Homework #2

Due in class on Monday 19 October 1998

1. Consider the following code, with the assumption that each executes in a separate thread:

   ```
   i=o;                 i=0;
   while (i < 10)       while (i > –10)
       i++;                 i—;
   printf("a wins");    printf("b wins");
   ```

   a. Is it <u>guaranteed</u> that someone wins? Explain.

   b. Is it <u>possible</u> for someone to win? Explain.

2. Implementing semaphores by disabling interrupts has several problems, one of which is that it doesn't work with on multiprocessors. Explain why this statement is true.

3. In several situations, such as when a thread tries to release a lock that it hasn't acquired, it's recommended that the Nachos ASSERT function be used to crash the operating system. Is this a good idea or a bad idea? Explain your answer.

4. (5.3 from OSC, parts (a) through (c) only, modified as follows) The processes are assumed to have arrived in the order <u>P5, P4, P3, P2, P1</u>, all at time 0.

5. (Exercise 5.6 from OSC) What advantage is there in having…

On the homework in this class, I expect clear, well-focused answers to the questions asked, answers that balance brevity and detail. While brevity is desirable, one-sentence "answers" will probably not suffice. While detail and examples are desirable, two-page answers to each question are probably unnecessary. Finally, simply copying material directly from the book or my notes is not acceptable, as that doesn't convince me that you even understand your own answer; I want you to answer the questions in your own words.