ECE 354: Part 4

Group: ECE.354.S11-G031
Members: Ben Ridder (brridder), Casey Banner (cccbanne), David Janssen (dajjanss)

This document is concerned with the high level design of our six testing processes. These processes are used by three test cases to check the validity of different parts of the operating system. The first test case is dedicated to testing the functionality of the delayed send primitive. Next, the memory management system is tested with regards to the blocked queue and allocation. Last, the third test case examines multiple situations where the operating system is expected to return an error. The final process is used as a test management system that tracks the success and failures of the prior test cases which inherently tests message passing.

Delayed Send Tests

This test case makes use of two processes: `test_delay_sender()` and `test_delay_receiver()`. The first process sends six delayed messages with unique delays to the receiving process. These messages are not sent in the order they are expected to be received in. Next, the receiver process runs and attempts to receive these messages in the correct order. If all the messages are received in the correct order a `TEST_SUCCESS` message is sent to the test management process, otherwise a `TEST_FAILURE` message is sent. If there is a problem with the message sending of the `delayed_send()` primitive the testing harness will hang.

Memory Management Tests

This test uses two processes, `test_memory_watchdog()` and `test_memory_allocator()`, to verify the functionality of blocked queues and memory allocation. First, the watchdog process allocates a block and then sends a message to the allocator process. Upon receiving this message the allocator requests as many memory block as are available. Once it is blocked the watchdog process will continue to run. The next step of the watchdog process is to release the memory block that it requested, and then release itself. At this point the allocator process becomes unblocked and is now able to release all of its memory blocks. If no deallocations fail then `TEST_SUCCESS` is sent to the the test management system, else `TEST_FAILURE` is sent.

Error Checking Tests

Test Management System