Yazan Halawa

ECEn 425

Homework 4

Dr. Archibald

1. I would use Round-Robin. Because there are no priorities involved. The pedestrians and the cars both get the same priority using the timer. So we just use polling to see if the timer has reached the required value and then switch the lights and reset the timer. It is very basic. Now say if specific lanes needed to have priority over others, then the matter would be different and I would switch to Round-Robin with interrupts or other architecture.
2. This is function-queue-scheduling. This is because the interrupts differ in their individual priority levels. Some interrupts have higher priorities than others. Moreover, They use a shared variable which is the wHighestPriorityFcn which decides which interrupt should be serviced next. Finally, there is a loop in the main that transitions to the interrupt that needs to be serviced next. All of these are attributes of function-queue-scheduling.
3. a. This macro returns the larger of two numbers.

b. Advantages: It is faster because since the macro is preprocessed, it avoids the overhead of a function call. Furthermore, macros are not limited to any specific parameter type.

Disadvantages: When passing expressions as the argument, the result could be an undefined behavior. For example passing x++ as one of the parameters to the specified macro could lead to incrementing the variable twice. Furthermore, writing multi line macros is a pain because it requires additional syntax like ‘\’ at the end of each line. Finally, order of operations could get weird depending on the parameters given.

c. both i and j are incremented twice rather than once because of the way the macro works.

d. I do not get the right result. for example if i call square (y+1) when y is 2, then the result should be square(3) = 9. But i get 5. This is because the function does not take into account order of operations. When we call square(y+1), we get y+1\*y+1 = y+ y + 1 = 2 + 2 + 1 = 5.

e. To fix the macro, we need to add parentheses around each x. So it becomes

#define square(x) (x)\*(x)

      4) a. argWord is at address bp+4 which has the value 0x1243

   argByte is at address bp+6 which has the value 0x20

   localWord is at address bp-2 which has the value 0x0321

   localByte is at address bp-4 which has the value 0x02B9

b. return address is at address bp+2 which is 0x02EA

c. 0xFFF6