

Lab 3: Creating a Messagebox System

Karl El Hajal

November 8, 2017

Documentation

- The messagebox directory contains the messagebox.c file where I implemented all the system calls.
- messagebox.h contains the messagebox struct I used along with the message struct which I added to simplify the transfer of messages.

Design Choices:

1. topic, description, len and mb_id were left as shown in the assignment's description.
2. To implement subscribing to a messagebox, I added a subscriber_id array which holds the ids, a number_of_subscribers variable which keeps track of the current number of subscribers, and a max_subscribers variable that stores the max number of subscribers for the messagebox.
3. To implement sending messages, I added a message struct that holds the message's id, the length of the message, the message itself, its type (1 for Broadcast, 2 for Unicast and 3 for both), and the destination's ID in case of a Unicast.

The messagebox struct has a variable that keeps track of the number of messages and one which identifies the max number of messages, and it holds a pointer to a pointer of messages which will hold the messagebox's messages.

- I also added the syscall_64.tbl and syscalls.h files to which I added the newly implemented system calls.
- messagebox_test.c contains the program I used to test the messagebox system as I was implementing it; I call every system call in it and make sure to check that they work correctly.
- chinese_whispers.c contains the required Chinese Whispers game where 8 processes each take turns to modify a randomly chosen character to another randomly chosen character from an initially chosen string.