**Name:** _____ **USC ID:** _____

# CSci 555 Final Exam

# Fall 2007

**Instructions:**

Show all work. No electronic devices are allowed. This exam is open book, open notes. You have **120 minutes** to complete the exam.

Please prepare your answers on separate sheets of paper. You may write your answers on the sheet of paper with the question (front and back). If you need more space, please attach a separate sheet of paper to the page with the particular question. **Do NOT extend your answer on the back of the sheet for a different question, and do NOT use the same extra sheet of paper to answer more than one question**.

In particular, **each numbered questions must appear on separate pieces of paper so that the exam can be split for grading**.

Be sure to include your **name** and **USC ID** number **on each page**.

There are **100 points** in all and **3 questions.**

| | Q1 | Q2 | Q3 | Total | Letter |
|---|---|---|---|---|---|
| Score | | | | | |

**Name:** _____ **USC ID:** _____

1. (35 points) Synchronization and File Systems
    a) Leases – For each of the following approaches to consistency, if they were to be implemented as a lease, list the corresponding lease term, and the rules for breaking the lease (i.e. if the normal rules for breaking a lease are not provided by the system, what are the effective rules of the mechanism. (16 points)
        a. AFS-2/3 Callback
        b. AFS-1 Check-on-use
        c. Time to live in the domain name system
        d. Locks in a transaction system

**Name:** _____ **USC ID:** _____

b) Log Structured File System – (19 points)

a. Discuss the similarity between a transaction system and the log structure file system. (4 points).

b. How does the log structure file system improve the performance of writes to the file system? (5 points)

c. Why does it take so much less time to recover from a system crash in a log structured file system than it does in the traditional Unix file system? How is recovery accomplished in the log structure approach? (10 points)

**Name:** _____ **USC ID:** _____

2. (25 points) Kernels

For a general purpose operating system such Linux, discuss the placement of services, listing those functions that should be provided by the kernel, by the end application itself, and by application level servers. Specifically, what OS functions should be provided in each location? Justify your answer and state your assumptions.

a) In the Kernel itself (5 points)

b) In the application itself (5 points)

**c)** In servers outside the kernel (5 points)

**d)** For a system supporting embedded applications, such as process control, what changes would you make in the placement of OS functions (i.e. what would be different than what you described in a-c). Justify your answer. (10 points)

**Name:** _____ **USC ID:** _____

3. (40 points) Design Problem

You have been hired to build a system to manage ticket sales for large concerts. This system must be highly scalable supporting near simultaneous request from the "flash crowds" accessing the system the instant a new concert goes on sale. The system must accept requests fairly, so that ticket consolidators are unable to "game the system" to their advantage through automated programs on well placed client machines located close to the servers in terms of network topology. To handle the load will require multiple servers all with access to the ticketing database, yet synchronization is a must as we can't sell the same seat to more than one person. The system must support several functions, among which are providing venue and concert information to potential attendees, displaying available seats, reserving seats, and completing the sale (collecting payment, recording the sale, and enabling the printing of a barcode ticket).

a) Describe the architecture of your system in terms of the allocation of functions across processors. Will all processors be identical in terms of their functionality, or different servers provide different functions, and if so which ones and why? (10 points)

b) Explain the transactional characteristics of your system. In particular, when does a transaction begin, and when does it commit or abort, and which processors (according to the functions described by you in part a) will be participants in the transaction. (10 points)

c) What objects will have associated locks and when will these object be locked and unlocked. (5 points)

d) How will you use replication in your system and how will you manage consistency of such replicated data (10 points)

e) How will you use distribution in your system (5 points)