به نام خدا

گزارش کار پروژه دوم شبکه

*فاطمه غفارپور:810197634
*زهرا یزدانی: 810197641

پروژه به زبان c++ زده شده است.
توضیحات مربوط به پیاده سازی پروژه و فرضیات در نظر گرفته شده:

ساخت سویچ ها و سیستم ها:

سویچ ها و سیستم ها به کمک پراسس ساخته می شوند. زمانی که دستور ساخت سوئیچ یا سیستم می آید، به کمک fork پراسس ساخته میشود و در فرزند،اگر دستور ساخت سوییچ باشه Swich.out اجرا می شود و اگر دستور ساخت سیستم باشد System.out اجرا میشود.

تبادل اطلاعات بین سیستم و سویچ:

نحوه تبادل داده به صورت پایپ در نظر گرفته شده است.دو نوع پایپ named و unnamed به این منظور در نظر گرفته شده است.برای تبادل کامند ها (char* buf) به سیستم و سوئیچ از unnamed پایپ استفاده شده است.به این صورت که هنگامی که سیستم یا سوئیچ ساخته میشود، یک unnamed پایپ برای این سیستم و سویچ ساخته میشود تا توسط این پایپ بتوان اطلاعات دستورات را به سوئیچ و سیستم فرستاد. برای تبادل فریم ها نیز، named پایپ در نظر گرفته شده است.دو named پایپ داریم.در یکی از آنها سیستم مینویسد و سویچ از آن می خواند و در دیگری سویچ مینویسد و سیستم از آن می خواند یا در تبادل بین دو سوئیچ، یک سوئیچ روی یک پایپ مینویسد و دیگری از آن پایپ میخواند و بالعکس.

دستور connect: دو نوع دستور connect در نظر گرفته شده است.
حالت اول connect سوئیچ به سیستم که ترتیب کامند وارد شده مطابق دستور کار پروژه است.زمانی که این دستور وارد میشود دو named پایپ ساخته می شود.این پایپ‌ها راه تبادل داده بین سوئیچ و سیستم را فراهم میکند.حال از طریق unnamed پایپ ساخته شده برای سوئیچ اطلاعات کامند connect و named پایپ‌هایی که با سوئیچ در ارتباط است برای سیستم فرستاده میشود.
همینطور از طریق unnamed پایپ که برای این سویچ که سیستم به آن وصل شده است،اطلاعات این connect و name پایپ‌هایی که بین این سوییچ و سیستم برقرار است فرستاده میشود.
حالت دوم connect سویچ به سویچ است.در این حالت دو named پایپ برای تبادل این دو سویچ ساخته میشود و اطلاعات این کامند را توسط unnamed پایپ به دو تا سوئیچ میفرستیم.

فرم دستور connect دو سوئیچ:

Connect_Switch_To_Switch source destination

دستور send:

اگر دستور send وارد شود،اطلاعات این دستور توسط unnamed پایپ مربوط به سیستم، به سیستم فرستاده میشود.

فرم دستور send :

send systemNumber destination filename

دستور receive:

اگر دستور receive وارد شود،اطلاعات این دستور توسط unnamed پایپ مربوط به سیستم، به سیستم فرستاده میشود.

نکته برای این دستور در پیاده سازی در نظر گرفته نشده است که منتظر بمانیم که فایلی که درخواست کرده حتما برسد.

فرم دستور receive:

receive systemNumber destination fileName

خواندن از پایپ ها:

برای اینکه از پایپ ها بخوانیم از select استفاده کرده ایم.بدین صورت که fd را برای named و unnamed در نظر گرفته ایم و بعد با استفاده از select که باعث میشود این دو پایپ بلاک نشوند، پایپ ها را میخوانیم. در هر دو فایل system.out و switch.out یک while وجود دارد که همواره چک می‌شود که اگر روی پایپی داده‌ای بود آن را هندل کنیم.

فرستادن اطلاعات بین سیستم ها:

اطلاعات به صورت frame انتقال داده میشود.به این صورت که یک استراکت frame داریم که شامل اطلاعات سایز داده، داده ، مبدا، مقصد، شماره فریم و تعداد بخش های کوچکتر فریم اصلی و اسم فایل است.

تبدیل داده بزرگ به داده های کوچکتر:

اگر سایز فایل از ۱۵۰۰ بزرگتر بود، فایل را به فریم هایی با سایز ۱۵۰۰ میشکانیم و یک فایل را با چند فریم ارسال میکنیم.

برای فرستادن این اطلاعات، این اطلاعات را به صورت string به هم میچسبانیم و با استفاده از _ اطلاعات را از هم جدا میکنیم.

ترتیب قرار گرفتن اطلاعات فریم به صورت string :

frame.dataSize _ frame.data_ frame.destination_ frame.source_ frame.num _frame.numberOfFrame _ frame.fileName

دریافت فریم ها و سرهم کردن فریم ها:

هر فریم با توجه به نام فایلی که خوانده شده و شماره فرستنده‌اش، از دیگری مجزا میشود.هر فریم همراه خود، مشخص میکند که اطلاعات فایلی که باید خوانده شود، به چند فریم کوچکتر شکسته شده است. پس باید به تعداد تکه های کوچکتر یک فایل را از پایپ بخوانیم. به وسیله یک لوپ که به اندازه تعداد فایل های کوچکتر است باید از پایپ بخوانیم تا همه ی تکه های کوچکتر برسند و بتوانیم merge کنیم و فایل اصلی را بسازیم. برای اینکه مطمئن شویم فریم فرستاده شده مربوط به این سیستم از مقصد فریم را با شماره ی سیستم چک میکنیم که برابر باشد و همچنین برای اینکه مطمئن باشیم فریم هایی که میرسند، مربوط به قسمتهای کوچک از یک فایل است از اسم فایل که در فریم ذخیره شده است استفاده میکنیم.در صورتی که در حالت خواندن از پایپ، فریمی بیاید که فریم مربوط به فایل ما نباشد آن را نیز می توان handle کرد.

زمانی که همه ی فریم های مربوط به یک فایل رسید، بر اساس num که مشخص میکند این فریم بخش چندم از فایل کوچک شده است، فریم ها را مرتب کرده و فایلی به فرمت out-fileName میسازیم و خروجی را داخل آن می نویسیم.

Lookup table:

در کلاس سوئیچ ، یک مپ برای Lookup table در نظر گرفته شده است.که این مپ شماره سیستم را میگیرد و named pipe مربوط به سوئیچ را میدهد.(فایلی که روی آن می نویسیم).

broadcast:

در این حالت بر روی تمام named pipe مربوط به سوئیچ می نویسیم.در واقع سیستم از این named pipe ها می خواند.

توضیح فایل های موجود در پروژه:

Main.cpp : در این فایل با استفاده از زمانی که دستور fork ساخت سوییچ و سیستم می آید، پراسس ساخته میشود و به ازای هر سوئیچ و سیستم نیز یک unnamed pipe میسازیم.همچنین دستورات وارد شده بررسی میشود و اطلاعات مربوط به دستور بسته به دستور به سوئیچ و سیستم فرستاده میشود.در زمان وصل شدن دو سوئیچ و یا سوئیچ و سیستم دو named pipe برای تبادل اطلاعات میسازیم.

Define.h: در این فایل struct هایی که در برنامه استفاده میشود، define شده است. استراکت ReadInfoFromUnnamedPipe شامل اطلاعات دستورات char* buf و valread که نشان دهنده ی وضعیت فایل خوانده شده است میباشد.
استراکت ReadInfoFromNamedPipe شامل فریم است زیرا از named pipe برای ارسال فریم ها استفاده میکنیم. valread که نشان دهنده ی وضعیت فایل خوانده شده است میباشد.

Function.h/Function.cpp: شامل توابعی که در بیشتر فایل ها بکار میرود.

Switch.cpp/Switch.h: در این فایل کلاس Switch را داریم.خواندن از پایپ unnamed و named ها، فرستادن فریم ها، connect شدن و ... بررسی میشود.
System.cpp/System.h: در این فایل کلاس System را داریم.خواندن از پایپ unnamed و named ها، ساختن فریم ها و فرستادن فریم ها، دریافت فریم ها و چسباندن آنها و نوشتن نتیجه در فایل، connect شدن و ... بررسی میشود.

Spanning tree:
اگر یک سیستم دارای دور باشد برای از بین بردن دور این الگوریتم را روی آن اجرا میکنیم. هر سوئیچ به بقیه سوئیچ ها اطلاعاتی مبنی بر اینکه root است ارسال میکند.root سویچی میشود که کمترین id را داشته باشد.اطلاعاتی که فرستاده میشود شامل ID of sender, ID of root, distance from sender to root است.هر سوئیچ اگر ID که دریافت می کند، کوچکتر از ID خودش باشد، متوجه میشود که خودش root نیست.root همه ی سوئیچ های دیگر را می پوشاند و loop را از بین میبرد. ممکن است root بعد از چند وقت از کار افتد که لازم است protocol دوباره اجرا شود.

Test:
دو تست فایل به نام های txt.1 و txt.2 در فایل های آپلود شده وجود دارد.
فایل txt.1 یک تست کیس کوچک است که فقط یک فریم است و خروجی آن در فایل out-1.txt است.
فایل txt.2 یک تست بزرگ شامل اعداد ۰ تا ۴۹۹ است که این فایل به دو فریم تبدیل میشود و نتیجه آن در فایل out-2.txt است.