Chair of Application and Middleware Systems
Department of Informatics
Technical University of Munich

| | |
|---|---|
| Personal sticker | S5336 |

**Compliance to the code of conduct**
I hereby assure that I solve and submit this exam myself under my own name by only using the allowed tools listed below.

______________________________
Signature or full name if no pen input available

# Distributed Systems Mock

| | | | |
|---|---|---|---|
| **Exam:** | IN2259 / DSMock | **Date:** | Friday 5th February, 2021 |
| **Examiner:** | Prof. Dr. Hans-Arno Jacobsen | **Time:** | 08:00 – 17:00 |

## Working instructions

- This exam consists of **10 pages** with a total of **7 problems**.
  Please make sure now that you received a complete copy of the exam.
- The total amount of achievable credits in this exam is 100 credits.
- Detaching pages from the exam is prohibited.
- Allowed resources:
  - one **non-programmable pocket calculator**
  - one **analog dictionary** English ↔ native language
- Subproblems marked by * can be solved without results of previous subproblems.
- **Answers are only accepted if the solution approach is documented.** Give a reason for each answer unless explicitly stated otherwise in the respective subproblem.
- Do not write with red or green colors nor use pencils.
- Physically turn off all electronic devices, put them into your bag and close the bag.

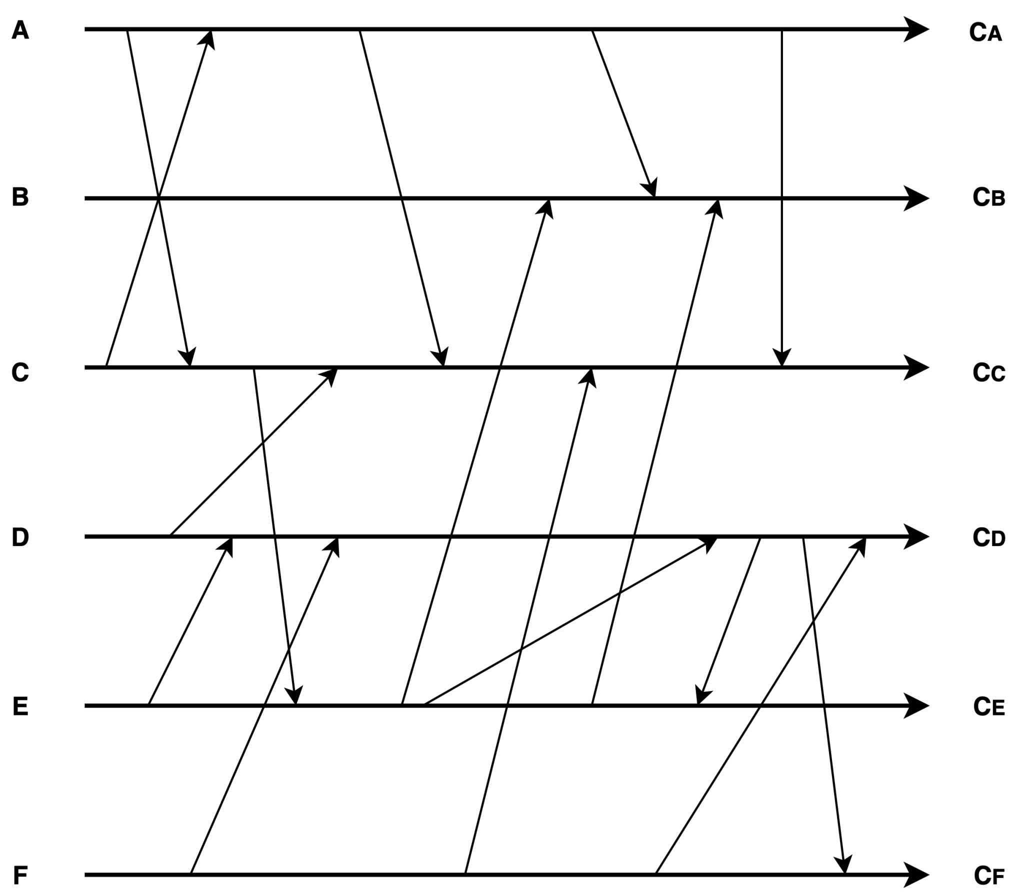
Left room from __________ to __________ / Early submission at __________

☐ Exam empty

## Problem 1 Lamport Logical Clock (12 credits)

For each process $p_i \in \{A, B, C, D, E, F\}$ determine the logical time ($\{C_A, C_B, C_C, C_D, C_E, C_F\}$) at the **end** of the time/event diagram according to **Lamport's Logical Clock** algorithm. Write the final value of clocks in the table below.

0 ☐
1 ☐
2 ☐
3 ☐
4 ☐
5 ☐
6 ☐
7 ☐
8 ☐
9 ☐
10 ☐
11 ☐
12 ☐

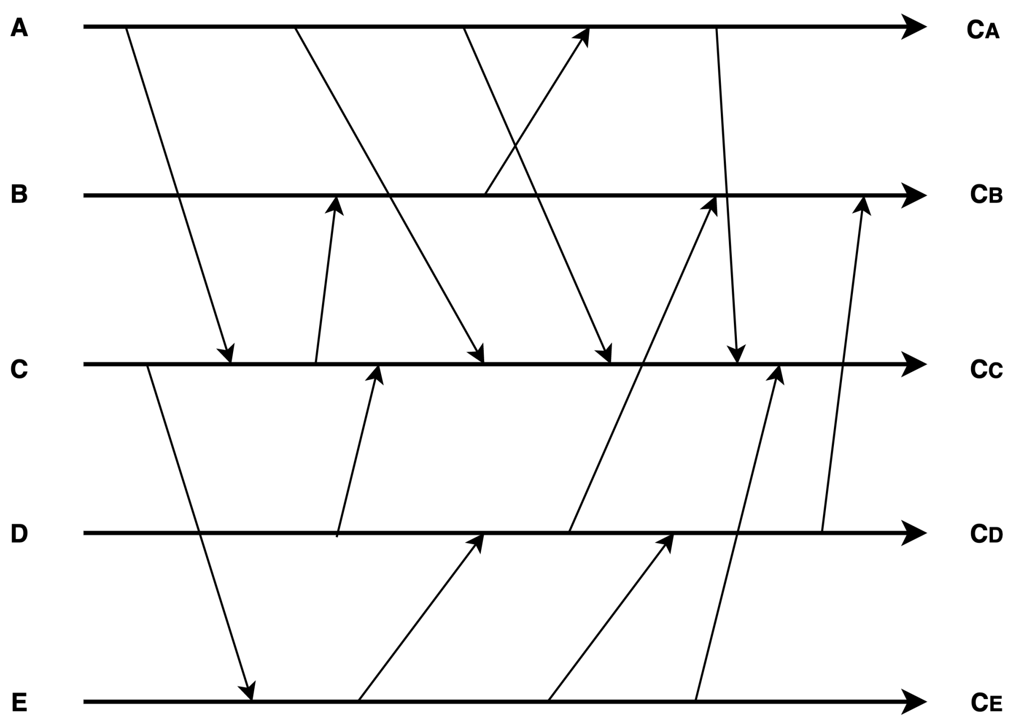


| | Logical Time |
|---|---|
| $C_A$ | |
| $C_B$ | |
| $C_C$ | |
| $C_D$ | |
| $C_E$ | |
| $C_F$ | |

## Problem 2 Vector Clock (12.5 credits)

For each process $p_i \in \{A, B, C, D, E\}$ determine the logical time ($\{C_A, C_B, C_C, C_D, C_E\}$) at the **end** of the time/event diagram according to the **Vector Clock** algorithm. Write the final value of clocks in the table below.

0 1 2 3 4 5 6 7 8 9 10 11 12



| | 1st Element | 2nd Element | 3rd Element | 4th Element | 5th Element |
|---|---|---|---|---|---|
| $C_A$ | | | | | |
| $C_B$ | | | | | |
| $C_C$ | | | | | |
| $C_D$ | | | | | |
| $C_E$ | | | | | |

## Problem 3 Consistency Models (17 credits)

Consider the execution depicted in the figure below and answer the following questions.

| | | | | |
|---|---|---|---|---|
| $P_1$ | R1(X)a | W1(Z)e | W1(Z)h | R1(X)j |
| $P_2$ | W2(z)d | R2(X)f | R2(X)j | R2(Y)k |
| $P_3$ | R3(Y)c | R3(Z)d | R3(Y)g | W3(Z)p |
| $P_4$ | W4(X)a | R4(Z)d | R4(Z)h | W4(Y)k |
| $P_5$ | W5(Y)c | W5(X)f | W5(X)j | R5(Y)k |
| $P_6$ | R6(Z)d | W6(Y)g | R6(Y)g | R6(X)j |

Time $\longrightarrow$

a) Is the execution **sequentially consistent** (Yes/No)? If Yes, give **complete** and valid execution sequence. If No, give an explanation.

b) Is it **causally consistent** (Yes/No)? If Yes, give **complete** and valid execution sequences for $P_2$, $P_4$, $P_6$. If No, give an explanation.

## Problem 4 Gossip-based Replication (22.5 credits)

Figure 4.1 shows lazy replication over a Gossip protocol. In Gossip architecture, messages (gossip) between servers are periodically exchanged to store data over a set of replicas redundantly. Clients can issue (1) write requests (update) and (2) read requests (query). In order to provide consistency, the clients and replicas keep track of clocks, $clock_c$ and $clock_{rep}$, where $clock_c$ is the client clock and $clock_{rep}$ is the replica clock.

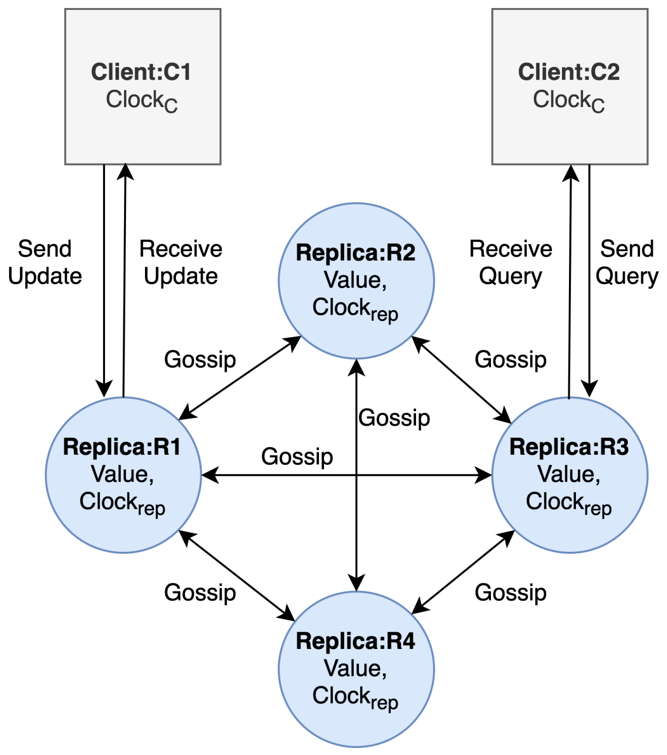

Client:C1
Clock$_C$
Client:C2
Clock$_C$
Send Update
Receive Update
Receive Query
Send Query
Replica:R2 Value, Clock$_{rep}$
Replica:R1 Value, Clock$_{rep}$
Replica:R3 Value, Clock$_{rep}$
Replica:R4 Value, Clock$_{rep}$
Gossip
Gossip
Gossip
Gossip
Gossip
Gossip

Figure 4.1: Gossip-based Replication

a) Assume you are asked to integrate replication to a system for collaboratively working on a shared document in real-time. Would you use the gossip-based replication? If yes, give a short motivation for your answer. If no, suggest an appropriate replication scheme discussed in the lecture and provide a short motivation for your answer.

0 1 2 3

b) Which level of consistency does the Gossip protocol implement (name the consistency model)? Briefly describe how this level of consistency is achieved with the help of timestamps (Give a short example).

0 1 2 3 4

☐ Page empty

c) The following messages are exchanged between the processes. Iterate through the steps manually and fill in the table provided below.

1. C1 sends update U1(Red) to R2
2. C2 sends update U2(Yellow) to R3
3. C2 sends query Q1 to R1
4. C2 sends query Q2 to R4
5. R2 sends gossip G1 to R1

| | | Initial | 1 | 2 | 3 | 4 | 5 |
|---|---|---|---|---|---|---|---|
| Client $C_1$ | $Clock_c$ | [0, 0, 0, 0] | | | | | |
| Client $C_2$ | $Clock_c$ | [0, 0, 0, 0] | | | | | |
| Replica $R_1$ | Value | - | | | | | |
| | $Clock_{rep}$ | [0, 0, 0, 0] | | | | | |
| | Pending | $\emptyset$ | | | | | |
| Replica $R_2$ | Value | 0 | | | | | |
| | $Clock_{rep}$ | [0, 0, 0, 0] | | | | | |
| | Pending | $\emptyset$ | | | | | |
| Replica $R_3$ | Value | - | | | | | |
| | $Clock_{rep}$ | [0, 0, 0, 0] | | | | | |
| | Pending | $\emptyset$ | | | | | |
| Replica $R_4$ | Value | - | | | | | |
| | $Clock_{rep}$ | [0, 0, 0, 0] | | | | | |
| | Pending | $\emptyset$ | | | | | |

## Problem 5 **Erasure Coding (12 credits)**

Consider a distributed data storage consisting of a cluster of several storage servers. To be able to recover corrupted data, we use *Reed-Solomon Encoding* for sharding the data. Each shared is stored on a different server. Assume that every data is divided into **4 shards**, and we can recover up to **3 shards**.

While querying the string below, we realized that some shards are corrupted (The corrupted parts are marked with underscores ( _ ), which has an ASCII value of 95). Construct the original string with the following assumptions (the string is case sensitive):

- Matrix Calculator (or any other calculator of your choice): https://matrixcalc.org/en/
- String to ASCII Converter: https://onlinestringtools.com/convert-string-to-ascii
- ASCII Table: http://www.asciitable.com/

Corrupted String (without quotation marks) = "Es_era_os _ublic"

$$\mathit{EncodingMatrix} = \begin{bmatrix} 3 & 2 & 1/4 & 2 \\ 2 & 1 & 1/2 & 2 \\ 0 & 0 & 1/4 & 1/5 \end{bmatrix}$$

$$\mathit{ParityMatrix} = \begin{bmatrix} 69 & 115 & 112 & 101 \\ 114 & 97 & 109 & 111 \\ 115 & 32 & 112 & 117 \\ 98 & 108 & 105 & 99 \\ 2639/4 & 763 & 792 & 3009/4 \\ 1011/2 & 559 & 599 & 1139/2 \\ 967/20 & 148/5 & 49 & 981/20 \end{bmatrix}$$

0 ☐ 1 ☐ 2 ☐ 3 ☐ 4 ☐ 5 ☐ 6 ☐ 7 ☐ 8 ☐ 9 ☐ 10 ☐ 11 ☐ 12 ☐

## Problem 6 Peer-To-Peer Systems (12 credits)

Consider the Pastry network, where every node ID $\in \{0, 1, 2, 3, 4, 5\}^6$ and contains the following nodes:

{ 002124, 012312, 111324, 112553, 120150, 122140, 140531, 144545, 202043, 205103, 211000, 211230, 221511, 222454, 223525, 224123, 225355, 225521, 230234, 233424, 235014, 240050, 240253, 240302, 242510, 243330, 244514, 245305, 250541, 251310, 251540, 300231, 314411, 335550, 405302, 405542, 413515, 420501, 422002, 424320, 424414, 443023, 444514, 514332, 551154 }

a) Complete the routing table of node **233424** in the table below:

Table 6.1: Routing Table

| RT | 0 | 1 | 2 | 3 | 4 | 5 |
|---|---|---|---|---|---|---|
| 0 | | | | | | |
| 1 | | | | | | |
| 2 | | | | | | |
| 3 | | | | | | |
| 4 | | | | | | |
| 5 | | | | | | |

b) Provided that the leaf-set has a size of **6**, give the leaf set of node **233424**:

## Problem 7 **Publish/Subscribe (12 credits)**

Consider an overlay broker network based on a **subscription-based** routing model, containing 8 brokers, as demonstrated in the figure below. Each broker maintains a Publication Routing Table (PRT) and subscription covering **is enabled**.

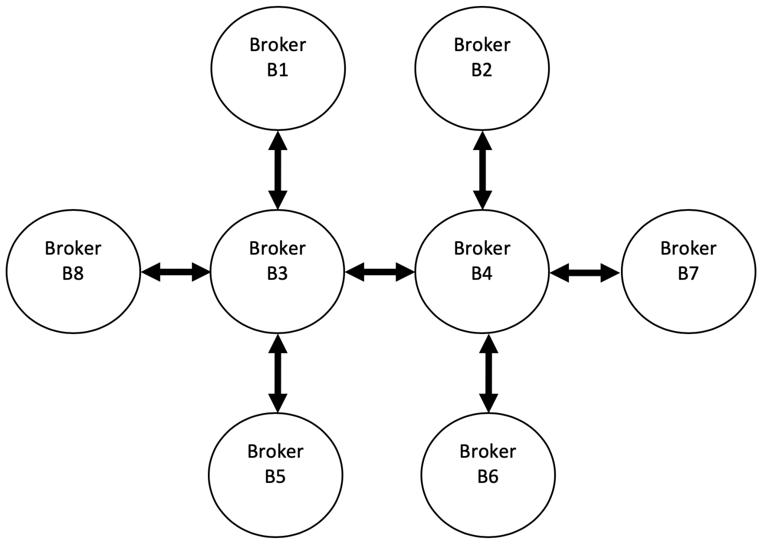


Figure 7.1: Broker overlay

Assume that we have several subscribers in the network, which can subscribe to any brokers. Subscribers issue the following subscriptions. Update the PRT table of brokers in the tables below.

1. $S2$ connects to $B4$ and subscribes to $[l, =, MUC], [temp, >, 30]$
2. $S4$ connects to $B1$ and subscribes to $[t, =, FILM], [r, <, 5]$
3. $S8$ connects to $B5$ and subscribes to $[t, =, FILM], [r, > 10]$
4. $S2$ connects to $B8$ and subscribes to $[l, =, MUC], [temp, >, 20]$
5. $S1$ connects to $B7$ and subscribes to $[t, =, FILM], [r, <, 7]$
6. $S4$ connects to $B3$ and subscribes to $[l, =, MUC], [temp, >, 10]$

Table 7.1: PRT of Broker **B3**

| | |
|---|---|
| | |
| | |
| | |
| | |
| | |
| | |

Table 7.2: PRT of Broker **B7**

| | |
|---|---|
| | |
| | |
| | |
| | |
| | |
| | |

**Additional space for solutions–clearly mark the (sub)problem your answers are related to and strike out invalid solutions.**