

Week #7: Synchronization
Exam Review Question
CS 241: Fall 2013

These questions are provided to you to help you study material covered in CS 241 that may appear on the final exam. These exact questions may or may not appear on the final exam, but the topics they cover will almost certainly be on the final exam.

1. Consider the code below:

```
int COUNT = 1000000, X = 0;

void *thread(void *ptr) {
    int i;
    for (i = 0; i < COUNT; i++) { X++; }
    return NULL;
}

void main() {
    pthread_t pid1, pid2;
    pthread_create(&pid1, NULL, thread, NULL);
    pthread_create(&pid2, NULL, thread, NULL);

    pthread_join(pid1, NULL);
    pthread_join(pid2, NULL);

    printf("The value of X: %d\n", X);
}
```

What is the value of **X** that is printed to **stdout**?

2. How does a mutex differ from a semaphore?
3. Explain how a semaphore can be used as a mutex to protect a critical section that should only be accessed by a single thread at a time.
4. Implement a semaphore using a single mutex and a single conditional variable.
5. What is the difference between `pthread_cond_signal()` and `pthread_cond_broadcast()`? Describe a situation where a broadcast is required and a conditional signal is not sufficient.
6. Under what conditions does each of the synchronization primitives block? Do any of the primitives *always* block when called?

7. Consider the code below:

```
void main() {
    pthread_mutex_t mutex;
    pthread_mutex_init(&mutex, NULL);

    pthread_cond_t cond;
    pthread_cond_init(&cond, NULL);

    pthread_cond_broadcast(&cond);
    pthread_cond_wait(&cond, &mutex);
}
```

Describe the result of running this program.