Jonathan W. Valvano November 9, 2005, 1 to 1:50pm

(10) Question 1. The **CRC** is a 15-bit field used to detect transmission errors.
(10) Question 2. The ID is 11 bits and the Data fields 32 bits, giving a total of 79 bits.
100,000 bits/sec * 1 frame/79bits * 4 bytes/frame = 5063 bytes/sec
(15) Question 3. Consider a producer/consumer problem linked by a FIFO queue.
Part a) The best choice is to arm the producer (CAN input), because the Fifo is empty and the first operation to occur will be to receive input data and Put it into the Fifo. If you were to arm just the consumer, that interrupt would occur resulting in the input being armed and the output disarmed. Similarly, if you were to arm both, the output channel would trigger and disarm itself because the fifo is empty. If you armed neither, then no input/output could ever occur.
Part b) The producer thread is the input channel, which should be rearmed when there is more space in the fifo on the next output interrupt (when the SCI output device is idle).
Part c) The consumer thread is the output channel, which should be rearmed when more data is put into the fifo, which occurs on the next input interrupt (when new CAN input is received).
(25) Question 4. The **`writer`** runs first, so semaphore initialization occurs here. **`CanBeRead`** will be 1, if **`TheData`** has new data and the **`reader`** is allowed to read it. **`CanBeWritten`** will be 1, if **`TheData`** is empty and the **`writer`** is allowed to write into it.

```
void writer(void){
  OS_InitSemaphore(&CanBeRead,0);
  OS_InitSemaphore(&CanBeWritten,0);
  wInit();   // initialization
  while(1){
    OS_Wait(&CanBeWritten);
    TheData=wProcess(); // body
    OS_Signal(&CanBeRead);
  }
}
```

```
void reader(void){

  rInit();   // initialization

  while(1){
    OS_Wait(&CanBeRead);
    rProcess(TheData); // body
    OS_Signal(&CanBeWritten);
  }
}
```

(30) Question 5. The goal of this problem is to design a cooperative thread switcher.
(10) Part a) All that **`OS_Switch`** needs to do is issue a SWI.

```
void OS_Switch(void){
  asm swi
}
```

(20) Part b) The SWI interrupt handler suspends the current thread and runs the next thread.

```
interrupt 4 void SWIhandler(void){
asm ldx RunPt
asm sts 2,x
  RunPt = RunPt->Next;
asm ldx RunPt
asm lds 2,x
}
```

(10) Question 6. We need to calculate the slew rate of the input $\mathbf{dV/dt = 2\pi f B \cos(2\pi ft)}$, where the maximum slew rate is $\mathbf{2\pi fA}$. The voltage error is $d\mathbf{V} = \mathbf{dV/dt} * d\mathbf{t} = \mathbf{2\pi f B}\, d\mathbf{t}$.