Mailbox Algorithm

The kernel module is split into four parts. One, contained in mailbox.c, manages the individual mailboxes and their functions. Another, contained in mailbox\_manager.c, manages the hashtable containing mailboxes. The third, contained in message.c, handles creating and deleting messages. Lastly, the code in module.c is the LKM interface, containing the system call overrides and the code to insert the overrides.

Module:

**Send Message:**

* Get mailbox (and “claim” a copy of it)
* Create message
* Add message to destination mailbox
* Unclaim mailbox

**Get message:**

* Get mailbox (and “claim” a copy of it)
* Get message
* Copy message information to userspace
* Unclaim mailbox

**Manage mailbox:**

* Get mailbox (and “claim” a copy of it)
* Copy number of messages to user
* If user wants to stop mailbox, stop mailbox
* Unclaim mailbox

**Exit thread:**

* If number of live threads is 1, remove mailbox
* Continue exit

**Exit group:**

* Remove mailbox
* Continue exit

Mailbox:

**Add message to mailbox:**

**Remove message from mailbox:**

**Stop mailbox:**

* Lock mailbox
* Set mailbox stopped to stopped
* Wake up all locked threads and let them exit
* Unlock mailbox

**Exit mailbox:**

* Same as stop mailbox, but set mailbox stopped to stopped *and* exiting

**Destroy mailbox:**

* Stop mailbox
* Lock mailbox
* Wait until all threads have stopped using the mailbox
* Remove all messages from message list and destroy them
* Unlock mailbox
* Wait until no more functions have claimed the mailbox
* Free memory cache for messages in the mailbox

Mailbox manager:

Init:

* For all the elements in hash table, init the hash list

Get mailbox:

* Try to find mailbox in hash table
* If mailbox not in hashtable, create mailbox, put mailbox in hash table
* Claim mailbox
* Return mailbox

Remove mailbox:

Tests