CSC360 Assignment 2
Design before Coding

Leanne Feng
V00825004

1. How many threads are you going to use? Specify the task that you intend each thread to perform.

Depends on how many flows are there in input file (eg. Flow.txt) since the number of threads is the same as the number of flows. Each flow has its own id, arrivalTime, transTime and priorty.

2. Do the threads work independently? Or, is there an overall “controller" thread?

Threads work independently. There is not an overall “controller" thread.

3. How many mutexes are you going to use? Specify the operation that each mutex will guard.

Only one mutex is going to be used. It is used for lock and unlock the transmitting flow to protect the current transmitting flow not being modified.

4. Will the main thread be idle? If not, what will it be doing?

Yes, the main thread will be idled by pthread_join().

5. How are you going to represent flows? what type of data structure will you use?

Each flow is represented by a typedef structure, each type has arrivalTime, transTime, priority and id. Then, all flows are stored in array flowList[MAXFLOW].

6. How are you going to ensure that data structures in your program will not be modified concurrently?

I am going to lock the transmitting flow to protect the flow not being modified and use pthread_cond_t_wait to let other flows wait until the transmitting flow finished. Then the data will not be modified.

7. How many convars are you going to use? For each convar:

Only 1 convars I am going to use.

(a) Describe the condition that the convar will represent.

When there exists a flow in transmitting, pthread_cond_wait(&trans_cvar, &trans_mtx) will be called to make other flows wait until the first flow in queue gets the signal pthread_cond_broadcast(&trans_cvar). So only one flow can be transmitted.

(b) Which mutex is associated with the convar? Why?

The transmitting mutex is associated with the convar. Because the wait function release mutex and cause the calling thread to block on the condition variable cond. Only by call signal or broadcast can let the thread to continue.

(c) What operation should be performed once pthread_cond_wait() has been unblocked and re-acquired the mutex?

It needs pthread_cond_broadcast() or pthread_cond_signal() and lock the transmitting mutex.

8. In 25 lines or less, briefly sketch the overall algorithm you will use.

After read the input file (eg. flow.txt) and extract the number and information of the flows, each flow will be stored in flowList. A pthread_create will create a thread and go to thrFunction since it is the start_routie. Then, a method getmyTime() will be used to calculate the total elapsing time for a flow, the method requestPipe() will be called: if a flow arrives first and queue is empty then it will immediately run and be the transmitting flow and then signal otheer flows to continue otherwise the the other flows will be added into queueList and a comparing method will be used to order the flows and identify which flow will be transmitted next. If there exists a flow in transmitting other flows will all be waiting until one of the flow (the top flow in queue) is signaled. Then the top flow in queueList will be the current transmitting flow.