UNIVERSITY OF LONDON
IMPERIAL COLLEGE OF SCIENCE, TECHNOLOGY AND MEDICINE

EXAMINATIONS 2000

MEng Honours Degree in Information Systems Engineering Part IV
MEng Honours Degree in Mathematics and Computer Science Part IV
MEng Honours Degrees in Computing Part IV
MSc in Advanced Computing
for Internal Students of the Imperial College of Science, Technology and Medicine

This paper is also taken for the relevant examinations for the Associateship of the City and Guilds of London Institute
This paper is also taken for the relevant examinations for the Associateship of the Royal College of Science

PAPER C437=I4.12

DISTRIBUTED ALGORITHMS

Tuesday 2 May 2000, 10:00
Duration: 120 minutes

Answer THREE questions

Paper contains 4 questions

1a Briefly state the *Leadership Election* problem and show that it cannot be solved in a synchronous ring system if all the processes are identical.

b The *TimeSlice* algorithm solves Leadership Election for a synchronous ring of N processes connected by unidirectional channels where each process has a unique identifier (UID). The algorithm has a communication complexity of N messages.

i) Give an informal description of the operation of the TimeSlice algorithm.

ii) Using the state machine notation from the notes, give a formal specification of TimeSlice.

iii) Comment on the practicality of the algorithm.

c A Leadership Election algorithm is required for a completely connected synchronous network of N processes (i.e. each pair of processes is connected by a bidirectional channel). Assuming each process has a unique identifier and that there are no process or channel failures:

i) Give an informal description of the operation of the algorithm

ii) State the precise communication and time complexity of the algorithm.

d Paranoid Computing PLC have built a system with seven completely connected computers. They have implemented a communication protocol for Byzantine agreement (from one sender to six receivers) that can tolerate up to two arbitrary processor crashes. They now require a protocol that can be used to choose one of the computers as a coordinator. Using their Byzantine agreement protocol, briefly describe an algorithm that chooses a coordinator such that each correct processor agrees on the same coordinator. Assume that each processor has a unique identity (UID) and that each processor knows the entire set of seven valid UIDs. Exactly how many messages are required to choose the coordinator? Is the coordinator your algorithm chooses guaranteed to be a correctly operating processor? – justify your answer.

The four parts carry, respectively, 20%, 40%, 20%, 20% of the marks.

2a i) There are two variations of the *Atomic Commitment* problem. State the correctness conditions (specifications) for each variation. For each condition, identify clearly whether it is a safety or liveness property.

ii) Which variation of the problem is solved by the *Two-Phase Commit* Algorithm? Give the system model assumptions.

b Two-Phase Commit is a *blocking* algorithm.

i) In which part of the algorithm can the *coordinator* process potentially block? Is there a way to avoid the coordinator blocking in practice? If so, explain how.

ii) In which part of the algorithm can a *participant* process potentially block? Is there a way to avoid the participant blocking in practice? If so, explain how.

c Consider a synchronous system where communication link failures may occur. In such a system, describe an execution (scenario) of the Three-Phase Commit Algorithm where the *Agreement* property is violated.

d The Termination Protocol of the Three-Phase Commit Algorithm is based on the fact that the identities of the participating processes are consecutive integers (e.g. 0, 1, 2, 3, ...). So, if process i (the coordinator in epoch i) crashes before the completion of round 3i+3, then process i+1 becomes automatically the coordinator in epoch i+1.

i) Propose an extension to the Three-Phase Commit Algorithm and its associated Termination Protocol, which works even in the case where the identities of the participating processes are unique but not consecutive integers (e.g. 2, 5, 12, 19, ...). Assume a synchronous, message passing system, with **n** processes completely connected by means of bidirectional links.
[Hint: You can use an algorithm from the lecture notes as a building block for the extended termination protocol.]

ii) Assume that up to two processes can crash while the Three-Phase Commit is being executed, in the above system (n>3). What is the total number of *rounds* and the total number of *messages* required in the *worst case* for the termination of Three-Phase Commit using the extended Termination Protocol (the one you suggested in (i))? Give exact numbers and not "big-O" notations.

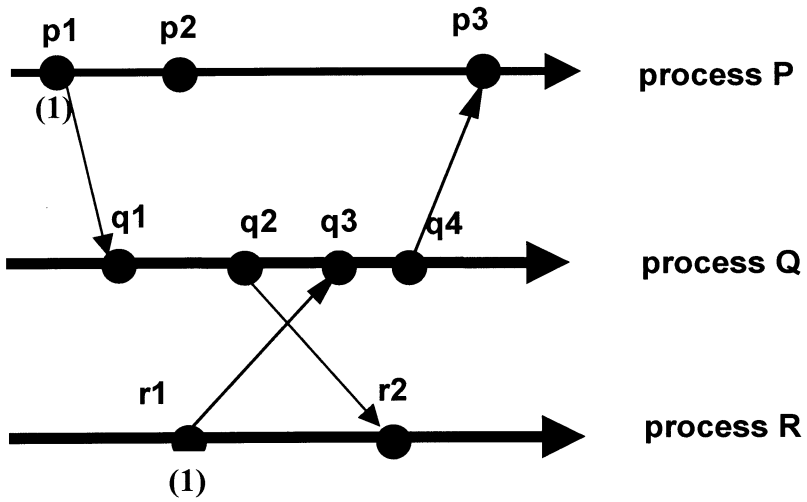
The four parts carry, respectively, 20%, 20%, 20%, 40% of the marks.

3a i) Given any two events a,b in a distributed system, define what is meant by the causal order relation a $\rightarrow$ b, i.e. "a happens before b".

ii) Define the implementation rules for Lamport's logical clocks that ensure for events a and b, a $\rightarrow$ b implies that C(a) < C(b) where C(a) and C(b) are timestamps.

iii) Explain why C(a) < C(b) does not necessarily imply that a $\rightarrow$ b.

iv) Annotate the process interaction diagram below with logical timestamp values for each event in each process P, Q and R.
[Note: redraw the diagram in your answer book.]



b The Ricart-Agrawala Algorithm solves the Distributed Mutual Exclusion problem.

i) Briefly describe the algorithm.

ii) Present an argument that the algorithm correctly implements mutually exclusive access to a resource.

iii) Lamport's Mutual Exclusion algorithm requires that channels between processes must be FIFO (first-in-first-out). Does Ricart-Agrawala require FIFO channels? Give reasons for your answer.

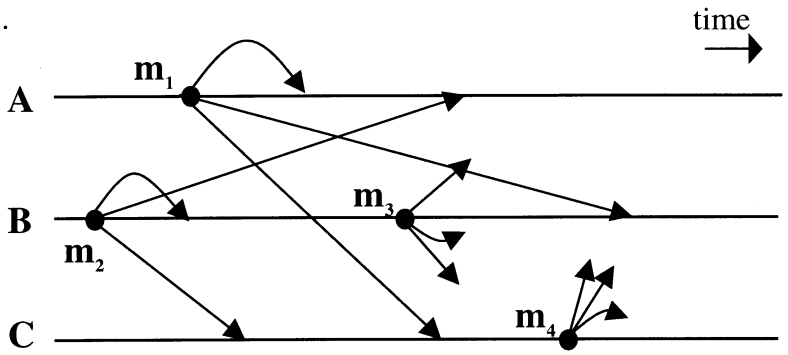
The two parts carry, respectively, 50%, 50% of the marks.

4a i) State the correctness conditions (specifications) of *Reliable Broadcast*. For each condition, identify clearly whether it is a safety or liveness property.

ii) Describe an algorithm (using pseudo-code notation) that implements Reliable Broadcast in asynchronous message passing systems. State any assumptions about the system model required for the correctness of the algorithm.

iii) State the additional property that allows a Reliable Broadcast to be characterised as a Timed Reliable Broadcast. Describe an algorithm that transforms any Reliable Broadcast primitive to a Timed Reliable Broadcast, in synchronous point-to-point networks.

b Assume that the messages in the diagram below are multicast using the ISIS CBCAST protocol. Redraw the diagram showing the delivery of the multicast messages m_3 and m_4 from processes B and C respectively. Annotate your diagram with the value of each process's vector clock *after* each multicast and delivery event. Assume the vector clock of each process is initialised to [0,0,0].



c The specification for Terminating Reliable Broadcast (TRB) for a synchronous system is as follows. Initially the sender has a value $v \in V$, $|V| \geq 2$ and the n-1 receivers do not know what value the sender has. Eventually each receiver process must irrevocably decide some value in V or the special value SF ("Sender Faulty"). Give the correctness conditions for Terminating Reliable Broadcast (TRB). Assuming no link failures, describe an algorithm (using pseudo-code notation) that satisfies TRB in a completely connected synchronous system in which no more than m processes can crash.

d Assume that there is an algorithm A_{TRB} that implements TRB in synchronous message passing systems with n processes and *no link failures*. The maximum number of process failures that may occur in the system is known and tolerated by A_{TRB}. Describe an algorithm (using pseudo-code notation) that solves Consensus in the assumed system model using A_{TRB} as a building block. [Hint: More than one instance of A_{TRB} can be executed concurrently.]

The four parts carry, respectively, 30%, 20%, 20%, 30% of the marks.