Real-Time Systems — EDA223/DIT162

Example Written Exam – Basic Part

Examiner:
Professor Jan Jonsson, Department of Computer Science and Engineering

Responsible teacher:
Jan Jonsson, phone: 031–772 5220
Visits the exam room on two occasions.

Aids permitted during the exam:
Chalmers-approved calculator
NOTE: Electronic dictionaries may not be used.

Content:
The **Basic part** of the exam consists of 3 problems worth a total of 30 points.
The **Advanced part** of the exam consists of 4 problems worth a total of 30 points.

Grading policy:
Obtaining less than 24 points in the Basic part leads to a **Fail** grade.
If you obtain at least 24 points in the Basic part, then your grade is determined by the total number of points in the Basic and Advanced parts combined as follows:

24–35 points ⇒ **grade 3**	24–43 points ⇒ **grade G** (GU)
36–47 points ⇒ **grade 4**	
48–60 points ⇒ **grade 5**	44–60 points ⇒ **grade VG** (GU)

Results:
When the grading is completed overall result statistics, and a time and location for inspection, will be announced on the course home page. Individual results will be available in Ladok.

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 is 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 we cannot read your solution, we will assume that it is wrong.

Good Luck!

BASIC PART

PROBLEM 1

The following sub-problem concerns processor-utilization analysis.

Consider a real-time system with three periodic tasks and a run-time system that employs preemptive single-processor scheduling using the rate-monotonic (RM) priority-assignment policy. The table below shows C_i (WCET), D_i (deadline) and T_i (period) for each task τ_i. All tasks arrive at $t = 0$.

	C_i	D_i	T_i
τ_1	3	20	20
τ_2	10	30	30
τ_3	25	60	60

a) Perform Liu & Layland's utilization-based analysis for the given task set. (2 points)

The following sub-problem concerns response-time analysis.

Consider a real-time system with three periodic tasks and a run-time system that employs preemptive single-processor scheduling using the deadline-monotonic (DM) priority-assignment policy. The table below shows C_i (WCET), D_i (deadline) and T_i (period) for each task τ_i. All tasks arrive at $t = 0$.

	C_i	D_i	T_i
τ_1	3	5	20
τ_2	10	25	30
τ_3	25	40	60

b) Perform response-time analysis for the given task set. The <u>final</u> (converged) response time should be calculated for each task in the task set, regardless of whether the response-time analysis for that task fails or not. (4 points)

PROBLEM 1 (cont'd)

The following sub-problems concern processor-demand analysis.

Consider a real-time system with three periodic tasks and a run-time system that employs preemptive single-processor scheduling using the earliest-deadline-first (EDF) priority-assignment policy. The table below shows C_i (WCET), D_i (deadline) and T_i (period) for each task τ_i. All tasks arrive at $t = 0$.

	C_i	D_i	T_i
τ_1	3	5	20
τ_2	10	25	30
τ_3	25	40	60

c) Show that the largest interval to examine is $L_{\max} = 60$ for the given task set. The solution should include calculations of the hyper-period upper bound as well as the upper bound proposed by Baruah, Rosier and Howell. (2 points)

d) Calculate the complete set of control points within $L_{\max} = 60$ for the given task set. (2 points)

e) Perform processor-demand analysis for the given task set, and the complete set of control points calculated in sub-problem d). The analysis should be performed for every control point in the set, regardless of whether the analysis in another control point fails or not. (4 points)

PROBLEM 2

The following sub-problems concern hyper-period analysis.

Consider a real-time system with three periodic tasks and a run-time system that employs preemptive single-processor scheduling. The table below shows O_i (offset), C_i (WCET), D_i (deadline) and T_i (period) for each task τ_i.

	O_i	C_i	D_i	T_i
τ_1	0	3	5	20
τ_2	0	10	25	30
τ_3	3	25	40	60

a) Simulate the execution of the task set using the deadline-monotonic (DM) priority-assignment policy, and construct a timing diagram that spans the entire hyper period. The arrival and deadline of each task instance should be clearly indicated in the diagram in the form of an up-arrow (arrival) and down-arrow (deadline), respectively. All task executions must be completed, regardless of whether a task instance misses its deadline or not. (5 points)

b) Derive a time table, corresponding to the timing diagram constructed in sub-problem a), that clearly states the start and stop times for each task instance (or task segment, if a task instance is preempted). (1 point)

PROBLEM 3

The following sub-problems concern analysis of worst-case execution time (WCET).

Consider the following C program code for function `calculate`:

```
int calculate(int y) {
   int i;
   int r;
   i = 0;
   r = y;
   while (i < 3) {
      r = r * y;
      i = i + 1;
   }
   r = r - 1;
   return r;
}
```

Assume the following costs for the language statements:

- Each declaration and assignment statement costs 1 μs to execute.
- Each evaluation of the logical condition in an `if`- or `while`-statement costs 2 μs.
- Each add and subtract operation costs 3 μs.
- Each multiply operation costs X μs.
- Each return statement costs 2 μs.
- All other language constructs can be assumed to take 0 μs to execute.

a) Derive the WCET of function `calculate` by using the analysis method proposed by Shaw. The WCET should be expressed in the form $\mathbf{k_1 \cdot y + k_2 \cdot X + k_3}$, where k_1, k_2 and k_3 are integer constants, y is the function parameter, and X is the cost for the multiply operation. (2 points)

b) Explain why is it preferred that WCET estimates for tasks in a real-time system are *pessimistic* as well as *tight*. (2 points)

The following sub-problems concern scheduling concepts.

c) Describe the meaning of *priority inversion*. (2 points)

d) State the major difference between the *priority inheritance protocol* (PIP) and the *deadline inheritance protocol* (DIP). (1 point)

The following sub-problems concern real-time programming.

e) Describe the meaning of *systematic time skew* in the context of implementation of periodic task executions. (2 points)

f) State the prominent feature of the `AFTER()` operation in TinyTimber that facilitates implementation of periodic tasks without systematic time skew. (1 point)
