Section A

1)

a) Define the term *scalability* as it is applied with respect to distributed systems. [4]

b) Briefly outline the issues that a distributed systems designer must consider if they are to build a scalable system. [6]

c) ACME systems, has decided that it will go into the business of providing service management facilities to people who purchase Bluetooth enabled products. Their customers will include a large number of people who are technically naïve, as well as some who are able and who will want to be able to create and install their own software. They expect that:

- Their market will be global
- There will be a wide range of devices with a wide range of different physical capabilities.
- There will be a wide range of applications, from small local, personal, applications, through to large cooperative multimedia applications.

Their remit is to ensure that all of the applications they are contracted to support are installed on all the devices for which they have a service contract. They expect that application updates and device registrations will happen frequently and at unpredictable intervals. They must also continue to operate in the presence of failures.

Given your answer to part (b), devise a system architecture and argue that it will allow them to fulfil their remit. Your answer should include:

i) A description of the components of the architecture and the ways in which they are connected.

ii) A description of events of importance in the system and the actions to be taken when those events occur.

iii) An argument that concludes that the system is scalable.

iv) Some statement about stability. [15]

[TURN OVER]

2)

a) A Lamport logical clock requires that for two events *a* and *b*, then, if $a \rightarrow b$, the logical clock value associated with *a* is less than the logical clock value associated with *b*. i.e. $a \rightarrow b \Rightarrow C_i(a) < C_j(b)$

Demonstrate that it is possible to devise a simple implementation of logical clocks that satisfies these properties. [7]

b) How may logical clocks be used to implement semaphores? Which of the following properties does your solution possess? Prove it.

i) Freedom from deadlock

ii) Freedom from starvation

iii) Fairness

iv) Fault Tolerance [11]

c) Logical clocks generate a causal ordering of events. Thus, for example, we may have five events ordered as:

$A \rightarrow C \rightarrow E$
$A \rightarrow B$
$C \rightarrow D$

A total ordering consistent with this would be: $A \rightarrow B \rightarrow C \rightarrow D \rightarrow E$, but an equally good ordering would be $A \rightarrow C \rightarrow E \rightarrow D \rightarrow B$. There are others.

i) Give an example of a situation in which it is necessary to ensure a total ordering of events in a system.

ii) How could you extend the concept of logical clocks to achieve this in a distributed fashion? [7]

[CONTINUED]

3)

a) What types of failure can there be in a simple bidirectional client-server interaction and what action is appropriate for each form of failure? [10]

b) Assuming you have UDP, write pseudocode to illustrate how you would implement three different forms of RPC call semantics: at most once, at least once, and exactly once. Ensure that your implementations are tolerant to the types of failure you identified in part (a), where appropriate. State any assumptions you make. [15]

[END OF SECTION A]