

## Assignment Description

Our submission consists of the following files :

context.c : This consists of all the functions building the context for thread.  
context.h

multiLevelQueue.c : This consists of the two queues, representing different levels of priority.  
multiLevelQueue.h

scheduler.c : Scheduler handles the scheduling of threads based on round robin.  
scheduler.h

threadStructure.c : This represents all functions modifying the thread, status updates etc.  
threadStructure.h

timer.c : This manages the timer for the scheduler to run the threads.  
timer.h

my\_pthread\_t.c : It contains all the standard pthread library functions.  
my\_pthread\_t.h

Maintainance Cycle : Using multilevelqueue we run the higher level priority queue for more but shorter

duration buffers. The lower level priority queue is run lesser number of times but for longer buffers. Once a thread passed its time slot in a queue, it is shifted to the next queue. Our maintainance cycle runs every 150 ms.

Priority Inversion : For priority inversion, we are verifying the priorities every time a thread is yielding.

A thread does not yield for a lower priority thread. In case such a situation arises, we invert priorities.

To run the code and to see the benchmarks:  
run the following commands:  
make  
build/demo

Included a file called ss.txt which shows the output of our program. We ran seven different

threads in parallel to test our code and this according to us should test all the conditions possible.

Thread1: Prints input arguments, normal thread.

Thread2: A thread that ends before it's time slice expires.

Thread3: Tests thread yield.

Thread4, Thread5: Checks mutexes

Thread6, Thread7: Checks join.

Tested on:

[java.cs.rutgers.edu](http://java.cs.rutgers.edu)

Submitted By :

Aditya Sastry Ambadipudi

Shubhank Varshney

Zhilmil Dhillon