EECE.4810/EECE.5730: Operating Systems

Spring 2017

Lecture 5: Key Questions
February 1, 2017

1. What is a thread?

2. Describe the basic differences between threads and processes.

3. In a multithreaded process, what information needs to be private to a thread? What can be shared?

4. What are the major benefits of multithreading?

5. Explain concurrency and parallelism.

6. **Example:** Your system has four processors available for scheduling. The application running on this system does the following:
   - Reads all input sequentially from a single file
   - Process input data and compute final results
     - Entirely CPU-bound during this part—no I/O
   - Print all output sequentially to a single file

To improve the performance of this application by multithreading it, determine:
- How many threads should you create to handle input and output? Why?
- How many threads should you create to handle computation? Why?

7. Describe how non-deterministic ordering can present problems in a multithreaded program.

8. In the example below, what are the possible outputs? What are the impossible outputs? What are these two threads sharing?

Thread 1	Thread 2
Print ABC	Print 123

9. In the example below, what are the possible results, assuming y is initially 10? What are these threads sharing?

Thread 1	Thread 2
x = y + 1	y = y * 2

10. In the example below, what are the possible results, assuming x is initially 0?

Thread 1	Thread 2
x = 0	x = 0
x++	x--

11. Explain what an atomic operation is.

12. In the example below, which thread finishes first? Is the "winner" guaranteed to print first? What's required to ensure that one thread actually does finish first?

Thread A

```
i = 0
while (i < 10)
   i++
print "A done"
```

Thread B

```
i = 0
while (i > -10)
   i++
print "B done"
```

13. Explain why synchronization is necessary.