Technical Seminar Presentation

On

“Real Time Operating Systems”

Submitted By

Chetan M Mudenoor

CONTENTS

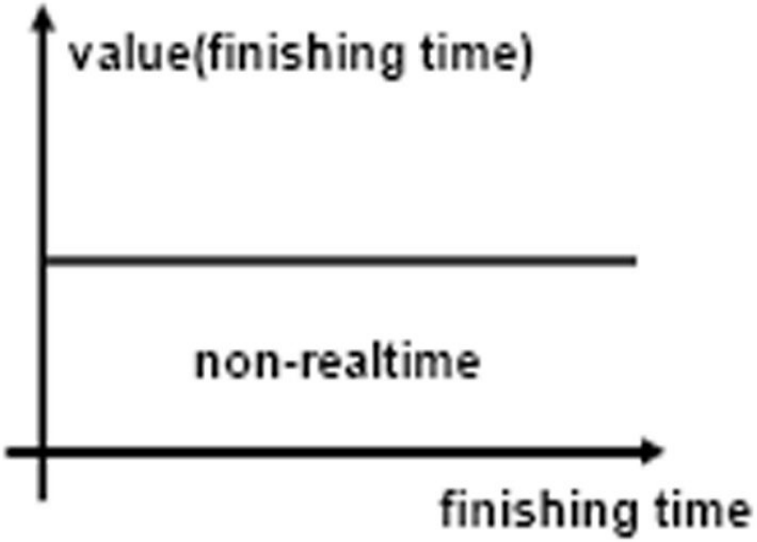
- What is RTOS?
- Types of RTOS
- RTOS Kernel
- Task Scheduling
- Task switching timing
- Inter task communication and synchronization
- Dynamic memory allocation
- Difference b/w RTOS & GPOS
- Where can real time systems to be found? or Advantages
- Application of RTOS

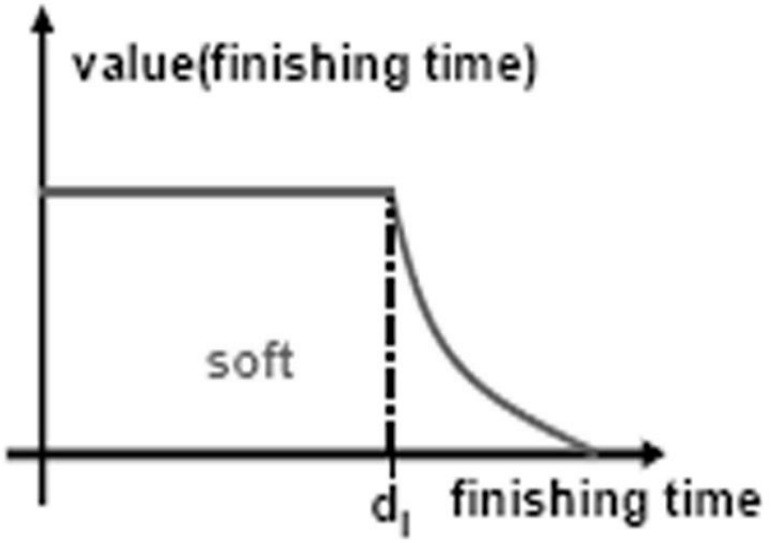
What is RTOS?

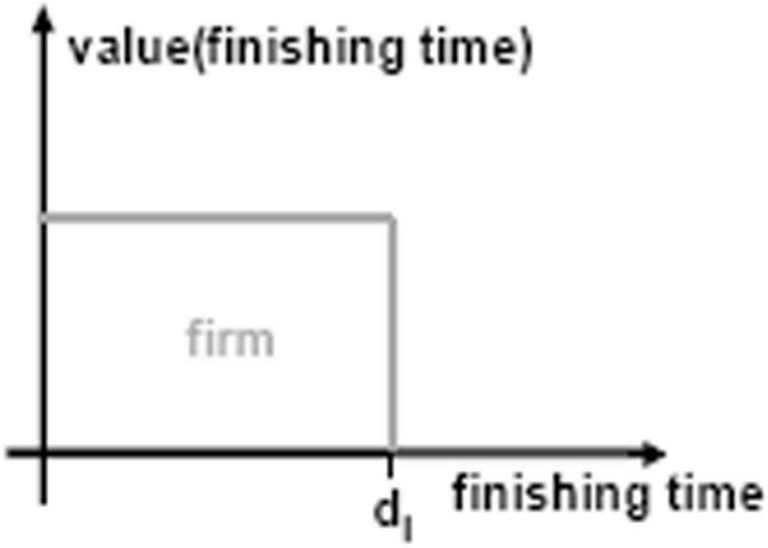
- Real time Operating Systems are very fast and quick respondent systems. These systems are used in an environment where a large number of events (generally external) must be accepted and processed in a short time.
- In real time operating system there is a little swapping of programs between primary and secondary memory. Most of the time, processes remain in primary memory in order to provide quick response, therefore, memory management in real time system is less demanding compared to other systems.

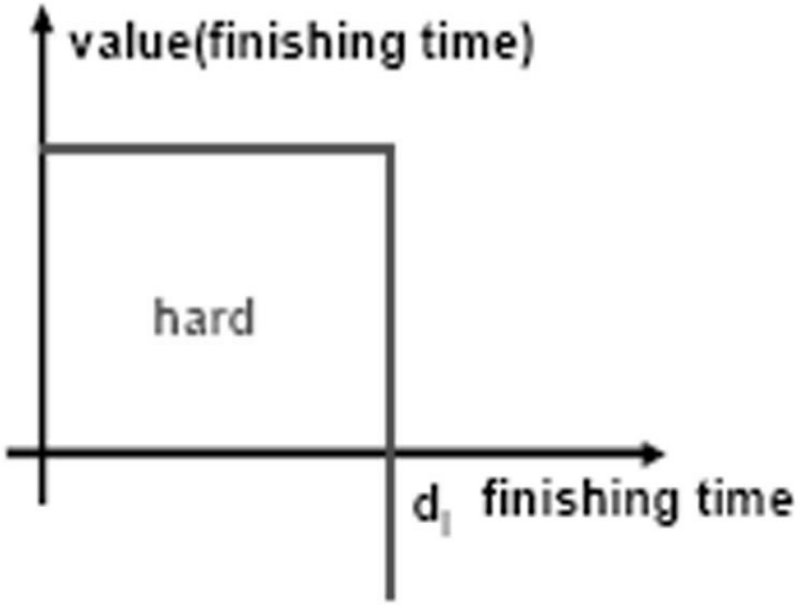
Types of RTOS

- Hard real time
- Soft real time
- Firm real time

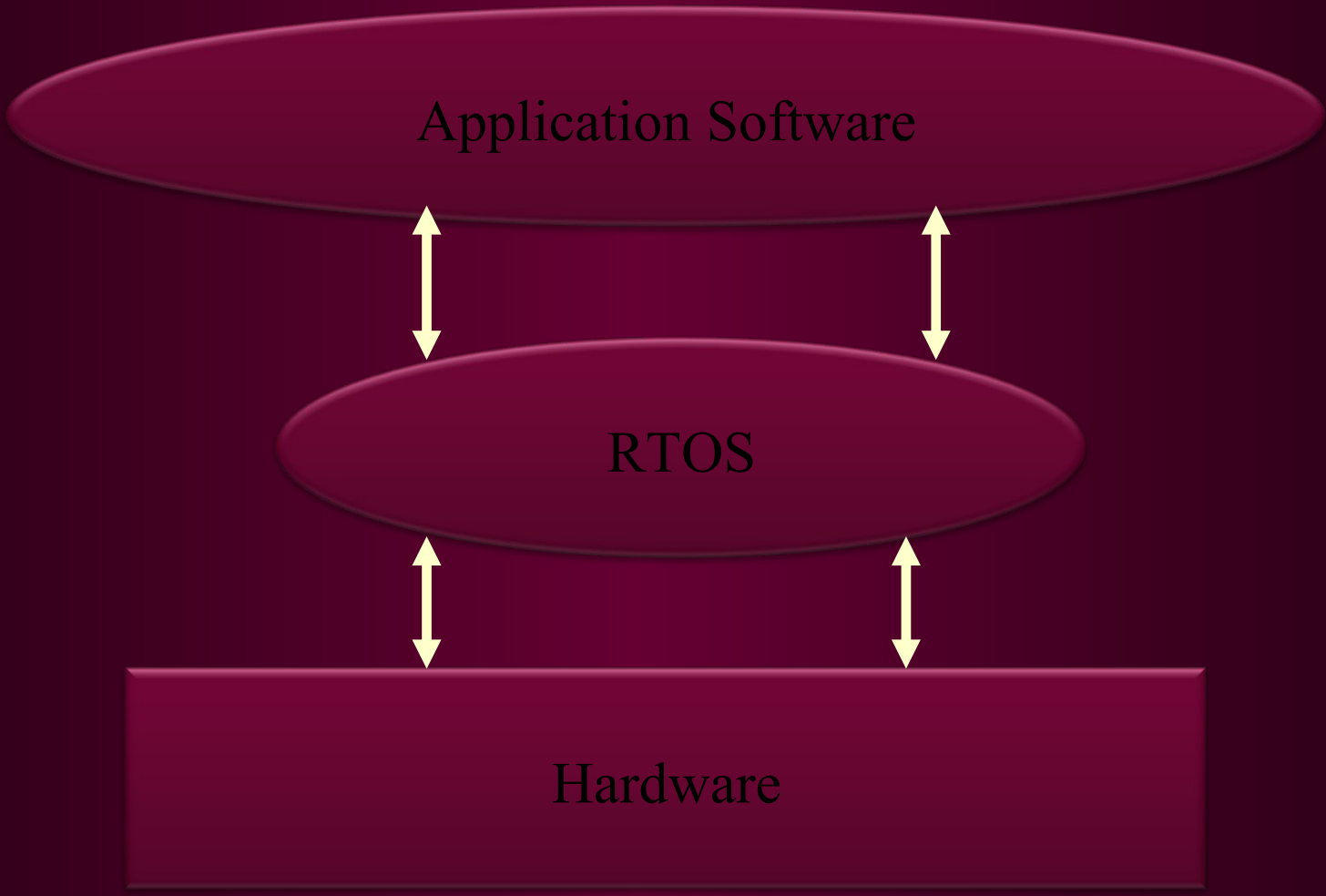

value(finishing time)
non-realtime
finishing time


value(finishing time)
soft
d_i
finishing time

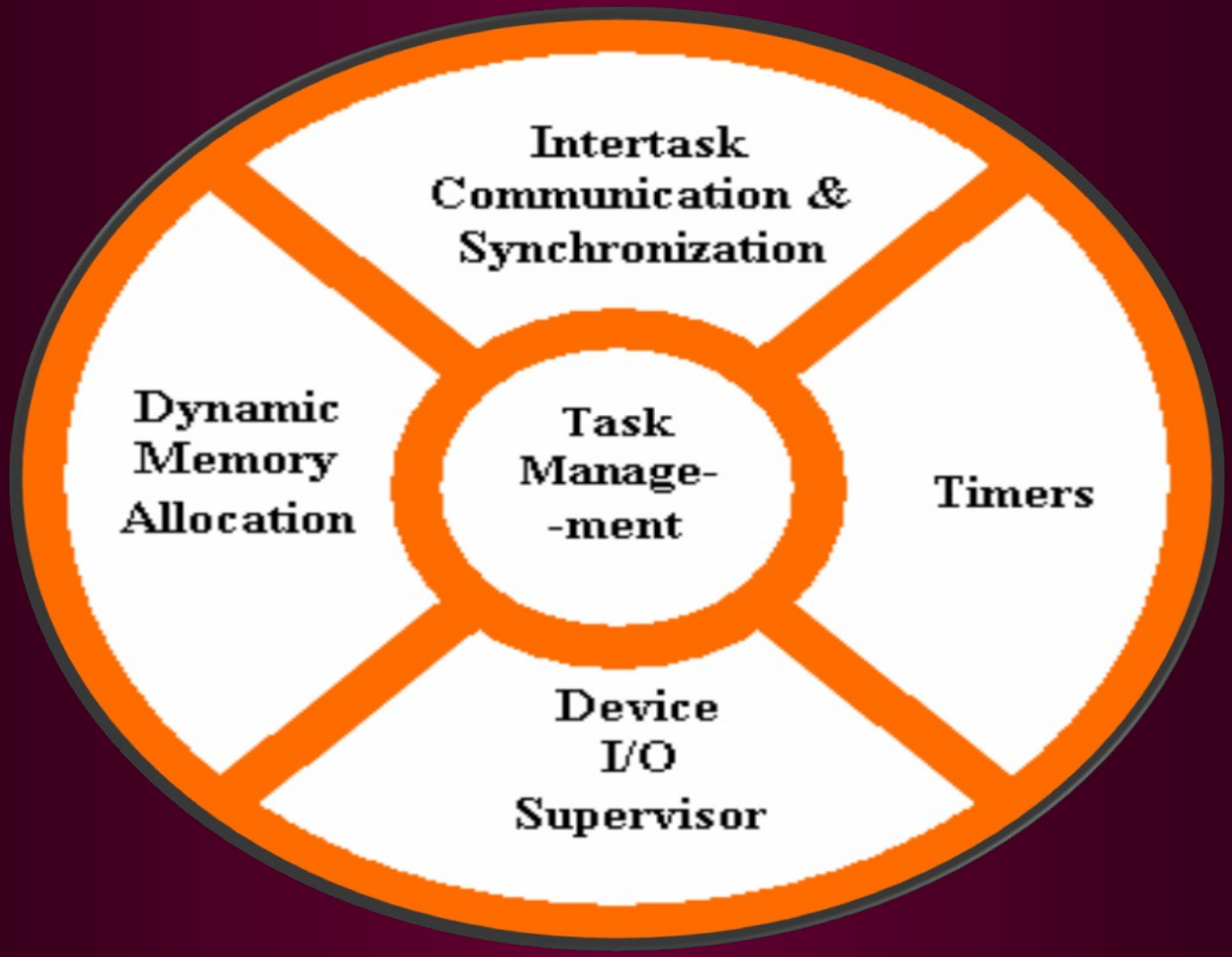

value(finishing time)
firm
d_i
finishing time


value(finishing time)
hard
d_i
finishing time

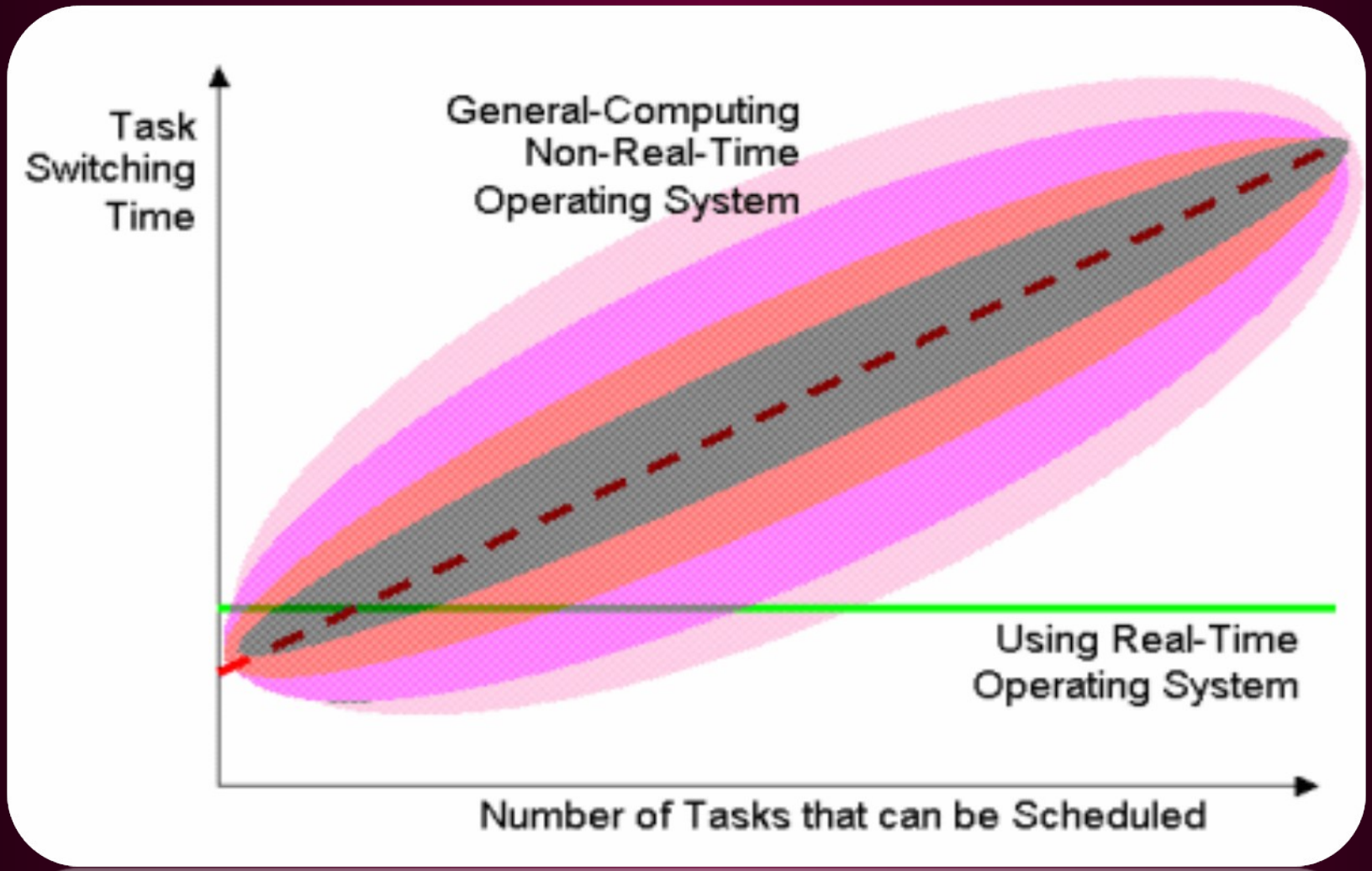
RTOS Kernel



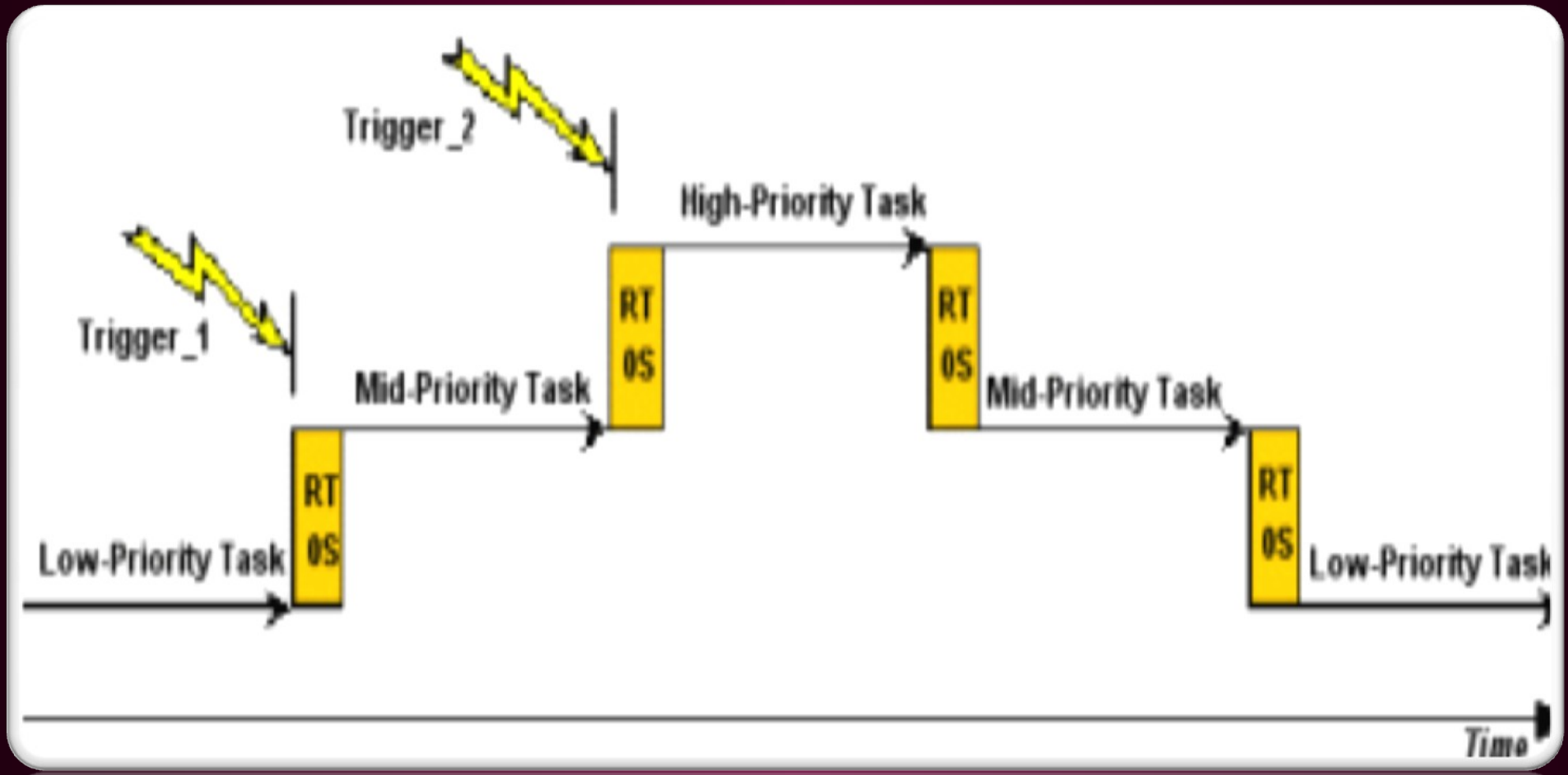
RTOS Kernel Functions



Task switching Timing

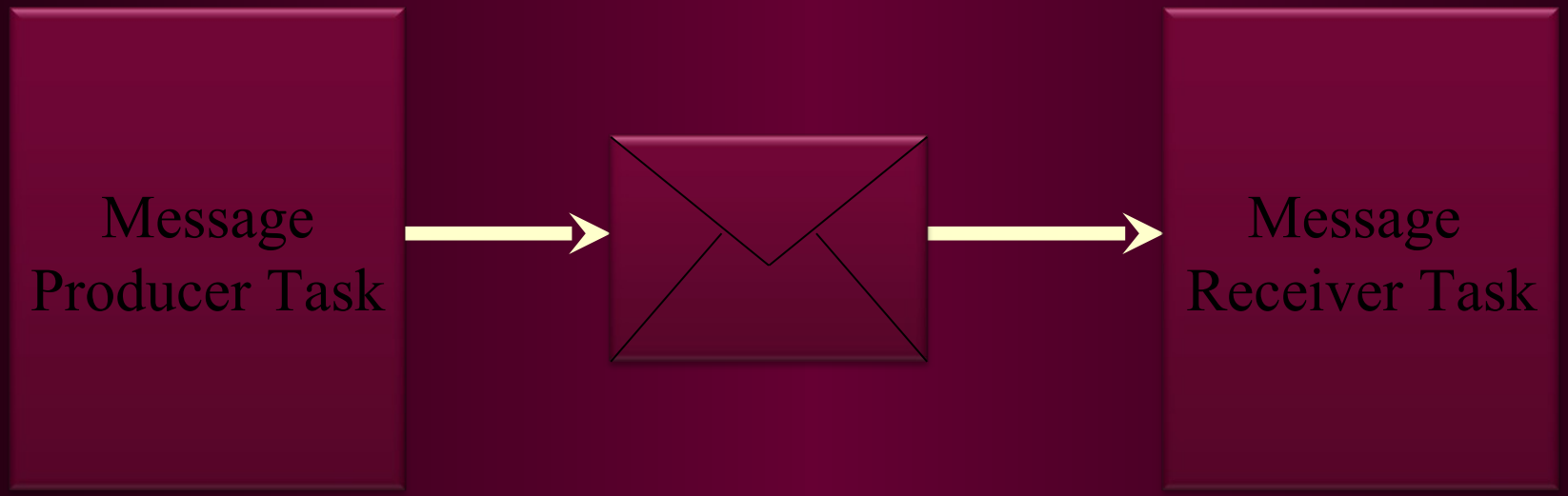


Task Scheduling



- Most RTOSs do their scheduling of tasks using a scheme called “priority-based preemptive scheduling”. Each task in a software application must be assigned a priority, with higher priority values representing the need for quicker responsiveness. Very quick responsiveness is made possible by the “preemptive” nature of the task scheduling. “Preemptive” means that the scheduler is allowed to stop any task at any point in its execution, if it determines that another task needs to run immediately.
- The basic rule that governs priority-based preemptive scheduling is that at every moment in time, “The Highest Priority Task that is Ready to Run, will be the Task that Must be Running”. In other words, if both a low-priority task and a higher-priority task are ready to run, the scheduler will allow the higher-priority task to run first. The low-priority task will only get to run after the higher-priority task has finished with its current work.

Inter Task communication & Synchronization



Dynamic memory allocation

Many (but not all) RTOS kernels provide Dynamic Memory Allocation services. This category of services allows tasks to “borrow” chunks of RAM memory for temporary use in application software. Often these chunks of memory are then passed from task to task, as a means of quickly communicating large amounts of data between tasks. Some very small RTOS kernels that are intended for tightly memory-limited environments, do not offer Dynamic Memory Allocation services.

Difference b/w RTOS and GPOS

RTOS	GPOS
Optimize worst case	Optimize average case
Predictable schedule	Efficient schedule
Simple executive	Wide range of service
Minimize latency	Maximize throughput

Where can Real time system be found?

RT-systems are everywhere:

1. Plant control
2. Automotive application
3. Flight control system
4. Environmental acquision and monitoring
5. Robotics
6. Military system
7. Space missions
8. Household application
9. Virtual/ augmented reality
10. Telecommunication systems ……etc

Applications

1. Hard RT application:
   automotive: power train control, air bag control, steer by wire, brake by wire
   aircraft: engine control, aerodynamic control

2. Firm RT application:
   weather forecast, decisions on stock exchange orders

3. Soft RT application:
   communication systems(voice over IPI)
   user interaction
   comfort electronics(body electronics in cars)

~~Real-Time = Fast~~

RT = PREDICTABLE

SOFT

STATIC

DYNAMIC

HARD

Conclusion

This presentation has addressed the past and present design techniques for RTOS but future design tend to be moving some tasks that usually reside slowly on the microprocessor to field programmable gate array (EPGA) device.

Thank you