COSC 3360 – 6310 – Fundamentals of Operating System
Assignment #3 for Fall 2018: The unidirectional network.
Due on Monday, December 3 at 11:59:59 PM

OBJECTIVE

This project will familiarize you with the use of pthreads, pthread semaphores, and pthread condition variables.

THE PROBLEM

Two computers (A and B) are linked together using a unidirectional channel. To guarantee that there are no packet collisions, the channel is operated using a time division scheme giving 150 ms to one computer followed by 150 ms of transition time to guarantee that all the packets currently using the channel can get to the end before the other computer starts using the channel for 150 ms. The goal of the assignment is to simulate the operation of the unidirectional channel using posix threads, posix semaphores, and conditional variables.

Your program should consist of:

1. The *main thread:* it creates the channel thread and the packet threads according to the input specifications.
2. The *channel thread*: it executes the following steps: a) allow packets to be transmitted from A to B for five seconds; b) prevent the access to the channel for five seconds; c) allow packets to be transmitted from B to A for five seconds; d) prevent the access to the channel for five seconds; and e) return to step a.
3. One *packet thread* per line in the input file (without including the first line): it simulates the packet that will be transmitted thru the channel. The channel only handles the following types of packets: FTP, HTTP, SSH, and SMTP.

The input to your program consists of the maximum number of packets that the channel can handle simultaneously (channel capacity), and an ordered list of packets to be transmitted thru the channel as in:

```
10              //      Channel capacity
FTP 1 AB 1      //      Packet 1
HTTP 2 BA 3     //      Packet 2
SSH 1 AB 1      //      Packet 3
```

The value in the first line of the input file represents the maximum number of packets that can use the channel at a particular time. The following lines represent the packets that the computers want to transmit. Each packet has the following format:

Packet Type: FTP, HTTP, SSH, or SMTP
Packet arrival time: it represents the elapsed amount of time (seconds) since the arrival of the previous packet.
Packet direction: AB = from computer A to B
BA = from computer B to A
Packet travel time: it is the amount of time (seconds) that the packet uses the channel.

Your program will terminate when all packets have been transmitted thru the unidirectional channel.

YOUR OUTPUT

Your program should track both the current state of the unidirectional channel and the packets using it. It should print a message each time the traffic direction changes and a message each time a packet: (a) arrives at the system; (b) enters the channel; and (c) leaves the channel. This second message should identify each packet by: (a) its sequence number in the input file (starting from 1); (b) its type (FTP, HTTP, SSH, and SMTP); and (c) the packet's direction (from A to B or from B to A). Based on the previous input, we have the following output:

```
The channel is now open from A to B.
Packet #1 (FTP) going to B arrives at the system.
Packet #1 (FTP) going to B is using the channel.
Packet #1 (FTP) going to B exits the channel.
Packet #2 (HTTP) going to A arrives at the system.
Packet # 3 (SSH) going to B arrives at the system.
Packet # 3 (SSH) going to B is using the channel.
The channel is now closed to ALL traffic.
Packet # 3 (SSH) going to B exits the channel.
The channel is now open from B to A.
Packet # 2 (HTTP) going to A is using the channel.
Packet # 2 (HTTP) going to A exits the channel.
```

At the end of the simulation, your program should also print a summary with:
1. The total number of packets (classified by destination and packet type);
2. The total number of packets that had to wait because the channel was full (classified by destination and packet type).

This summary could look like:

```
Packets transmitted:
        From A to B:
                1 FTP, 0 HTTP, 1 SSH, 0 SMTP.
        From B to A:
                0 FTP, 1 HTTP, 0 SSH, 0 SMTP.
Packets that waited:
        From A to B:
                0 FTP, 0 HTTP, 0 SSH, 0 SMTP.
        From B to A:
                0 FTP, 0 HTTP, 0 SSH, 0 SMTP.
```

Note: Your program must use I/O redirection. You can safely assume that the input files will always be in the proper format.

These specifications were written on **Friday, November 9, 2018.** Please refer to the course web site for corrections and updates.