

---

**The University of Queensland**  
**School of Information Technology and Electrical Engineering**  
**Semester Two, 2009**

**COMS3200/COMS7201 – Tutorial 2**

**Questions**

1. In a large computer network, there are three identical servers for a particular operation. How can the RPC model be adapted to handle this situation?
2. Parameter passing in RPC is easiest if all the computers (and operating systems) are identical. Imagine, however, that RPC is being used among machines whose binary representations for integers, floating point numbers, and characters are all different. Suggest two strategies for handling this problem and discuss their strengths and weaknesses.
3. An RPC might take a long time to complete. If it takes too long the client may time out and retransmit the request, leading to a variety of complicated situations. How can you reduce this problem?
4. Consider blocking and non-blocking send and receive message passing primitives. Which might you use to construct an RPC system with
  - (a) at most once semantics, and
  - (b) at least once semantics?How? (You should assume the existence of timeout primitives also.)  
What type of control information would be in the various messages passed?
5. (PD5.40) Suppose we were to implement remote file system mounting using an unreliable RPC protocol that offers zero-or-more semantics. If a message reply is received this improves to at-least-once semantics. We define `read()` to return the specified  $N$ th block, rather than the next block in sequence; this way reading once is the same as reading twice and at-least-once semantics is thus the same as exactly once.
  - (a) For what other file system operations is there no difference between at-least-once and exactly-once semantics? Consider `open()`, `create()`, `write()`, `seek()`, `opendir()`, `readdir()`, `mkdir()`, `delete()` or `unlink()`, and `rmdir()`.
  - (b) For the remaining operations, which can have their semantics altered to achieve the equivalence of at-least-once and exactly-once? What file system operations are irreconcilable with at-least-once-semantics?
  - (c) Suppose the semantics of the `rmdir()` system call are now that the given directory is removed if it exists, and nothing is done otherwise. How could you write a program to delete directories that distinguishes between these two cases?
6. Consider a network of automatic teller machines (ATMs) and banks. ATMs belong to a certain bank and only communicate with that bank. ATMs deal with a single withdrawal or balance request at a time, sending the request to their bank and waiting for a reply before taking some action (dispensing money or showing/printing an account balance). If an account-holder uses an ATM belonging to a bank other than their own, the bank which receives the request (i.e. ATM owner) must forward the request to the account holder's bank and forward the reply back to the ATM when it receives it. Banks must also send details of such forwarded transactions to a central clearinghouse (which will calculate the appropriate inter-bank fees the banks will charge each other). No replies are expected from the clearinghouse. All processes are single-threaded and use a single communication end-point.

---

Banks must be able to process multiple transactions at once. What interprocess communication primitives should the ATM, bank and clearinghouse processes use to send and receive messages?

7. (PD7.7) Give the big-endian and little-endian representations for the integers (a) 101 (b) 10,120 (c) 16,909,060. Assume a 32-bit representation.