CSSE 332 -- Operating Systems

Condition Variables II

Name: Solution Key

Question 1. (5 points) Consider the following sequence of events, we have three threads, $\mathbf{T}_1$, $\mathbf{T}_2$, and $\mathbf{T}_3$. Also, assume that $t_1 < t_2 < t_3$.

Time	Thread	Event
...	...	...
t_1	$\mathbf{T}_1$	`pthread_cond_wait(&c, &m);`
...	...	...
...	...	...
t_2	$\mathbf{T}_2$	`pthread_cond_wait(&c, &m);`
...	...	...
...	...	...
t_3	$\mathbf{T}_3$	`pthread_cond_signal(&c);`

Some time after t_3, which one of the waiting threads ($\mathbf{T}_1$ and $\mathbf{T}_2$) would wake up and start executing?

A. $\mathbf{T}_1$.

B. $\mathbf{T}_2$.

C. Neither $\mathbf{T}_1$ nor $\mathbf{T}_2$.

D. Other: Cannot determine

Question 2. (15 points) The following pieces of code contains errors, find and fix these errors.

```
pthread_cond_t c = PTHREAD_COND_INITIALIZER;
pthread_mutex_t lock = PTHREAD_MUTEX_INITIALIZER;

void *thread1(void *unused) {
  // some code here...

  // need to wait on a condition variable
  while(!ready) {
    pthread_cond_wait(&c, &m);
  }
}

void *thread2(void *unused) {
  // some code here

  ready = 1;
  pthread_cond_signal(&c);
}
```

Thread 1 is accessing the `ready` variable without having the lock `m`.

```
pthread_cond_t c = PTHREAD_COND_INITIALIZER;
pthread_mutex_t lock = PTHREAD_MUTEX_INITIALIZER;

void *thread1(void *unused) {
  // some code here...

  // need to wait on a condition variable
  pthread_cond_wait(&c, &m);
}

void *thread2(void *unused) {
  // some code here

  pthread_mutex_lock(&lock);
  ready = 1;
  pthread_cond_signal(&c);
  pthread_mutex_unlock(&lock);
}
```

Thread 1 does not check any conditions on the `ready` state variable.

```c
pthread_cond_t c = PTHREAD_COND_INITIALIZER;
pthread_mutex_t lock = PTHREAD_MUTEX_INITIALIZER;

void *thread1(void *unused) {
  // some code here...

  // need to wait on a condition variable
  pthread_mutex_lock(&lock);
  if(!ready) {
    pthread_cond_wait(&c, &m);
  }
  pthread_mutex_unlock(&lock);
}

void *thread2(void *unused) {
  // some code here

  pthread_mutex_lock(&lock);
  ready = 1;
  pthread_cond_signal(&c);
  pthread_mutex_unlock(&lock);
}
```

Thread 1 should **always** use a `while` loop before waiting on a condition variable.