CS241 Lecture 16 Lawrence Angrave
Semaphores

Condition Variables Review
1. I have two threads blocked on a condition variable 'cv1'
while(cloudy == 42) p_cond_wait(&cv1, &m);

How do I wake them both up?
p_m_lock(&m);
cloudy = 42;
_____
p_m_unlock(&m);

2. What must be locked before calling p_cond_wait ?

3. How do I use counting semaphores?

4. What is a ring buffer?

5. How can I use counting semaphores to implement a ring buffer?
pthread_mutex_t m = PTHREAD_MUTEX_INITIALIZER:
void init() {
  sem_init(_____, 0, _____);
  sem_init(_____, 0, _____);
}
void sync_enqueue(work_t *work) {

}
work_t* sync_dequeue(){

}

Some more C functions for you:
`sigprocmask pthread_sigmask pthread_self() atexit sigaction`

Psuedo code Candidate # 1

wait until your flag is lowered raise my flag *// Do Critical* *Section stuff* lower my flag	wait until your flag is lowered raise my flag *// Do Critical* *Section stuff* lower my flag

// Threads do other stuff and then will repeat
Problems with 1?

Candidate #2

raise my flag wait until your flag is lowered *// Do Critical* *Section stuff* lower my flag	raise my flag wait until your flag is lowered *// Do Critical* *Section stuff* lower my flag

// Threads do other stuff and then will repeat
Problems with 2?

Candidate #3

wait until my turn (turn==id?) *// Do Critical* *Section stuff* turn = *yourid*	wait until my turn (turn==id?) *// Do Critical* *Section stuff* turn = *yourid*

// Threads do other stuff and then will repeat
Problems with 3?

What have I made?

```
01  pthread_mutex_t m = PTHREAD=MUTEX_INITIALIZER;
02  pthread_cond_t cv = PTHREAD_COND_INITIALIZER;
03  int  cake = 0;
04
05  void  decrement() { // Waits if nonzero
06   lock(&m)
07   while(cake == 0) p_cond_wait(&cv, &m);
08   cake --;
09   unlock(&m);
10  }
11
12  void increment() {
13    lock(&m);
14    cake ++;
15    if( _____ ) p_cond_signal(&cv);
16    unlock(&m);
17  }
18
```