EECE.4810/EECE.5730: Operating Systems

Spring 2018

Lecture 6: Key Questions
February 14, 2018

1. (Review) What is a critical section? What are the requirements of a critical section?

2. (Example) Dekker's Algorithm is the first known solution to the critical section problem. Threads or processes sharing a critical section share the following variables:

```
boolean flag[2];       // Both initially false
int turn;
```

The structure of process P_i (i == 0 or 1) is shown below. The other process is P_j (j == 1 or 0). Prove the algorithm shown below satisfies the three requirements for a critical section:

```
do {
   flag[i] = true;

   while (flag[j] == true) {
      if (turn == j) {
         flag[i] = false;
         while (turn == j)
            ;       // do nothing
         flag[i] = true;
      }
   }

      /* critical section would be placed here */

   turn = j;
   flag[i] = false;

} while (true);
```

3. (Review) What is a lock, or mutex?

4. (Review) Describe how to implement a thread-safe queue with locks.

5. Suppose you wanted the dequeue() function to wait if the queue is empty. How can you avoid busy waiting?

6. Describe condition variables and the operations one can perform on them.

7. What is a monitor? How can you implement a thread-safe queue using monitors?