



SCHOOL OF SCIENCE AND TECHNOLOGY
Semester 2, 2024



FINAL EXAMINATION PAPER

ITC333: DISTRIBUTED SYSTEMS TECHNOLOGY

Important Instructions to Students:

1. Read the questions, instructions carefully and answer the questions.
2. You are allowed an **extra 10 minutes reading time** (during this time, you are not allowed to write with pen) and **180 minutes (3 Hours) to complete this paper**.
3. Silent, non-programmable calculators are allowed, but not supplied.
4. Write all your answers in the answer booklet provided. If you need extra sheets, please ask from the supervisor. Write your name, course code and student identification number in the answer booklet and in the attendance sheet.
5. This paper contains one (1) section – Section A. Answer ANY 20 questions out of the given 22.
6. Write all your answers legibly. The marker has the right to award zero for answers that are not legible.
7. This is a **Closed Book Examination**. The University views plagiarism, cheating and dishonest practice as serious offences and if found, offenders are penalized.

Examination Paper Content (Format):

<u>Section</u>	<u>Question</u>	<u>Content</u>	<u>Marks</u>	<u>Suggested Time</u>
A	1-22	Discussion Questions	100	180 minutes
		TOTAL	100	180 Minutes

SECTION A – DISCUSSION QUESTIONS

[100 MARKS]

Each question is worth 5 marks.

Answer ANY 20 questions out of the following 22.

1. Explain what is meant by (distribution) transparency and list eight (8) different types of transparency.
2. Explain: (a) Why is it sometimes so hard to hide the occurrence and recovery from failures in a distributed system? (b) Why is it not always a good idea to aim at implementing the highest degree of transparency possible?
3. Draw and explain a three-tiered client-server architecture.
4. Not every node in a peer-to-peer network should become *super peer*. (a) Draw a labelled diagram of *super peer* network clearly showing its hierarchical arrangement of nodes. (b) What are reasonable requirements that a *super peer* should meet?
5. Would it make sense to limit the number of threads in a server process?
6. Having only a single lightweight process per process is also not such a good idea. (a) Why not? (b) Describe a simple scheme in which there are as many lightweight processes as there are runnable threads.
7. When a collection of computers is organized in a (logical or physical) ring, explain how could you guarantee a (a) maximum end-to-end delay? (b) minimum end-to-end delay?
8. (a) One way to handle parameter conversion in RPC systems is to have each machine send parameters in its native representation, with the other one doing the translation, if need be. The native system could be indicated by a code in the first byte. However, since locating the first byte in the first word is precisely the problem, can this work? Explain. (b) Does it make sense to implement persistent asynchronous communication by means of RPCs? Explain.
9. In the centralized approach to mutual exclusion (Fig. 5.14), suppose that the coordinator crashes. Does this always bring the system down? If not, under what circumstances does this happen? Is there any way to avoid the problem and make the system able to tolerate coordinator crashes?

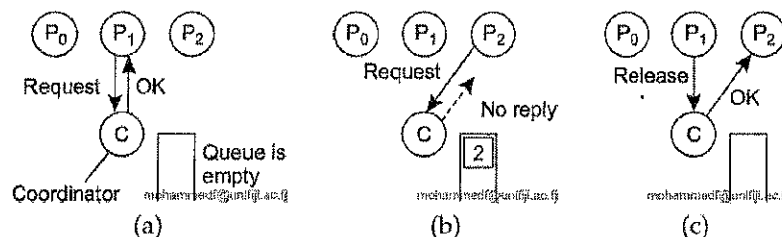


Figure 5.14: (a) Process P_1 asks for permission to access a shared resource. Permission is granted. (b) Process P_2 asks permission to access the same resource, but receives no reply. (c) When P_1 releases the resource, the coordinator replies to P_2 .

10. (a) When searching for files in an unstructured peer-to-peer system, it may help to restrict the search to nodes that have similar files as yourself. Explain how gossiping can help to find those nodes.
- (b) Consider a communication layer in which messages are delivered only in the order that they were sent. Give an example in which even this ordering is unnecessarily restrictive.
11. A distributed system may have multiple, independent resources. Imagine that process 0 wants to access resource *A* and process 1 wants to access resource *B*. Can Ricart and Agrawala's algorithm lead to deadlocks? Explain your answer.
12. Consider the *Chord* system in Fig. 6.2 and assume that node 7 has just joined the network. What would its finger table be and would there be any changes to other finger tables?

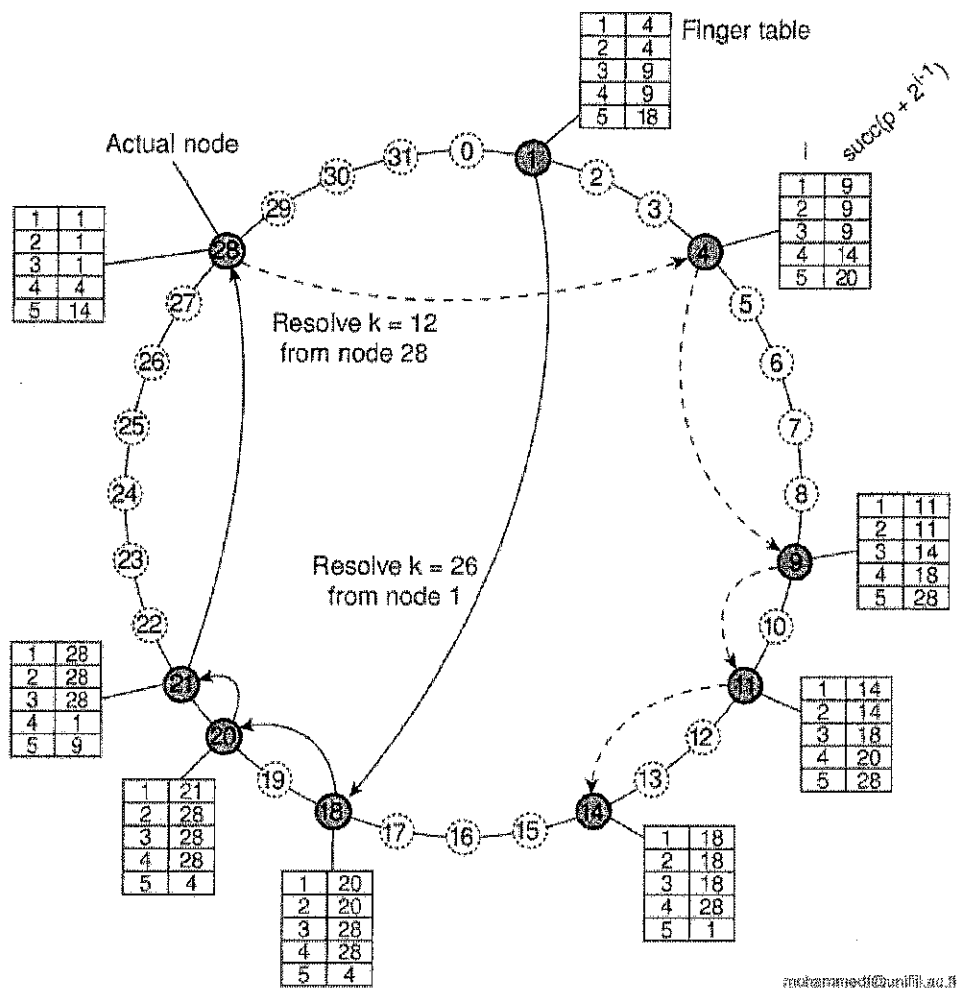


Figure 6.2: Resolving key 26 from node 1 and key 12 from node 28 in a Chord system.

13. Suppose that two processes detect the demise of the coordinator simultaneously and both decide to hold an election using the *Bully algorithm*. (a) What happens?
 (b) In the Fig 5.21, we have two ELECTION messages circulating simultaneously. While it does no harm to have two of them, it would be more elegant if one could be killed off. Devise an algorithm for doing this without affecting the operation of the basic election algorithm.

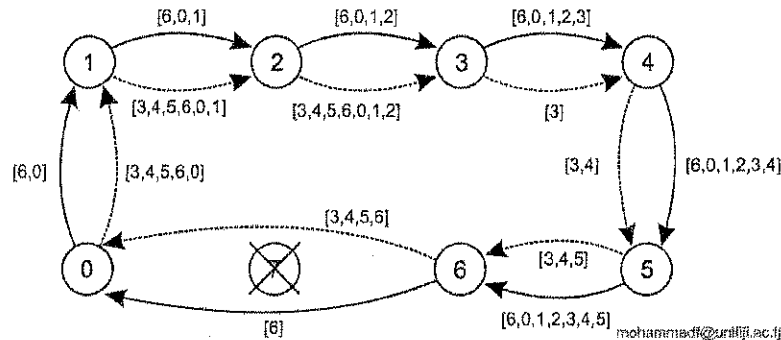


Figure 5.21: Election algorithm using a ring. The solid line shows the election messages initiated by P_6 ; the dashed one those by P_3 .

14. Consider a distributed file system that uses per-user name spaces. In other words, each user has his own, private name space. Can names from such name spaces be used to share resources between two different users?
15. To implement totally ordered multicasting by means of a sequencer, one approach is to first forward an operation to the sequencer, which then assigns it a unique number and subsequently multicasts the operation. Explain two alternative approaches and compare the three solutions.
16. A file is replicated on 10 servers. List all the combinations of read quorum and write quorum that are permitted by the *voting algorithm*.
17. What are the principles of fault tolerance and briefly list the steps that the development team needs to take in designing a fault tolerant distributed system.
18. How many failed elements (devices plus voters) can Fig. 8-2 handle? Give an example of the worst case that can be masked.

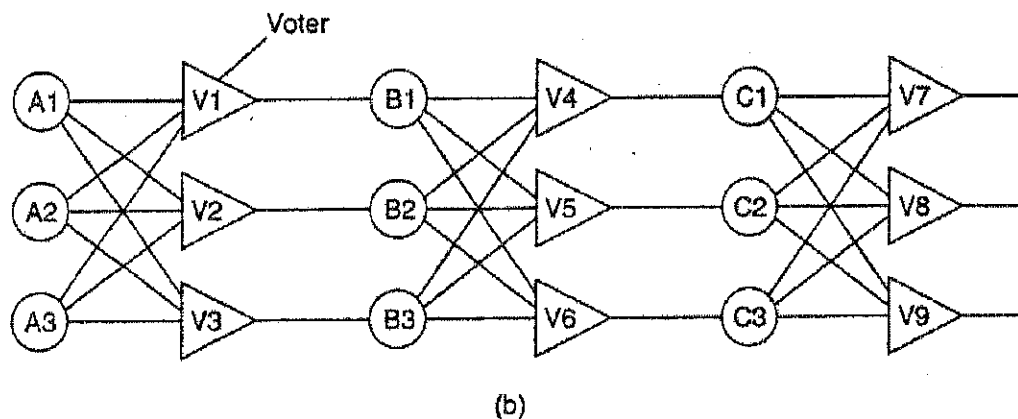


Figure 8-2. Triple modular redundancy.

19. What are the permissible delivery orderings for the combination of *FIFO* and *total-ordered multicasting* in Fig. 8-15?

Process P1	Process P2	Process P3	Process P4
sends m1	receives m1	receives m3	sends m3
sends m2	receives m3	receives m1	sends m4
	receives m2	receives m2	
	receives m4	receives m4	

Figure 8-15. Four processes in the same group with two different senders, and a possible delivery order of messages under FIFO-ordered multicasting.

20. Explain the role of the timestamp in message 6 in Fig. 9.16, and why does it need to be encrypted?

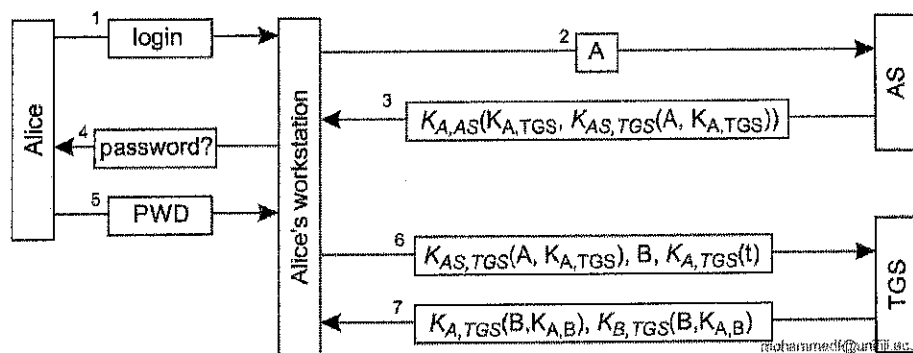


Figure 9.16: Authentication in Kerberos.

21. Assume Alice wants to send a message m to Bob. Instead of encrypting m with Bob's public key K_{+B} , she generates a session key $K_{A,B}$ and then sends $[K_{A,B}(m), K_{+B}(K_{A,B})]$. Why is this scheme generally better? (*Hint*: consider performance issues.)
22. Explain distributed systems technologies in use by Facebook data centre.

The End

