4. a) A race condition is a situation that is undesirable which occurs when a device or system tries to perform more than one operation simutaniously but the operations must be done in a certain sequence in order to be correct.

An example of a race condition is when two threads try to change the same value. One thread does a “check-then-act” meaning check the condition and then do an action while the other thread changes the value between checking of the condition and the action.

b) Unlike in a uniprocessor system where using CLI prevents a race condition from occuring, in a multiprocessor systems when executing a CLI instruction only disables interrupts on a particular processor thus leaving the other processors free to handle interrupts.

This does not prevent race condtions because the other processors are able to handle interrupts and are able to start executing the offending interrupt handler.

In order to prevent a race condition from occuring in a multiprocessor system, CLI and locks must be used simutaniously.

5. A semaphore is a way to lock a resource in order to prevent a race condition. A semaphore is like a mutex however a semaphore can support a fixed number of simultaneous callers. Once the limit is hit, it will block the rest of the callers until one of the slots are freed.

6.

7. a) <http://www2.cs.uregina.ca/~hamilton/courses/330/notes/allocate/allocate.html>

b)