Scheduling Review Chapter 5

5.1, 5.2, 5.3, 5.4, 5.5, 5.6, 5.10, 5.11, 5.13

1. What was it useful that the previous Linux scheduler (the O(1) scheduler) was O(1) runtime? What specific task was O(1) in runtime?

2. Why is a red-black tree used in the CFS Linux Scheduler? What benefits does the red-black tree provide over a priority queue or set of priority queues?

3. How was priority provided in the O(1) scheduler?

4. How is priority provided in the CFS scheduler?

5. What is the O( ) runtime to insert or delete a node from a red-black tree (average case, worst case)?

6. Explain why the workload of a webserver vs the workload of a desktop may require a different scheduling algorithm. Which evaluation criteria are most important in each situation?

7.What does it mean for a scheduling algorithm to be *fair*?

8. What is meant by *real-time scheduling*?

9. Why would the Linux Scheduler prefer to schedule a task on the same core that that task was last schedule on (in other words, why doe the Linux Schedule implement processor affinity)?

10. If you notice the CPU on your Lab machine is running at 90% capacity, do you think the scheduler is doing a good job or bad job of scheduling tasks? Justify your answer.