This is an open-book, open-note, closed-internet, closed-cellphone and closed-computer exam. Your exam should have 6 pages with 5 questions totaling 100 points. You have 75 minutes. Please write your answers in the space provided on the exam paper. If you unstaple your exam, please put your initials on all pages. You may use the back of pages if necessary, but please indicate if you do so so we know where to look for your solution. You may ask us for additional pages of scratch paper. You must submit all sheets you use with your exam. However, we will not grade what you scribble on your scratch paper unless you indicate you want us to do so. Answers will be graded on correctness and clarity. You may lose points if your solution is more complicated than necessary.

Name (printed) ___________________________________________________

I accept the letter and the spirit of the Virginia Tech graduate honor code – I have not given or received aid on this exam.

(signed) _______________________________________________________
1. **RPC.**

An RPC architecture, such as OMG’s CORBA, performs a number of services or tasks that allow a client A to invoke a procedure provided by a server object B. A key goal is to provide transparency, i.e., that client A should be unaware of where server object B is located.

a) Give two examples of tasks or services that an RPC system does *not* have to perform if A and B happen to be located on the same machine, but in different processes.

b) Give two examples of tasks or services that an RPC system does *not* have to perform if A and B happen to be located within the same process.
2. **Eraser.**

Consider the following code fragment. Recall the naïve version of lockset refinement (i.e., without considering read-sharing and read-write sharing separately.) Consider the sets locks\_held(t) for a thread t and the candidate lockset C(x) for a shared variable x.

a) Complete the following table, adding entries whenever locks\_held(t) or C(x) changes. Write down the value of the respective set after the execution of the corresponding statement in the left-most column.

<table>
<thead>
<tr>
<th>locks_held(t)</th>
<th>candidate set C(x)</th>
</tr>
</thead>
<tbody>
<tr>
<td>mutex mu1, mu2; lock(mu2); if (x == 0) { lock(mu1); unlock(mu2); if (x == 0) { x = 5; } unlock(mu1); } else { unlock(mu2); }</td>
<td>Ø</td>
</tr>
</tbody>
</table>

b) Will Eraser flag a race condition? Say why or why not.
3. Logical Clocks.
Consider a system of two processes that are exchanging messages. Ignore internal events for the purposes of the problem: all events of interests are either sending or receiving of messages.

a) Construct a scenario (timeline) in which for two events $a$ and $b$ the following is true. (Show your work.)
   - $a < b$ according to the total order $<$ provided by Lamport timestamps with process-id tiebreaker.
   - $a$ and $b$ are concurrent, that is, neither $a \rightarrow b$ nor $b \rightarrow a$ is true.

b) Now assume the processes used vector clocks instead. Give the vector timestamps for $a$ and $b$ and explain how they show $a \parallel b$. 
4. Proportional Share Scheduling (VTRR).

Nieh’s paper defines the service time error of a client $A$ during an interval $(t_1, t_2)$ as:

$$E_A(t_1, t_2) = W_A(t_1, t_2) - (t_2 - t_1) \frac{S_A}{\sum_i S_i}$$

Assume a system with only two clients, $A$ and $B$, which are both runnable during $(t_1, t_2)$. Express $B$’s service time error $E_B(t_1, t_2)$ as a function of $E_A(t_1, t_2)$ (Show your work.)
5. **End-to-end Arguments.**

Describe one design decision from one of the systems we discussed in class in which designers followed Saltzer’s end-to-end argument. Describe the decision and outline the choice taken by the designers as well as the alternative choice that would have run counter to the end-to-end argument. For partial credit, give an example not mentioned in the papers we read, but merely rehashing an example included in Saltzer’s paper will get you no credit.