the awake signal always assumes that it also has the lock for qacc this is important.

my solution also assumes always at least one person of each gender because wo that this solution is just not needed

Unisex bathroom problem

For the purposes of this solution I assume that there is at least one person of each gender.

My solution to this problem relies on a batch let-in idea. Basically we take a look at the current queue, and let in all the people waiting from one gender from the current queue. Then we do the same for the other gender.

This ensures fairness because every time you are unable to access the bathroom one time, you are guaranteed to be able to access it the next time that it is open for your gender. And because we always switch this will happen eventually.

Methodology

There are really two separate queues, one for men and women, they both have their own semaphores. The queues are made of all the threads that are waiting on the awake semaphore to be released. To ensure that only the already present threads are let in, before the semaphore that wakes up the threads is increased, the semaphore that allows for the addition to queues is decremented.

After that all the threads let in increase the counter of inside, and wait until crucially wait until all of the waiting threads are inside. I just have each one check if the queue is open again, but checking if there waiting counter is also ok, but makes them busy-wait.

After that they all exit the bathroom by decrementing the inbathroom counter. All but the last one will go to beginning of their loops to sleep. The last one hands tells the other gender to wake up and start going into the bathroom.

This is done by waiting until there is actually someone inside the other queue, and then locking their access to it, and decrementing the awake semaphore. Then all the threads will pass each other the “awake baton” and only the ones already in the queue will be able to enter.

If any thread is done, it also triggers a release of both semaphores that wait just to make sure it all works.

Here is a sample execution

```
mint@mint-Legion-7-16ACHg6:~/Documents/concurrent_programing/HM3$ ./unisex 10
gender: 0, number inside 311
your gender waiting 0, other gender waiting 235
sleeping 85, those who didn't get into queues 9

gender: 1, number inside 255
your gender waiting 0, other gender waiting 95
sleeping 245, those who didn't get into queues 45

gender: 0, number inside 111
your gender waiting 0, other gender waiting 85
sleeping 408, those who didn't get into queues 36

gender: 1, number inside 94
your gender waiting 0, other gender waiting 84
sleeping 429, those who didn't get into queues 33

gender: 0, number inside 100
your gender waiting 0, other gender waiting 78
sleeping 419, those who didn't get into queues 43

gender: 1, number inside 85
your gender waiting 0, other gender waiting 96
sleeping 436, those who didn't get into queues 23

gender: 0, number inside 109
your gender waiting 0, other gender waiting 73
sleeping 405, those who didn't get into queues 53

gender: 1, number inside 83
your gender waiting 0, other gender waiting 104
sleeping 427, those who didn't get into queues 26

gender: 0, number inside 109
your gender waiting 0, other gender waiting 94
sleeping 406, those who didn't get into queues 30

gender: 1, number inside 114
your gender waiting 0, other gender waiting 90
sleeping 367, those who didn't get into queues 69


The execution time is 0.048309 sec
mint@mint-Legion-7-16ACHg6:~/Documents/concurrent_programing/HM3$
```