

National Exams May 2003

98-Comp-B10, Distributed Systems
3 hours duration

Notes:

1. If doubt exists as to the interpretation of any question, the candidate is urged to submit a clear statement of any assumptions made with the exam paper.
2. This is a Closed Book exam, no calculators are allowed.
3. This exam consists of 7 parts. All parts are mandatory and thus need to be completed.
4. **Please do not put any questions on the covering page. i.e. Start exam questions on page two.**

5. (7 Points) It appears that the three properties that a *secure digest function* $H, h = H(M)$ need to satisfy is: A - Given M , it is easy to compute h ; B - Given h , it is hard to compute M ; C - Given M , it is hard to find another message M' , such that $H(M) = H(M')$. Assume we also have a *symmetric cryptographic algorithm* $E, M_K = E(M, K)$, where K is the shared secret key. Substituting H with E , does E satisfy the same three properties? If so, what is the difference between H and E after all? If not, which property it does not satisfy?

Part 2 - *Failure Models* (10 Points)

In order to illustrate the effects of failures on the design of protocols in a distributed system, we may use the following example. Two processes, A and B , communicate by sending and receiving messages on a bidirectional channel. Neither process can fail. However, the channel can experience transient failures, resulting in the loss of a subset of the messages that have been sent. We wish to devise a protocol where either of two actions α and β are possible, and either (i) both processes take the same action; or (ii) neither takes both actions. It is known that this problem has no solution, i.e., such a protocol is impossible to devise. Please prove the correctness of this claim.

Part 3 - Security (10 Points)

Before Alice sends a message to Bob, she wishes to first digitally sign the message, and then encrypt it for better security. For both signing and encrypting the message, she only considers using asymmetric cryptographic algorithms. In addition, she wishes to use secure digest functions to improve efficiency. Assuming that the complexity of digital certificates are not involved, explain the steps that are needed to be performed on *both Alice and Bob* to complete the delivery of the message.

Part 4 - *Reliable Broadcasts* (10 Points)

In an asynchronous distributed system, if we assume that:

(1) Processes in the system may only suffer from benign failures (those who may fail are said to be *faulty*, otherwise they are *correct*);

(2) Links between processes are reliable;

(3) The point-to-point inter-process communication primitives (*send* and *recv*) are available from the lower layers that satisfy the following properties:

(3A) *Validity*: if p sends m to q , and both p and q are correct, then q eventually delivers m ;

(3B) *Integrity*: For any message m , q receives m at most once from p , and only if p has previously sent m to q .

Under these assumptions, we consider implementing *reliable broadcast* using *send* and *recv*. Answer the following three questions.

1. Part 4A (5 Points): How do we define the *agr ement*, *validity* and *integrity* properties of reliable broadcasts?

2. Part 4B (5 Points): Before any implementations of reliable broadcast are possible, we need to make *one further assumption*. What is this critical assumption, without which reliable broadcasts can not be implemented even in synchronous distributed systems?

Part 5 - Concurrency Control (15 Points)

Assume that we have two transactions in a distributed system, T_1 and T_2 . We use the notation $w_k(x)$ to indicate a *write* operation in T_k on object x , and $r_k(y)$ to indicate a *read* operation in T_k on object y . c_k denotes the *commit* operation in T_k , and a_k denotes the *abort* operation in T_k . We have three objects, x , y and z . Consider the following schedules:

Schedule S_1 : $r_2(x)w_1(x)w_2(x)w_2(y)w_1(x)r_1(y)c_2c_1$

Schedule S_2 : $r_2(x)r_1(x)w_1(y)w_1(x)r_2(z)w_2(z)c_1c_2$

Schedule S_3 : $r_1(x)r_2(x)r_1(y)r_2(z)w_1(y)a_1w_2(y)w_2(z)c_2$

1. Part 5A (5 Points): Is schedule S_1 *conflict serializable*? If so, please give the serial schedule that S_1 is conflict equivalent to. Otherwise, please show your reason.

2. Part 5B (5 Points): Will schedule S_2 be generated by the execution of the *two-phase locking* mechanism? If so, explain the detailed steps of acquiring, promoting and releasing locks. Otherwise, justify your answer.

3. Part 5C (5 Points): Will schedule S_3 be generated by the execution of the *strict two-phase locking* mechanism? If so, explain the detailed steps of acquiring, promoting and releasing locks. Otherwise, justify your answer.

3. Part 7C (4 Points): How does the vector clock supply the *gap detection property*?