ID2201 Distributed Systems

The exam consists of three parts: A, B and C. Part A consists of twenty-four multiple choice questions where a correct answer is rewarded one point and a wrong answer zero points. Answer the questions by writing A, B, C or, D in the boxes below. It's the boxes below that count, not the things you write elsewhere. If you think the question is wrong or if you think that two answers are correct then make a small note of this but do write one answer in the boxes below.

Part B consists of eight short answer questions that could give you two points each. You should use the the space given in the exam to answer these questions. Make sure to write short answers that are to the point.

Part C consists of three essay question where I want you to reason about what problems we might have, different solutions to the problems and the pros and cons of your solutions. You should use the the space given in the exam to answer these questions i.e. the page the question is written on. The answers should be well formulated and well written. You should show that you can identify the critical problems and make use of the knowledge gained in the course.

Write your name on all pages and hand in the whole exam.
You are allowed to have a dictionary (book, not digital) but this must be examined by the staff at the beginning of the exam.
The requirements for grades are as follows (part A, B, C)

- grade E (16, -, -)
- grade D (20, 6, -)
- grade C (20, 10, -)
- grade B (20, 10, 4)
- grade A,(20, 10, 8)

A1:	A2:	A3:	A4:	A5:	A6:
A7:	A8:	A9:	A10:	A11:	A12:
A13:	A14:	A15:	A16:	A17:	A18:
A19:	A20:	A21:	A22:	A23:	A24:

Part A

A:1 1p. What aspect makes distributed systems most different from non-distributed systems?

- [A] harder to handle failure
- [B] marshaling of data structures
- [C] have to handle concurrency
- [D] privacy and authentication

A:2 1p. What is meant by location transparency?

- [A] a replicated resources is accessed exactly as if it was a single object
- [B] a resource will handle all request equal independent of location of client
- [C] remote resources are accessed using location independent names
- [D] local and remote resources are access using the same operation

A:3 1p. What is provided by TCP?

- [A] a best effort delivery of messages
- [B] transactional control and persistence
- [C] a guarantee that messages will always reach its destination
- [D] a full-duplex stream between two processes

A:4 1p. What is a good reason for choosing UDP rather than TCP?

- [A] you have small messages that should be sent with little delay
- [B] UDP will guarantee the delivery of a message
- [C] you need to know that a message is handled by the remote application
- [D] you have large messages or a sequence of messages

A:5 1p. What is meant by *time uncoupling* in a communication framework?

A A sender does not need to know the name or identifier of the receiver.

B Messages are resent if lost, providing a reliable service.

C The sender does not wait for a acknowledgment from the receiver.

D Sender and receiver need not be active at the same time.

A:6 1p. What is meant by *causal ordering* in a communication framework?

A If two clients send messages *m1* and *m2* independently of each other, then a receiver will receive these in the real-time order in which the send operations where issues.

B Clients will be able to send messages in a *round-robin* order thus preserving inter-ordering relationships.

C If a client is delivered two messages *m1* and *m2*, then *m1* could not have been sent by someone after having being delivered *m2*.

D If a sender is sending two messages *m1* and *m2*, then a receiver will be delivered *m1* before *m2*.

A:7 1p. If a RPC call with *at least once* semantics fails, we know that:

A the call has not been executed at all

B the call has either been executed once or not at all

C the call has been executed at least once

D the call might have been executed at least once

A:8 1p. Does Erlang provide a form of location transparency?

A Yes - a process can use a process identifier without having to know the address of the node where the process lives.

B No - since the address of a registered process contains the IP address of the node, there is no transparency in the system.

C No - you always need to know the node address of a process.

D Yes - there are no explicit node addresses in the system.

A:9 1p. What accuracy can be provided using Christian's algorithm?

A $\pm(T_{round} \div 2)$

B $\pm(T_{round} \div 2 - \text{minimum latency})$

C $\pm(T_{round} \times 2 - \text{minimum latency})$

D $\pm(T_{round})$

A:10 1p. What does the reply from a NTP server contain?

A the send time of the reply

B the latency of the request and the send time of the reply

C send and receive time of request and send time of reply

D the delta of the client time stamp and the server time

A:11 1p. What is true for events A and B?

A if A occurred in real-time before B then A *happened before* B

B if A occurred in real-time before B then A caused B

C if A caused B then A *happened before* B

D if A *happened before* B then A caused B

A:12 1p. What can we know if we use Lamport clocks?

A if $L(a) < L(b)$ then a must have caused b

B if $L(a) < L(b)$ then a occured in real-time before b

C if $L(a) < L(b)$ then a *happened before* b

D if $L(a) < L(b)$ then a could have caused b

A:13 1p. When is it problematic to use vectors clocks?

A if we have an asynchronous system

B when we do not have an elected leader process

C when we have a dynamic set of processes

D if nodes are not synchronized using for example NTP

A:14 1p. What is the definition of a consistent cut?

- A if e and f are in the cut then e and f have a causal order
- B if e is in the cut and f happened-before e then f is in the cut
- C all events in the cut are strictly ordered in a *happened before* order
- D if e *happened before* f and e is in the cut then f is in the cut

A:15 1p. What is the definition of a stable global state predicate?

- A the predicate will eventually hold true in all linearizations
- B the predicate will never hold true in any consistent state reachable from the original state
- C if a system enters a state where the predicate holds true it will remain true in all future states
- D the predicate holds true in all consistent global states

A:16 1p. What is the benefit of a reliable multicast?

- A messages are guaranteed to be delivered to all correct processes
- B messages are delivered in total order
- C messages are delivered within a upper bound time
- D the sender will not crash during an operation

A:17 1p. What is the difference between *uniform agreement* and *non-uniform agreement* for multicast?

- A Non-uniform agreement does not include the reliability requirement.
- B Uniform agreement is the combination of total and causal ordering.
- C In non-uniform agreement correct nodes can decide different.
- D Uniform agreement includes the behaviour of non-correct node.

A:18 1p. What does Lamport clocks give us when implementing distributed mutual-exclusion?

- A request are granted in *happened before* order

B at most one process may enter the critical section at a time (safety)

C the system is prevented from entering a dead-lock

D requests will be granted in real time order

A:19 1p. What is two-phase locking in a transaction?

A reserving all locks before taking the first

B taking locks in a strict order

C not taking any locks once a lock has been released

D taking a read lock that is strengthened to a write lock

A:20 1p. What does it mean that a transaction meets the isolation property?

A effects of the transaction are isolated from server failures

B the transaction can safely be duplicated resulting in the same server state

C intermediate results must not be visible to other transactions

D either all or no operations in the transaction are performed

A:21 1p. In a distributed transaction one often use two-phase commit, why is this better than one-phase commit?

A We prevent locks from being taken once we have released the first lock.

B We ensure that if one transaction aborts then no transaction will commit.

C We prevent dirty-read operations to cause a cascading abort scenario.

D One-phase commit can suspend indefinitely if the the coordinator dies.

A:22 1p. In a view-synchronous group membership protocol, a process that enters the group, and is included in the delivered view, will be guaranteed to be delivered:

A only the messages delivered before it joined

B all messages starting from the view where it is included in the group

C all messages in the history of the group

D all messages starting from the active view when it issues its request to join

A:23 1p. What is sent by the primary replica manager to the backup replica manager in a passive replicated system?

A A snapshot of the state after each request has been handled.

B A copy of the request time stamped with a vector clock that provides the information on how to order the request.

C A copy of the original request together with a unique identifier.

D A unique identifier, the state change and the response.

A:24 1p. What is the purpose of hashing in a DHT?

A to reduce the original identifiers to a smaller key space

B to authenticate a request

C to generate uniformly distributed keys

D to more quickly find the responsible node

Part B

B:1 2p. When defining a request message layer, one has the option of choosing to provide *at-most-once* or *at-least-once* semantics. Describe an advantage with the *at-least-once* semantics but also describe what it means to the application layer.

B:2 2p. Briefly describe a situation that meets the requirements of total order multicast but not to FIFO order multicast.

B:3 2p. If you need to replicate a server that mainly handles compute intensive read request i.e. request that do not change the state of the server, what would then be the best strategy for replication, active or passive, and why?

B:4 2p. Describe an efficient algorithm that allows you to detect a distributed dead-lock.

B:5 2p. In a quorum based algorithm the quorum is often chosen to be the majority of processes. You could however choose other quorums, what is the most important criteria when you divide nodes into quorums?

B:6 2p. Describe with a message diagram the difference between a service that is *linearizable* and one that only provides *sequential consistency*. Why would we settle for an service that only gives us sequential consistency?

B:7 2p. What would the advantage be to use a gossip protocol when building a replicated service? What would we maybe have to sacrifice?

B:8 2p. Why would you use a snap-shot algorithm, what is collected and what can you determine?

Part C

C:1 4p. What we are often looking for is what caused something to happen in a distributed system. To our help we have Lamport clocks, vector clocks and real-time clocks. How are these clocks related to each other and how are they related to causality? Describe the pros and cons of using these clocks, and how good they are in helping us track down the events that caused something to happen.

C:2 4p. Assume that we have a distributed system where processes communicate by sending messages. We have an *unstable predicate* that we want to determine if it is true during an execution. How would we go about to answer this question? Outline what an implementation would look like and what limitations there would be. What could you do to limit the amount of messages and the size of messages?

C:3 4p. You're building a high performance transaction server using time-stamp concurrency control. Describe what information you save for each object in the data-base and the rules for the read and write operations.

When will a transaction suspend and when will it have to abort? Why would it be an advantage to save multiple committed values and how would you make sure that you're not storing data that is no longer needed?