است و در انتهای آن نیازی به بایت پوچ نیست. در صورتی که اتصال به منبع خارجی موفقیت‌آمیز نباشد، کارگزار باید بعد از ارسال این بسته، ارتباط خود را با کارخواه قطع کند.

1	2	3	4	5	6	7	8
Null	Status	DST PORT		DST IP			

شکل ۳: ساختار بسته‌ی پاسخ در SOCKS

پس از اتصال موفق به منبع خارجی، کارگزار باید به انتقال داده بین کارخواه و منبع خارجی بپردازد. در صورتی که در حین اجرای این وظیفه، اتصال کارخواه با کارگزار قطع شود، باید پیغام زیر نمایش داده شود:

```
The client from [IP]:[port] was disconnected
```

در واقع این پیام باید پس از بستن اتصال با کارخواه چاپ شود (حتی اگر اتصال با منبع خارجی برقرار نشده باشد). بعد از نمایش این پیغام، کارگزار اتصال با منبع خارجی را هم (در صورت وجود) به طریق مناسب قطع می‌کند. در صورتی که اتصال کارگزار با منبع خارجی با مشکل مواجه شود (یا پس از چاپ پیغام فوق بسته شود)، باید پیغام زیر نمایش داده شود:

```
The connection with [IP]:[port] was lost
```

^۴ در پروتکل SOCKS این بایت‌ها توسط کارخواه نادیده گرفته می‌شوند و مقدار آنها اهمیت ندارد ولی در این تمرین شما باید آن‌ها را برابر صفر قرار دهید.

بعد از نمایش این پیغام، کارگزار اتصال با کارخواه را هم (در صورت وجود) به طریق مناسب قطع می‌کند. به عبارت دیگر نیازی نیست تفاوتی میان حالت‌های بروز خطا در نظر بگیرید. پس از هر خطا، اتصال با کارخواه و منبع خارجی (در صورت برقرار شدن) به طریق مناسب قطع شده و با بستن هر سوکت، پیغام خاتمه‌ی ارتباط مربوط به آن سوکت نمایش داده می‌شود.

۶. کارخواه و منبع خارجی

پیاده‌سازی برنامه‌ی کارخواه و منبع خارجی بخشی از تمرین نیست. برای بررسی عملکرد برنامه‌ی کارگزار می‌توانید از یک مرورگر به عنوان کارخواه و از هر کارگزار HTTP دلخواهی به عنوان منبع خارجی استفاده کنید. برای این کار باید تنظیمات proxy در مرورگر را طوری تغییر دهید که از پروتکل SOCKS 4A استفاده کند. همچنین می‌توانید از ابزار SOCAT استفاده کنید.

۷. پیاده‌سازی

با توجه به اینکه ارزیابی و نمره‌دهی تمرین به صورت خودکار خواهد بود، توصیه می‌شود جهت ارتقا نمره‌ی خود مراحل زیر را برای پیاده‌سازی دنبال کنید.

۱. ابتدا باید یک سوکت به منظور گوش دادن بر روی پورت مورد نظر ایجاد کنید و درخواست‌های مشتری‌ها را بپذیرید و خروجی صحیح را چاپ کنید.

۲. در مرحله‌ی بعد بایستی یک ارتباط با مشتری‌ها برقرار کنید و یک ریسه برای ارتباط با مشتری ایجاد کنید. سپس باید اولین پیام که حاوی اطلاعات منبع خارجی است دریافت کنید. با دریافت این پیام باید اطلاعات صحیح مربوط به درخواست را چاپ کنید.

۳. در صورتی که پیام اول حاوی نام دامنه باشد، باید آدرس IP مرتبط با آن را واکاوی کنید.

۴. بعد از مشخص شدن آدرس، باید تلاش کنید با منبع خارجی ارتباط برقرار کنید و وضعیت ارتباط را چاپ کنید.

۵. در مرحله‌ی بعد باید وضعیت اتصال را به مشتری اطلاع دهید. این کار باید با استفاده از پیام پاسخ با ساختار معرفی شده انجام شود.

۶. تا به اینجا ارتباط‌های مورد نیاز برقرار شده و حالا باید به ارسال داده در هر دو جهت پردازید. برای اینکار باید یک ریسه‌ی دیگر ایجاد کنید. به این ترتیب هر ریسه وظیفه‌ی انتقال داده‌ها را در یک جهت خواهد داشت.

۷. چاپ خروجی صحیح مربوط به قطع ارتباط با مشتری و منبع خارجی و بستن همه‌ی سوکت‌ها آخرین مرحله‌ی کار است.

۸. نکات ضروری

- به علت اینکه نمره‌ی تمرین به صورت خودکار داده می‌شود، ساختار پیام‌های گفته شده باید دقیقاً به صورت گفته شده باشد.
- نحوه‌ی ارزیابی ممکن است دچار تغییراتی شود و تست‌های دیگری اضافه شوند.
- در صورتی‌که هر مشکل یا پرسشی داشتید که فکر می‌کنید پاسخ آن برای همه مفید خواهد بود، آن را به گروه اینترنتی درس ارسال کنید.
- از فرستادن جواب تمرین به گروه اینترنتی درس خودداری کنید.
- تمام برنامه‌ی شما باید توسط خود شما نوشته شده باشد. فرستادن کل یا قسمتی از برنامه‌تان برای افراد دیگر، یا استفاده از کل یا قسمتی از برنامه‌ی فرد دیگری، حتی با ذکر منبع، تقلب محسوب می‌شود!
- پس از اتمام کارتان لازم است که فایل‌های خود را به همراه Makefile فشرده کرده و از طریق [وبسایت پرتو](#) ارسال نمایید.
- فایل‌ارسالی شما باید یک فایل zip باشد که پوشه‌ای در آن نیست و در آن فقط فایل‌های .cpp و .h. شما به اضافه‌ی یک Makefile است که با اجرای دستور make فایل اجرایی server.out تولید می‌شود.