American University of Beirut

Department of Computer Science
CMPS 372 – Distributed Systems
Spring 2003-2004

Final Exam

Date: June 3rd, 8:00 – 10:00am.
Instructors: Jihad Boulos

Problem 1 (9 points): System Models.

Consider a communication service used in asynchronous distributed systems. In this service, messages may be lost, duplicated, delayed, or delivered too faster for the recipient to handle them, but those that are delivered arrive with the correct contents. In addition, checksums apply to both headers and bodies. Very briefly answer the following questions:

a) Describe the classes of failure exhibited by this service.
b) Classify their failures according to their effect on the properties of validity and integrity.
c) Can this service be described as a reliable communication service?

Problem 2 (9 points): Multiprocessing.

Suppose you have a large source program containing m files that you want to compile. You have a cluster of n "shared-nothing" workstations, where $n > m$, on which you may compile your files. At best you will get an m-fold speedup compared to a single processor. List at least three reasons as to why the actual speedup might be less than this.

Problem 3 (9 points): Networking.

You are developing a network protocol for the reliable delivery of fixed-sized messages over unreliable networks. You are using a sequence number in each message to allow the receiver to eliminate duplicates, but you still have three design alternatives to consider.

The design alternatives are: (1) the sender must receive an acknowledgement for the previously sent message before it can send the next message in the sequence, (2) the sender can transmit up to *n* unacknowledged messages, but the receiver will discard any messages that are received out of sequence (in other words, it will only acknowledge a message if it is received in sequence), and (3) the sender can transmit up to *n* unacknowledged messages, and the receiver will acknowledge each on receipt, even if they arrive out of order.

For each alternative, answer the following question:

- Explain what state the receiver must keep around to implement each of the three alternatives (remember, the receiver must be able to detect and discard duplicates).

Problem 4 (8 points): Remote Method Invocation.

a) Why do we need to consider the number of times of invocations and location transparency when we extend local method invocations to remote method invocations?

b) What kind of invocation semantics are there for remote method invocations? Briefly explain each semantic.

Problem 5 (8 points): Serving multiple clients.

There are two main approaches to organizing a server daemon, such as a web server:

(a) Create a new kernel thread for each client (for each web browser connection);

(b) Use a single process responding to all clients, usually based on the select() system call.

Briefly compare and contrast these two approaches.

Problem 6 (10 points): Distributed Deduction.

An island is inhabited by 1000 natives. 700 of these have brown eyes, the rest have pink eyes (this fact is not known to the islanders). The natives are extremely intelligent, but deeply superstitious. They believe that pink eyes are the sign of the devil: if one ever finds out that he or she has pink eyes, he or she will commit suicide in the dark at midnight. Being kind people though, and not wanting to force others to suicide, nobody ever talks about any other person's eye color. There are no mirrors on the island. (The ocean is too polluted to reflect well). Each islander sees each other islander every day.

One day, a foreigner arrives on the island and gains the complete trust of all the inhabitants. Not knowing the customs here, just before he leaves he said pink eyed person I saw today in his farewell party. All islanders were present in this party, and were naturally shocked. What happens, when and why? Please by brief.

Problem 7 (16 points): Predicate Checking.

Consider the global predicate $B = (x_1 < x_2) \wedge (x_3 < 20)$ where x_i is on process P_i. Your task is to design an algorithm to detect this predicate in an offline fashion. In an offline algorithm, a trace is generated for each process. After the computation is finished all the traces are sent to a checker process. Checker process is responsible for detecting the predicate. You need to show what information should be kept as part of the trace. Also, give the algorithm for the checker process. What is the time complexity of checker process if a process has at most m state intervals?

Problem 8 (15 points): Leader Election.

We studied leader election protocols that satisfy the Safety and Liveness conditions given on page 432 of text. Argue whether (or not) the leader election problem is solvable in an asynchronous system. (Hint: How does one show that a given problem is harder to solve than consensus?)

Problem 9 (16 points): Read-Write Mutual Exclusion.

Consider a file F that is present in a distributed system of N processes ($N \geq 4$). The mutual exclusion required on this file has the following safety and liveness conditions (different from those discussed in lecture):
Safety: At most one process may obtain write access to the file at a time. At most 4 processes may obtain read access to the file at a time. If a process has write access to the file, no other process should be able to read or write it.
Liveness: Requests to access and release the resource eventually succeed.

a) Describe a token ring-based algorithm for the above problem.
b) Show that your algorithm guarantees both Safety and Liveness properties defined above.
c) What are the bandwidth usage, client delay and synchronization delay for your algorithm?