Embedded System Design
Readers Writers

Max Thrun — Ian Cathey — Mark Labbato

October 29, 2012

Description

The purpose of this lab was to create a system where one task, called the "writer", continuously writes words into a circular buffer while multiple "reader" tasks read all the words that have been written. The reader tasks are to execute simultaneously each each other but exclusive from the writer task. To do this the system uses a combination of mutexes to control access to the shared buffered. The first issue is that we need to keep track of how many readers are currently accessing the buffer. To do this we use a variable called "num_readers" and a mutex called "mutex_num_readers". Each reader pends on access to this mutex and once it gains access it increments the "num_readers". When the reader is done accessing the shared buffer it tries to gain the mutex and then decrements "num_readers" since it is now finished. In order to lock out the writer task each reader checks the value of "num_readers" when it first gains access to it. If it sees that it is the first reader (num_readers == 0) it pends on "mutex_wr" which is a mutex that gets locked by the writer task when it is writing. In order to ensure that the writer is not starved we use a third mutex called "mutex_rd". The writer waits on this mutex which causes it execute as soon as possible since it has the highest priority. The reader tasks lock the "mutex_rd" during the critical section of keeping track of how many readers are running but it releases it immediately after. The delays in the readers are varied by their priority to ensure we are able to observe proper operation.