به نام خداوند بخشنده‌ی مهربان

دانشگاه صنعتی شریف
مدرس: مهدی خرازی

تمرین برنامه‌نویسی یکم*
شبکه‌های رایانه‌ای

دانشکده‌ی مهندسی کامپیوتر
خزان ۱۳۹۳

۱. هدف‌ها

- آشنایی با چارچوب کاربری پرتو (PARTOV)،
- آشنایی با واکاوی نام دامنه،
- آشنایی با سرآیندهای Ethernet، IP و UDP.

۲. مقدمه

سامانه‌ی نام دامنه (ساناد)[۱]، مجموعه‌ای از کارگزارهای نام دامنه را در برمی‌گیرد که در یک ساختار سلسله‌مراتبی به هم مرتبط می‌شوند. هدف این سامانه، ایجاد تناظر میان نام دامنه‌ها و آدرس IP مرتبط با آن‌هاست. به فرآیند تبدیل یک نام دامنه به یک آدرس IP «واکاوی نام دامنه»[۲] می‌گویند.

در این تمرین قصد داریم بخش محدودی از یک اکوسیستم مشابه DNS را در سامانه‌ی پرتو شبیه‌سازی کنیم. توجه کنید که پروتکل مورد بحث در این تمرین تفاوت‌های فراوانی در جزییات با پروتکل اصلی DNS دارد و بخش‌هایی از پروتکل اصلی حذف شده‌اند و تغییر یافته‌اند.

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

[۱] Domain Name System (DNS)

[۲] Domain Name Resolution

۳. آشنایی با پرتو

پرتو سامانه‌ی شبیه‌ساز شبکه‌های کامپیوتری است که در این تمرین به‌کارگرفته می‌شود. این سامانه به صورت کارخواه‌ـ‌کارگزار عمل می‌کند. کارگزار پرتو یک توپولوژی شبکه و معماری گره‌ها را برنامه‌ریزی می‌کند و کارخواه‌ها، گره‌های خاص را برنامه‌ریزی می‌کنند. شما می‌توانید با اجرای یک کارخواه به یک گره‌ی مجازی که بر روی کارگزار پرتو شبیه‌سازی شده است، متصل شوید. بدین ترتیب کارگزار پرتو بسته‌های شبیه‌سازی‌شده را به کارخواه‌ها می‌رساند و بسته‌های ارسالی آن‌ها را هم دریافت کرده و در اختیار گره‌های مجازی قرار می‌دهد. چارچوب کارخواه[۳] پرتو، کتابخانه و کلاس‌های نرم‌افزاری از پیش نوشته‌شده‌ایست که نقش شبیه‌سازی دستگاه‌های شبکه را ایفا می‌کند. کاربر با استفاده از آن‌ها و در محیط آن، توابع، متدها و دیگر نیازهای برنامه‌نویسی خود را تامین می‌کند.

در این تمرین، هر کارگزار DNS را بر روی یکی از گره‌های مجازی پرتو اجرا می‌کنیم و این گره‌ها می‌بایست از طریق تعامل با یکدیگر مطابق پروتکل DNS پاسخ پرسمان‌های دریافتی را بیابند.

برای آشنایی بیشتر با نحوه‌ی استفاده از چارچوب کارخواه پرتو می‌توانید به مستند راهنمای آن مراجعه کنید.

۴. واکاوی نام دامنه

فرآیند واکاوی نام دامنه، از طریق تبادل چند پیغام میان کارخواه و کارگزارهای DNS صورت می‌گیرد. این فرآیند، با ارسال درخواست به کارگزار DNS ریشه شروع می‌شود و در هر گام کارخواه به سمت یک کارگزار DNS اختصاصی‌تر هدایت می‌شود، تا در نهایت آدرس IP متناظر با نام دامنه‌ی دلخواه را بیابد.

نکته: همان‌طور که در طول درس دیدیم[۴]، یک کارگزار جهت پاسخ‌گویی به یک پرسمان بازگشتی، همانند یک کارخواه عمل کرده و مراحل واکاوی نام را به ترتیب طی می‌کند.

برای مثال، در فرآیند واکاوی نام دامنه‌ی `ce.sharif.edu.`، ابتدا درخواست به یک کارگزار ریشه ارسال می‌شود، که پاسخ آن ما را به سمت کارگزار مرجع دامنه‌ی `edu.` هدایت می‌کند. با ارسال درخواست به این کارگزار، به سمت کارگزار مرجع دامنه‌ی `sharif.edu.` هدایت می‌شویم. با ارسال درخواست به این کارگزار، پاسخ نهایی را (که IP متناظر دامنه‌ی `ce.sharif.edu.` است) دریافت می‌کنیم.

نکته: این روند را با اجرای دستور `dig ce.sharif.edu. +trace +all` در shell مشاهده کنید.

نکته: به نقطه‌ی انتهای نام دامنه توجّه کنید. این نقطه در واقع نشان‌دهنده‌ی دامنه‌ی ریشه است که به طور معمول به جهت سادگی، آن را حذف می‌کنند. در این تمرین عدم حضور این نقطه به معنای نامعتبر بودن نام دامنه است.

[۳] Client Framework

[۴] Translating Addresses

۱.۴. جزییات

در طول فرآیند واکاوی نام دامنه نکات زیر را رعایت کنید:

- سرآیندهای Ethernet، IP و UDP را مطابق استاندارد پیاده‌سازی کنید.
- برای ساختن پرسمان، مقدار اوّلیه‌ی فیلد Source Port را برابر ۸۰۰۰ قرار دهید و به ازای هر پرسمان آن را یکی افزایش دهید.
- دریافت پاسخ هیچ پرسمانی نباید بیش از ۳ ثانیه طول بکشد. بنابراین در صورتی که ۳ ثانیه پس از ارسال یک درخواست پاسخی دریافت نکردید، آن فرآیند واکاوی را لغو کنید و در خروجی عبارت `Timeout!` را چاپ کنید.
- کارگزار باید در پاسخ یک پرسمان رکوردی را بازگرداند که پسوندی از دامنه‌ی مورد پرسش باشد. در صورت وجود چند رکورد با این خاصیت، باید رکورد با طولانی‌ترین نام دامنه را بازگرداند و در صورتی که چنین رکوردی وجود نداشته باشد، نباید پاسخی برگرداند (بنابراین پرسمان منجر به timeout خواهد شد).
- در واقعیت کارگزارهای DNS می‌توانند رکوردهای متعدّدی برای یک نام دامنه داشته باشند و از این قابلیت برای توزیع بار میان کارگزارهای مختلف استفاده می‌شود. ولی در این تمرین به جهت سادگی می‌توانید فرض کنید که هر نام دامنه صفر یا یک رکورد روی یک کارگزار دارد.
- به هیچ وجه نتایج میانی پرسمان‌ها را cache نکنید. برای نمونه حین پاسخ‌گویی به پرسمان `ce.sharif.edu.` دامنه‌ی `tld.net.` بارها واکاوی می‌شود. به عبارت دیگر نتیجه‌ی یک پرسمان را به هیچ وجه (حتّی در طول یک عملیات واکاوی) برای تسریع پرسمان‌های بعدی ذخیره نکنید.
- کارگزار شما باید بتواند به پرسمان‌های هم‌زمان پاسخ بدهد.

۵. انواع رکوردهای منبع

هر کارگزار شامل تعدادی رکورد منبع[۵] است و هر رکورد تناظری میان نام دامنه و نحوه‌ی یافتن آدرس مربوطه برقرار می‌کند. رکوردها انواع زیادی دارند، اما در این تمرین تنها با سه نوع رایج‌تر سر و کار خواهیم داشت:

۱. `A` : این رکورد شامل آدرس IP مرتبط با نام دامنه است. برای نمونه تناظر دامنه‌ی `ce.sharif.edu.` با آدرس `213.233.170.2` به وسیله‌ی این نوع رکورد معلوم می‌شود.

۲. `CNAME` : این نوع رکورد یک دگرنامی از نام دامنه را مشخّص می‌کند. برای نمونه `mail.yahoo.com.` یک دگرنامی از `login.yahoo.com.` است. یکی از فواید این نوع رکورد، برای کارگزارهایی است که چندین دامنه را میزبانی می‌کنند. چرا که در صورت تغییر آی‌پی آن‌ها، کافیست یک رکورد تغییر یابد.

[۵] Resource Record

۳. `NS` : این رکورد شامل نام دامنه‌ی یک کارگزار DNS است که ادامه‌ی روند واکاوی باید از طریق آن پی‌گیری شود. همان‌طور که گفته شد، DNS یک سامانه‌ی توزیع‌شده است و داده‌ها در آن به صورت مرکزی ذخیره نمی‌شود. این امکان از طریق نوع رکورد `NS` فراهم می‌شود. برای نمونه در صورت درخواست ترجمه‌ی `ce.sharif.edu.` از کارگزار مرجع `edu.` ، پاسخ `NS` دریافت خواهیم کرد که ما را به سمت `ns.sharif.edu.` راهنمایی می‌کند.

نکته: شاید متوجّه شده باشید که در مثال آخر، هدایت شدن به طرف `ns.sharif.edu.` منجر به ایجاد یک حلقه می‌شود! در این موارد، از یک رکورد پیوندی[۶] استفاده می‌شود. به این صورت که کارگزار، IP مرتبط با رکورد NS را نیز نگه‌داری می‌کند و به این ترتیب از شکل‌گیری حلقه جلوگیری می‌شود.

۶. پروتکل

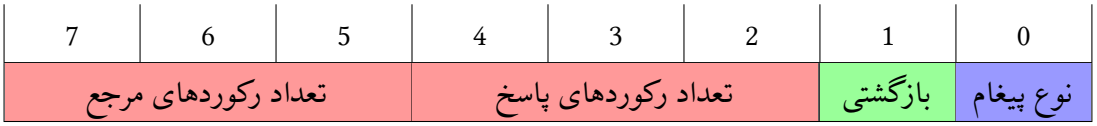
هر پیغام (درخواست یا پاسخ) DNS در قالب یک و تنها یک بسته‌ی UDP جابه‌جا می‌شود و کارگزار به درخواست‌هایی که بر روی پورت ۵۳ دریافت می‌کند پاسخ می‌دهد. دقّت کنید که برخی فیلدها مشترک هستند، برخی تنها در درخواست معنی دارند و برخی تنها در پاسخ.

هر پیغام DNS به ترتیب شامل بخش‌های زیر است:

در ادامه محتوای هر یک از این بخش‌ها را به تفصیل بررسی می‌کنیم.

۱.۶ سرآیند

سرآیند ۸ـبیتی پیغام DNS به صورت زیر است (بیت 7 پرارزش‌ترین بیت است):



نوع پیغام: در پرسمان مقدار `0` و در پاسخ پرسمان مقدار `1` می‌گیرد.

بیت بازگشتی: نشان‌دهنده‌ی نوع پرسمان است. در صورت گرفتن مقدار `1` پرسمان بازگشتی است.

[۶] glue record

تعداد رکوردهای پاسخ: نشان‌دهنده‌ی تعداد رکوردهایی است که در بخش پاسخ (مطابق بخش ۳.۶) آورده می‌شوند. این مقدار در بسته‌های پرسمان برابر `0` است (چون بخش پاسخ و مرجع در پرسمان وجود ندارند). اگر کارگزار بخواهد رکورد `A` یا `CNAME` را بازگرداند، مقدار این فیلد برابر `1` خواهد بود. در این حالت مقدار فیلد بعدی `0` است.

تعداد رکوردهای مرجع: نشان‌دهنده‌ی تعداد رکوردهایی است که در بخش رکوردهای مرجع پاسخ پرسمان (مطابق بخش ۴.۶) آورده می‌شوند. این فیلد در بسته‌های پرسمان و همچنین در بسته‌هایی که فیلد قبل برابر `1` باشد، مقدار `0` را می‌گیرد. در غیر این صورت، این فیلد مقدار `1` را می‌گیرد.

۲.۶. رکورد مورد پرسش

این بخش تنها در پرسمان می‌آید و شامل یک رشته‌ی null-terminated است که همان دامنه‌ی مورد پرسش است.

۳.۶. رکوردهای پاسخ

هر رکورد به ترتیب شامل سه فیلد زیر است:

نوع رکورد: یک فیلد ۱۶ـبیتی است که نوع رکورد را مشخّص می‌کند. در این تمرین تنها یکی از مقادیر `0x01` (رکورد `A`)، `0x02` (رکورد `NS`) یا `0x05` (رکورد `CNAME`) را می‌گیرد.

نام دامنه: یک رشته‌ی null-terminated که نشان‌دهنده‌ی نام دامنه‌ی رکورد است.

داده‌ی رکورد: برای نوع `A` یک فیلد ۴ـبیتی شامل IP است و برای رکوردهای `CNAME` یا `NS` یک نام دامنه‌ی دیگر (به صورت null-terminated) است.

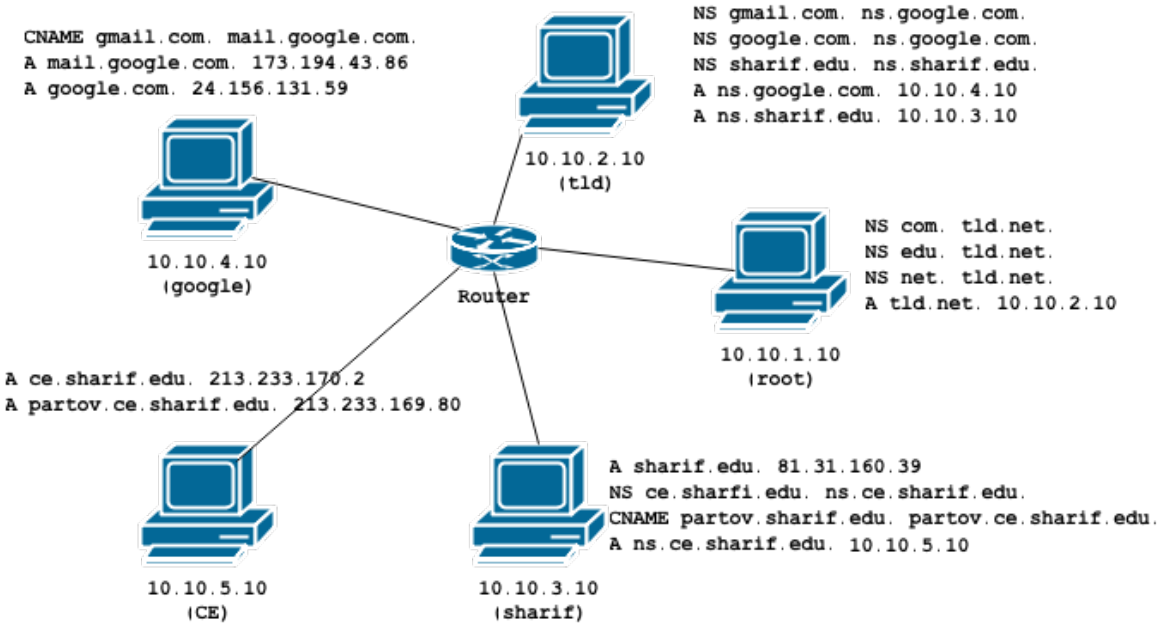
۴.۶. رکوردهای مرجع

رکوردهایی که در این بخش می‌آیند، از نوع `NS` و نشان‌دهنده‌ی nameserverهایی هستند که ادامه‌ی روند واکاوی نام باید از طریق آن‌ها پی‌گیری شود. فرمت رکوردها همانند بخش ۳.۶ است.

برای نمونه وقتی درخواستی برای واکاوی نام دامنه‌ی `ce.sharif.edu.` به کارگزار نام دامنه‌ی `edu.` ارسال شود، پاسخ پرسمان شامل صفر رکورد در بخش پاسخ (چون کارگزار `edu.` مرجع نام دامنه‌ی مورد نظر نیست) و یک رکورد در بخش مرجع خواهد بود. این رکورد اعلام می‌کند که کارگزار مرجع نام دامنه‌ی `sharif.edu.` دارای نام `ns.sharif.edu.` است (زیرا `sharif.edu.` طولانی‌ترین پسوند نام مورد جست‌وجو است). آنگاه در ادامه‌ی واکاوی، باید نام `ns.sharif.edu.` (با شروع دوباره از کارگزار نام دامنه‌ی ریشه) واکاوی شود تا آدرس IP آن برای ادامه‌ی واکاوی به‌کاربرده شود.

۷. توپولوژی شبکه

شکل ۱ نشان‌دهنده‌ی وضعیت گره‌های پرتو به همراه رکوردهای مرتبط با آن‌ها است. البته برنامه‌ی شما نباید به تعداد کارگزارها یا اطلاعات روی آن‌ها که در شکل آورده شده‌اند، وابسته باشد. این شکل تنها یک نمونه است.



شکل ۱ : نقشه‌ی شبکه‌ی شبیه‌سازی‌شده‌ی نمونه شامل پنج کارگزار نام دامنه

۸. ورودی و خروجی

موارد زیر از طریق بخش Custom Information پرتو در اختیار شما قرار می‌گیرد:

- آدرس MAC روتر متصّل به کارگزار
- آدرس IP کارگزار DNS ریشه
- لیستی از رکوردهایی که بر روی این کارگزار قرار گرفته. رکوردهای مختلف توسّط , (کاما) جدا شده‌اند.

برنامه‌ی شما باید پس از شروع، به عنوان یک کارگزار DNS به پرسمان‌هایی که روی پورت ۵۳ دریافت می‌کند پاسخ بدهد.

همچنین، در هر خطّ ورودی یک دستور به صورت `dig <domain-name>` می‌آید که باید با آن همانند یک پرسمان بازگشتی برخورد کنید. پاسخ این پرسمان را در یک خط چاپ کنید. برای نمونه[۷]:

```
dig ce.sharif.edu.
213.233.170.2
dig partov.sharif.edu.
partov.ce.sharif.edu.
dig edu.sharif.edu.
Timeout!
```

۹. توصیه‌ها

- ابزارهای مورد نیاز خود را در کنار فایل سرآیندها که به شما داده می‌شود پیاده‌سازی کنید. این ابزارها شامل توابع، کلاس‌ها، ساختارها و ماکروهای مورد نیاز برای کار بر روی بسته‌های شبکه و قابلیت نمایش اطّلاعات به صورت قابل فهم برای انسان است.
- تا جای ممکن تعداد ریسه‌ها را در برنامه‌ی خود کم نگه دارید. در زمینه‌ی همگام‌سازی نهایت تلاش خود را انجام دهید.
- دقّت کنید که بسته‌هایی که در محیط شبیه‌ساز دریافت می‌کنید برای خودتان نیستند[۸]! همچنین به constructor و copy constructorها توجّه کنید تا از مشکلات مربوط به کار با حافظه تا جای ممکن به دور باشید.
- در هنگام استفاده از داده‌ساختارها و کتابخانه‌های زبان C++ به ویژه STL بسیار دقّت کنید. در مورد چگونگی کارکرد آن‌ها اطمینان حاصل کنید و از ساختارهای پیچیده تا جای ممکن پرهیز کنید. اگر چندین نمونه شی را با استفاده از templateها به صورت تودرتو استفاده می‌کنید، دقّت کنید تا با مشکلات زیادی روبه‌رو نشوید.
- به نظم نمایش اعداد (Byte Endianness) توجّه کنید و در یک نظم واحد کار کنید.
- امکانات C++11 می‌توانند به سادگی کد شما کمک کنند. جهت استفاده از آن‌ها می‌توانید گزینه‌ی `-std=c++11` را به انتهای متغیّر `FLAGS` در Makefile خود اضافه کنید.

[۷] رنگ نوشته‌ها فقط برای تمایز ورودی و خروجی است. ورودی و خروجی برنامه‌ی شما نباید رنگی باشد.

[۸] در متد `processFrame()` یک frame به شما داده شده که buffer آن برای دریافت همه‌ی بسته‌ها به صورت مشترک استفاده می‌شود. در صورتی که نیاز به نگه‌داری بسته دارید آن را کپی کنید و در حافظه‌ای که خود در اختیار دارید قرار دهید.

۱۰. نکات ضروری

- 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 پوشه‌ی user را به همراه Makefile فشرده کرده (می‌توانید این کار را با اجرای دستور `make submit` انجام دهید) و از طریق وب‌سایت پرتو ارسال نمایید.