

Name: \_\_\_\_\_

---

CS 4/53201

**Exam #2**

**Operating Systems**

**Friday 13 November 1998**

---

1. Define each of the following terms. (5 points each = 25 points)

a. mutual exclusion

b. atomic operation

c. busy waiting

d. CPU burst

e. hold and wait

Name: \_\_\_\_\_

2. Consider this definition of semaphores, which we will call version 1:

<u>wait(s):</u> s = s - 1 if (s < 0) block the thread that called wait else continue into critical section	<u>signal(s):</u> s = s + 1 if (s <= 0) wake up a waiting thread
---	---

Also consider this definition, which we will call version 2:

<u>wait(s):</u> if (s <= 0) block the thread that called wait s = s - 1 continue into critical section	<u>signal(s):</u> if (a thread is waiting) wake up a waiting thread s = s + 1
--	--

How do these two definitions compare? Assuming they're implemented correctly (as atomic operations, etc.), do they both work? If not, which one doesn't work, and why? If they do both work, is there any difference between them? (10 points)

3. Lock ACQUIRE and RELEASE operations can be implemented fairly directly using semaphore WAIT and SIGNAL operations, respectively. Can condition variable WAIT and SIGNAL operations also be implemented using semaphore WAIT and SIGNAL operations, respectively? Explain your answer. (15 points)

Name: \_\_\_\_\_

4. For each of the following scheduling algorithms, (i) clearly indicate when the CPU scheduler is invoked, and (ii) briefly explain the criteria used by the CPU scheduler to pick the next process to execute. (5 points each =15 points)

a. FCFS scheduling

b. Nachos scheduler

c. SRT scheduling

5. One method for dealing with deadlock is detection and recovery.

a. Explain how deadlock can be detected in a system with single instances of each resource type. (15 points)

Name: \_\_\_\_\_

- a. Explain how deadlock can be detected in a system with multiple instances of each resource type. (20 points)