University of Minnesota
Department of Computer Science
CSci 5105 - Spring 2015 (Instructor: Tripathi)
Midterm Exam — Date: March 11, 2015 (4:00 – 5:15 pm)
(Time: 75 minutes) Total Points – 100
This exam contains four problems.
(It has 12 pages and the last two pages are blank.)
CLOSED BOOK/CLOSED NOTES – NO Laptops, PDAs, or Cell Phones
Please write your answer in the space provided with each question.

STUDENT NAME:
STUDENT ID:

<table>
<thead>
<tr>
<th>Problem 1</th>
<th>Problem 2</th>
<th>Problem 3</th>
<th>Problem 4</th>
<th>Total</th>
</tr>
</thead>
<tbody>
<tr>
<td>66</td>
<td>14</td>
<td>10</td>
<td>10</td>
<td>100</td>
</tr>
</tbody>
</table>
Problem 1: (66 points)
(a) (6 points) When a failed leader comes back up in a group of servers, would it be more efficient for it to use the Bully algorithm or the Ring algorithm to regain leadership? Explain your answer briefly.

(b) (6 points) Explain why we can adjust the Lamport clock of a machine by a large amount upon message arrival, but must adjust the physical clock gradually to catch up with the correct time?
(c) (6 points) What is the superserver *inetd*? Briefly describe the concept and its benefit.

(d) (6 points) What is the function of *portmapper*? What information is maintained by it?
(e) (6 points) An RPC system can be implemented using either TCP or UDP (i.e. datagram) as the transport level protocol. Under what conditions it is more desirable to use UDP as the underlying transport protocol for an RPC system?

(f) (6 points) In quorum-based protocol for managing multiple replicas, why is the write quorum size required to be at least a majority of the replicas, and why is this requirement not necessary for the read quorum size? Under what conditions the read quorum size can be less than the majority?
(g) **(6 points)** Updates can be propagated in three ways: (i) send a full copy, (ii) send a notice of the change, and (iii) send just the update operation. When would each be preferred?

(h) **(6 points)** Protocols for synchronizing physical clocks avoid setting a clock back in time. What kind of problems can arise if the clock at a node is set back in time. Give at least three problems.
Consider a primary-backup system in which all writes are performed at the primary copy. The primary copy uses lazy (i.e. asynchronous) propagation of updates to the backup copies, and all updates are applied at the backups according to their sequence order at the primary copy. Read operations can be performed at any copy. Select the correct answer for the following questions:

(a) Do all processes reading data observe a causally consistent view?  
   YES  NO

(b) Is this system sequentially consistent?  
   YES  NO

(c) Does it have *read-your-writes* property?  
   YES  NO

(d) Does it have *monotonic-writes* property?  
   YES  NO

(e) Does it have *monotonic-reads* property?  
   YES  NO

(f) Does it have *writes-follow-reads* property?  
   YES  NO

Consider a server in which it takes 15 milliseconds to process a request if the data needed is in the main memory. If a disk operation is needed, as is the case one-third of the time, an additional 75 millisecond is needed, during which time the thread sleeps. How many requests/second can the server handle:

If it is single-threaded?  
If it is multi-threaded?
(k) (6 points) Consider the scenario shown in the figure below for synchronizing the physical clocks of node $P_1$ with that of node $P_0$. It is known that clock oscillator frequencies at these nodes may deviate by tolerance level $\rho$, which means the clocks at these nodes may be running at a rate bounded by $(1 - \rho)$ and $(1 + \rho)$. Node $P_1$ sends a request message to $P_0$, which then sends reply message containing its clock value at the time of sending the reply. As shown in the figure, $P_0$ includes value $T$ in the message. Node $P_1$ also measures, using its own clock, the round-trip time $D$ from the time of sending of the request message to receiving the reply message.

What is the range of possible values for the clock of process $P_0$ at the time when $P_1$ receives the reply message?
Problem 2: (14 points) Shown below is a scenario of communication between three processes: $P_1$, $P_2$, and $P_3$. The initial vector clock values for all these three processes are $<0, 0, 0>$. The messages being communicated at $m_1$, $m_2$, $m_3$, $m_4$.

(a) (6 points) Show the vector clock values for each process immediately after sending or receiving a message. Show your answer in the figure above.

(b) (4 points) Does this scenario satisfy the causal delivery order for all messages? Explain your answer.

(c) (4 points) How can we use vector clocks to detect violation of causal delivery order at the time of a message delivery? Briefly describe a scheme.
Problem 3: (10 points) Consider a system using the majority voting based approach for managing replicated data where each copy has one vote and a majority of votes is needed for reading and writing. Give the rules for implementing the read and write operations in this system to ensure *sequential consistency*. Clearly identify what metadata you will need to maintain with each copy. You may skip the details of the voting part of the protocol. Outline the steps in reading and writing.
Problem 4: (10 points) Consider a system with three processes $P_1$, $P_2$, and $P_3$. They all access a shared variable $x$ to perform read or write operations. The notation $W(x)a$ means writing of value $a$ in variable $x$. $R(x)a$ indicates a read operation on $x$ returning value $a$. Suppose that in this system, the following sequence of operations were performed by these three processes.

Process $P_1$: $W(x)b$ ; $R(x)a$ ; $R(x)c$

Process $P_2$: $W(x)a$;

Process $P_3$: $R(x)a$ ; $R(x)b$ ; $W(x)c$

(a) (5 points) Is this system sequentially consistent? If yes, then give a sequential execution sequence for these operations. If not, then explain your answer by indicating the operations which do not satisfy this property.

(b) (5 points) Is this system causally consistent? If yes, then show a causally consistent happened-before relationships between writes and reads. If no, then show a violation of causal consistency.
All blank page.
All blank page.