8.2

to allow the master thread to change the thread ID -> jobid need a write mode pthread\_rwlock to protect.

A job’s thread ID could be changed during working. So a thread could remove a job whose threadID not equal to itself’s.

We could add a count condition and a mutex to make master change the job’s thread ID that no thread is processing.

8.4

1. Lock a mutex (pthread\_mutex\_lock).

2. Change the condition protected by the mutex.

3. Signal threads waiting on the condition (pthread\_cond\_broadcast).

4. Unlock the mutex (pthread\_mutex\_unlock).

Some threads may executing before mutex unlock.

1. Lock a mutex (pthread\_mutex\_lock).

2. Change the condition protected by the mutex.

3. Unlock the mutex (pthread\_mutex\_unlock).

4. Signal threads waiting on the condition (pthread\_cond\_broadcast).

Already executing(not awaken by signal) thread can get the mutex between 3 and 4,then it can access the protected condition variable. After that, we signal threads waiting on the condition. However the condition variable could be changed.