Name:

Operating Systems (COP 6611)
Spring 2009
Final Exam

Closed books, notes, cell phones, PDAs, iPods, laptops, etc. No headphones, please. You may use a simple calculator (but it is unlikely you will need it).

You have 120 minutes to solve 8 problems, all equal weight. The first 6 problems are required, the last 2 you can choose from 4 options. As with any exam, you should read through the questions first and start with those with which that you are most comfortable. If you believe that you cannot answer a question without making some assumptions, state those assumptions in your answer.

Points will be taken off for irrelevant information included in your answers.

Partial credit will be offered for meaningful progress towards solving the problems.

In recognition of and in the spirit of the academic code of honor, I certify that I will neither give nor receive unpermitted aid on this exam.

Signature:_______________________________________

1	
2	
3	
4	
5	
6	
7	
8	
Total	

1. **Short Attention Span:**
    a. In the context of operating systems, what do the following acronyms stand for?
        - NAS
        - SAN
        - TLB
        - MTTF
        - NFS
        - RAID

    b. **Your favorite**: Show how to emulate general (counting) semaphores using binary semaphores.

2. **Short-term Working Memory (yours):** Based on the results that you found while working on Project 3 (File System Measurements), explain for each of the following, how each would have affected the path you took for finding the file system properties.
    a. Pre-fetching disabled
    b. Caching disabled
    c. Write-through cache policy

3. **Memory**: Can the size of the working set of a process affect how much CPU time it gets and how often it gets the CPU? Support your answers with specific examples.

4. **Overview**: One new trend in operating systems is motivated by the wide-spread adoption of mobile devices, such as smart phones, PDAs, etc. Discuss the various aspects of an operating system that need to be re-designed to run on these devices. In your discussion, include topics such as performance metrics for various components of an OS, objectives that may differ from traditional systems, which components can be used unmodified from traditional OS and which components need to be changed. Justify your answers.

5. **Distributed File Systems:** In distributed file systems, what is the difference between location independence and location transparency? Under what circumstances would a client prefer a location-transparent distributed file system? Under which circumstances would a client prefer a location-independent distributed file system? Discuss the reasons for these preferences.

6. **Distributed Barriers:** *Barriers* are a synchronization mechanism that requires all processes in a program to reach a given point before proceeding. Each process calls `barrier(b)` and each process blocks until all processes have made the barrier call; when the last process makes the barrier call, all processes are unblocked.

   Assume that you are writing distributed programs that run on a network of UNIX workstations. These workstations support standard stream (TCP/IP) and datagram (UDP) message communications.

   a. How would you implement barrier synchronization in this environment? Assume that you want to make the operations as fast as possible (minimal delay).
   b. Is load on the network (assume a typical Ethernet) likely to be a problem in your solution? If yes, describe in what cases this will occur. If no, explain why not.

Answer any 2 questions from the remaining 4:

7. **File Systems:** Consider the Log Structured File System (LFS) proposed by Rosenblum and Ousterhout.
    a. Describe the basic design and principles of LFS.
    b. What are the assumptions about trends in hardware and workloads that motivate the LFS design? Are these trends likely to make LFS more or less desirable in the next few years? Defend your claim.

8. **Deadlock in Distributed Systems:** Your company is building a distributed system, and you are asked to develop a scheme for dealing with the deadlock problem.

    a. Would you use a deadlock-detection scheme or a deadlock-prevention scheme? Explain your choice.
    b. If you were to use a deadlock-prevention scheme, which one would you use? Explain your choice.
    c. If you were to use a deadlock-detection scheme, which one would you use? Explain your choice.
    d. Suppose in a distributed systems there are several resource allocators, each responsible for some set of resource classes. A client process sends each request to the appropriate allocator. (Assume there is some mechanism for clients to figure out the appropriate allocator for each request.) A `check_for_deadlock` request can be sent to *any* allocator. Allocators may exchange messages with each other to detect deadlock. Outline a deadlock detection algorithm in this environment. You may ignore the possibility of process or communications failures, but assume that communication between allocators is relatively expensive.

9. **Atomic Transactions in Distributed Systems:** Consider a failure that occurs during the 2-Phase Commit (2PC) of a transaction. For each possible failure, explain how 2PC ensures transaction atomicity despite the failure.

10. **Global-state:** In the last paper read (Chandy & Lamport), the algorithm presented makes the following assumptions: infinite buffers, error-free channels, and ordered messages. Explain how eliminating each one of these assumptions affects the algorithm. Can you propose a modification to the global snapshot algorithm such that it still works under the relaxed assumptions?