Name: Ajita Shree

Roll No e.g. 170001: 20111262

Dept. e.g. CSE: Computer Science Engineering

Indian Institute of Technology Kanpur
CS637 Embedded and Cyber-Physical Systems
Homework Assignment 4
Deadline: November 8, 2020

Instructions:

Total: **40 marks**

1. This question paper contains a total of 9 pages (9 sides of paper). Please verify.
2. Write your name, roll number, department, section on **every side of every sheet** of this booklet
3. Write final answers **neatly** in the given boxes.

Problem 1. (10 points) Work out Problem 4 in the Exercises of Chapter 10 in [LS15].
[LS15] Edward A. Lee and Sanjit A. Seshia, Introduction to Embedded Systems, A Cyber-Physical Systems Approach, Second Edition, http://LeeSeshia.org, ISBN 978-1-312-42740-2, 2015.

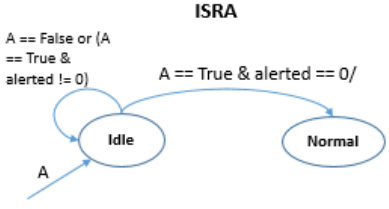
Solution

The dashboard displays "normal" when the brakes in the car operates normally and "emergency" when there is a failure. The intended behavior is that once "emergency" has been displayed, "normal" will not again be displayed. That is, "emergency" remains on the display until the system is reset. ISRA is the interrupt service routine that is invoked when brakes are applied by the driver. ISRB is invoked if a sensor indicates that the brakes are being applied at the same time that the accelerator pedal is depressed. ISRB has higher priority than ISRA. Assuming each line of code to be atomic, answer the following questions.

```
volatile static uint8_t alerted;
volatile static char* display;
void ISRA() {
    if (alerted == 0) {
        display = "normal";
    }
}
void ISRB() {
    display = "emergency";
    alerted = 1;
}
void main() {
    alerted = 0;
    ...set up interrupts...
    ...enable interrupts...
    ...
}
```

- **a)** Does this program always exhibit the intended behavior.
  Ans - Yes, The program will be able to exhibit the behavior but additional statements can be added to remove the ambiguity. As per the program, it is clear that ISRA will not be invoked after ISRB, the alerted == 0 condition will take care of that. Once we reset the program, alerted will be equal to 0 and ISRA can be enabled until ISRB.

- **b)** Construct a determinate extended state machine modeling ISRA. Input is signal A that when present indicates interrupt request for ISRA. Display is output of char*



- **c)** Size of state space of the solution is 2 as shown in the above diagram

Name: Ajita Shree

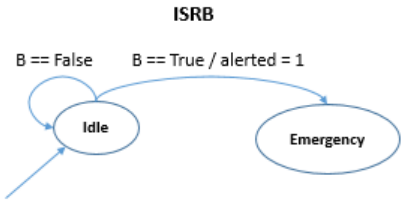
Roll No e.g. 170001: 20111262

Dept. e.g. CSE: Computer Science Engineering

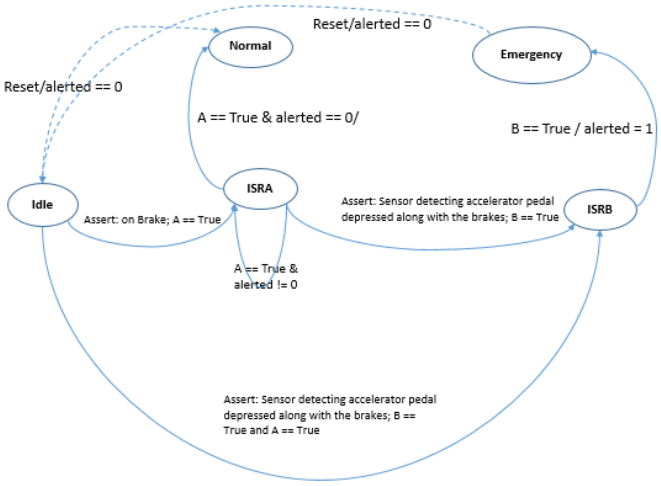
Indian Institute of Technology Kanpur
CS637 Embedded and Cyber-Physical Systems
Homework Assignment 4
Deadline: November 8, 2020

- **d)** Explain your assumptions about when the state machine in b reacts. Is this time triggered or event triggered?
  The state machine in b will react when the ISRA interrupt is triggered i.e. when the brakes are applied. An additional condition of alerted = 0 is to ensure that the state responses only when this condition is met. Alerted will be set to 1 once ISRB is envoked.It is important to check alerted as we don't want to envoke ISRA after ISRB. This is event triggered.

- **e)** Construct a determinate state machine modelling ISRB. Input is pure signal B that when present indicates an interrupt request for ISRB

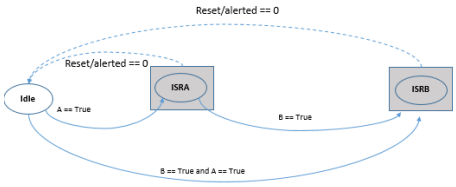


- **f)** Construct a flat determinate extended state machine describing the joint operation of these two ISRs. Use your model to argue the correctness of your answer to part a



The state machine is ensuring that ISRA is not executed after ISRB unless the system is reset. The state machine also is ensuring invoke of ISRA or ISRB at any time depending on conditions for display of "normal" or "emergency".

- **g)** Below is an equivalent hierarchical state machine corresponding to the above flat extended machine.



It is depicting 3 states i.e. idle, ISRA invoke and ISRB invoke based on the specifications given in the program.

Name: Ajita Shree

Roll No e.g. 170001: 20111262

Dept. e.g. CSE: Computer Science Engineering

Indian Institute of Technology Kanpur
CS637 Embedded and Cyber-Physical Systems
Homework Assignment 4
Deadline: November 8, 2020

Problem 2. (10 points) Work out Problem 2 in the Exercises of Chapter 11 in [LS15].

Solution

In the following program, design a procedure 1 such that lock b is acquired before lock a by any thread that acquires both the locks to ensure deadlocks don't happen. Give a design for procedure 1 that minimizes unnecessary acquisitions for lock B. Does your solution eliminate unnecessary acquisitions of lock B? Is there a solution that does this?

```
int a, b;
pthread_mutex_t lock_a
    = PTHREAD_MUTEX_INITIALIZER;
pthread_mutex_t lock_b
    = PTHREAD_MUTEX_INITIALIZER;

void procedure1(int arg) {
  pthread_mutex_lock(&lock_a);
  if (a == arg) {
     procedure2(arg);
  }
  pthread_mutex_unlock(&lock_a);
}

void procedure2(int arg) {
  pthread_mutex_lock(&lock_b);
  b = arg;
  pthread_mutex_unlock(&lock_b);
}
```

Yes,there exist a design that can eliminate unnecessary acquisitions for lock B. Following is the pseudo code for the same.

void procedure1(int argument) {
pthread_mutex_lock(&lock_a);
if (a == arg) {
set Flag = True}
pthread_mutex_unlock(&lock_a);

pthread_mutex_lock(&lock_b);
if (Flag == True) {
procedure2(arg)
pthread_mutex_unlock(&lock_b);
}

Explanation: Lets say there are two processes and are trying to execute procedure 1. Only one of the process can access shared variable **a** at a moment using $lock_a$. The flag variable defined in the routine will be set based on the condition. While calling procedure 2, the condition can be tested as Flag == True and using only one $lock_b$ for variable i.e. **b**. Hence, $lock_b$ is only acquired when the condition a == arg is met without having to unnecessarily block $lock_a$ and $lock_b$ together even when the condition is not met.

Name: Ajita Shree

Roll No e.g. 170001: 20111262

Dept. e.g. CSE: Computer Science Engineering

Indian Institute of Technology Kanpur
CS637 Embedded and Cyber-Physical Systems
Homework Assignment 4
Deadline: November 8, 2020

Name: Ajita Shree

Roll No e.g. 170001: 20111262

Dept. e.g. CSE: Computer Science Engineering

Indian Institute of Technology Kanpur
CS637 Embedded and Cyber-Physical Systems
Homework Assignment 4
Deadline: November 8, 2020

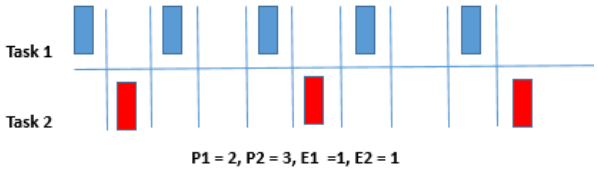
Problem 3. (10 points) Work out Problem 3 in the Exercises of Chapter 12 in [LS15]

Solution

Consider two tasks with periods p1 = 2 and p2 = 3 and execution times e1 = e2 = 1. Assume that the deadline for each execution is end of the period.

- Give RM schedule for this task set and find the processor utilization. How does this utilization compares to Liu and Layland utilization bound?

  Ans: As RM schedules works by assigning higher priority to the lower period task, hence, task 1 will be preferred over task 2. The schedule will be as follows.The task 1 will be repeating with period p1 and task 2 with period p2. Both the tasks are able to finish execution within their deadline (i.e. end of the period). The processor utilization is 75 percent (1 in 4 times the processor is not utilization because of the periods of task 1 and 2). The Liu and Layland utilization for n = 2 is <= 0.828 i.e. 82 percent. **Schedule (Task 1 is scheduled first then Task 2 and series as follows based on the tasks period)**



- Show that any increase in e1 or e2 makes the RM schedule infeasible.
  - Below diagrams shows that when the execution times of each task is increased by 1 how does it affect the schedules.In the first case, when e1 is increased by 1, Schedule is infeasible as Task 2 will not at all get a chance to get scheduled and only Task 1 will be repeated at regular time intervals of p1 i.e.2 units.

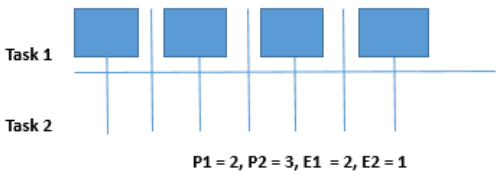
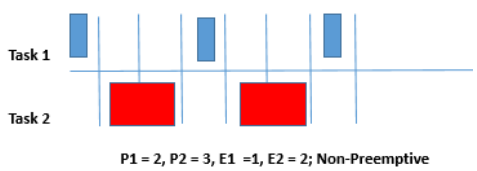
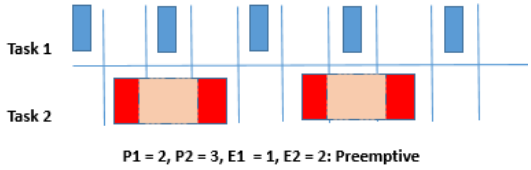


Figure 1: e1 is increased by 1

  - When e2 is increased by 1, the schedule is infeasible because a) In 1st case, task 1 is not able to finish off in its period e.g. task 1 came for processor at time 1, 3, 5 time units, but due to unavailability it gets delayed and is not able to complete within the deadline. Whereas in case b) if task 1 is able to preemt, task 2 requests for processor at 2nd, 5th, 8th time units but is not able to complete before its deadline. Hence, the schedule is infeasible.





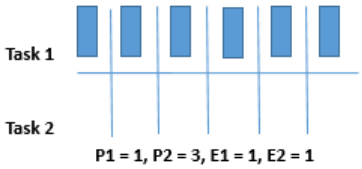
Name: Ajita Shree

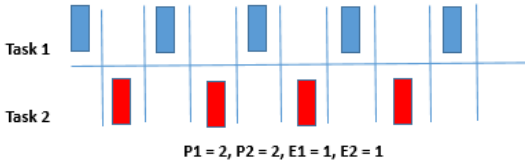
Roll No e.g. 170001: 20111262

Dept. e.g. CSE: Computer Science Engineering

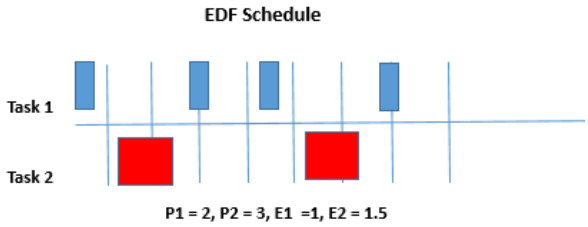
Indian Institute of Technology Kanpur
CS637 Embedded and Cyber-Physical Systems
Homework Assignment 4
Deadline: November 8, 2020

- If you hold e1 = e2 = 1 and p2 = 3 constant, is it possible to reduce p1 below 2 and still get a feasible solution? By how much? If you hold e1 = e2 = 1 and p1 = 2 constant, is it possible to reduce p2 below 3 and still get a feasible solution? By how much?
  - Case1: p1 = 1 : Task 2 will not be able to get schedule as task 1 will be repeatedly schedule as its period is 1.The solution is not feasible.
  - Case 2: p2 = 2: The solution is feasible with 100 percent processor utilization. So, increase in processor utilization compared to a) is 33.3 percent.





- Increase the execution time of task 2 to be e2 = 1.5 and give an EDF schedule. Is it feasible? What is processor's utilization?
  The schedule is feasible as shown below. The processor utilization is 87.5 percent (as the processor is occupied 7 / 8 times as shown in the diagram below).



Name: Ajita Shree

Roll No e.g. 170001: 20111262

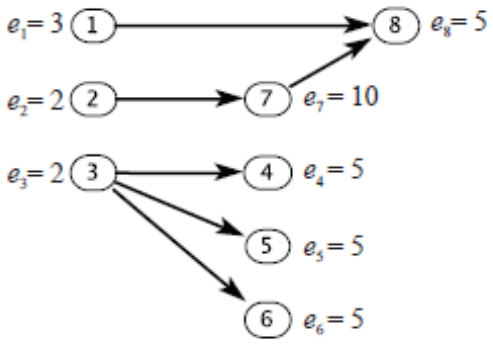
Dept. e.g. CSE: Computer Science Engineering

Indian Institute of Technology Kanpur
CS637 Embedded and Cyber-Physical Systems
Homework Assignment 4
Deadline: November 8, 2020

Problem 4. (10 points) Work out Problem 6 in the Exercises of Chapter 12 in [LS15].

Solution

The task with lower id has higher priority and there is no preemption.



- Schedule these tasks on two processors. Draw the schedule for this tasks and report the Makespan.
  - P1: 111777777777766666
  - P2: 2233444445555588888
  - Makespan: 19
- Schedule on 3 processors. Report Makespan. Is it bigger or smaller than part a.
  - P1: 11166666
  - P2: 22444447777777777
  - P3: 3355555− − − − − − − − − −−88888
  - MakeSpan: 22
  - Makespan is greater than part a
- If execution time is reduced by 1 unit for all the tasks and scheduled on 2 processor. Report Makespan. Is it bigger or smaller than part a.
  - P1: 1144446666− − − − −8888
  - P2: 235555777777777
  - Makespan: 19
  - Makespan is equal to that of part a

Name: Ajita Shree

Roll No e.g. 170001: 20111262

Dept. e.g. CSE: Computer Science Engineering

Indian Institute of Technology Kanpur
CS637 Embedded and Cyber-Physical Systems
Homework Assignment 4
Deadline: November 8, 2020

Name: Ajita Shree

Indian Institute of Technology Kanpur
CS637 Embedded and Cyber-Physical Systems
Homework Assignment 4
Deadline: November 8, 2020

Roll No e.g. 170001: 20111262

Dept. e.g. CSE: Computer Science Engineering

BLANK SPACE: Any answers written here will be left ungraded.
No exceptions.
You may use this space for rough work.