Exam TI2720-C/TI2725-C Embedded Software

Thursday July 3 2014 (14.00 - 17.00)

Koen Langendoen

TU Delft

In order to avoid misunderstanding on the syntactical correctness of code fragments in this examination, we will always assume that we are dealing with pseudo-code, although we might have syntactically correct code in some cases. We assume that the required variables, semaphores, tasks, timers, etc. are always declared and initialized correctly.

Further, we assume the following abbreviations to be known:

- RR = Round Robin,
- RRI = Round Robin with Interrupts,
- FQS = Function Queue Scheduling,
- RTOS = Real-Time Operating System,
- IR = interrupt and
- ISR = interrupt service routing.

In this exam, we use the following definitions, unless stated otherwise:

```
void delay(int ms) {
  !! do some CPU computation for ms milliseconds
}

void putchar(char c) {
  while (!! UART tx buffer not empty);
  !! send c to UART tx buffer
}

void puts(char *s) {
  !! print string s using putchar
}
```

To pass this written exam, you need to correctly answer at least 20 questions. The relationship between the number of correctly answered questions and your mark is given below.

#correct	≤ 8	9	10	11	12	13	14	15	16	17	18	19
Mark	1	1.5	2	2.5	3	3	3.5	4	4.5	5	5	5.5

#correct	19	20	21	22	23	24	25	26	27	28	29	30
Mark	5.5	6	6.5	7	7	7.5	8	8.5	9	9	9.5	10

Question 1
Embedded programming is in essence more difficult than “classical” programming because of a. the lack of recursion b. the event-based programming model c. the thread-based programming model d. none of the above
Question 2
Which of the following statements is correct? Using interrupts improves … a. task response time b. memory response time c. system response time d. processor response time
Question 3
Which of the following statements is correct? Interrupts can be disabled in order to a. protect a critical section b. enable context switches c. protect other interrupts d. disable a critical section
Question 4
How can a low priority task prevent itself from being interrupted by a high priority task? a. By using critical sections b. By avoiding critical sections c. By enabling interrupts d. By disabling the interrupts
Question 5
Which of the following statements is correct? An interrupt service routine is supposed to a. clear the context upon entrance b. restore the context and return c. save the context to disk upon entrance d. restore the lowest-priority interrupt and return
Question 6
The primary shortcoming for RRI architecture is: a. all the tasks have the same priority b. it is more complex than RR c. critical sections must be used d. the memory footprint increases
Question 7
Which of the following statements is correct? An interrupt service routine a. must not reference global variables (shared data problem) b. must be written in assembler (to avoid compiler optimizaitions) c. must not call routines that perform I/O d. must return within 100 ms to meet user perception
Question 8
Which architecture has the best response time for task code? a. RRI b. FQS c. RR d. RTOS
Question 9
Which of the following statements is correct? The shared data problem can be solved through a. using semaphores b. enabling interrupts c. using the extern keyword d. using the static keyword
Question 10
Which of the following statements holds for the FQS architecture? a. the code length of the lower priority tasks does not affect the execution time of other tasks b. the code length of the higher priority tasks does not affect the execution time of other tasks c. to reduce the response time of higher priority tasks, the code for lower priority tasks should be small d. to reduce the response time of higher priority tasks, the code for lower priority tasks should be large

Question 11
Which factor determines the best response time of a system reacting on an interrupt? a. The shortest period of time during which the interrupt is enabled. b. The shortest period of time during which the interrupt is disabled. c. The shortest period of time during which the context of the current task is restored. d. The shortest period of time during which another high-priority task is executing.
Question 12
Who handles the signaling between the interrupt routines and the task code in RTOS? a. the user b. the ISR c. the RTOS d. it is done automatically
Question 13
Which of the following statements is correct? a. An RR architecture provides interrupts and interrupt priorities. b. We cannot have a shared data problem in an RRI architecture. c. An FQS architecture supports task priorities. d. An RTOS provides interrupt preemption whereas an FQS architecture does not.
Question 14
In an RTOS, tasks can be in state BLOCKED, READY or RUNNING. Which of the following statements is true? a. A task starts in the state RUNNING b. A task ends in the state READY c. A task can transition directly from BLOCKED to RUNNING d. A task can transition directly from READY to BLOCKED
Question 15
For which kind of software architecture for embedded systems is the worst response time for task code the execution time for the longest task code plus the execution time for interrupt routines? a. Round-robin without interrupts b. Round-robin with interrupts c. Function-queue-scheduling d. Real-time operating system
Question 16
Which of the following statements is **false**: a. interrupts can be used for signaling between tasks b. interrupts can be disabled in order to protect a critical section c. semaphores can be used for solving shared data problems d. semaphores can be used for signaling between tasks
Question 17
Which of the following statements is correct? A reentrant function a. may never call other reentrant functions b. may only use hardware in an atomic way c. may not be shared by different tasks d. cannot be stored on the stack
Question 18
Tasks in an RTOS are often structured as: a. state machines with states stored in global variables and messages in their queues don't have any role b. state machines with states stored in private variables and messages in their queues acting as events c. RR architectures with messages triggering state changes d. RRI architectures with interrupts advancing the state machine
Question 19
Which of the following principles of RTOS-based design is **false**? a. turning time-slicing off solves the data sharing problem b. short interrupt routines are needed for a responsive system c. more tasks help sometimes to encapsulate data more efficiently d. it is recommended to use just the minimum necessary functionality from an RTOS
Question 20
Why should time-slicing be avoided in an RTOS? a. It creates too many context switches b. It raises the complexity of the scheduler c. It makes the response time of tasks less predictable d. Equally important tasks require equal processor attention

Question 21

Given the following RTOS (pseudo) code:

```
int temp1, temp2;

void isr_buttons(void) // arrive here if a button is pressed
{
   temp1 = X32_PERIPHERALS[PERIPHERAL_TEMP1];
   temp2 = X32_PERIPHERALS[PERIPHERAL_TEMP2];
   …
}

main() {
   …
   while (! program_done) {
      X32_display = ((temp1 & 0xff) << 8) | (temp2 & 0xff);
      if (temp1 != temp2) {
         !! shutdown plant
      }
   }
}
```

Which of the following statements is correct?

a. There is a potential deadlock problem in this code
b. There is a potential shared data problem in this code
c. There is a potential problem with respect to reentrancy in this code
d. There is no potential problem in this code

Question 22

Given the following (pseudo) code:

```
int done;

void isr_button (void) // arrive here when button pressed/released
{
   if (!! button pressed)
      done = 1;
}

void main (void)
{
    done = 0;
    while  (!done) {
       printf (“Hello World\r\n”);
       delay(10000);
    }
    printf (“done\r\n”);
}
```

The program is uploaded onto the embedded system in the conventional way. Which of the following statements is correct?

a. Sometimes, the program does not end if the button is not debounced.
b. The program does not always show the same output if it is restarted without uploading it.
c. Sometimes, the program does not end.
d. The program ends always as soon as the button is pressed.

Questions 23 – 25

Given is the following (pseudo) code, which reads the current values of 4 different buttons and acts accordingly. The 4 buttons are all mapped to bits 0..3 of the button register. The buttons are already debounced.

```
void f1(void) { delay(1000); }
void f2(void) { delay(2000); }
void f3(void) { delay(3000); }
void f4(void) { delay(4000); }

void main (void) {
   while (1) {
      if (buttons & 0x01) f1();
      if (buttons & 0x02 ) f2();
      if (buttons & 0x04 ) f3();
      if (buttons & 0x08 ) f4();
      delay(1000);
  }
}
```

Question 23

None of the buttons has been pressed. Which of the following statements is correct? The longest time that button #3 must be pressed in order to activate f3() once is

a) 1 second
b) 2 seconds
c) 3 seconds
d) 4 seconds

Question 24

None of the buttons has been pressed. 500 ms after button #3 has been pressed, button #1 and button #4 are pressed (in this sequence) and held down. Which of the following statements is correct? The sequence of executed functions is

a) f3(), f1(), f4()
b) f3(), f4(), f(1)
c) f3(), and the remaining sequence cannot be determined
d) arbitrary

Question 25

At a given moment in time, the system is in an arbitrary state. Which of the following statements is correct?

a) Button #1 must be pressed at most 9 s in order to activate f1().
b) Button #1 must be pressed at least 10 s in order to activate f1().
c) Button #1 must be pressed at most 10 s in order to activate f1().
d) Button #1 must be pressed at most 11 s in order to activate f1().

Question 26

Given the following RTOS (pseudo) code.

```
…

void isr_button(void)  { // arrive here when button pressed
   delay(40); // wait for debounce
   // do something -- takes 10 ms
}

void task(void)  {
   while (1) {
      delay(100);    // synchronize
      f();               // perform function that takes 100 ms
   }
}
```

Pressing the button during synchronization will ...

a. extend the duration of synchronization by 10 ms
b. extend the duration of the synchronization by 20 ms
c. extend the duration of the synchronization by 50 ms
d. not have any delaying effect

Questions 27 – 28

Given is the following RTOS (pseudo) code with decreasing priorities T1 > T2 > T3:

```
void T1(void) {
   while (1) {
      OS_Pend(s_status);
      !! control chemical process according to status
      OS_Post(s_status);
   }
}

void T2(void) {
   while (1) {
      OS_Pend(s_temp);
      !! read and update plant temperatures
      OS_Post(s_temp);
   }
}

void T3(void) {
   while (1) {
      OS_Pend(s_status);
      !! calculate and update the status
      OS_Post(s_status);
   }
}
```

Question 27

Which of the following statements is correct? There is …

a) a potential deadlock between T1 and T2
b) a potential deadlock between T1 and T3
c) a potential deadlock between T2 and T3
d) no potential deadlock

Question 28

Imagine that the priorities are exchanged, so that T1 < T2 < T3. Which of the following statements is correct? There is …

a) a potential deadlock between T1 and T2
b) a potential deadlock between T1 and T3
c) a potential deadlock between T2 and T3
d) no potential deadlock

Question 29

Which of the following statements is correct regarding calling RTOS functions?

a. a task may call the OS_pend() routine, but not the OS_post() routine
b. an ISR may call the OS_pend() routine, but not the OS_post() routine
c. an ISR may call the OS_post() routine, but not the OS_pend() routine
d. an ISR may neither call the OS_post(), nor the OS_pend() routine

Question 30

Which of the following statements is correct about ISRs?

a. The lowest-priority ISR runs **before** the highest-priority task
b. The lowest-priority ISR runs **after** the highest-priority task
c. ISRs are easier to debug than tasks
d. ISRs exhibit less bugs/nr lines of code than normal routines