



تمرین برنامه‌نویسی صفرم^۱ شبکه‌های کامپیوتری

مدرس: مهدی خرازی

پائیز ۱۳۹۰

در این تمرین شما باید برنامه‌ای را به صورت Client/Server برای اجرای دستورات رو یک کامپیوتر راه دور و دریافت خروجی آن با استفاده از مفاهیم Socket در محیط UNIX به زبان C یا C++ پیاده‌سازی کنید.

هدف

آشنایی با:

- محیط سیستم‌عامل‌های UNIX؛
- محیط برنامه‌نویسی Eclipse؛
- POSIX Socket API؛
- نوشتن Makefile؛
- ساختن پوشه، نام‌گذاری و فشرده‌سازی و ارسال صحیح تمرین.

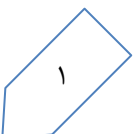
مقدمه

گاهی اوقات، کامپیوتری که قرار است دسترسی به آن صورت پذیرد، به صورتی فیزیکی در مکان دوردستی قرار دارد و امکان دسترسی مستقیم به آن نیست؛ برای مثال ممکن است کارگزار آ‌های وب‌سایت یک شرکت ایرانی در کشور کانادا قرار داشته باشند. در چنین شرایطی امکان دسترسی فیزیکی به این سیستم‌ها دشوار و غیرممکن می‌نماید. برای حل این مشکل راه‌حل‌ها گوناگونی وجود دارد. یکی از ساده‌ترین و قوی‌ترین راه‌ها، وجود برنامه‌ای است که در یک سو فرامین موردنظر را از کارخواه^۳ دریافت کند و

^۱ با تشکر از بهنام مؤمنی، امیر شیخها، مهدی احمدی‌نژاد، کامیار اللهوردی، سجاد فولادی و مهرداد مرادی

^۲ Server

^۳ Client



در سوی دیگر آن‌ها را بر روی کامپیوتر راه دور^۴ اجرا نماید و خروجی آن دستورات را به کارخواه بازگرداند. به این ترتیب، کارخواه قادر است فرامین موردنظر خود را اجرا نماید.

به این منظور، پروتکل‌های مختلفی مانند Telnet و Secure Shell (SSH) ایجاد شده‌اند. در این تمرین قصد داریم یک جفت برنامه Client و Server ایجاد کنیم تا این وظیفه را بر عهده گیرند.

برنامه کارگزار

این برنامه بر روی کامپیوتر راه دور اجرا می‌شود و وظیفه دارد که فرمان‌های دریافت شده را اجرا نماید. همچنین این برنامه وظیفه تصدیق هویت کارخواه را نیز برعهده دارد که شرح آن در ادامه می‌آید.

این برنامه به صورت زیر باید اجرا شود:

```
$ ./server <PORT> <FILE_NAME>
```

در این دستور، PORT شماره پورت TCP است که برنامه‌ی کارگزار باید بر روی آن Listen کرده و منتظر بماند تا یک برنامه کارخواه به آن متصل گردد.

همچنین، FILE_NAME آدرس فایل‌ی است که در آن نام کاربری و رمز عبور کاربران مجاز وارد شده است و هر نام کاربری، رمز عبور و Shell مربوط به کاربر با فاصله از هم جدا شده است و هر کاربر در یک خط آمده است. به عنوان مثال:

```
amir 112233 /bin/bash
mahtab 123456 /bin/bash
mehrdad 653421 /bin/sh
sadjad rsdtjkfff /bin/csh
kamyar highway /bin/bash
```

ستون سوم، پوسته‌ای است که باید فرامین را در آن اجرا کنید. جزئیات مربوط به این مورد در ادامه آمده است.

برنامه‌ی کارگزار بر روی <PORT> منتظر برقراری ارتباط TCP از سوی کارخواه می‌ماند. بعد از برقرار شدن این اتصال، کارگزار پیام Connection established. را به کارخواه ارسال می‌کند و باید این پیام از سوی کارخواه نمایش داده شود.^۵

تصدیق هویت در برنامه کارگزار

اکنون برنامه‌ی کارخواه قادر است تا دستورالعمل‌ها را - به فرمتی که در ادامه توضیح داده خواهد شد - به کارگزار ارسال کند. اما تا پیش از تصدیق هویت^۶ کارگزار نباید هیچ دستوری را اجرا کند. برای تصدیق هویت، از سه دستور استفاده می‌کنیم. دستور اول، دستور HELO است. بعد از برقراری ارتباط و دریافت پیام موفقیت از سوی کارگزار ابتدا کارخواه با ارسال دستور HELO برقرار شدن اتصال را تایید می‌کند. پاسخ این پیام از سوی کارگزار Please authenticate. خواهد بود. ارسال هر دستوری پیش از این دستور باید با پیام Not authenticated. همراه شود.

^۴ Remote

^۵ تمام رشته‌های ارسالی باید null-terminated باشند. همچنین تمامی فرامین ارسالی به کارگزار باید به n \ ختم شوند.

^۶ Authentication

سپس کارخواه، نام کاربری خود را با دستور **USER username** به کارگزار ارسال می‌کند که در آن **username** نام کارخواهی است. در صورتی که این نام کارخواهی در **<FILE_NAME>** موجود نباشد، کارگزار به کارخواه پیام

Invalid username.

را ارسال می‌کند و دوباره منتظر دستور **USER** می‌ماند. در غیر این صورت پیام

Username accepted, enter password.

را به کارخواه می‌فرستد. در این حالت کارخواه باید با دستور **PASS password** رمز عبور خود را ارسال کند. در صورتی که رمز عبور نادرست باشد، پیام **Invalid password.** ارسال می‌گردد و کارخواه لازم است دوباره از دستور **USER** استفاده کند.

در صورت صحت رمز عبور، کارگزار پیام

Welcome, username.

را برای کارخواه ارسال می‌کند.

ارسال هر دستور دیگر یا وارد کردن دستورات به ترتیب نادرست باید با پیام **Not authenticated.** رو به رو گردد. دقت کنید که در این حالت عملیات تصدیق هویت قبلی ادامه می‌یابد (مثلاً اگر دستور **USER** اجرا شده بود، نیازی به اجرای مجدد آن نیست).

برنامه کارخواه

برنامه‌ی کارخواه به صورت زیر اجرا می‌شود:

```
./client <IP> <PORT>
```

در این دستور، مقادیر **IP** و **PORT** مربوط به کامپیوتری است که برنامه کارگزار در آن اجرا شده است.

در ابتدا کارخواه به کارگزار متصل می‌گردد. بعد از برقراری اتصال، کارخواه قادر است که دستورات خود را برای کارگزار ارسال کند. اما تا پیش از تصدیق هویت، هیچ‌کدام از دستورات اجرا نخواهند شد. بعد از تصدیق هویت کارخواه قادر است دستورات موردنظر خود را به صورت رشته‌های مختوم به صفر (**null-terminated**) به کارگزار ارسال کند. همچنین باید در انتهای رشته‌ی مربوط به تمام دستورات ارسالی از سوی کارخواه کاراکتر **\n** قرار گرفته شود. کارگزار آن دستورات را اجرا کرده و خروجی موردنظر را به کارخواه باز می‌گرداند. دستورات از ورودی استاندارد خوانده می‌شوند و خروجی آنها بعد از دریافت از برنامه کارگزار در خروجی استاندارد باید چاپ شوند.

در نهایت کارخواه با دستور **EXIT** به اتصال پایان می‌دهد و از سمت خودش به اتصال پایان می‌دهد. در این حالت کارگزار نیز باید به اتصال از سوی خودش پایان دهد و دوباره منتظر برقراری اتصال جدید بماند.

اگر کاربر دستور **TERMINATE** را اجرا کند، هم برنامه کارخواه و هم کارگزار باید به کار خود خاتمه دهند.

دقت کنید که بعد از پایان نشست، سرور باید دوباره به **Listen** کردن ادامه دهد. همچنین هر بار تنها یک **Client** وجود خواهد داشت (بنابراین نیازی به استفاده از **Thread**ها در سمت سرور برای مدیریت چند کاربر همزمان نیست).

دقت کنید نیازی نیست برای دستورات فرمت خاصی در نظر بگیرید؛ تمام دستورات همانگونه که تایپ شده‌اند به کارگزار ارسال شده و کارگزار آنها را در Shell مورد نظر اجرا می‌نماید و خروجی را دریافت می‌کند. بنابراین نیازی نیست که نگران رشته‌های ورودی به عنوان فرمان باشد؛ کافی است آنها را همان‌گونه که هستند (به همراه \n انتهایی) به کارگزار ارسال نمایید.

یک نمونه از یک نشست بین کارگزار و کارخواه به صورت زیر است (پیام‌های آبی ارسالی از کارگزار و سبز ارسالی از کارخواه است):

```
Connection established.
HELO
Please authenticate.
USER justin
Invalid username.
USER sadjad
Username accepted, enter password.
ls
Not authenticated.
PASS rsdtjkff
Welcome, sadjad.
ls
.      ..   temp  files  games
rm -rf home/
date
Wed Aug 5 11:15:30 IRDT 2011
cat /home/sadjad/temp/1.txt
There are four PAs:
1. PA0
2. PA1
3. PA2
4. PA6
EXIT
```

اجرای دستورات کاربر

نکته‌ای که باید ذکر شود آن است که برای اجرای برنامه‌ها نباید تک تک دستوراتی که وارد می‌شوند را با فرمان `exec` اجرا کنید. این کار باعث می‌شود تا تمام برنامه‌هایی که نیاز به تعامل با کاربر (مثلاً گرفتن ورودی) دارند، نتوانند به وسیله برنامه شما و از راه دور کار کنند.

در عوض، برای هر کاربر، سرور یک Shell که آدرس فایل آن در ستون سوم فایل تنظیمات سرور آمده است، باز می‌کند و تمام دستوراتی که کاربر وارد می‌کند را به آن ارسال کرده و خروجی را از آن دریافت می‌کند. به این ترتیب، برای اجرای تمام دستورات یک کاربر کافی است که یک باید عملیات `fork/exec` انجام شود تا shell مورد نظر ایجاد شود.

برای این کار، به جای دستور `fork` از دستور `forkpty` استفاده می‌کنیم. این دستور به عنوان آرگومان اول، یک متغیر از نوع `int*` دریافت می‌کند و بعد از `fork` کردن یک `file descriptor` را داخل این متغیر قرار خواهد داد. به کمک این `file descriptor` و استفاده از دستورات `read` و `write` می‌توانید به سادگی با پروسه‌ی فرزند ایجاد شده تعامل کنید. با استفاده از دستور `write` فرامین را به shell مورد نظر ارسال کرده و با استفاده از `read` خروجی‌های نوشته شده در این پروسه را دریافت کنید.

دقت کنید که بعد از استفاده از `forkpty` برنامه‌های خود را باید با سوئیچ `lutil`– کامپایل کنید. مثلاً `gcc server.c –lutil` در غیر این صورت به خطای `undefined reference to 'forkpty'` برخورد خواهید کرد. کاربران سیستم‌عامل `Mac OS X` نیازی به این سوئیچ برای کامپایل ندارند.

برای اطلاعات بیشتر در مورد `forkpty` لینک‌های زیر می‌توانند مفید باشند:

- <http://www.win.tue.nl/~aeb/linux/lk/lk-10.html>
- <http://linux.die.net/man/3/forkpty>

ارسال

برای ارسال این تمرین باید یک `Makefile` بنویسید. برای این کار می‌توانید از راهنمای نوشتن `Makefile` که در سایت درس قرار دارد استفاده کنید. برنامه شما باید با دستور `make` کامپایل شود و فایل‌های اجرایی `client` و `server` را تولید کند.

نکات ضروری

- در صورتیکه هر مشکل یا پرسشی داشتید که فکر می‌کنید پاسخ آن برای همه مفید خواهد بود، لطفاً آن را به گروه پستی درس ارسال کنید.
- از فرستادن جواب تمرین به گروه پستی جداً خودداری کنید.
- فرستادن کل یا قسمتی از برنامه‌تان برای افراد دیگر، یا استفاده از کل یا قسمتی از برنامه فردی دیگر به نام خود، تقلب محسوب می‌شود.
- پس از اتمام کارتان لازم است که فایل‌های خود را به همراه `Makefile` فشرده کرده و بر روی سیستم خودکار داوری⁷ `upload` کنید. از ارسال فایل‌های باینری خودداری کنید.

موفق باشید

⁷ <http://partov.sharif.edu/judge>