

Question 3: Spock and Kirk

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

When both are running and you kill Spock, Kirk will continue running and is still able to send messages. However, since Spock isn't there to receive them, the any messages that Kirk sends will stay in the message queue. The queue will maintain the messages until it reaches capacity or Spock starts running again.

When both are running and you kill Kirk with SIGINT, Spock remains running since the message queue is still in place. When both are running and you kill Kirk with CTRL+D and `msgctl()`, Spock will also terminate since the message queue Spock is reading is now non-existent and the connection is lost.

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

When you run two copies of Kirk, Spock is able to process and display messages sent from both instances of Kirk in the same message queue. This works because both instances of Kirk have the same key generated by `ftok()`, so it connects to the same message queue in Spock. However, when one instance of Kirk sends CTRL+D to terminate the message queue, the other instance of Kirk cannot send any more messages since its queue was removed.

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

When you run two copies of Spock, the instances race to display the messages from Kirk before the other instance. Each instance is independent of each other, however, meaning they don't know another instance exists. The behavior that arises is an alternating pattern between the instances of Spock in the order of which they were executed. Each instance is sharing the same queue that Kirk is sending messages to.