UNIVERSITY OF DUBLIN

TRINITY COLLEGE

FACULTY OF ENGINEERING & SYSTEM SCIENCES

DEPARTMENT OF COMPUTER SCIENCE

B.A. (Mod.) Computer Science Trinity Term 1999

SS Examination

4BA8 - Distributed Systems

Thursday 27th May Regent House 14.00 – 17.00

Mr. Brendan Tangney
Mr. Stephen Barrettt
Dr. Vinny Cahill

Answer 5 Questions

Q1.

Describe in detail the architecture of a distributed system to perform control of road traffic in the greater Dublin area. Give an overall architectural diagram and explain how scale is handled. How robust is the system in the face of partial failure? What are the requirements on the network infrastructure?

Q2.

i) How does NFS handle the caching of server and client data.

ii) How can a single network wide file system be constructed using NFS. Why does the UID have to be sent with each NFS operation request and what ramifications does this have on the autonomy of node administrators in the assigning of UIDs to users.

Q3.

i) In distributed programming what is meant by **binding**? How is binding done with each of the following and what information is needed in each case: Unix Pipes, Sockets, Orbix.

ii) What are the fundamental differences between local and remote procedure calls.

iii) Outline the core idea at the centre of the "End-to-End Argument". Give a practical example of its application.

Q4.

What well-known problem with the two phase commit protocol is addressed by the three phase commit protocol? To what extent is the problem solved by three phase commit?

Outline the operation of the three phase commit protocol paying particular attention to the consequences of failure of the co-ordinator and/or individual participants at any time during the execution of the protocol.

Outline the possible consequences of network partition during the execution of the protocol. What does this behaviour tell you about the consequences of partitions in general?

Q5.

Define the ordering and reliability guarantees provided by each of the following ordered reliable multicast protocols:

- unsafe (i.e., non-dynamically uniform) totally ordered atomic multicast;
- safe (i.e., dynamically uniform) totally ordered atomic multicast;
- unsafe causally ordered atomic multicast;
- safe causally ordered atomic multicast;
- causally and totally ordered atomic multicast.

Given a causally ordered atomic multicast primitive, describe how that primitive might be used to implement a multicast protocol that is both totally and causally ordered.

Q6.

Distributed transactions are traditionally used when accessing (i.e., reading and updating) data that is shared by many concurrent and distributed users. Outline the four essential properties of an atomic transaction - the so-called ACID properties.

Outline the technique of *strict two-phase locking* and describe how this technique avoids problems such as lost updates, inconsistent retrievals, unrepeatable reads and dirty reads. Is the use of *two-phase locking* sufficient to avoid all of these problems?

Consider the design of a system to support the operation of a national lottery system based on the use of a single centralised database. Clients of the lottery purchase tickets corresponding to their choice of a sequence of numbers. Possession of a valid ticket guarantees that a client will be entitled to a share in any prize due to the sequence of numbers displayed on that ticket depending on how many other clients have chosen the same sequence of numbers. The single centralised database is to be used to record information about ticket sales. Distributed ticket agents will contact the central server as they sell each ticket. Discuss whether or not the use of atomic transactions is an appropriate paradigm for implementing such a system. In doing so, outline the important requirements on the design and any differences between the requirements for this system and other systems in which transactions are traditionally used.

Q7. Answer each of the following

(a) Why do computer systems fail?
(b) Outline the rationale for the technique of sender-based message logging and the main implications of the design.
(c) What are the short fallings of sockets as an IPC abstraction.
(d) A hypercube consisting of a number of processors connected by a fast network is not considered to be a distributed system - why not?