3. a) Discuss and evaluate what happens when you're running both in separate windows and you kill one or the other.

When I run both of them in separate windows, whatever kirk process says, Spock process hears it back. However, the message of Spock is not visible in the kirk process (windows).

"^D" kills the kirk process. If we kill kirk via typing "^D", both the kirk and the spock processes end. However, the kirk process doesn't end if we kill the spock process via typing "^D". If we quit either kirk or spock process via typing "^C", only the corresponding process ends and the other process remains unimpacted.

b) Discuss what happens (and why) when you run two copies of kirk.

If we run two versions of the kirk process, for the second process, we get an "*undefined reference to main*" error. The msgget() function returns the System V message queue when given the parameter of key and the memory location permission. Since the kirk process has set the permission to 0644, the memory location cannot be written by other processes. Because 0644 permission only gives the owner to read and write but not execution, while the others can only read. As a result, due to "*an undefined reference to main*," it returned 1 (error).

c) Discuss what happens (and why) when your run two copies of spock.

If we run two versions of the spock process, for the second process, we get two processes distinct spock processes. Both connected to the queue. The interesting result it every time, kirk process sends a message, the first message goes to the spock processes and the second message goes to the spock_2 process. Since 0644 gives read access to others, an infinite number of spock processes can be connected to the queue. Since the top process in the queue receives the most recent process, the first spock process (for my case spock) receives the first message and then the second message is received by the second spock process (for my case spock_2). Because once the first message is taken by spock, the top process in the queue shifts, and as a result, spock_2 becomes the top process and takes the second message.