

6 - Communication

Tutor version

Dr Leonardo Mostarda,
School of Science and Technology,
Camerino University, London
Version 1.0, 10 Mar 2013

Question 1

Describe how connectionless communication between a client and a server proceeds when using sockets.

Solution 1

Both the client and the server create a socket, but only the server binds the socket to a local endpoint. The server can then subsequently do a blocking read call in which it waits for incoming data from any client. Likewise, after creating the socket, the client simply does a blocking call to write data to the server. There is no need to close a connection.

Question 2

Suppose that you could make use of only transient asynchronous communication primitives, including only an asynchronous receive primitive. How would you implement primitives for transient synchronous communication?

Solution 2

Consider a synchronous send primitive. A simple implementation is to send a message to the server using asynchronous communication, and subsequently let the caller continuously poll for an incoming acknowledgment or response from the server. If we assume that the local operating system stores incoming messages into a local buffer, then an alternative implementation is to block the caller until it receives a signal from the operating system that a message has arrived, after which the caller does an asynchronous receive.

Question 3

Suppose that you could make use of only transient synchronous communication primitives. How would you implement primitives for transient asynchronous communication?

Solution 3

This situation is actually simpler. An asynchronous send is implemented by having a caller append its message to a buffer that is shared with a process that handles the actual message transfer. Each time a client appends a message to the buffer, it wakes up the send process, which subsequently removes the message from the buffer and sends it its destination using a blocking call to the original send primitive. The receiver is implemented in a similar fashion by offering a buffer that can be checked for incoming messages by an application.

Question 4

Does it make sense to implement persistent asynchronous communication by means of RPCs?

Solution 4

Yes, but only on a hop-to-hop basis in which a process managing a queue passes a message to a next queue manager by means of an RPC. Effectively, the service offered by a queue manager to another is the storage of a message. The calling queue manager is offered a proxy implementation of the interface to the remote queue, possibly receiving a status indicating the success or failure of each operation. In this way, even queue managers see only queues and no further communication.