Real-Time Systems — EDA222/DIT161

Final exam, March 12, 2008 at 14:00 – 18:00 in the V building

---

Examiner:

Associate professor Jan Jonsson
Phone: 031–772 5220

Aids permitted during the exam:

Excerpt from *Ada 95 Reference Manual*
R. Riehle, *Ada Distilled*
Quick Reference, *Ada vs. Java*
Chalmers-approved calculator

Content:

The written exam consists of 5 pages (including cover), containing 7 problems worth a total of 60 points.

Grading policy:

24–35 ⇒ grade 3
36–47 ⇒ grade 4
48–60 ⇒ grade 5

Solution:

Posted on the course home page on Thursday, March 13, 2008 at 09:00.

Results:

Posted on the course home page on Monday, March 31, 2008 at 09:00.

Inspection:

Room 4128, Rännvägen 6 B, on Monday, March 31, 2008 at 13:00–15:00. Inspection at another occasion could be arranged by contacting the course examiner.

Language:

Your solutions should be written in English.

IMPORTANT ISSUES

1. Use separate sheets for each answered problem, and mark each sheet with the problem number.
2. Justify all answers. Lack of justification can lead to loss of credit even if the answer might be correct.
3. Explain all calculations thoroughly. If justification and method is correct then simple calculation mistakes do not necessarily lead to loss of credit.
4. If some assumptions in a problem are missing or you consider that the made assumptions are unclear, then please state explicitly which assumptions you make in order to find a solution.
5. Write clearly! If I cannot read your solution, I will assume that it is wrong.

Good Luck!

PROBLEM 1

State whether the following propositions are TRUE or FALSE. Each correct statement will give 0.5 points; each erroneous statement will give -0.5 points; an omitted statement gives 0 points. Although a motivation for a correct answer is not required, a convincing one gives another 0.5 points, while an erroneous/weak one gives another -0.5 points. **Quality guarantee**: The total result for this problem cannot be less than 0 points. (6 points)

a) A protected entry i Ada 95 must always be accompanied by a *barrier* (boolean expression).

b) The fundamental task states in a real-time kernel are *Running*, *Waiting* and *Ready*.

c) Deadlock can never be completely avoided in systems where tasks require access to more than one exclusive resource at a time.

d) If a given task set is known to be not schedulable, a *sufficient* feasibility test will always report the answer "no" when applied to that task set.

e) The purpose of *memory mapping* in a real-time kernel is to offer flexibility concerning on what primary memory addresses a task's program code and data should be located.

f) By *priority inversion*, we mean a situation where the static priority assigned to a task is inversely proportional to the period of the task.

PROBLEM 2

Most scheduling analysis techniques assume the worst-case execution time (WCET) to model the computational demand of a real-time task. One of the earliest methods for WCET analysis was presented by Shaw in the end of the 1980s.

a) Assume that a procedure `Main` (see below) is used as part of a real-time program and that the procedure, when called, is allowed to take at most 100 μs to execute.

Derive WCET for procedure `Main` (see below) by using Shaw's method and check whether the procedure's deadline with be met or not.

Assume that each assignment statement, return statement and comparison operation costs 1 μs to execute. A function call costs 1 μs plus WCET for the function in question. Each addition and subtraction operation costs 2 μs. Each multiplication operation costs 5 μs. All other language constructs can be assumed to take 0 μs to execute. (6 points)

```
procedure Main is
 A : Natural := 3;
 F : Natural := 0;

 function Calculate (Z : in Natural) return Natural is
   R : Natural := 0;
 begin
   if Z == 0 then
     R := 1;
   else if Z == 1 then
     R := 1;
   else
     R := Calculate(Z-1)*Z;
   end;
```

```
   return R;
  end Calculate;

 begin
   F := Calculate(A);
 end;
```

b) Now assume that the data types for variables `A`, `F`, `Z`, `R`, and `F`, and the return type of function `Calculate`, are changed to `Float`. Since floating point operations require more advanced hardware and algorithms to implement than integer operations, the cost for executing the arithmetic operations will now increase.

If each floating point addition and subtraction operation costs 5 μs, decide the largest allowed cost for the floating point multiplication operation while not exceeding the deadline (100 μs) of the procedure. (4 points)

PROBLEM 3

Consider the following type declaration:

```
type BIT_TYPE is range 0..1; -- Named type and min..max values
for BIT_TYPE'SIZE use 1;     -- Object type  needs a bit
type BITFIELD8 is
record
    b0:  BIT_TYPE;
    b1:  BIT_TYPE;
    b2:  BIT_TYPE;
    b3:  BIT_TYPE;
    b4:  BIT_TYPE;
    b5:  BIT_TYPE;
    b6:  BIT_TYPE;
    b7:  BIT_TYPE;
end record;
```

a) Show a representation clause using BIG endian bit ordering. (2 points)

Now, also assume the following type and variable declarations:

```
type UINT8 is integer range 0..255;
for UINT8'size use 8;

a : BITFIELD8;
b : UINT8;
```

b) Show how you would do assignments from one variable to the other (`a` to `b`, `b` to `a`), by bypassing the compiler's type checking. (2 points)

c) Explain the purpose of the ADA pragma `Volatile`. Give an example of a proper use of the pragma in a program. (3 points)

d) Consider the following C-prototype declaration:

```
void can_init( char * port );
```

Show how to make this function usable from an ADA95 program. (1 points)

PROBLEM 4

The following questions are related to communication networks used for industrial real-time applications.

a) Describe how different types of message transmissions can co-exist on a FlexRay network. (2 points)

b) State whether communication medium access in the Controller Area Network (CAN) is token-based or contention-based. Motivate your answer. (1 points)

c) Describe how response-time-based schedulability analysis can be adapted to message scheduling in CAN. (3 points)

PROBLEM 5

Consider a real-time system with two periodic tasks and a run-time system that employs static scheduling using a time table. The table below shows C_i (WCET), D_i (deadline) and T_i (period) for the two tasks. Both tasks arrive the first time at time 0.

	C_i	D_i	T_i
τ_1	2	5	5
τ_2	4	7	7

a) Describe how the run-time system works for time-table based scheduling. (3 points)

b) Contruct a time table for the execution of the two tasks τ_1 and τ_2. Assume that the tasks are allowed to preeempt each other. Your solution should clearly indicate the start and stop times for each task (or task segment, if a task is preempted). In addition, the total length of your time table (in time units) should be given. (6 points)

c) Evaluate the schedule described by your time table in sub-problem b). That is, does it constitute the best possible schedule, or does there exist a superior one? (1 points)

PROBLEM 6

Consider a real-time system with two periodic tasks and a run-time system that employs preemptive rate-monotonic (RM) scheduling. For each task τ_i it applies that its deadline D_i is equal to the period T_i. Both tasks arrive the first time at time 0. It is known that the tasks' utilizations $U_i = C_i/T_i$ are $U_1 = 0.75$ och $U_2 = 0.25$, respectively. However, the individual execution times C_i for the two tasks are not known.

a) Explain the principle behind the RM priority assignment approach. (2 points)

b) If the tasks' periods T_i are not known either, is it possible to decide if the tasks are schedulable with RM? Motivate your answer. (3 points)

c) If we know that $T_2 = 2T_1$, is it then possible to decide whether the tasks are schedulable or not with RM? Motivate your answer. (3 points)

PROBLEM 7

Consider a real-time system with four periodic tasks and a run-time system that uses preemptive earliest-deadline-first (EDF) scheduling. The table below shows C_i (WCET), D_i (deadline) and T_i (period) for the four tasks. All tasks arrive the first time at time 0.

	C_i	D_i	T_i
τ_1	5	15	20
τ_2	5	55	60
τ_3	10	25	30
τ_4	5	20	20

a) Show that the tasks are schedulable under EDF scheduling using processor-demand analysis. (4 points)

b) Do a robustness analysis of the given system by assuming that a task can increase its execution time to $C_i \cdot \alpha_i$. Derive, for each task τ_i, the maximum scaling factor α_i for which the system is still schedulable when that, and only that, task increases its execution time. (4 points)

c) Perform the same robustness analysis as in subproblem b), but now assume that the tasks are executed according to deadline-monotonic (DM) scheduling. (4 points)
