

پروژه دوم درس شبکه های کامپیوتری

(شبیه سازی کارکرد شبکه)

سینا سلیمیان 810197528

کیاوش جمشیدی 810197486

## توضیح پروژه:

در این پروژه به طراحی سیستمی مشابه با کارکرد یک شبکه اینترنتی به کمک تعدادی سویچ و سیستم پرداخته ایم.

این پروژه شامل 3 فایل `switch.cpp` ، `system.cpp` ، `manage.cpp` است.

در `manage.cpp` کارهای مربوط به گرفتن ورودی و ارجاع دادن به دو فایل دیگر در هنگام `send` ، `create switch` ، `create system` و ... انجام می شود.

تمام فایل های `named pipe` داخل فولدری به نام `cache` در کنار فایل ها ساخته می شوند و تست کیس ها نیز در فولدر `steps` قرار گرفته اند.

برای اجرای کد دستورات داخل فایل `command_in_terminal.txt` اجرا شوند.

## دستورات ورودی:

دستور `MySwitch` و `MySystem` :

در هنگام ساختن یک سویچ و یا سیستم جدید، به کمک سیستم `fork()` یک `process` دیگر ساخته می شود و به کمک سیستم `exec()` ، ادامه اجرا به فایل مربوطه ارجاع داده می شود. به این صورت سویچ و یا سیستمی جدید به طور موازی همیشه در حال اجرا می باشد.

برای برقراری ارتباط بین `process` ها (IPC) از مکانیزم `named pipe` استفاده می کنیم. به این صورت که برای هر سویچ و هر سیستم جدید، یک فایل جدید نیز ساخته می شود و به آن اختصاص داده می شود.

دستور send:

در این دستور ، از طریق named pipe مربوط به سیستم به او اجازه ی ارسال فایل داده می شود و سیستم نیز مقصد مورد نظرش را به سوییچ متصل به خود ارسال می کند تا در آن جا مسیر مناسب برای رسیدن به مقصد انتخاب شود.

دستور connect :

در این دستور هم به سوییچ مشخص شده و هم به سیستم مشخص شده از طریق name pipe اطلاع می دهیم که connection جدید برای چه پورتهای ( در صورت نیاز) و چه سیستم یا سوییچی ساخته شده است و آن را ذخیره می کنیم.

دستور connect\_Switch :

این دستور مشابه دستور connect است با این تفاوت که دو سوییچ را از طریق پورت های مشخص شده به یکدیگر متصل می کنیم.

## سیستم یادگیری Table در سویچ ها:

هر سویچ دارای یک وکتور تک بعدی به نام LookUpTable است که در صورتی که از شماره سیستم یا سویچ یک پورت در میان پورت های متصل به سویچ مطمئن بودیم، مقدار آن پورت را از 1- به شماره سیستم یا سویچ تغییر می دهیم.

همچنین هر سویچ دارای وکتوری تک بعدی به نام connections است که پورت هایی را که اشغال شده اند را نشان می دهند.

در صورتی که فایلی از سیستمی به سویچ فرستاده شود، سویچ مربوطه، مقدار پورت ورودی داده را در وکتور LookUpTable به شماره سیستمی که فایل را ارسال کرده تغییر می دهد. به این صورت به تدریج آدرس سیستم تمام پورت ها مشخص می گردند.

در هنگام ارسال فایل به سیستم مقصد، در صورتی که آدرس پورت در LookUpTable مشخص نبود، داده به تمام پورت ها broadcast می گردد.

برای ارسال فایل ها، اندازه هر frame را برابر با 1025 در نظر گرفته ایم. در صورتی که اندازه فایل بیش از این مقدار بود، آن را به قسمت های 1025 تایی تقسیم می کنیم و طی چند مرحله ارسال خواهیم کرد.

## : Spanning tree

در صورت وجود دور در میان سویچ ها، در فایل `manage.cpp` درخت پوشای کمینه را پیدا می کنیم و `connection` هایی را که غیر فعال می شوند را به سویچ ها اطلاع رسانی می کنیم. به همین منظور در هر سویچ یک وکتور به نام `Enables` داریم که پورت های فعال سویچ ها را به سویچ نشان می دهد و در صورت فعال بودن پورت، به آن ارسال می کند.