UNIVERSITÄT SIEGEN

Fakultät IV
Betriebssysteme und verteilte Systeme
Prof. Dr. rer. nat. Roland Wismüller

Excercise Sheet 10

(To be processed until 18.01.)

Lecture Distributed Systems
Winter Term 2023/24

Exercise 1: Totally Ordered Multicast / Lamport time stamps

In a bank implementation scenario, a totally ordered multicast is required, i.e., a multicast operation in which all messages are delivered to all recipients in the same order. Lamport time stamps are used to implement the multicast in a distributed manner. When a process receives a message, it is placed in a local queue, which is organized according to time stamps.

The recipient sends a multicast acknowledgement to the other processes. If we follow Lamport's algorithm to adjust the local clocks, the time stamp of the received message is smaller than the time stamps of the acknowledgement. After a short time, it turns out that the local queues in which the messages and notifications are located are the same for all processes, since it can be assumed that the delivery takes place in the same order as the one in which they were sent. A process can only deliver a message if it is at the beginning of the queue and has been acknowledged by all other processes.

Is it strictly required here that each message is acknowledged in order to achieve totally ordered multicasting with Lamport timestamps?

Exercise 2: Transactions

Briefly explain what is meant by the term transactions.

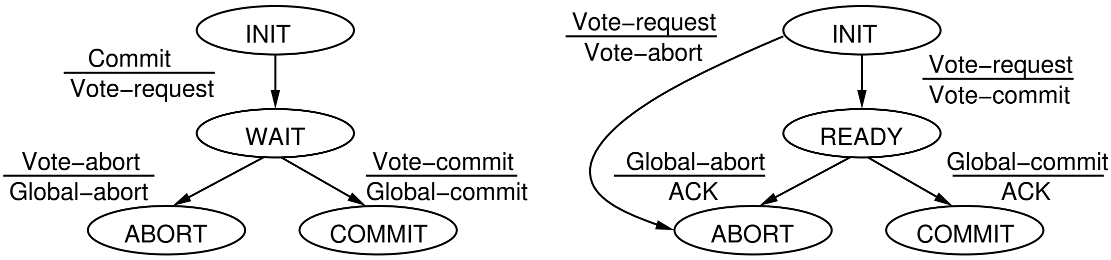
a) What are the main characteristics of transactions?

b) How can deadlocks arise in this context?

Exercise 3: Two-Phase Commit

In a system where errors may occur, several problems may occur when using the basic two-phase-commit protocol. First, note that both the coordinator and the participants may have states in which they are blocking and waiting for incoming messages. So the protocol can easily fail if one process crashes, because the other processes wait indefinitely for a message from that process. The figure below shows the state machine for a coordinator and for a participant in this context. For each transition, the evant (local request or incoming message) is shown on the top, while the corresponding action (outgoing message) is shown at the bottom.

Look at this diagram and name a total of three possible states in which a coordinator or participant could block.



State machine for the coordinator

State machine for a participant

Exercise 4: Consistency Models (Mandatory exercise for 6 CP)

a) In the lecture (chapter 8.2) a number of consistency models were presented. Why are these different forms of consistency introduced? Explain in your own words what is the most important reason to consider weakened consistency models.

b) Given are the two subsequent computations in a distributed system. Which of the consistency models: strict, sequential and causal allows such a memory behavior? Justify your statement.

P1:	W(x)a	W(x)b				
P2:					R(y)a	R(x)a
P3:			R(x)b	W(y)a		

P1:	W(x)a		W(x)b						
P2:		R(x)a		W(x)c			R(y)a		
P3:					R(x)a	W(y)a		R(x)c	R(x)b

Notations:

- $W(x)a$: the value a is written into the variable x.
- $R(x)b$: Variable x is read, the result is b.

Exercise 5: Sequential Consistency (Mandatory exercise for 6 CP)

Is the memory underlying the subsequent execution of two processes sequentially consistent? It is assumed that all variables are initially set to zero.

$P_1 : R(x)1; R(x)2; W(y)1$

$P_2 : W(x)1; R(y)1; W(x)2$

Exercise 6: Causal Consistency

a) Show that the following execution is not causally consistent:

$P_1 : W(x)0; W(x)1$

$P_2 : R(x)1; W(y)2$

$P_3 : R(y)2; R(x)0$

b) What kind of consistency would you use to implement electronic stock trading? Explain your answer.