

Major Program 2: Condition variables and the readers and writers problem

Due: Friday March 3 at 11:59 pm

The objective of this assignment is to give you some practical experience using Condition Variables in constructing a strict FIFO solution to the Readers and Writers problem.

As with the last assignment, I have provided the "main" program. It is in the `assns/mp2` directory and its name is `rdrwrtr.c`. There is also a header file named `enqdeq.h` that you will need to use in that directory.

Your mission is to create a module called `enqdeq.c` which contains the functions `enqueue()` and `dequeue()` that are used to allocate and free the resource. Your function must provide for:

- 1 - concurrent reading of the resource
- 2 - strict FIFO queueing (i.e. Readers MUST NOT pass waiting Writers)
- 3 --Only one writer can write at a time and writing can't occur while reading is ongoing.

When `enqueue()` is called and the resource is not available `pthread_cond_wait()` must be called to block the calling thread.

When `dequeue()` is called and it is possible for a blocked thread to now read or write `pthread_cond_broadcast()` must be called to wake up ALL blocked threads.

When a thread wakes up it must ensure that its *qel* is at the front of the queue before it proceeds. A reading thread must reissue `pthread_cond_broadcast()` before returning from `enqueue()` if the new head of the wait queue is also a reader.

The file enqdeq.h contains function prototypes and data structures. Your solution MUST NOT require any modifications to my main program.. You may modify or enhance enqdeq.h (as long as you don't change the prototypes or necessitate mods to the main function.)

Your routine will be tested using various configurations of the main program provided. In particular:

- 1 - The numbers of readers and writers will be varied
- 2 - The duration of the microsleeps will vary

How to submit your program:

NOTE: This procedure has **NOTHING** in common with "handin" nor "sendlab"
Do **NOT** even **TRY** to think about how they fit into this procedure because
THEY DON'T!!

<<<Do NOT turn in any core files, makefiles etc.>>>

You must turn in 2 files: enqdeq.c and enqdeq.h

1. From any departmental Solaris system *ssh* to workstation *jmw*
2. The submission directories lie in the directory `/local/jmw2/322/mp2` which is available **ONLY IF YOU HAVE LOGGED INTO WORKSTATION jmw**. Each student has a subdirectory of `/local/jmw2/322/mp2`. The name of your subdirectory is your userid (in the example we will assume your id is *wjsmith*).
3. copy (via the `cp` command) required file to your subdirectory in `/local/jmw2/322/mp2`

For example:

```
cp /home/wjsmith/322/mp2/enqdeq.* /local/jmw2/322/mp2/wjsmith
```

Here you would (hopefully) obviously need to replace
`/home/wjsmith/322/mp2/enqdeq.*`
with wherever you have your program.

4. Don't modify the permissions on your subdirectory. They are set so that **ONLY** you can access your directory.

After you think you have turned your programs in, its a good idea to
`cd /local/jmw2/322/mp2/wjsmith`
and make sure your files are there and they still compile and work correctly.