

گزارش پروژه دوم

ملیکا مرافق 810197581
نازنین یوسفیان 810197610

دستورات پیاده سازی شده:

```
MySystem <id>  
MySwitch <number_of_ports> <id>  
Connect <system_id> <switch_id> <port_number>  
ConnectSwitch <switch1_id> <switch2_id> <switch1_port> <switch2_port>  
Send <sender_id> <receiver_id> <filename>  
Receive <receiver_id> <sender_id> <filename>  
STA
```

توضیح برنامه:

در ابتدا یک برنامه اصلی به نام `main_process` داریم که دستورات را دریافت می کند. به ازای هر دستور `MySystem` یک پروسس جدید `fork` و `exec` می شود و یک سیستم جدید خواهیم داشت. به ازای هر دستور `MySwitch` نیز یک پروسس `switch` جداگانه خواهیم داشت. بین `main_process` و هر یک از پروسس های جدیدی که ساخته می شود یک `unnamed pipe` وجود دارد تا اطلاعات دستورات دیگر از این طریق به برنامه ها داده شود. هرگاه دستور `Connect` وارد شود، دو `named pipe` ساخته می شود تا سیستم و سوئیچ بتوانند تبادل اطلاعات کنند. یکی از این پایپ ها از سمت سیستم به سوئیچ و دیگری از سمت سوئیچ به سیستم است. برای دستور `ConnectSwitch` نیز همین گونه است و دو پایپ بین دو سوئیچ ساخته می شود که یکی از سوئیچ 1 به 2 و دیگر از سوئیچ 2 به 1 است.

هر سوئیچ همواره در حال گوش کردن روی پایپ های ورودی است و اگر روی هر کدام از پایپ ها چیزی نوشته شود، آن را می خواند و در یک صف، `store` می کند (به آخر صف اضافه می کند). هم چنین اگر این صف خالی نباشد، اولین عنصر آن را پاک کرده و `forward` می کند. قبل از `forward` کردن، ابتدا `lookup table` خود را بررسی می کند. اگر آیدی فرستنده در جدول وجود نداشت، آن را به جدول اضافه می کند. برای اینکه یک فریم فرستاده شود دو حالت وجود دارد: یا آیدی گیرنده در جدول وجود دارد یا خیر. اگر وجود داشت، که پایپ متصل به آن را داریم و فقط در همان پایپ فرستاده

می شود. در غیر این صورت، در تمامی پایپ های خروجی به جز پایپ مربوط به فرستنده فریم فرستاده می شود.

هر سیستم بعد از وصل شدن به سوئیچ، روی پایپ ورودیش همواره گوش می دهد. برای اینکه سیستم فایلی را بفرستد، فایل را خوانده و آن را به یک سری فریم تبدیل می کند. فریم ها به این صورت هستند که 8 کاراکتر اول آن ها هدر است و در حالت send برابر با 11110101 است. پس از آن آیدی گیرنده، آیدی فرستنده و سایز فریم است که هر کدام از 5 کاراکتر تشکیل شده اند. در انتها نیز دیتا قرار می گیرد که 1500 کاراکتر است به جز فریم آخر که ممکن است از این مقدار کمتر باشد. با این فرمت فریم، سوئیچ ها و سیستم ها میتوانند اطلاعات مورد نظر را استخراج کنند.

پس از اینکه فایل به فریم ها تبدیل شد، سیستم آن ها را روی پایپ خروجی می نویسد. هر فریمی که از پایپ ورودی به سیستم برسد، سیستم ابتدا چک می کند که آیدی گیرنده با آیدی خودش برابر باشد. اگر برابر بود، بر اساس آدرس فرستنده آن را به یک map اضافه می کند. کلید این مپ برابر با آیدی فرستنده و محتوای آن وکتوری از فریم هاست. وقتی که فریمی وارد شد که سایز دیتای آن از 1500 کمتر بود می توان فهمید که فایل به صورت کامل دریافت شده است و در فولدری که به سیستم اختصاص داده شده و نام آن برابر با آیدی سیستم است، فریم ها در یک فایل نوشته می شوند. (فرض شده که فقط فایل های text جابجا می شوند).

هنگامی که سیستمی دستور receive را وارد کند، یک فریم از این سیستم روی پایپ خروجیش قرار می گیرد. این فریم به این صورت است که 8 کاراکتر اول آن هدر است و برابر با 11111111 است. سپس آیدی مقصد، آیدی خودش و سایز فریم، 15 کاراکتر بعدی آن هستند. در انتها نیز اسم فایل به عنوان دیتا نوشته می شود. سیستم مقصد که این فریم را دریافت می کند ابتدا هدر را بررسی می کند و اگر برابر با 8 تا 1 بود، send را آغاز می کند.

با دریافت دستور STA الگوریتم spanning tree اجرا می شود. فرض بر این است که هر سیستم به یک سوئیچ و هر پورت یک سوئیچ حداکثر به یک دستگاه (سوئیچ یا سیستم) متصل است و گرافی همبند داریم. ابتدا کوچکترین شناسه را در میان سوئیچ ها پیدا می کنیم. سپس با توجه به اینکه در صورتی که چند مسیر تا root وجود داشت کوتاه ترین فاصله تا root را باید انتخاب کنیم BFS را اجرا می کنیم. هنگام پیمایش بر روی همسایه های یک node اگر آن را visit کرده باشیم یعنی این مسیر یا طولانی تر است یا پدرش id کوچکتری نسبت به فعلی داشته. بنابراین این یال را حذف می کنیم و node های دو سر یال را نگه می داریم تا در انتهای الگوریتم اتصال بین آنها را قطع کنیم. برای حالتی که فاصله از ریشه برابر بود باید سوئیچ با id کوچکتر را انتخاب کنیم. برای این کار هنگامی که تمامی سوئیچ ها در صف یک فاصله از root قرار دارند آن ها را بر اساس id به صورت صعودی مرتب می کنیم. پس از اتمام الگوریتم، توسط main_process به سوئیچ هایی که دو سر پورت هایی قرار دارند که باید حذف شوند، یک پیام disconnect فرستاده می شود و پایپ های مربوطه حذف می شوند. اینگونه بین این دو سوئیچ دیگر کانال ارتباطی وجود ندارد.

کلاس Network

در این کلاس تعداد پورت های یک سوئیچ- id سیستم ها و سوئیچ ها و یک map که id سوئیچ و id دستگاه(سوئیچ یا سیستم) را به پورت متناظر سوئیچ مپ می کند داریم. در هر دستور Connect یا ConnectSwitch چک میکنیم که دستگاه ها موجود باشند و سوئیچ آن پورت را داشته باشد و پورت متصل به دستگاه دیگری نباشد. در صورتی که دستور بدون اشکال بود شماره پورت و سوئیچ ها را به map اضافه می کنیم.

و همان طور که گفته شد در انتهای الگوریتم spanning tree شماره پورت های اتصالاتی که باید off شوند را از map حذف می کنیم.