Lucas Kushner

Homework #2

Problem 3.16

By failing to implement the “at most once” paradigm in an RPC system, it would be possible for a user to execute the same call many times, a situation that is unlikely, and would arise from network errors or accidental repeat sending. By checking and logging the timestamp on messages, the server ensures that each message it receives is executed once at most.

In “exactly once”, the possibility that the server never receives the message is accounted for by confirming with the client when the message is received. If no confirmation is received, the client will resend the message. Without this paradigm in place, it is possible for the client to continue along a series of calls despite some of them not having been executed, which could lead to complicated issues.

If an RPC system were to have neither the “at most once” nor the “exactly once” semantic implemented, possible uses could still include systems where function calls don’t have an effect on the data that could cause conflicts (i.e. an RPC that simply displays data from the system. In this case, sending and servicing multiple requests is of little consequence)

Problem 3.17

Output at Line X: `0 -1 -2 -3 -4`
Output at Line Y: `0 1 2 3 4`

Problem 3.18

a) The advantage of synchronous communication is that the program will not continue until a successful send or receive has occurred, which minimizes errors with missing data due to communication errors. The upside of asynchronous communication is that other actions can be performed while the communication is happening, resulting in more efficient program execution. On the programmer level, it can sometimes be easier to program a synchronous solution because of its linear nature.

b) The advantage of explicit buffering is that messages are guaranteed to be sent, because the buffer has no capacity so the sender must block until it is received. However, an automatic buffering system allows for more flexibility in terms of managing when to send messages.

c) The benefit of doing a pass by copy is that editing the data that is passed to the new process will have no effect on the original process, which may be critical depending on the situation. However, pass by reference has much less memory and communication overhead. On the programmer level, there is little difference between the two.

d) Fixed-sized messages make the system level implementation very straight forward since it is standardized, but the programmer level implementation more difficult because of having to adhere to the standard. On the other hand, variable-sized messages are easier to program, but require a more advanced implementation on the system level to accommodate the different possibilities.

Programming Problem 3.25
See attached Java files

Problem 4.8
b and c are shared across threads

Problem 4.11
It is possible to have concurrency but not parallelism, because concurrency allows multiple tasks to perform by scheduling execution such that all tasks make progress, whereas parallelism means the ability to perform several tasks at exactly the same time (i.e. having multiple cores). An example of such a system is a single-core processor that is executing several tasks with an even distribution of processor time.

Problem 4.13
a) Task parallelism
b) Both Task and Data parallelism
c) Data parallelism
d) Task parallelism