

Operating Systems

Assignment 5

(10 Points) Review Questions :

- Individual task
- **Due:** Next Section (one night before if you submit by email)

Chapter 5 Exercises (Starting at page 242):
No. 2, 3, 4, 5, 7, 10, 14, 15, 16, 17, 20, 23

(5 Points) Programming Task:

- Group task, (3 students at Max)The instructor should review within the lab
- **Due: Next Section** (email submission is not allowed)

2. A barrier is a tool for synchronizing the activity of a number of threads. When a thread reaches a barrier point, it cannot proceed until all other threads have reached this point as well. When the last thread reaches the barrier point, all threads are released and can resume concurrent execution.

Assume that the barrier is initialized to N —the number of threads that must wait at the barrier point:

```
init(N);
```

Each thread then performs some work until it reaches the barrier point:

```
/* do some work for awhile */  
barrier_point();  
/* do some work for awhile */
```

Using the Mutex lock of either POSIX API or Windows API, construct a barrier that implements the following API:

- **int init(int n)**—Initializes the barrier to the specified size.
- **int barrier_point(void)**—Identifies the barrier point. All threads are released from the barrier when the last thread reaches this point.

The return value of each function is used to identify error conditions. Each function will return 0 under normal operation and will return -1 if an error occurs. A testing harness is provided in the source code download to test your implementation of the barrier.