CPS 536: Exam 1 **Name:**_____

Instructor: Dr. Clark February 12, 1996

This test is closed book and closed notes. You may use a calculator. Answer the questions in the space provided. When answering questions, please state any and all assumptions you are making. For questions involving calculation, indicate your final answer by circling it.

Part 1 (20 points)

1. What is the difference between supporting multiprogramming and supporting time-sharing in an operating system? What might you do differently when designing a system for one versus the other?

2. In class we discussed 8 components of an operating system. List and briefly describe **5** of those components.

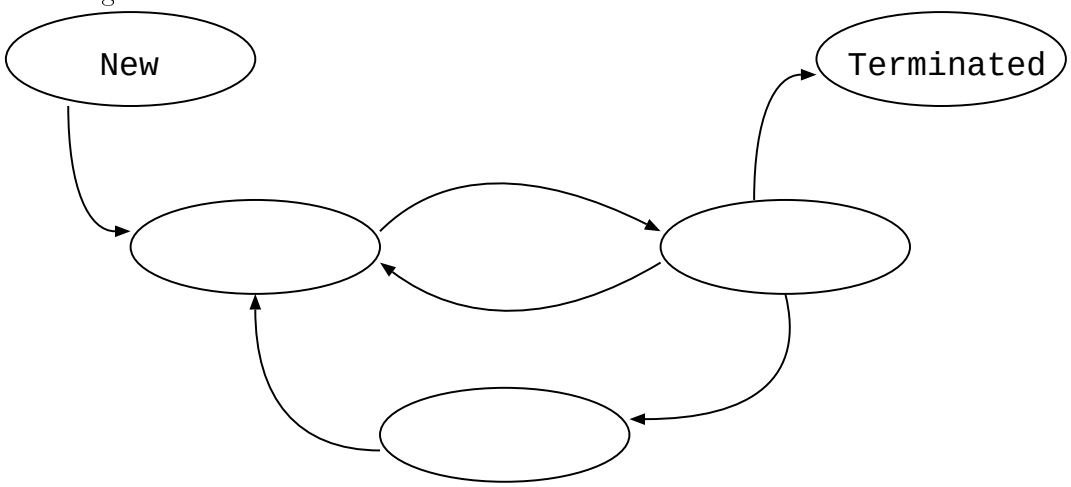
CPS 536: Exam 1 **Name:**_____

Part 2 (30 points)

1. What is the difference between a process and a thread?

2. What are the advantages and disadvantages of writing programs using threads rather than processes?

3. The following is a process state transition diagram. Complete the diagram by labelling the states and indicating the cause of each transition.



Part 3 (20 points)

In general, the three synchronization primitives we discussed are considered to be equivalent since any mutual exclusion problem that can be solved with one can also be solved with the others. Despite this equivalence in power, the primitives each have a unique flavor that makes them more appropriate for specific problems.

1. We defined locks to be very similar to semaphores but with some specific restrictions in how they are both implemented and used. What are those restrictions?

2. If I write a program using locks and then replace the locks with semaphores, would my program be guaranteed to work as it did before? Explain.

3. Why do these restrictions on locks help the programmer using locks in a program? (Note: I'm not referring to the person implementing locks. I'm referring to the person using the locks.)

Part 4 (20 points)

1. We discussed several different criteria for comparing scheduling algorithms. For which criteria can it be shown that the Shortest Job First scheduling algorithm is optimal?

2. If it is optimal, why don't we use SJF in every operating system?

3. What is the difference between Round Robin scheduling and First Come First Serve scheduling?

Part 5 (10 points)

1. In an operating system it is often necessary to support some mechanism for providing atomic operations. How do we guarantee atomicity in NACHOS?

2. Would this approach work in a multi-processor system? Explain your answer.