One and a half hours

UNIVERSITY OF MANCHESTER
SCHOOL OF COMPUTER SCIENCE

Fundamentals of Distributed Systems

Date: Monday 6th June 2016

Time: 14:00 - 15:30

Please answer any TWO Questions from the THREE Questions provided.

Use a SEPARATE answerbook for each QUESTION.

This is a CLOSED book examination

The use of electronic calculators is permitted provided they are not programmable and do not store text

1. a) Imagine you are a software engineer in Goodlet, a web-based company whose main business process relies on aggregating adverts of properties for sale. These adverts come from a large number of estate agents. Goodlet makes money by charging each estate agent a fee every time that a Goodlet website user converts into a prospective customer (say, by requesting further details on a property, or by requesting to visit a property, etc.).

   i) Jack, a colleague of yours, has argued that, *from the viewpoint of the estate agents*, the main reason for them to participate in the distributed system created by Goodlet, is *non-functional*. State whether or not you agree with Jack and briefly explain why.

   (3 marks)

   ii) Jill is another colleague of yours. She has argued that Goodlet should expand its business processes by developing a RESTful API. This would allow Goodlet to charge third-party applications (say, interior decoration companies, etc.) for access to the estate agent data they hold. Goodlet would then pay the estate agents a share of the revenue that had been generated in this way. Explain why this new initiative would characterize, *from the viewpoint of the estate agents*, a *functional* reason to participate in the distributed system created by Goodlet.

   (3 marks)

   iii) Assume that Goodlet adopted Jill's proposal (in 1(a)ii above) for exposing a RESTful API for third-party applications to access their data. Argue that, *in this extended Goodlet system*, all the components have the features required in the definition of distributed system discussed in this course unit. (5 marks)

   b) In this course unit, we placed five computation models in a spectrum, from [1] the basic centralized model to [5] those distributed systems that rely on middleware. Choose two of the remaining models and explain what their distinctive features are. (3 marks)

   c) *Multi-CPU* systems are a kind of of modern chip multiprocessor (CMP) in which processing units that are physically separate rely on a separate interconnect (in other words, a bus) to communicate. For each one of the eight axioms of distributed computing discussed in this course unit, briefly explain whether that axiom applies or not to a multi-CPU CMP. (In your answer, you must take the bus that connects the multiple CPUs to be what in the axioms is referred to as "the network".)

   (8 marks)

   d) An operating system (OS) uses *priority-based preemptive scheduling*. This means that an OS takes into account the priority of a job before it suspends the execution of that job in order to allow another process its share of CPU time. Given the above, briefly explain why graphical user interfaces (GUIs) are likely to be treated as high priority by OS schedulers. (3 marks)

2. a) The five-layer Internet protocol stack was briefly discussed in this course unit.

   i) Briefly characterize the difference between the transport, the internet and the link layers. (3 marks)

   ii) State three differences between TCP (Transmission Control Protocol) and UDP (User Datagram Protocol). (3 marks)

   iii) State whether you agree with the claim that *"HTTP is stateless because it implicitly manages a global state for a connection, thus removing the need for clients and servers to manage resource state."* and briefly explain your reasons. (2 marks)

   b) Briefly explain how MAC addresses, IP addresses and domain names, together, provide a global mechanism for addressing servers on the Internet. (3 marks)

   c) Consider Figure 1. It depicts eleven distributed components `C1` to `C13`. Let a dashed arrow (such as the one from `C5` to `C4`) denote a message send (or request) and let a solid arrow (such as the one from `C4` to `C2`) denote a message receive (or response).

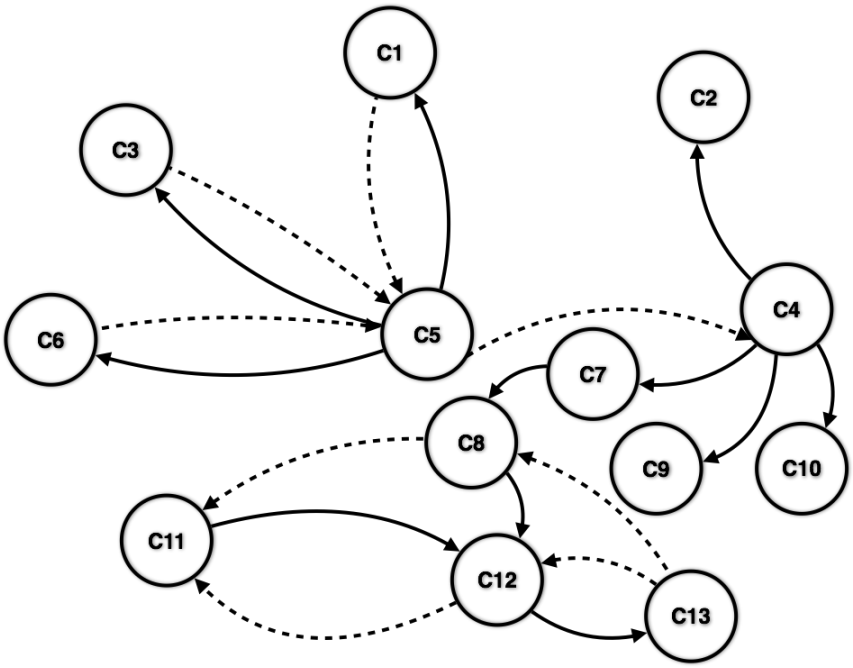

Figure 1: Distributed Components

   i) Identify the subset of components in Figure 1 that seem to be in *a client-server relationship* motivating your answer in terms of the request-response pattern that applies to that subset of components. (3 marks)

   ii) Identify the subset of components in Figure 1 that seem to be in *a peer-to-peer relationship* motivating your answer in terms of the request-response pattern that applies to that subset of components. (3 marks)

iii) Identify the subset of components in Figure 1 that seem to be engaged in *mediated message exchange* motivating your answer in terms of the request-response pattern that applies to that subset of components. (3 marks)

d) Consider the functions `foo` and `bar` in Figure 2. Note that they share the global variables `GX` and `GY`.

```
def foo(x,y):
  global GX
  global GY
  GX = x + 1
  GY = x**2

def bar(x,y):
  global GX
  global GY
  GY = y + 1
  GX = x / 3  # <- integer division
```

Figure 2: Variable-Sharing Functions

i) Reason whether the calls `foo(GX,GY)` and `bar(GX,GY)` can be scheduled in any order without the risk of the final outcome being different for different schedules. You can, as a test case, use `GX = GY = 0` as the initial system state. (In your argument, you do not need to show each step in computations.) (2 marks)

ii) Consider now the expression `foo(GX,GY) + bar(GX,GY)`. Can we evaluate the left and right terms of this expression concurrently? Justify your answer. (3 marks)

3. a) Consider the sequence diagram in Figure 3. It depicts processes `P1` and `P2` and their interactions over time with resources `R1` and `R2`. The vertical arrows are process and resource timelines, therefore the interactions denoted by the horizontal arrows take in place in time sequence from top to bottom. The interaction types (namely, `request x-lock`, `grant`, `release`, and `wait`) are denoted by the labels on the horizontal arrows and their meaning is assumed to be obvious (where `x-lock` is shorthand for *exclusive lock*, i.e., a mechanism for mutual exclusion over access to the resource that grants the lock).

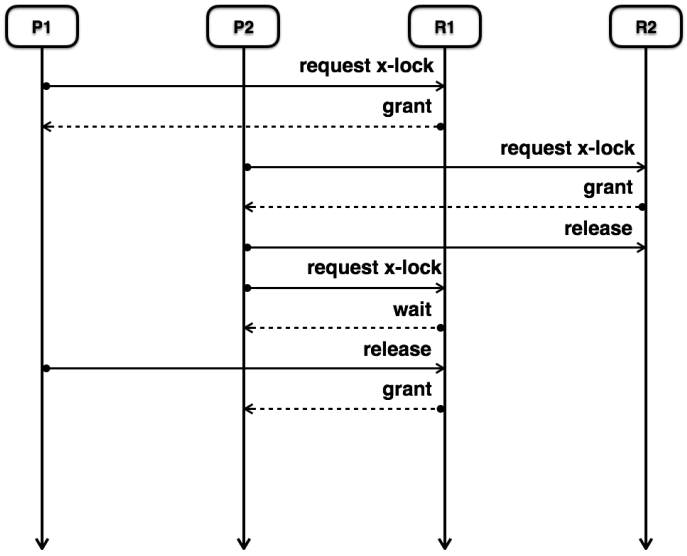

Figure 3: Process-Resource Interactions

i) State and briefly define the four Coffman conditions for deadlock to occur. (4 marks)

ii) Reason from the Coffman conditions and conclude that the system in Figure 3 is not deadlocked. Make explicit any assumptions you need for your argument, if any. (3 marks)

iii) Draw another sequence diagram over the same processes and resources as well as the same interaction types in Figure 3 so that it shows a deadlocked system. Explain why the deadlock arises using the Coffman conditions. Make explicit any assumptions you need for your argument, if any. (6 marks)

b) Assume that, when using a logical clock system, we have the following relationships between events A, B, C and D in a distributed system:

$$\begin{array}{ccc} A & \rightarrow & B \\ C & \rightarrow & D \\ D & \rightarrow & A \\ B & \rightarrow & D \end{array}$$

Reason from the meaning of a logical clock that this is (or is not) a consistent set of assertions. (4 marks)

c) The next two subparts are about the Chord distributed hash table.

   i) Briefly explain what is the main purpose of a *finger table* in Chord. (2 marks)

   ii) On an N-node network, of what order is the number of nodes that must be contacted to find a successor? (1 mark)

d) The next three subparts are about distributed architectures in the context of the Doom and Quake video games that were discussed in this course unit.

   i) Of the architectural paradigms discussed in the course unit, which one was Doom essentially configured as? (1 mark)

   ii) How did Quake differ from Doom in terms of architectural paradigm? (2 marks)

   iii) Two limitations arising from Doom's architectural choices were overcome by Quake's adoption of a different architectural paradigm. Explain which two limitations were these. (2 marks)

END OF EXAMINATION