This exam has two parts: Part I consists of true/false questions and Part II consists of short answer problems. In Part I, there are 11 true/false questions with justification. Each question is worth 2 points if you answer correctly and 0 points, you answer incorrectly.

In Part II, there are many problems. Do all of them. Note that some problems can be answered more quickly than others.

Please write your answers legibly so that we can understand your answers. If a problem seems ambiguous, please state your assumption explicitly and solve the problem. Obviously, your assumption should be reasonable and should not trivialize the problem.

Please sign the pledge at the back of the answerbook. Good luck!
Part I. [2 points for each question] Answer True/False.

1. A file is replicated on 10 servers. The following combinations of (read quorum, write quorum) are permitted by the voting algorithm: (1, 10), (2, 9), (3, 7).
   
   **Answer:** False, (3,7) is not valid.

2. The model of triple modular redundancy described in the text and lecture can handle any type of failure, including Byzantine failure.
   
   **Answer:** True. The whole discussion assumed that failing elements put out random results, which are the same as Byzantine failures.

   We also gave 2 points for answering False since it may be possible to (incorrectly) interpret the question for the number of failure.

3. Suppose multicasing is used to send an image to a group by multicasting images in small fragments, where each fragment contains the (x, y) coordinate as part of its data. For this to work correctly every one must received the image fragments in the order that they are sent.
   
   **Answer:** False. No msgs ordering is needed since image fragments can be placed in correct positions as they are received.

4. The minimum number of processes needed for priority inversion to occur is three processes.
   
   **Answer:** True, since needs at least three processes to occur

5. The primary goal of the Byzantine Generals problem is to make it possible to reach an agreement even if some node crashes.
   
   **Answer:** False, it is to deal with liars.

6. A stronger consistency model is prefered as long as it meets the application’s performance requirements since a stronger consistency model allows less concurrency than a weaker consistency model, but makes it easier to reason correctness.
   
   **Answer:** True.

7. Suppose that two processes can communicate only by exchanging messages. Even if any message can be lost, it is possible for these two processes to reach a trivial agreement by exchanging a finite number of messages.
   
   **Answer:** True, in fact, no msgs are needed.

8. Update operations on replicated data store can be done using a pull-based or push-based protocol. The push-based approach is more efficient if read-to-update ratio is low; i.e., many updates between reads.
   
   **Answer:** False, no need to get a new value if it is not going to be used.

9. One of the contributions of ”The Structure of the THE-Multiprogramming System” by Dijkstra is the introduction of the notion of monitors.
   
   **Answer:** False, it introduces semaphores with P and V. Monitors are by Hoare.
10. Lamport’s happen-before logical clock guarantees that if the event E1 happened before the event E2, then the timestamp of event E1 is less than the timestamp of event E2, but not vice versa.

**Answer:** True.

11. It is possible to solve the Byzantine generals problem for a system with 2 faulty processes if there are at least 5 correctly working processes, for a total of 7 processes.

**Answer:** True, need \(2m + 1\) correct ones for a total of \(3m + 1\) for \(m\) faulty processes.

**Part II. [60 Points] Short answers. DO ALL.**

1. (5 points)

Consider the behavior of two machines in a distributed system. Both have clocks that are supposed to tick 1000 times per millisecond. One of them actually does, but the other ticks only 990 times per millisecond. If UTC updates come in once a minute, what is the maximum clock skew that will occur?

**Answer:** A: The second clock ticks 990,000 times per second, giving an error of 10 msec per second. In a minute this error has grown to 600 msec. Another way of looking at it is that the second clock is one percent slow, so after a minute it is off by 0.0160 sec, or 600 msec.

2. (5 points)

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.

**Answer:** Imagine the transfer of a large image which, to that end, has been divided into consecutive blocks. Each block is identified by its position in the original image, and possibly also its width and height. In that case, FIFO ordering is not necessary, as the receiver can simply paste each incoming block into the correct position.

3. (5 points)

Consider the following three concurrently-executing processes, assume \(x, y, z\) are (distributed) shared variables and initially all 0:

<table>
<thead>
<tr>
<th>Process P1</th>
<th>Process P2</th>
<th>Process P3</th>
</tr>
</thead>
<tbody>
<tr>
<td>(x := 1;)</td>
<td>(y := 2;)</td>
<td>(z := 3;)</td>
</tr>
<tr>
<td>(\text{print}(y,z);)</td>
<td>(\text{print}(x,z);)</td>
<td>(\text{print}(x,y);)</td>
</tr>
</tbody>
</table>

Assuming sequentially consistent data store, identify all possible execution sequences that would produce: 231312, where 23 means that \(y\) and \(z\) were 2 and 3, respectively, when \(P1\) executes its print, 13 means that \(x\) and \(z\) were 1 and 3, respectively, when \(P2\) executes its print, and 12 means that \(x\) and \(z\) were 1 and 2, respectively, when \(P3\) executes its print.
Answer: 3 x 2 (before any print)
3 x 2 for three print statements
So, 6 x 6 = 36

4. (5 points)
What is “Causal Consistency” model? For the above example, is it possible to produce: 000000 under causal consistency?
Answer: Writes that are potentially causally related must be seen by all processes in the same order. Concurrent writes may be seen in a different order on different machines. It is weaker than sequential consistency.
Yes, if and when prints are executed before writes are propagated.

5. (5 points)
To implement totally ordered multicasting by means of a sequencer, one approach is to first forward a msg to the sequencer, which then assigns it a unique number and subsequently multicasts the msg. Describe an alternative approach, and compare the two solutions.
Answer: Two additional approaches are possible.
The second approach is to multicast (including to the sequencer) a msg, but defer delivery until the sequencer has subsequently multicast a sequence number for it. The latter happens after the msg has been received by the sequencer. A third approach is to first get a sequence number from the sequencer, and then multicast the msg.
The first approach (send operation to sequencer), involves sending one point-to-point message, and a multicast message. The second approach requires two multicast messages: one containing the msg, and one containing a sequence number. The third approach costs one point-to-point message with the sequence number, followed by a multicast message containing the msg.

6. (5 points)
In the two-phase commit protocol, explain why blocking can never be completely eliminated, even when the participants elect a new coordinator.
Answer: After the election, the new coordinator may crash as well. In this case, the remaining participants can also not reach a final decision, because this requires the vote from the newly elected coordinator, just as before.

7. (5 points)
Explain how the write-ahead log in distributed transactions can be used to recover from failures.
Answer: The log contains a record for each read and write operation that took place within the transaction. When a failure occurs, the log can be replayed to the last recorded operation. Replaying the log is effectively the opposite from rolling back, which happens when the transaction needed to be aborted.
8. (5 points)
Devise a simple authentication protocol using signatures in a public-key cryptosystem. Describe the algorithm for the case that Alice wants to authenticate Bob.

**Answer:** A: If Alice wants to authenticate Bob, she sends Bob a challenge $R$. Bob will be requested to return encrypting with his private key $K_B^-(R)$, that is, place his signature under $R$. If Alice is confident that she has Bob’s public key, decrypting the response back to $R$ should be enough for her to know she is indeed talking to Bob.

9. (5 points)
Assume Alice wants to send a message $m$ to Bob. Instead of encrypting $m$ with Bob’s public key, she generates a session key $K_{A,B}$ and then sends $[K_{A,B}(m), K_B^+(K_{A,B})]$, where

- $K_B^-(m)$: ciphertext by encrypting msg $m$ with a private key belong to B.
- $K_B^+(m)$: ciphertext by encrypting msg $m$ with a public key belong to B.
- $K_{A,B}(m)$: ciphertext by encrypting msg $m$ with a shared key between and A and B.

Why is this scheme generally better in terms of performance?

**Answer:** The session key has a short, fixed length. In contrast, the message $m$ may be of arbitrary length. Since a symmetric cryptosystem is much more faster than a public-key criptosystem, the combination of using a session key and applying public-key cryptography to a short message will generally provide much better performance than using only a public key on a large message.

10. (5 points)
Name a few advantages and disadvantages of using centralized servers for key management.

**Answer:** An obvious advantage is simplicity. For example, by having N clients share a key with only a centralized server, we need to maintain only N keys. Pairwise sharing of keys would add up to $N(N - 1)/2$ keys. Also, using a centralized server allows efficient storage and maintenance facilities at a single site. Potential disadvantages include the server becoming a bottleneck with respect to performance as well as availability. Also, if the server is compromised, new keys will need to be established.

11. (5 points)
Is it possible to achieve a non-trivial agreement among four generals in which at most one of them is a traitor using only one-round of messages, where during one-round, each general sends (point-to-point communication) its value to the other generals? To simplify the argument, assume that the agreement decision is based on a majority vote with evenly spitted votes considered no majority; for example, the decision on values $(1,1,1,0)$ is 1, whereas the decision on values $(1,1,0,0)$ is 0.

If yes, prove it. If no, explain your answer with an example.

**Answer:** No. Here is a counter-example. Let’s label the four generals as A, B, C, D. Assume A is a traitor and their values are $(-,1,1,0)$. During the first round: A sends 1 to B, 0 to C, 1 to D. B sends 1 to A, C, D. C sends 1 to A, B, D. D sends 1 to A, B, C. After the
first round, B has (1,1,1,0), C has (0,1,1,0) D has (1,1,1,0). Thus, using the majority decision procedure described in the problem statement, B decides 1, C decides 0, D decides 1. That is, B, C, D have not agreed on a common value.

In general, if one value is critical in making a decision and if the traitor sends different values to different generals, there will be a case where decisions can not be uniform.

12. (5 points)

Describe the property of vector clocks.

**Answer:**

Vector clocks guarantees the following: the event $E_1$ happened before the event $E_2$ iff the timestamp of event $E_1$ is less than the timestamp of event $E_2$.

Illustrate how it works by filling in the values of a,b,c,...z in the following example:

**Answer:** a,b,f,c,d,m,l,p,r,t,u,v are 0;
n,k,o,q,s,w,y are 1;
x, z are 3;
e,g,h,i,j are 4.