The University of Melbourne
COMP90020: Distributed Algorithms

Sample Exam 2020 SM1

Exam Duration: 3 hours
Total marks for this exam: 60

Authorised materials:

This is an open book assessment.

You are **allowed to**:

- Use the textbook, lecture slides, and any other reading materials used or referenced throughout the course.
- Use explanations or examples given in the lectures.
- Use the solutions given for all the tutorial questions.

While you are undertaking this assessment you **must not**:

- Use any messaging or communications technology.
- Act in any manner that could be regarded as providing assistance to another student who is undertaking this assessment, or will in the future be undertaking this assessment.

The work you submitmust be based on **your own knowledge and skills**, without assistance from any other person.

Instructions to Students:

- There are **12 question**s in this exam, **attempt all** of them.
- All questions require you to upload a file with your answer.
- Write your answer on a sheet of paper, scan or photograph your answer, and upload the corresponding file.
- You may answer on any clean, blank paper that you have available.
- Alternatively, you can write your answers digitally using a tablet and pen, save the file as a picture, and upload the file as your answer.
- **Write legibly** and be as concise as possible.
- For further information on Canvas Quizzes please see: https://students.unimelb.edu.au/your-course/manage-your-course/exams-assessments-and-results/exams/exam-types.

Question 1 **[3 Marks]**

Using Cristian's method for synchronizing clocks where we use a time server, we record the round-trip time and the timestamp returned by the server as 4 sec and 09:55:28 (hh:mm:ss), respectively.

(A) What time should we set the local clock to? (1 mark)

(B) What is the accuracy of this setting? (1 mark)

(C) What is the accuracy of the setting if we know for a fact that the minimum round trip time is 1 second?

Show your calculations and notation clearly.

Question 2 **[3 Marks]**

Why is the implementation of a reliable failure detector only possible in a synchronous distributed system?

Question 3 **[6 Marks]**

Give an example execution of the ring-based mutual exclusion algorithm to show that processes are not necessarily granted entry to the critical section in happened-before order.

Question 4 **[3 Marks]**

Consider a set of distributed processes p_1, p_2, and p_3 that use the Ricart and Agrawala's algorithm for mutual exclusion. Assume that p_3 is currently in the critical section (CS) and there is no other process in the WANTED state.

(A) Now consider requests from p_1 and p_2 (in that order) to enter CS. Show the state and queue entries at each process. (1.5 marks)

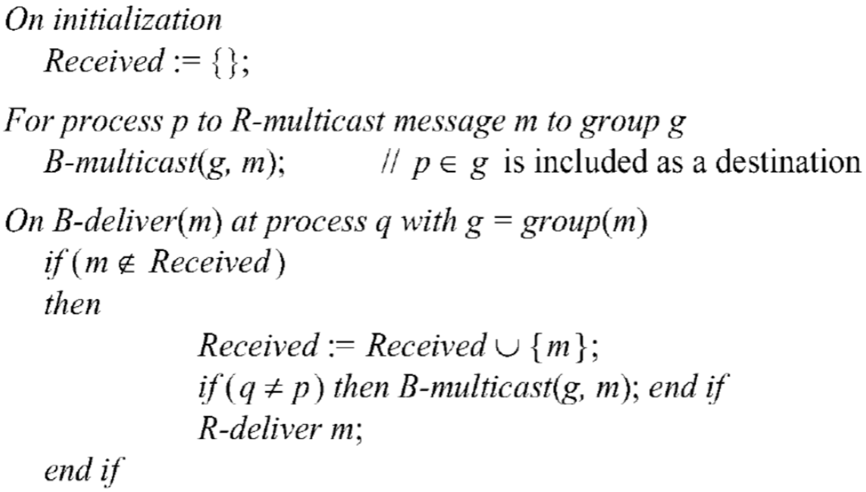
(B) Now, p_3 exits the CS and informs all relevant processes that CS is released. Show the state and queue entries at each process, at this stage. (1.5 marks)

Question 5 **[9 Marks]**

In a certain system, each process typically uses a critical section many times before another process requires it. Explain why Ricart and Agrawala's multicast-based mutual exclusion algorithm is inefficient for this case, and describe how to improve its performance. Does your adaptation satisfy liveness condition ME2?

Question 6 **[3 Marks]**

In the following reliable multicast algorithm (Figure 1) that we saw in class, explain briefly what would happen if we were to have 'R-deliver m' before the 'if (q $\neq$ p) then...' statement.



```
On initialization
    $Received := \{\}$;

For process p to R-multicast message m to group g
    B-multicast(g, m);        // $p \in g$ is included as a destination

On B-deliver(m) at process q with g = group(m)
    if ($m \notin Received$)
    then
            $Received := Received \cup \{m\}$;
            if ($q \neq p$) then B-multicast(g, m); end if
            R-deliver m;
    end if
```

Figure 1: Reliable multicast algorithm.

Question 7 **[6 Marks]**

The figure below (Figure 2) shows some multicast messages happening for three processes on different machines. Is the ordering of these multicast messages CO, FIFO and/or TO? Explain your answer.

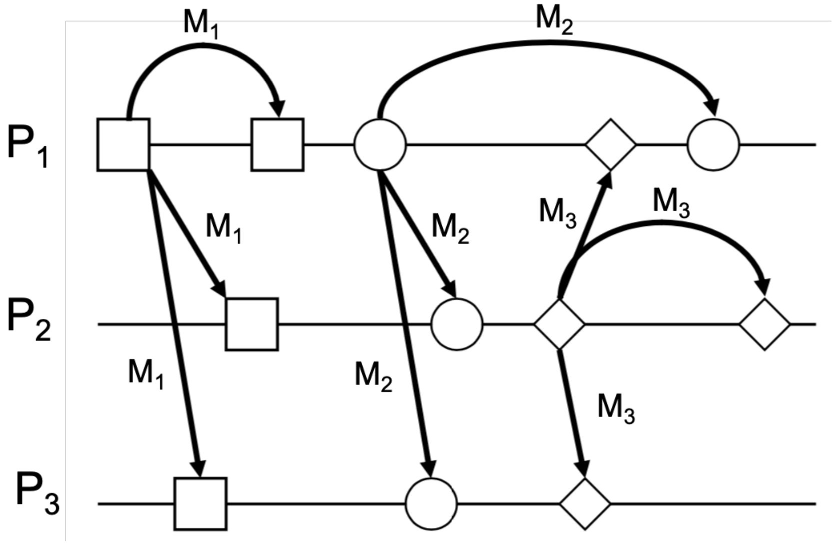


Figure 2: Ordered multicast example.

Question 8 [6 Marks]

Consider a distributed system involving 5 processes, p_1, p_2, p_3, p_4 and p_5. Process p_1 is appointed as the general for this run and its proposed value is a. Process p_3 has been hijacked by a hacker, and will relay the value b to processes p_2 and p_5, and c to the rest. The system does not have a digital signature mechanism. Show that processes p_2, p_4 , and p_5 can still reach an agreement. How many cycles are mandatory for reaching the agreement?

Question 9 [6 Marks]

For distributed transactions, we discussed the one phase commit protocol. What are the disadvantages of this protocol? How can the two phase commit (2PC) protocol overcome these disadvantages? Explain your answer.

Question 10 [6 Marks]

Consider the following wait-for-graph (Figure 3). Show that the system is in a deadlock situation. What is the most efficient way (in terms of number of terminations) to get out of the deadlock?

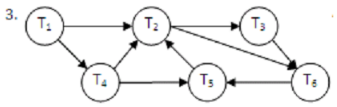
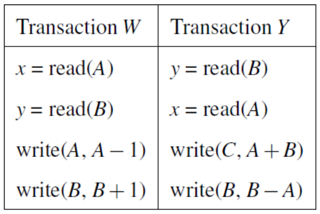


Figure 3: Wait-for-Graph

Question 11 [6 Marks]

Consider transactions W and Y (Figure 4):



Transaction W	Transaction Y
x = read(A)	y = read(B)
y = read(B)	x = read(A)
write(A, $A-1$)	write(C, $A+B$)
write(B, $B+1$)	write(B, $B-A$)

Figure 4: Transactions W and Y.

show a concurrent execution of these transactions that has a dirty read.

Question 12 **[3 Marks]**

Explain how optimistic concurrency control can increase the system concurrency? In what situation optimistic concurrency control fails to enhance the systems throughput?