

- A) When you run both programs, kirk takes in lines of text and spock echos these lines, I am assuming through a piece of shared memory. When you kill kirk with ^D, it kills spock as well as removes the identifier for spock(?). However when you kill spock with ^C, kirk continues to run.
- B) When you run two processes of kirk, spock continues to echo the messages made from both kirk processes. I believe this is because spock just reads when something new is put into the shared memory and prints whatever is in there so it would not matter how many instances of kirk there are running as long as they all place the inputted string into the same piece of shared memory.
- C) When you run two instances of spock, the two instances take turns echoing the line of text inputted into kirk. Each instance echoes every other line. I'd imagine this is because only one instance can access the piece of shared memory at a time so they both take turns accessing it, one after the other. Both instances are killed at the same time when you cancel kirk with ^D.