Performance Evaluation of Software Architectures

Exercises

José Costa

Software for Embedded Systems

Departamento de Engenharia Informática (DEI)
Instituto Superior Técnico

2015-10-06

Outline

- Software architectures - exercises

Exercise 1

Consider an embedded system with 4 devices whose software to handle the devices is as follows:

```
/* interrupt routine */
void interrupt handle_deviceA(){
  /* Take care of device A */
  ...
  fDeviceA = 1;
}
...
/* interrupt routine */
void interrupt handle_deviceD(){
  /* Take care of device D */
  ...
  fDeviceD = 1;
}
```

```
main() {
  while(1) {
    if (fDeviceA) {
      fDeviceA = 0;
      HandleDataA();
    }
    if (fDeviceB) {
      fDeviceB = 0;
      HandleDataB();
    }
    if (fDeviceC) {
      fDeviceC = 0;
      HandleDataC();
    }
    if (fDeviceD) {
      fDeviceD = 0;
      HandleDataD();
    }
  }
}
```

What type of software architecture is it?

Answer: Round-robin with interrupts

Exercise 1

Software

```
main() {
  while(1) {
    if (fDeviceA) {
      fDeviceA = 0;
      HandleDataA();
    }
    if (fDeviceB) {
      fDeviceB = 0;
      HandleDataB();
    }
    if (fDeviceC) {
      fDeviceC = 0;
      HandleDataC();
    }
    if (fDeviceD) {
      fDeviceD = 0;
      HandleDataD();
    }
  }
}
```

Requisites of the system and times for each device are:

Device	Max Latency time	Time handling device	Time handling data
A	2600	150	1000
B	2300	350	1500
C	2800	250	2000
D	2300	250	300

Assume that only occurs one interrupt per cycle of the main loop.
In the worst case, which device(s) restriction(s) (latency times) may not be met?

Answer: Device B

Assume that it can occur two interrupts at the same time in an iteration of the main loop.
In the worst case, which device(s) restriction(s)

Exercise 2

Consider a system with a simple Round-Robin Architecture with 5 tasks

Task	Execution times [ms]
T1	1.5
T2	[3 - 4.5]
T3	0.5
T4	[4 - 5]
T5	0.1(does watchdog refreshment)

The system has a watchdog timer with 10 ms period.

The system has apparently random but frequent malfunctions. **Why?**
Main cycle time? T1 + T2 + T3 + T4 + T5 = ?
9.1ms $\leq$ T1 + T2 + T3 + T4 + T5 $\leq$ 11.6ms
Sometimes T5 does not arrive quickly enough to refresh the watchdog.

How do we solve the problem? We have two alternatives:

- Increase the watchdog period from 10 to 12 ms;
- Include in the main loop an additional refreshment of the watchdog (another call to T5).

Where do we insert the extra call? Not between calls to T1 and T2.
The new main cycle time would be:
9.2 ms $\leq$ T1 + T2 + T5 + T3 + T4 + T5 $\leq$ 11.7 ms .
The maximum time between refreshments would be:
max (T1+T2, T3+T4) = max (6 ms, 5.5 ms) = 6 ms.

Exercise 3 (1/3)

Considere an embedded system program that interacts with 4 external devices:

```
/* Interrupt rotines */
void interrupt handle_device_A (void) {
  /* Handles device A */
  Insert_in_queue (function_A);
}
void interrupt handle_device_D (void) {
  /* Handles device D */
  Insert_in_queue (function_D);
}
/* Main cycle */
void main (void) {
  while (TRUE) {
    call Get_from_queue (); /* Call function from head of queue and handles device
  }
}
```

Type of architecture: Function queue.

Exercise 3 (2/3)

System requirements - Latency time from interrupt until beginning of processing information [μs]:

Device A	1.000
Device B	1.600
Device C	4.000
Device D	10.000

Response times of the system[μs]:

handleA	100	Process_dataA	400
handleB	50	Process_dataB	1.000
handleC	40	Process_dataC	500
handleD	35	Process_dataD	200
Latency time from interrupts		8 (hardware)	

When in operation the system showed some sporadic errors. **Why?**

Justify the occurence of errors by computing the latency times of the devices.
latency time of X = t_handleX + t_main_cycle
t_main_cycle = max(t_Process_DataY) = 1000μs (task B)

latecy time of A = 100+1000=1100μs
latecy time of B = 50+1000=1050μs
latecy time of C = 40+1000=1040μs
latecy time of D = 35+1000=1035μs

Latency time of A violates requirement. So, there are situations where service of device A is not properly handled.

Exercise 3 (3/3)

- **Sugest modifications to the initial program, maintaining the general structure, in order to eliminate intermittent faults.**

- The only way to solve the problem is to decompose the most time consuming functions in smaller modules, if the application permits.
- In this case it's task B (Process_dataB) that poses problems due to its duration so it should be decomposed into two tasks, where neither can not take more than 900μs

Exercise 4 (1/4)

We want to build an electronic scale that communicates with the exterior by asynchronous serial communication at 19200 baud.

A piezoelectric sensor generates a voltage proportional to the weight to be measured. this voltage is converted into a digital value by an analog-digital converter incorporated in the microcontroller balance. This microcontroller also includes a timer and an UART.

The system should do weight measurements at intervals of 20 ms. Whenever it receives a request for serial communication, the system sends an average of the 3 latest measurements, indicating also if the weight is stable.

The response should not take longer than 80 ms.

Exercise 4 (2/4)

- **Should this system be considered a real-time system?**

- Although there are several specified response times with time constraints, it is not clear, for lack of context of the application, that its violation can not occur without serious flaws.
- On the other hand, removing the need to maintain communication at 19200 baud, the temporal requirements are not very tight and therefore we do not consider this a real-time system.

Exercise 4 (3/4)

- **Propose a software architecture for the system. Justify.**

- Round-robin with interrupts. It is the simplest architecture with interruptions.
- The presence of interrupts in this case will simplify the construction of the application, namely the service of the timers and the serial channel.

Exercise 4 (4/4)

- **Which tasks do you need to the software and what should be the role of each task.**
- Tasks:
- T1: Service of the asynchronous serial channel
- T2: Reading the weight
- T3: Computation of the weight information to send.
- Interrupt handlers:
- H1: Transmission of characters by the channel. (Possibly two interrupt routines, one for reception, one for sending.)
- H2: Real-time clock. Mark the periods of the weight reading.
- H3: Reading the weight. Necessary routine if the analog-digital converter is synchronized with the processor using interrupts.

Exercise 5

Consider an embedded system with 4 devices whose software architecture to handle the devices is round-robin with interrupts.
Assume also that, for each device in the system, the response times for the interrupt routine, the response time for handling data and the period are:

Device	Time handling device	Time handling data	Period
A	150	1000	6000
B	150	1500	6000
C	250	2000	6000
D	50	250	6000

Is it possible to always meet all deadlines?
Worst case: All interrupts happen at the same time.
In that case, the time it takes for everything to complete is
150+150+250+50+1000+1500+2000+250 = 5350
All deadlines are met!

In the worst case, what is the latency time of Device A?
Worst case: All interrupts happen at the same time.
And software has finished testing device A.
In that case the time it takes from the interruption of device A to start handling its data is:
150+150+250+50+1500+2000+250 = 4350

Next class

- Multitasking Operating Systems