HW4

Assigned	9/20/05
First possible pop quiz	9/27/05
Last possible pop quiz	9/29/05

1. A commonly used group synchronization mechanism is called **barrier.** Some computations are divided into phases and have the rule that no thread may proceed into the next phase until all threads are ready to proceed to the next phase. The behavior may be achieved by letting each thread execute *barrier.done(n)* at the end of each phase, where n is the number of threads in the computation. A call to *barrier.done* blocks until all of the n threads have called *barrier.done.* Then all thread proceed. Write a monitor that implements Barrier.

2. Synchronization within monitors uses condition variables and two operations, **wait** and **signal.** A more general form of synchronization would be to have a single primitive, **waituntil,** that had an arbitrary Boolean predicate as parameter. Thus, the **signal** primitive would no longer be needed. Explain why this scheme is not used. (Think about the implementation)

3. (1) Can a system be in a state that is neither deadlocked nor safe?
(2) A system has p processes each needing a maximum of m resources and a total of r resources available. What condition must hold to make the system deadlock free.

4. A system has 4 processes and 5 allocatable resource. The current allocation and maximum needs are as follows:

	Allocated	Maximum	Available
Process A	1 0 2 1 1	1 1 2 1 3	0 0 x 1 1
Process B	2 0 1 1 0	2 2 2 1 0	
Process C	1 1 0 1 0	2 1 3 1 0	
Process D	1 1 1 1 0	1 1 2 2 1	

What is the smallest value of x for which this a safe state.

5. Compare the circular-wait scheme with the various deadlock-avoidance schemes (like the banker's algorithm) with respect to the following issues:
    a. Runtime overheads

b. System throughput.