Message Slot

Load to kernel

Compile code and then load to kernel:

```
make
sudo insmod message_slot.ko
```

Then run `dmesg` to check which major the driver is associated with

Create device

Create device (with correct major, should be 240) and give it permissions

```
sudo mknod /dev/msgslot1 c 240 1
sudo chmod o+rw /dev/msgslot1
```

Send / Receive messages

First, compile the send/receive scripts:

```
gcc -O3  -Wall -std=c11 message_sender.c -o message_sender
gcc -O3  -Wall -std=c11 message_reader.c -o message_reader
```

Then, try sending and receiving some messages, for example:

```
./message_sender /dev/msgslot1 4 "Hello this is channel 4"
./message_reader /dev/msgslot1 4
```

And ensure that after executing the last command, it prints "Hello this is channel 4"

Try playing around with sending data to different channels, overriding data from an existing channel etc...

Testers (from the Google Drive)

C Tester (tester.c)

First, be sure the tester code uses the correct IOCTL macro you defined in message_slot.h

Then, compile the tester.

And then, ensure you have a CLEAN driver with the following by removing it if it's already loaded:

```
sudo rmmod message_slot
sudo rm /dev/msgslot1
```

Create a NEW device file and run the tester. If `/dev/msgslot1` already exists, first remove it using `sudo rm /dev/msgslot1` :

```
gcc -O3 -Wall -std=c11 tester.c -o tester
sudo mknod /dev/msgslot1 c 240 1
sudo chmod o+rw /dev/msgslot1
./tester /dev/msgslot1
```

C Tester 2 (tester2.c)

First, be sure the tester code uses the correct IOCTL macro you defined in message_slot.h

Then, you must ensure that you have a CLEAN driver and device, so if your driver is already loaded and you already have a device, run:

```
sudo rmmod message_slot
sudo rm /dev/msgslot1
```

compile the tester, create a NEW device file and run the tester. If `/dev/msgslot1` already exists, first remove it using `sudo rm /dev/msgslot1` :

```
gcc -O3 -Wall -std=c11 tester2.c -o tester
sudo insmod message_slot.ko
sudo mknod /dev/msgslot1 c 240 1
sudo chmod o+rw /dev/msgslot1
./tester2 /dev/msgslot1
```

Python Tester (tester.py)

Open the Terminal and run the following (when they ask to press Y/n, enter Y):

```
sudo apt-get update
sudo apt install python3-pip
pip3 install colorama
```

And then, cd into the project's directory, compile the reader and sender with the following:

```
gcc -O3 -Wall -std=c11 message_sender.c message_slot.h -o message_sender.o
gcc -O3 -Wall -std=c11 message_reader.c message_slot.h -o message_reader.o
```

Then, insure your driver is loaded using the following:

```
sudo insmod message_slot.ko
```

One the reader and sinder are compiled, and the driver is loaded, run the tester (make sure the tester.py file is in the directory): `python3 ./tester.py`

This tester might ensure more than what was required. It checks the receiver and sender output and expects to see the correct error messages which means that it assumes you're using perror/strerror and that your driver correctly sets the errno.