Software Design

Hossam Adel

SPRINTS

Who’s your instructor !

- Hossam Adel (+7 year of Experience)
  - Graduated 2011 (Benha University)
  - Master Degree in Embedded System
  - IST Industries (2 year) - CAI, EGY
  - Avelabs (2 year) – CAI, EGY
  - Thirdwayv (1 year) – CAI, EGY
  - Delphi Technologies (+1 year) – MI, USA
  - Freelancer Instructor Resala, AMIT, ITI, eCore and Sprints (+5 year)

Agenda

- Introduction
- Static Design
- Dynamic Design
- TMU (Timer Magnt Unit)
- BCM (Basic Com Module)
- Small OS
- Application Design using RTOS
- References
- Q&A

SPRINTS

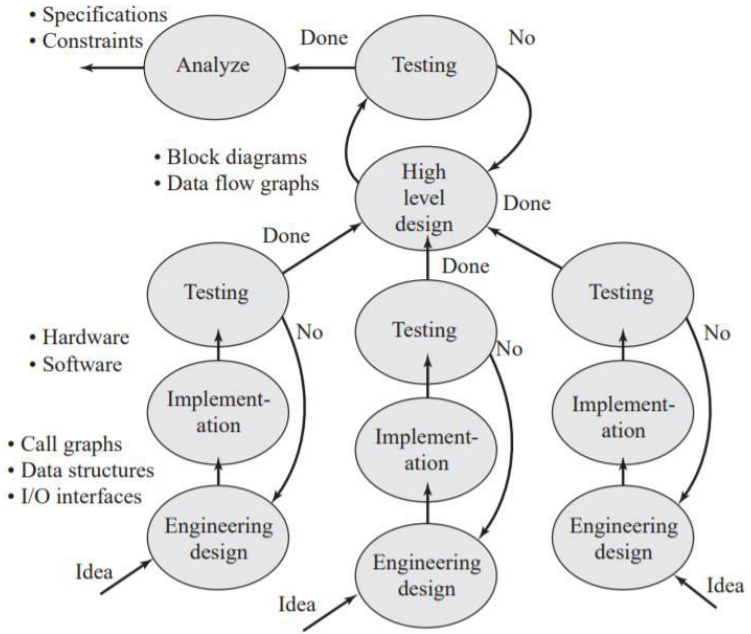
Introduction

- SW Design
  - Software design is the process of envisioning and defining software solutions to one or more sets of problems. One of the main components of software design is the software requirements analysis (SRA).
- Design Specs
  - **Compatibility** (backward-compat
  - Extendibility (add new code)
  - Modularity
  - Maintainability (easy for debug)
  - Reusability
  - Robustness (work hard)
  - Usability (user interface)
  - Scalability (increase input data)

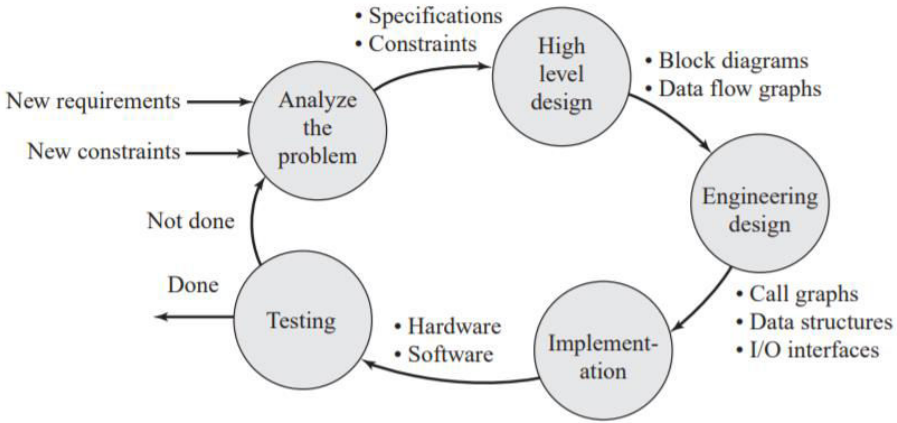
SPRINTS

Software Design

Bottom Up Design



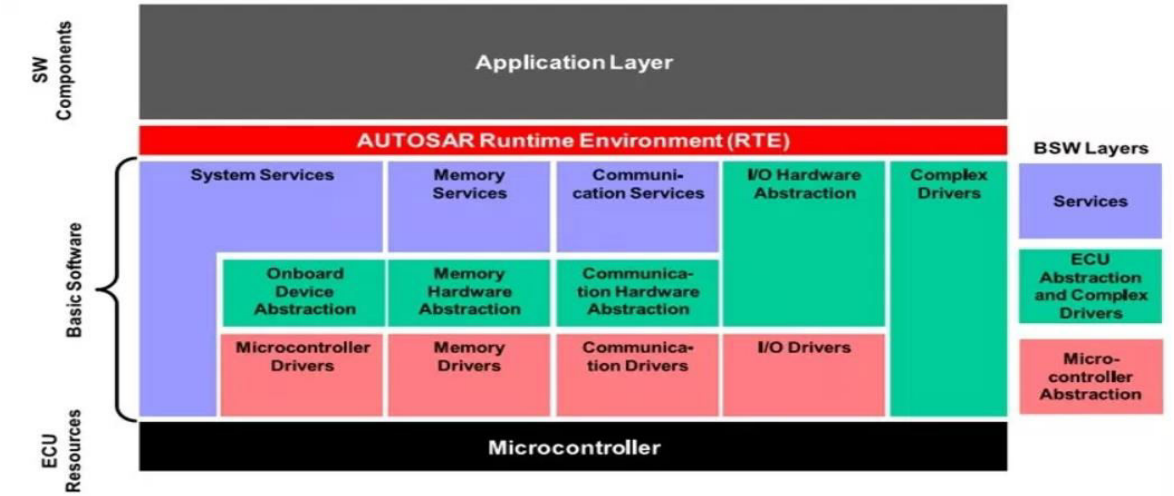
Top-Down Design



SPRINTS

Static Design (software architecture)

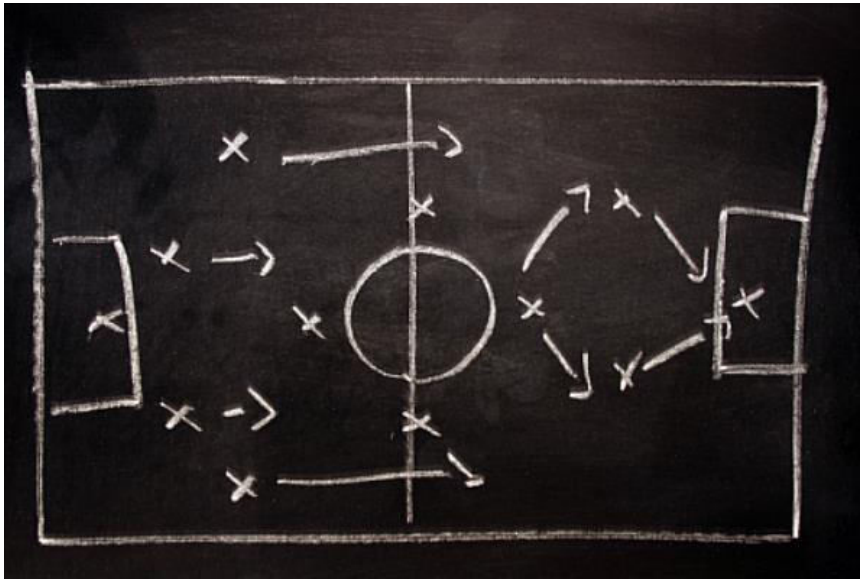
- Static Design have an answer for ..
  - What are the layers we have ?
  - Where’s the module allocate ?



SPRINTS

Dynamic Design

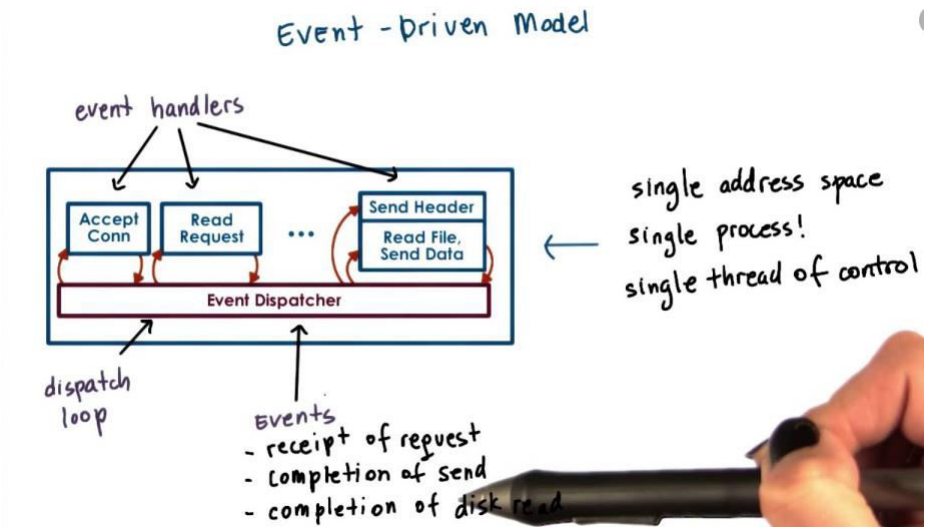
- Dynamic Design always answer how the modules talk with each other !



SPRINTS

Dynamic Design

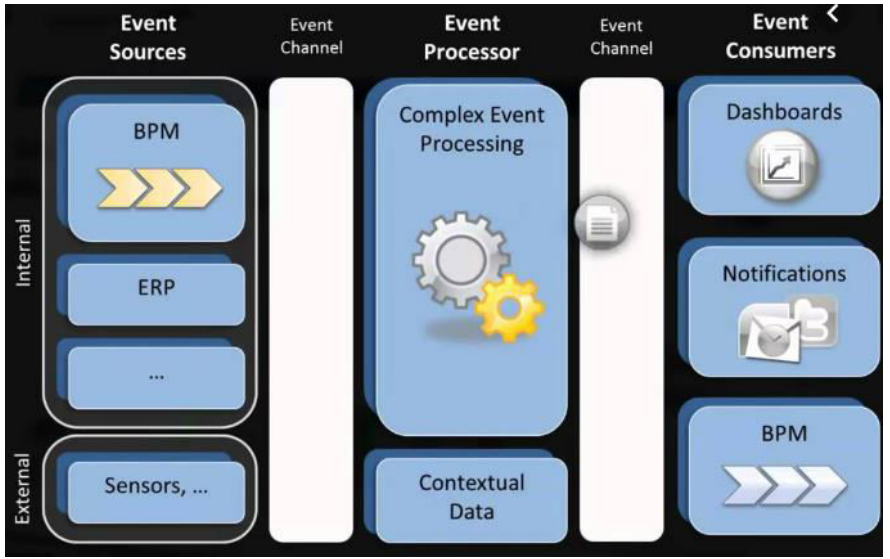
- Super Loop Design
  - Event driven
    - Interrupt base Design
    - Event base Design
- RTOS Design
  - Timing base Design



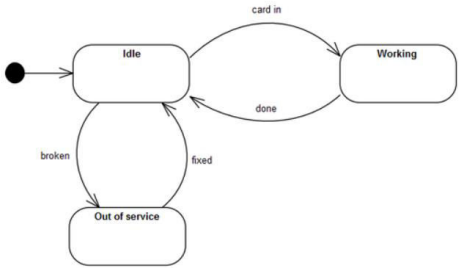
SPRINTS

Event Driven Design (EDA)

- Event Creator
- Event Manager
- Event Consumer



- Sequence Diagram.
- Finite state machine.



SPRINTS

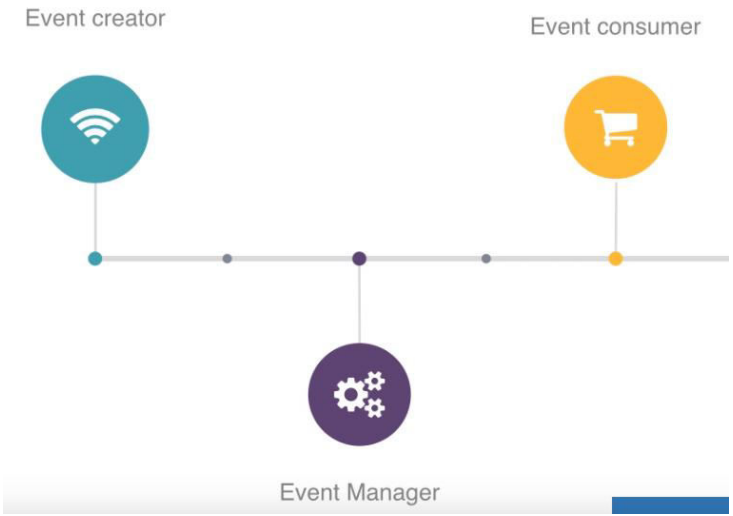
Super Loop Design

```c
void System_Init(void)
{
    TMU_Init();
    BCM_Init();
}

int main(void)
{
    /*init area*/
    System_Init();

    /*Super Loop*/
    while(1)
    {
        /* call the dispatch periodic functions */
        /* First has higest Priority */
        TMU_Dispatcher();
        BCM_Dispather();


        CPU_Sleep();
    }
}
```

SPRINTS

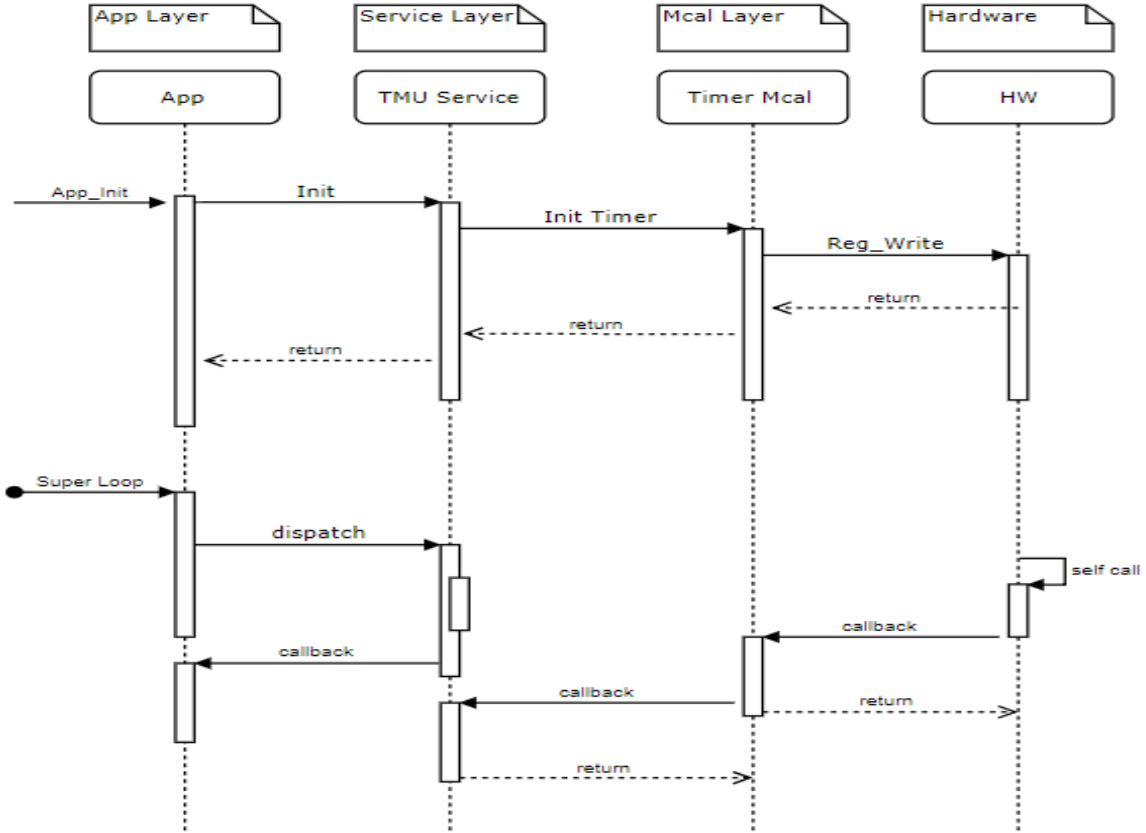
TMU (Timer Management Unit)

- TMU_Init
- TMU_Dispatcher
- TMU_Start_Time
- TMU_Stop_Time
- TMU_DeInit



SPRINTS

TMU (Timer Management Unit)



SPRINTS

BCM (Basic Com Module)

How to send 1000 byte ?

Draw the Sequence Diagram !

SPRINTS

BCM (Basic Com Module)

- BCM_Init
- BCM_Send
- BCM_Setup_Receive
- BCM_RxDispatcher
- BCM_TxDispatcher
- BCM_RxUnlock
- Packet [Command - Size – Data – CS]

SPRINTS

CPU Load

- How to calculate the CPU utilization ?! In Super Loop Design Mechanism.

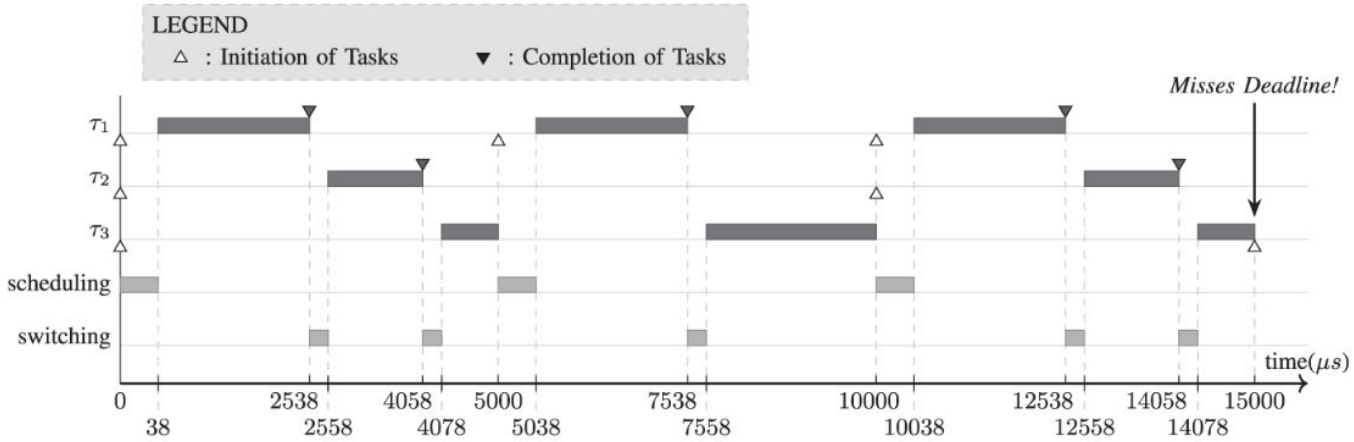
- Assume we have two Dispatcher functions (TMU and BCM).

- CPU Load = ??
- Twdg = ??

SPRINTS

OS Design

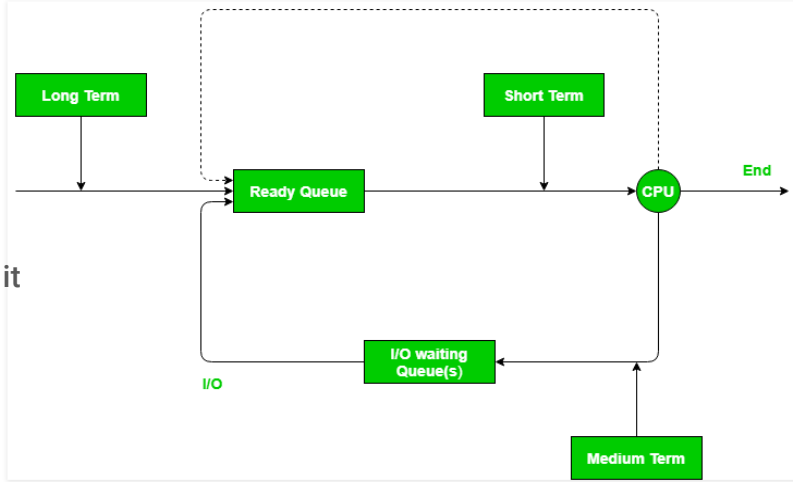
- Why we use OS ?!
- Where should OS be in the SW architecture ?
- Is the OS adding system overhead !



SPRINTS

Dispatcher VS Scheduler

- Schedulers are special system software which handle process scheduling in various ways. Their main task is to select the jobs to be submitted into the system and to decide which process to run. There are three parts (Long, Medium and Short term).

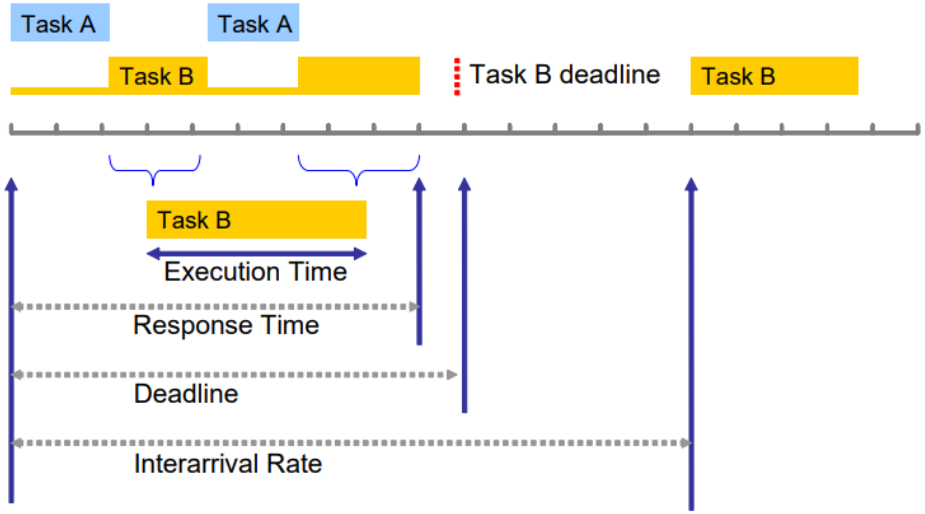


- **Dispatcher** is a special program which comes into play after the scheduler. When the scheduler completes its job of selecting a process, it is the dispatcher which takes that process to the desired state/queue. The dispatcher is the module that gives a process control over the CPU after it has been selected by the short-term scheduler. This function involves the following:
  - Switching context
  - Switching to user mode
  - Jumping to the proper location in the user program to restart that program

SPRINTS

Small OS Design

- Sos_Init
- Sos_Run
- Sos_Create_Task
- Sos_Delete_Task



- Task Design and Draw the Sequence Diagram ! ☺

SPRINTS

Application Design Using RTOS

SPRINTS

Design Tasks

- Architect need to decide :
  - N: number of tasks in the system
  - C : Task Capacity
  - T: Task periodicity
  - Ttick : System tick
  - Twdg : Wdg Timer period

SPRINTS

RMS (Rate-monotonic scheduling)

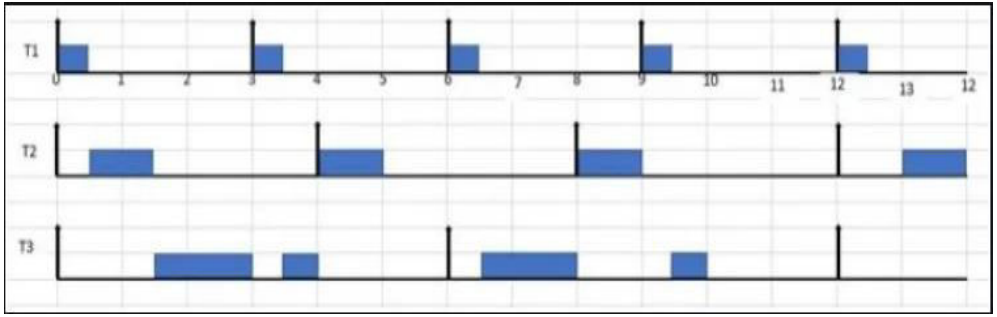
- is a priority assignment algorithm used in real-time operating systems (RTOS) with a static-priority scheduling class.

- The static priorities are assigned according to the cycle duration of the job, so a shorter cycle duration results in a higher job priority.

SPRINTS

RMS (Rate-monotonic scheduling)

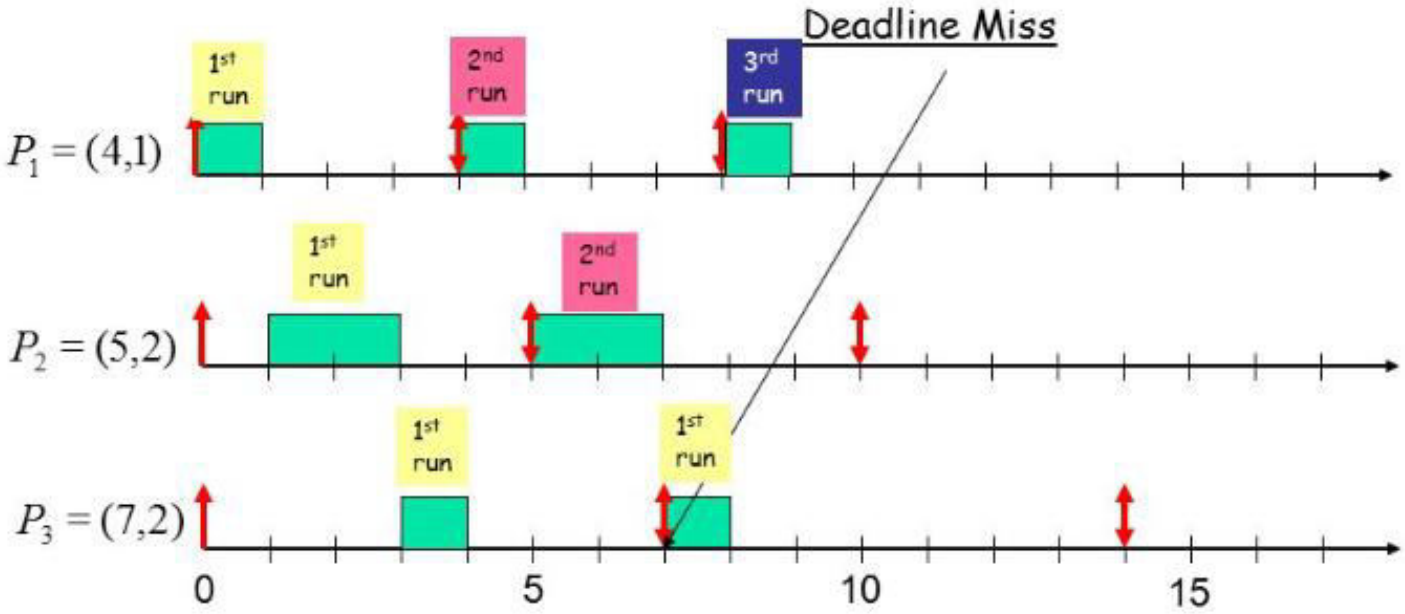
- Ti (Pi , Ci) P : Period , C : Capacity (Processing Time)
  - T1 (3 , 0.5) - Highest
  - T2 (4 , 1) - Medium
  - T3 (6 , 2) - Lowest



SPRINTS

RMS (Rate-monotonic scheduling)

$P_i = (T_i, C_i)$ T_i = period C_i = processing time



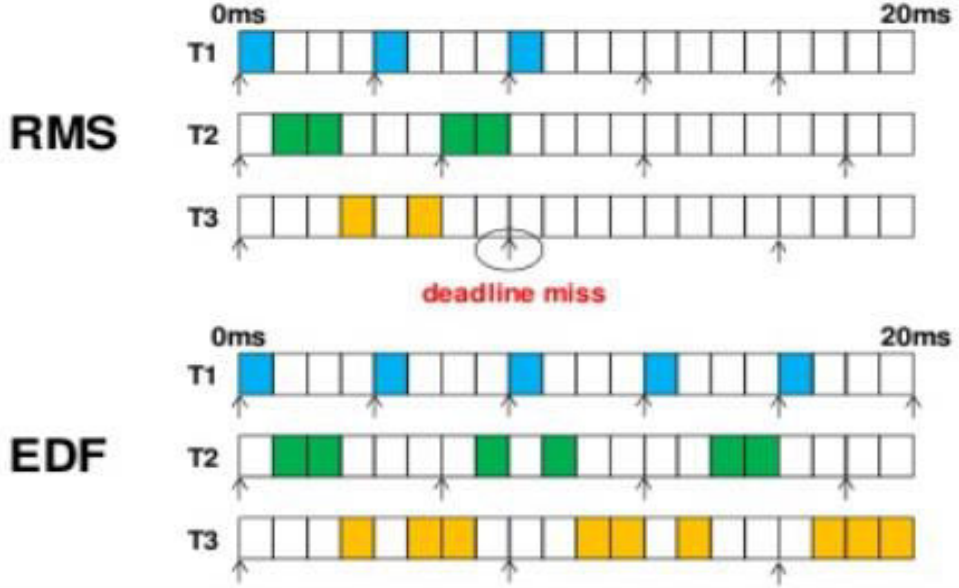
SPRINTS

RMS vs EDL

Compared with the RMS scheduling

- Task1: budget 1ms period 4ms
- Task2: budget 2ms period 6ms
- Task3: budget 3ms period 8ms

CPU usage



SPRINTS

RMS (Rate-monotonic scheduling)

- Liu & Layland (1973) proved that for a set of *n* periodic tasks with unique periods, a feasible schedule that will always meet deadlines exists if the CPU utilization is below a specific bound (depending on the number of tasks). The schedulable test for RMS is:

$$U = \sum_{i=1}^{n} \frac{C_i}{T_i} \leq n(2^{1/n} - 1)$$

$$\lim_{n \to \infty} n(\sqrt[n]{2} - 1) = \ln 2 \approx 0.693147\ldots$$

- U : CPU Utilization
- C: Task Capacity
- T: Task Periodicity
- N : Number of tasks

SPRINTS

References

- Use https://www.draw.io/ to draw the sequence Diagram

- http://s1.nonlinear.ir/epublish/book/Embedded_Microcomputer_Systems_Real_Time_Interfacing_1111426252.pdf
- https://pdfs.semanticscholar.org/a312/430dc5a54a1cae78e19f06c3ffa8509015a7.pdf
- https://en.wikipedia.org/wiki/Rate-monotonic_scheduling

SPRINTS

THANK YOU

Hossam Adel

SPRINTS