CSIE 5452, Spring 2024: Homework 1

Due March 19 (Tuesday) at Noon

When you submit your homework, select the corresponding page(s) of each question. Points will be deducted if no appropriate intermediate step is provided.

1 Timing Analysis of the CAN Protocol: Part I (12pts)

Given a set of periodic messages μ_0, μ_1, μ_2 with their priorities, transmission times, and periods as follows:

Message	Priority (P_i)	Transmission Time (C_i) (msec)	Period (T_i) (msec)
μ_0	0	10	50
μ_1	1	30	200
μ_2	2	20	100

The worst-case response time R_i of μ_i can be computed as

$$R_i = Q_i + C_i, \tag{1}$$

and

$$Q_i = B_i + \sum_{\forall j, P_j < P_i} \left\lceil \frac{Q_i + \tau}{T_j} \right\rceil C_j, \tag{2}$$

where $\tau = 0.1$ in this question. You can consider using the following tables to help you.

1. (4pts) What is the worst-case response time of μ_0?

Iteration	LHS (Q_0)	B_0	RHS	Stop?
1	30	30	30	Yes

2. (4pts) What is the worst-case response time of μ_1?

Iteration	LHS (Q_1)	B_1	j	$Q_1 + \tau$	T_j	$\left\lceil \frac{Q_1+\tau}{T_j} \right\rceil$	C_j	RHS	Stop?
1	30	30	0	30.1	50	1	10	40	No
2	40	30	0	40.1	50	1	10	40	Yes

3. (4pts) What is the worst-case response time of μ_2?

Iteration	LHS (Q_2)	B_2	j	$Q_2 + \tau$	T_j	$\left\lceil \frac{Q_2+\tau}{T_j} \right\rceil$	C_j	RHS	Stop?
1	20	20	0	20.1	50	1	10	60	No
			1	20.1	200	1	30		
2	60	20	0	60.1	50	2	10	70	No
			1	60.1	200	1	30		
3	70	20	0	70.1	50	2	10	70	Yes
			1	70.1	200	1	30		

1.

1.

$Q_0 = B_0 = 30$

$R_0 = 30 + 10 = 40$

$30 = 30 + 0$

$R_1 = 30 + 10 = 40$ #

2.

$Q_1 = B_1 = 30$

$30 \neq 30 + \left(\left\lceil \frac{30 + 0.1}{50} \right\rceil\right) \times 10$

$40 = 30 + \left(\left\lceil \frac{40 + 0.1}{50} \right\rceil\right) \times 10$

$R_2 = 40 + 30 = 70$ #

3.

$Q_2 = B_2 = 20$

$20 \neq 20 + \left(\left\lceil \frac{20 + 0.1}{50} \right\rceil\right) \times 10 + \left(\left\lceil \frac{20 + 0.1}{200} \right\rceil\right) \times 30$

$60 \neq 20 + \left(\left\lceil \frac{60 + 0.1}{50} \right\rceil\right) \times 10 + \left(\left\lceil \frac{60 + 0.1}{200} \right\rceil\right) \times 30$

$70 = 20 + \left(\left\lceil \frac{70 + 0.1}{50} \right\rceil\right) \times 10 + \left(\left\lceil \frac{70 + 0.1}{200} \right\rceil\right) \times 30$

$R_3 = 70 + 20 = 90$ #

2 Timing Analysis of the CAN Protocol: Part II (36pts)

Please download the benchmark "input.dat" from NTU COOL. In the benchmark, the first number is n, the number of messages. The second number is τ. Each of the following lines contains the priority (P_i), the transmission time (C_i), and the period (T_i) of each message. You are required to do two things in your submission:

1. You should print out n numbers (one number per line) representing the worst-case response time (R_i) of those messages. Note that you need to follow the message ordering in the benchmark, *e.g.*, the first number in the list is the worst-case response time of the first message in the benchmark.
2. You should also print out your source codes. (For your information, my implementation is less than 100 lines.) We may ask you to provide your source codes which must be the same as those on your printout. If the worst-case response times above are correct but the source codes are clearly wrong implementation, it is regarded as academic dishonesty.

It is highly recommended to write your codes well (*e.g.*, capable of dynamically allocating memory based on n) so that you can reuse them in Homework 2. Ideally, you can test your implementation with the small benchmark in Question 1 and verify its solution by your implementation. Just do not make the same mistake in Questions 1 and 2.

3 Timing Analysis of Preemptive Fixed-Priority Scheduling (16pts)

The CAN protocol is based on non-preemptive fixed-priority scheduling. For tasks on an Electronic Control Unit (ECU), they are usually scheduled by preemptive fixed-priority scheduling. The worst-case response time R_i of a task τ_i can be computed as

$$R_i = C_i + \sum_{\forall j, P_j < P_i} \left\lceil \frac{R_i}{T_j} \right\rceil C_j, \tag{3}$$

where P_i, C_i, and T_i are the priority, the computation (execution) time, and the period of τ_i, respectively. Given a set of periodic tasks τ_0, τ_1, τ_2 with their priorities, computation times, and periods as follows:

Task	Priority (P_i)	Computation Time (C_i) (msec)	Period (T_i) (msec)
τ_0	0	10	50
τ_1	1	30	200
τ_2	2	20	100

1. (4pts) What is the worst-case response time of τ_0?

Iteration	LHS (R_0)	C_0	RHS	Stop?
1	10	10	10	Yes

2. (4pts) What is the worst-case response time of τ_1?

Iteration	LHS (R_1)	C_1	j	R_1	T_j	$\left\lceil \frac{R_1}{T_j} \right\rceil$	C_j	RHS	Stop?
1	30	30	0	30	50	1	10	40	No
2	40	30	0	40	50	1	10	40	Yes

2.

1.

```
lambo@Ubuntu-Lambo:~/Desktop/LEC/HW1$ ./HW1
Index of Msg is 0 and its response time is 1.44
Index of Msg is 1 and its response time is 2.04
Index of Msg is 2 and its response time is 2.56
Index of Msg is 3 and its response time is 3.16
Index of Msg is 4 and its response time is 3.68
Index of Msg is 5 and its response time is 4.28
Index of Msg is 6 and its response time is 5.2
Index of Msg is 7 and its response time is 8.4
Index of Msg is 8 and its response time is 9
Index of Msg is 9 and its response time is 9.68
Index of Msg is 10 and its response time is 10.2
Index of Msg is 11 and its response time is 19.36
Index of Msg is 12 and its response time is 19.8
Index of Msg is 13 and its response time is 20.32
Index of Msg is 14 and its response time is 29.4
Index of Msg is 15 and its response time is 29.76
Index of Msg is 16 and its response time is 30.28
```

2.

```
#include <iostream>
#include <fstream>
#include <vector>
#include <string>
#include <sstream>
#include <cmath>
using namespace std;

struct Msg{
    int    priority;
    double transmissionTime;
    double period;
};

double computeResponseTime(int id, vector<Msg>& vecMsg, double tau);

int main(void)
{
    ifstream file("input.dat");
    if (!file) {
        cerr << "Unable to open file\n";
        return 1;
    }

    vector<Msg> vecMsg;
    int num;
    double tau;

    file >> num;
    file >> tau;

    string line;
    Msg msg;
    while(file>> msg.priority >> msg.transmissionTime >> msg.period){
        vecMsg.push_back(msg);
        if (vecMsg.size() >= num) break;
    }
    file.close();

    // compute response time
    int id = 0;
    vector<double> vecResponseTime;
    while(id < num){
```

```
    while(id < num){
        double responseTime = computeResponseTime(id, vecMsg, tau);
        vecResponseTime.push_back(responseTime);
        id += 1;
    }

    int count = 0;
    for(auto& time : vecResponseTime){
        cout << "Index of Msg is "<< count << " and its response time is " << time << endl;
        count += 1;
    }
    return 0;
}

double computeResponseTime(int id, vector<Msg>& vecMsg, double tau)
{
    double responseTime = 0;
    double Q = 0, B = 0;

    // set block time
    for(auto& msg : vecMsg){
        if(msg.priority >= vecMsg[id].priority && msg.transmissionTime > B){
            B = msg.transmissionTime;
        }
    }
    Q = B;

    while(1){
        double rhs = 0.0;
        rhs += B;
        for(auto& msg : vecMsg){
            if(msg.priority < vecMsg[id].priority){
                rhs += ceil( (Q+tau)/msg.period ) * msg.transmissionTime;
            }
        }

        if(rhs + vecMsg[id].transmissionTime > vecMsg[id].period){
            cerr << "The system is not scheduable!" <<endl;
            return 1;
        }
        else if(Q == rhs){
            responseTime = Q + vecMsg[id].transmissionTime;
            return responseTime;
```

```
            responseTime = Q + vecMsg[id].transmissionTime;
            return responseTime;
        }

        else{
            Q = rhs;
        }
    }
    return 1;
}
```

3. (4pts) What is the worst-case response time of τ_2?

Iteration	LHS (R_2)	C_2	j	R_2	T_j	$\left\lceil \frac{R_2}{T_j} \right\rceil$	C_j	RHS	Stop?
1	20	20	0 1	20	50 200	1 1	10 30	60	No
2	60	20	0 1	60	50 200	2 1	10 30	70	No
3	70	20	0 1	70	50 200	2 1	10 30	70	Yes

4. (4pts) Compared with non-preemptive fixed-priority scheduling, preemptive fixed-priority scheduling is expected to be disadvantageous to the lowest-priority message/task. Explain why the the worst-case response time of τ_2 is smaller than the worst-case response time of μ_2 in Question 1.

4 Timing Analysis of TDMA-Based Protocols (12pts)

Following the assumptions (each time slot has the same length, each time slot serves exactly one frame, and a frame is transmitted only if the whole time slot is available) in the lecture, please compute the worst-case response time of the "asynchronous" message with the frame arrival pattern $(4, 10, 0, 3, 5, 6)$ and the schedule pattern $(2, 5, 1, 2)$ by completing the following steps.

1. (2pts) Please duplicate the schedule pattern (hint: $(4, 10, 1, 2, \ldots)$). No intermediate work is needed here.
2. (2pts) Please duplicate the arriving times of frames in the frame arrival pattern but fix $m = 4$ and $p = 10$. No intermediate work is needed here.
3. (2pts) Please duplicate the starting times of time slots in the schedule pattern but fix $n = 4$ and $q = 10$. No intermediate work is needed here.
4. (4pts) Please complete the following table:

k	$\max_{1\le j\le n}(s_{j+k} - s_j)$	=	$\min_{1\le i\le m}(a_{i+k-1} - a_i)$	=	(Column-3) − (Column-5)
1	$\max_{1\le j\le 4}(s_{j+1} - s_j)$	4	$\min_{1\le i\le 4}(a_i - a_i)$	0	4
2	$\max_{1\le j\le 4}(s_{j+2} - s_j)$	5	$\min_{1\le i\le 4}(a_{i+1} - a_i)$	1	4
3	$\max_{1\le j\le 4}(s_{j+3} - s_j)$	9	$\min_{1\le i\le 4}(a_{i+2} - a_i)$	3	6
4	$\max_{1\le j\le 4}(s_{j+4} - s_j)$	10	$\min_{1\le i\le 4}(a_{i+3} - a_i)$	6	4

5. (2pts) Please compute the worst-case response time (which is waiting time plus transmission time) of the message.

5 MILP Linearization (12pts)

We will prove or make the following propositions are equivalent so that we can transform constraints to linear forms and thus apply the Mixed Integer Linear Programming (MILP). Note that "$\Longleftrightarrow$" denotes "equivalence" and "$\land$" denotes "logical conjunction" (AND).

3

1.

$R_0 = C_0 = 10$

$10 = 10 + 0 = 10$

$R_0 = 10$ #

2.

$R_1 = C_1 = 30$

$30 \neq 30 + \left(\left\lceil \frac{30}{50} \right\rceil\right) \times 10 = 40$

$40 = 30 + \left(\left\lceil \frac{40}{50} \right\rceil\right) \times 10 = 40$

$R_1 = 40$ #

3.

$R_2 = C_2 = 20$

$20 \neq 20 + \left(\left\lceil \frac{20}{50} \right\rceil\right) \times 10 + \left(\left\lceil \frac{20}{200} \right\rceil\right) \times 30 = 60$

$60 \neq 20 + \left(\left\lceil \frac{60}{50} \right\rceil\right) \times 10 + \left(\left\lceil \frac{60}{200} \right\rceil\right) \times 30 = 70$

$70 = 20 + \left(\left\lceil \frac{70}{50} \right\rceil\right) \times 10 + \left(\left\lceil \frac{70}{200} \right\rceil\right) \times 30 = 70$

$R_2 = 70$ #

5.

μ_2 response time = 90

τ_2 response time = 70

Non–preemptive:

$$R_i = Q_i + C_i,$$

$$Q_i = B_i + \sum_{\forall j, P_j < P_i} \left\lceil \frac{Q_i + \tau}{T_j} \right\rceil C_j,$$

Preemptive :

$$R_i = C_i + \sum_{\forall j, P_j < P_i} \left\lceil \frac{R_i}{T_j} \right\rceil C_j,$$

本題是要探討 low priority task 的 worst-case，所以可以知道 preemptive 的情況一定會被 high priority 的 task 擋好擋滿，而 non-preemptive 的情況下也會是同樣的，在看看公式可以推知，preemptive 和 non-preemptive 計算 Σ 內的結果應該大致相同，而 μ_2 的 response time 會大於 τ_2 的 response time 的原因從公式中可以看出是 non-preemptive 需要額外多考慮 block time

4

1.

4、10、1、2、6、7

2.

a = 0、3、5、6、10、13、15、16

3.

S = 1、2、6、7、11、12、16、17

5.

$6 + 1 = 7$ #

1. (4pts) Given α, β, γ which are binary variables, prove

$$\alpha + \beta + \gamma \neq 2 \Longleftrightarrow \alpha + \beta - \gamma \leq 1 \wedge \alpha - \beta + \gamma \leq 1 \wedge -\alpha + \beta + \gamma \leq 1$$

by filling "T" (True) or "F" (False) in the following table (if LHS=RHS in all cases, then LHS and RHS are equivalent):

α	β	γ	LHS	$\alpha + \beta - \gamma \leq 1$	$\alpha - \beta + \gamma \leq 1$	$-\alpha + \beta + \gamma \leq 1$	RHS	LHS=RHS?
0	0	0	T	T	T	T	T	Equivalent
0	0	1	T	T	T	T	T	Equivalent
0	1	0	T	T	T	T	T	Equivalent
0	1	1	F	T	T	F	F	Equivalent
1	0	0	T	T	T	T	T	Equivalent
1	0	1	F	T	F	T	F	Equivalent
1	1	0	F	F	T	T	F	Equivalent
1	1	1	T	T	T	T	T	Equivalent

2. (4pts) Given α, β, γ which are binary variables, prove

$$\alpha\beta = \gamma \Longleftrightarrow \alpha + \beta - 1 \leq \gamma \wedge \gamma \leq \alpha \wedge \gamma \leq \beta$$

by filling "T" (True) or "F" (False) in the following table (if LHS=RHS in all cases, then LHS and RHS are equivalent):

α	β	γ	LHS	$\alpha + \beta - 1 \leq \gamma$	$\gamma \leq \alpha$	$\gamma \leq \beta$	RHS	LHS=RHS?
0	0	0	T	T	T	T	T	Equivalent
0	0	1	F	T	F	F	F	Equivalent
0	1	0	T	T	T	T	T	Equivalent
0	1	1	F	T	F	T	F	Equivalent
1	0	0	T	T	T	T	T	Equivalent
1	0	1	F	T	T	F	F	Equivalent
1	1	0	F	F	T	T	F	Equivalent
1	1	1	T	T	T	T	T	Equivalent

3. (4pts) Given β which is a binary variable, x, y which are non-negative real variables, and a constraint $x \leq 2022$, select a value of M to guarantee

$$\beta x = y \Longleftrightarrow 0 \leq y \leq x \wedge x - M(1-\beta) \leq y \wedge y \leq M\beta,$$

where you can refer to the following table:

β	LHS	$0 \leq y \leq x$	$x - M(1-\beta) \leq y$	$y \leq M\beta$	RHS
0	$0 = y$	$0 \leq y \leq x$	$x - M \leq y$	$y \leq 0$	$x - M \leq y = 0 \leq x$
1	$x = y$	$0 \leq y \leq x$	$x \leq y$	$y \leq M$	$0 \leq y = x \leq M$

6 Signal Packing (12pts)

Bit stuffing does not need to be considered in this problem, *i.e.*, you can assume that the length of a message is the length of its data field plus 44 plus 3. Note that the length of a data field must be 8, 16, 24, ..., or 64 bits, even if the message itself is shorter. Assume that there are 4 Electronic Control Units (ECUs), $\varepsilon_0, \varepsilon_1, \varepsilon_2, \varepsilon_3$, and 4 messages, $\mu_0, \mu_1, \mu_2, \mu_3$, as follows:

5.

3.

根據題目 $x \le 2022$

(1)

$\beta = 0$

$LHS = 0 \cdot x = y \Rightarrow y = 0$

$RHS = 0 \le y \le x$ 且 $\underbrace{x - M \le y}_{M\text{只要} \ge 2022\text{即可滿足式子}}$ 且 $y \le 0$

$\because 0 \le y$ 且 $y \le 0$

$\therefore y = 0$

$LHS = RHS$ #

(2)

$\beta = 1$

$LHS = 1 \cdot x = y \Rightarrow 0 \le x = y$ (大於等於0的原因是 x.y 是非負實數)

$RHS = 0 \le y \le x$ 且 $x \le y$ 且 $y \le M$

$\because y \le x$ 且 $x \le y$

$\therefore 0 \le y = x \le M$

M 若是 ∞,

就可以使 LHS 完全相等於 RHS #

$\therefore$ 根據 (1).(2) 得證 #

Message	Sender	Receiver(s)	Number of Bits (Data Field)	Period (msec)
μ_0	ε_0	ε_1	6	50
μ_1	ε_0	ε_1	10	50
μ_2	ε_1	$\varepsilon_2, \varepsilon_3$	10	50
μ_3	ε_0	ε_3	16	100

A system designer redesigns the messages as follows:

Message	Sender	Receiver(s)	Number of Bits (Data Field)	Period (msec)
μ_0'	ε_0	ε_1	16	50
μ_2	ε_1	$\varepsilon_2, \varepsilon_3$	10	50
μ_3	ε_0	ε_3	16	100

where the first 6 bits of μ_0' are the bits from μ_0 and the following 10 bits of μ_0' are the bits from μ_1.

1. (4pts) Regarding the number of bits which need to be transmitted, do you think that the new design is better? Please explain.
2. (4pts) Can you further merge μ_2 into μ_0'?
3. (4pts) In most cases, it does not hurt to have more frequent messages, but it is not allowed to have less frequent messages. Following this policy, can you further improve the number of bits which need to be transmitted? Please explain.

6

1.

Yes, 在沒有超過一個 message 所能負荷的量之前都可以藉由合併來save headers, 以此來提升效率 ($8+47+16+47=118 > 16+47=63$)

2.

No, 因為 μ_2 負責發送和接收的ECU與 μ_0' 不同

3.

因為題目沒有給任何的限制

所以我將 μ_3 切一半分為 α、β (各 8 bits)

並且前 50 msec 送 α, 後 50 msec 送 β (整体来看 period 依然是 100 msec)

接著 α、β 分別 μ_0' 一同發送

以下來計算一下我提出的方法是否有改善

原本方法:

$$(16+47)+(16+47)+\frac{1}{2}(16+47)=157.5 \text{ bits}/50\text{ msec}$$

改善後:

$$(24+47)+(16+47)=134 \text{ bits}/50\text{ msec}$$

$134 < 157.5$, 故可以更加改善 #