

# REAL-TIME SYSTEMS — EDA222/DIT160

Final exam, December 20, 2006 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, December 21, 2006 at 09:00.

**Results:**

Posted on the course home page on Monday, January 8, 2007 at 09:00.

**Inspection:**

Room 4128, Rännvägen 6 B, on Monday, January 8, 2007 at 13:00–15:00. Inspection at another occasion could be arranged by contacting the course examiner.

**Language:**

Your solutions should be written in Swedish or 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) Compilers for Ada95 always provide support for dynamic task priorities.
  - b) By the term *critical instant* in feasibility analysis we mean a point in time where all tasks deadlines coincide.
  - c) A `delay` statement cannot be used in the body of a protected procedure i Ada95.
  - d) An important assumption in worst-case execution-time (WCET) analysis is that interferences from higher-priority tasks are accounted for.
  - e) If a given task set is known to be schedulable, a *necessary* feasibility test will always report the answer “yes” when applied to that task set.
  - f) The Controller Area Network (CAN) protocol uses time slots to arbitrate between different senders.
- 

## PROBLEM 2

The following questions are related to estimation of worst-case execution times (WCET) of tasks.

- a) Explain why is it preferred that WCET estimates for tasks in a real-time system are *pessimistic* as well as *tight*. (2 points)
  - b) Explain what the term *false path* refers to in the context of WCET analysis. Also, describe why it is important to identify such paths. (3 points)
  - c) Describe the basic ideas behind the method used for performing *timing analysis* of programs when a processor with pipelines and cache memories is used. (3 points)
- 

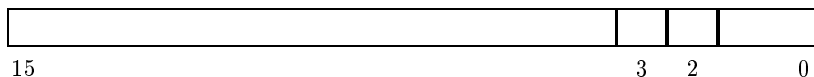
## PROBLEM 3

For real-time systems it is recommended to use a programming language with support for *low-level programming*.

- a) Describe the four general types of support that a language must provide to enable low-level programming. (4 points)
  - b) In Ada95, interrupt handlers are implemented using protected objects. Explain why it is important to assign *ceiling priorities* to such protected objects. (4 points)
-

#### PROBLEM 4

The interface to an external unit consists of four 12-bit registers, connected via a multiplexer to a 16-bit port located at memory address FFFFF080. The structure of the port is given below. The multiplexer is shown on the last page of this thesis.



field	no of bits	position	function	R/W
<i>mxValue</i>	12	15..4	contains the value of the selected register if <i>mxValid</i> = 1	R
<i>mxValid</i>	1	2	indicates whether <i>mxValue</i> contains a valid value or not	R
<i>mxSelect</i>	2	1..0	selects one of the four registers via the multiplexer	W

When a new value is written to *mxSelect*, the multiplexer is set-up to choose one of the four registers. The operation is fast, but not momentary. Before the value of the register (*mxValue*) can be read, the software must check that the multiplexer set-up has been completed (*mxValid*). In order to guarantee that a read operation of an arbitrary register is atomic, this operation must be encapsulated in a protected object.

The registers are updated from software via calls to a function `refresh` available in a C library:

```
void refresh(int register);
/* Update contents in "register" (0-3) */
```

Write a protected object `ProtectedRead` that exports the following function:

```
function ReadRegister(register_number : in integer) return integer;
```

The function should return the updated value of the register with the given number. (8 points)

#### PROBLEM 5

Write an Ada package for administration of 10 identical resources. Identical resources means that you only have to keep track of the number of resources available. The user should be able to request 1, 2, or 3 resources. The following package specification is given.

```
procedure request(needed : in integer);
-- Request needed number of resources.
-- Blocking operation. Returns when the resources are available

procedure trytorequest(needed : in integer; OK : out boolean);
-- Request needed number of resources.
-- Operation do not block. Returns immediately.
-- OK is is true the resources are available, otherwise false.

procedure release(given : in integer);
-- Returns given number of resources.
-- It is assumed that the resources have been previously requested.
```

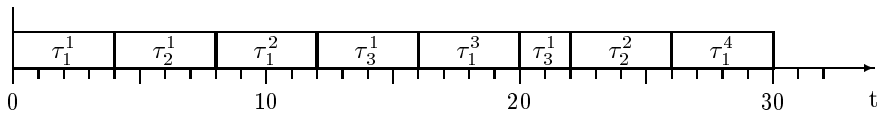
The package must be able to handle concurrent calls from different tasks in a correct way. (8 points)

### PROBLEM 6

Consider a real-time system with periodic tasks and a run-time system that employs preemptive earliest-deadline-first (EDF) scheduling.

- a) Describe Liu & Layland's exact feasibility test for EDF scheduling. (2 points)
- b) Describe under what assumptions the EDF feasibility test in sub-problem a) apply. (2 points)
- c) Describe a suitable EDF feasibility test for task sets where the assumptions in sub-problem b) do not apply. (3 points)
- d) Decide, based on the table and timing diagram below, whether the schedule was generated by an EDF scheduler or not. Motivate your answer. In the timing diagram  $\tau_i^k$  is used for representing instance  $k$  of the periodic task  $\tau_i$ . (3 points)

	$C_i$	$D_i$	$T_i$
$\tau_1$	4	6	8
$\tau_2$	4	12	16
$\tau_3$	6	24	32



### PROBLEM 7

Consider a real-time system with five tasks, whose timing properties are listed in the table below. Tasks  $\tau_3$  and  $\tau_4$  cooperate in the sense that  $\tau_4$  uses the result produced by  $\tau_3$ , and they must both complete within a common end-to-end deadline of  $D_{global} = 75$ . The global deadline should be divided between these two tasks, and  $X$  represents the part of the deadline that is assigned to  $\tau_3$ . To guarantee that valid data is available at the beginning of its execution, the offset for  $\tau_4$  is set to  $X$ . The same offset is also used for task  $\tau_5$ . All other tasks are supposed to be independent, with an offset of 0.

	$C_i$	$O_i$	$D_i$	$T_i$
$\tau_1$	5	0	6	15
$\tau_2$	6	0	11	25
$\tau_3$	9	0	$X$	75
$\tau_4$	15	$X$	$75 - X$	75
$\tau_5$	15	$X$	25	25

The tasks will assigned to two separate processors that are connected through a shared-bus network. Processor 1 executes tasks  $\tau_1$ ,  $\tau_2$  and  $\tau_3$ , and employs preemptive earliest-deadline-first (EDF) scheduling (that is, dynamic priorities.) Processor 2 executes tasks  $\tau_4$  and  $\tau_5$ , and employs preemptive deadline-monotonic (DM) scheduling (that is, static priorities.) The overhead for sending data between the processors over the network is assumed to be negligible.

Derive the range of allowed (maximum and minimum) values of  $X$  for which all tasks will meet their deadlines. (12 points)