This project requires that you create four tasks. After the Init() task does all of it's initialization duties it should delete itself.

Task 1 should run for five seconds displaying "This is task one!" once every second. (Use the printf stdio routine for all output in this project.) At the end of five seconds, it should release a semaphore and then block waiting on an event.

Task 2 should run for four seconds displaying "This is task two!" once every second. Initially it is blocked waiting for Task 1 to release the semaphore. After four seconds Task 2 should send a message to Task 3 via a message queue. The message should be "The semaphore has been acquired N times." where N is the number of times Task 2 has acquired the semaphore.

Task 3 should run for three seconds displaying the message it received from Task 2 once every second. Initially Task 3 is blocked waiting for a message from Task 2. After three seconds Task 3 signals Task 1 via an event and the whole process repeats.

Task 4 monitors the keyboard waiting for the letter 'c' to be pressed by the user. A watchdog timer should be reset when the letter 'c' is entered. If a 'c' is not entered within a 30 second interval then the watchdog timer should send a signal to shutdown to task 1. Task 1 should delete Task 4 and delete the semaphore, which Task 2 should take as an indication to delete the message queue and then delete itself. When the message queue is deleted Task 3 should send an event to Task 1 and then delete itself. After Task 1 receives this final event notification it should shut the system down with a call to exit. Each task should display a “Task N received a shutdown notification.” (where N is the task number) before deleting itself or exiting.

Your application should to use as few resources as possible.

Submit printout of your source code. Submit your source code and Makefile in a zip or tar archive by email to richardson.tony@gmail.com. In the email subject line use “EE458 Project 4 – Your Name”.