Adapted for CS-502 at Cisco System, Fall 2011
from a submission by Alfredo Porras, Summer 2011

Important notes:
1) Many debugging printk() lines are left in the kernel code because they give a good idea of what is going on in the kernel. Their output can be read in the message log.
2) To use the mailbox API make sure to include the provided "mailbox.h" in your test program.
3) Each output file has comments to help understand what is going on. To find these comments search the file for " <-- " (that is, 'less than', 'dash', 'dash' with no spaces in between).

…

Test programs

Test programs are named testmailboxN, where N is the number of the test program (1 through 7), and each has its out accompanying files, called testN.txt which shows relevant output from the kernel.

To compile all the test programs, the command "make all", or you can also "make" each file individually.

testmailbox1
This program simply tests if a single message can be sent and received. The main process forks a child (which blocks until it has received a message) and sends a message to it. Once the child receives the message it prints it. Output from running this program is shown in **test1.txt**.

testmailbox2
This program tests how processes behave when they have to wait until able to complete their system call. The parent process forks 50 children, all of which try to send a message to the parent. At the same time the parent tries to receive all 50 messages, but one process cannot easily keep up with 50 so it becomes very likely that the children will have to wait until able to send their messages. Output from running this program is shown in **test2.txt**.

testmailbox3
This program tests the behavior of stopped mailboxes. First the parent forks 50 children and sends a message to each one. Each children replies back to the parent. The parent sleeps for enough time for all of the children to try sending their message, and not all of the messages will fit into the parent's mailbox. Once the parent awakes it stops its own mailbox and retrieves all the messages that were inside. After the mailbox is empty it tries to retrieve another message (block == true). Then a new child is forked and it tries to send a message to its parent's stopped mailbox. Output from running this program is shown in **test3.txt**.

testmailbox4
This program tests the flushing of messages done by a mailbox that is being destroyed. The parent forks a child, which sleeps for 3 seconds. The parent then has enough time to send two

messages to the child. It then waits for 5 seconds to allow for enough time for the child to wake up and exit. When the child exits it should flush the two messages that are in its mailbox. Output from running this program is shown in **test4.txt**.

testmailbox5
This program test if a program can delete its mailbox before processes that were waiting to send a message are able to exit. The parent process forks 50 children, all of which try to send a message to the parent. 18 of the children will have to wait to be able to complete sending the message. The parent sleeps for enough time for the children to be waiting to send the message, and once it wakes up it exits. The children that were waiting should be able to get out before the parent destroys the mailbox. Output from running this program is shown in **test5.txt**.

testmailbox6
This program is similar to testmailbox5, but this one tests whether a mailbox is only destroyed when the last thread of the process dies. The main process spawns 5 threads each of which sleeps for one second more than the previous one. The main process forks children and has them send messages to its mailbox. The threads then start dying one by one. Output from running this program is shown in **test6.txt**.

testmailbox7
This program is similar to testmailbox5, but it does not give the children enough time to reply before trying to delete the parent's mailbox. This guarantees that there are many pointers in many different places in the system call that sends messages to the parent. The intention is to stress the subtle race condition that occurs when deleting a mailbox. To increase the stress a greater number of children can be forked and the parent can sleep for less time (or none at all) before trying to exit. I have been able to run this program successfully a few times. Output from running this program successfully is shown in **test7.txt**.