```
/****************************************************************************
 * SysTick_Handler
 * Interrupt Service Routine for system tick counter
 * The name of this function cannot be changed - it establishes the linkage
 ****************************************************************************/
    .equ TIMER3_CMD,      0x40010C04
    .equ GPIO_PD_DOUTCLR, 0x40006080

    .syntax unified
    .text
    .thumb
    .thumb_func
    .align      4
    .globl      SysTick_Handler
    .type       SysTick_Handler, %function
SysTick_Handler:
    push   {r4-r11, lr}

    ldr r8,=TIMER3_CMD      // timer 3 command register
    ldr r0,=2               // stop timer command
    str r0,[r8]             // timer 3 now stopped

    ldr    r6,=SystemTick  // r6 is address of SystemTick
    ldr    r7,[r6]         // r7 is current tick
    add    r7,r7,#1        // increment tick
    str    r7,[r6]         // save new tick
    //keep this below
    ldr    r4,=CurrentTask // r4 is address of current task
    ldr    r5,[r4]         // r5 is current task
    str    sp,[r5,#0]      // stack pointer is first thing in TCB
    bl     scheduler
    str    r0,[r4]         // save new CurrentTask
    ldr    sp,[r0,#0]      // get sp from new current task

    ldr r1,[r0,#4]          // get timer3_on value
    str r1,[r8]             // start timer 3 if appropriate

    pop    {r4-r11, pc}

    .thumb_func
    .align      4
    .globl      SVC_Handler
    .type       SVC_Handler, %function
SVC_Handler:
    push   {r4-r11, lr}

    ldr r8,=TIMER3_CMD      // timer 3 command register
    ldr r0,=2               // stop timer command
    str r0,[r8]             // timer 3 now stopped

    ldr    r4,=CurrentTask  // r4 is address of current task
    ldr    r5,[r4]          // r5 is current task
    str    sp,[r5,#0]       // stack pointer is first thing in TCB
    bl     scheduler
    str    r0,[r4]          // save new CurrentTask
    ldr    sp,[r0,#0]       // get sp from new current task

```

```
    ldr r9,=GPIO_PD_DOUTCLR  // Port D “clear” register
    ldr r10,=0x1f           // clear bits 0-4
    str r10,[r9]            // and clear
    add r9,#-4              // point to Port D “set” register
    ldr r10,=0x20           // set bit 5 (context switch bit)
    str r10,[r9]            // and it is set
                            // save sp into current TCB, call the scheduler, etc.
    ldr r1,[r0,#8]          // get task_mask from current TCB
    str r1,[r9]             // set task_mask bits on port D
    add r9,#4               // point to port D “clear” register
    str r10,[r9]            // clear bit 5 (context switch bit)
                            // possibly start timer 3

    ldr r1,[r0,#4]          // get timer3_on value
    str r1,[r8]             // start timer 3 if appropriate

    pop    {r4-r11, pc}

    .thumb_func
    .align      4
    .globl      Yield
    .type       Yield, %function
Yield:
    svc #0 // raise SVC interrupt
    bx lr  // return from subroutine

    .end

```