

425 HW 10

1. Longest critical section
  1. Result: 250
  2. How determined: We determined by inspection that the critical section in YKQPost() was the longest, due to the fact that it calls so many functions and has the opportunity to be very long with multiple tasks in the suspend list. We then counted the lines of assembly code, including the maximum number of loop iterations that could happen in this lab in each loop in this section.
2. Worst case interrupt latency
  1. Result: 271
  2. How determined: The longest ISR runs 21 lines of machine code before reaching the first line of the interrupt handler.
3. Overhead to post to a semaphore
  1. Result: 82 machine instructions
  2. How determined:
    1. I counted the lines of assembly code. The scheduler takes 14 instructions if there is no change in task, and the dispatcher does not run. I also did not count the branch that moves TCBs to the ready list, since we are assuming no task is made ready.
4. Overhead to pend on a semaphore
  1. Result: 17 machine instructions
  2. How determined: I counted the lines of assembly code. Assuming the semaphore is available the scheduler will not be called.
5. Overhead to switch contexts with YKDelayTask
  1. Result: 157 machine instructions
  2. How determined: I counted the lines of assembly code. Up until the scheduler is called, YKDelayTask goes through 56 instructions. 39 of those come from removeFirstTCBFromRdyList(). Then the dispatcher will take 44 instructions. That plus the rest of the Scheduler code is 59 instructions. Those added together plus the first instruction of the new task adds up to 157.
6. Size of YAK C source code
  1. Result: 870 lines
  2. How determined: I ran the command `wc -l *.c`
7. Size of YAK assembly source code

1. Result: 194 lines
2. How determined:  
I ran the command `wc -l ISRs.s yaks.s`

8. Amount of memory required in bytes

1. Result: 11.3 kB
2. How determined: I looked at the size of the .bin file.