Real-Time Embedded Systems

1st Assignment Report

Code:

https://github.com/TheSolipsist/Real-Time-Embedded-Systems/blob/master/Assignment_1/prod-cons.c

Orestis Farmakis
8262

For the 1st assignment of this course, we were instructed to change the C code in a given file "*prod-cons.c*", in order to add a number of function pointers to a FIFO buffer, which would in turn be removed and the corresponding functions executed. The pointers are added with the use of N_PRODUCER producer threads, and they are removed and executed with the use of N_CONSUMER consumer threads. The aforementioned functions simply print the string "*General Kenobi! You are a bold one.\n*"

The number of producers is always given as 4 using the #define directive, while the number of consumers is (in a similar manner) given through the terminal while compiling the program, with the -D preprocessor option. The LOOP macro is given as 1000, which means that a total of 1000 * 4 = 4000 function pointers will be added to the buffer.

The goal of the present assignment is to find the number of consumer threads which minimizes the mean time that it takes for a consumer thread to remove a function pointer that a producer added. We predict this number to be big, since, the more threads that we have, the less time it will take for a consumer to receive a function pointer (note that, after we exceed the number of our CPU's hardware threads, the total execution time should be getting higher, but the mean time from addition to receipt of a function pointer should not, since more consumer threads will be waiting for the mutex unlock).

The following bar plot illustrates the relationship between the mean time and the number of consumer threads, as taken by executing the C program (which is linked on GitHub on the 1st page of this report) on an Intel i7 4770K @ 3.5GHz (4C, 8T). We find the optimal number of consumer threads to be 22, while testing 1-24.