Real-Time Operating System

Tanushree Das
Department of Electrical and Computer Engineering
University of Western Ontario
London, Canada
tdas4@uwo.ca

***Abstract*—The technological world surrounding us primarily depend on the Embedded Systems for its functionalities. The applications of Embedded Systems can be found from consumer electronics and military applications to NASA's crucial projects. These Embedded System often have to perform a required operation within a set deadline, therefore Embedded Operating System is quoted as the Real-Time Operating System (RTOS). This paper attempts to explore the functional details of Real-time Operating Systems, its types, crucial features and discuss its application in Embedded System Technology.**

Keywords—Embedded System, Real-time Operating System (RTOS), VxWorks.

I. INTRODUCTION

Embedded System is defined as a combination of software-hardware that is designed to perform a particular functionality and can be a part of a larger system [1]. The applications of Embedded System can be seen everywhere in our day-to day life such as microwave oven, traffic controllers to advanced military defence mechanism and space exploration robots. Embedded System are required to perform a specified task within a stipulated time and therefore the general purpose operating system are not suitable for embedded systems as they do not guaranty a deadline for the task; therefore, embedded system utilise Real-time operating system(RTOS) for its functioning and management. The rest of the paper is organized as follows: Section II provides the introduction to RTOS, its type, architecture and underlying functionality. Section III provides an insight into 'VxWorks', one of the most marketable and efficient RTOS used in the industry. In the end, section IV states the conclusion drawn from the research work.

II. REAL-TIME OPERATING SYSTEM

This section provides description of RTOS, RTOS types and architecture.

A. What is RTOS?

A **real-time operating system** (**RTOS**) is an operating system (OS) that has been designed to serve and support real-time applications or process real time data as it comes in, generally without any buffer delays. Major RTOS used in the industry are freeRTOS, Microsoft's Windows CE and VxWorks [2].
The key features associated with RTOS are [3]:

1) *Deterministic:* An RTOS guarantees task completion within a set target deadline making it reliable.
2) *Responsiveness:* RTOS offers minimum interrupt and thread switch latency making it fast.
3) *Priority:* Tasks must be able to be given a priority. So that the critical tasks with high priority can be executed earlier by the processor.
4) *Scheduling:* An RTOS should follow some scheduling mechanism so as to guarantee response-time.

B. Types of RTOS

Real-time System is broadly divided into two categories as listed below:

1) *Hard RTOS:* Hard RTOS is used to support systems in which the task has to be definitely completed within the specified deadline, that is failure to deadline achievement is not acceptable as it will result in complete system failure or catastrophic effect. Example of Hard RTOS is VxWorks that is used to support system such as satellite launch, air-traffic controller [3].
2) *Soft RTOS:* Soft RTOS is used to support system in which meeting the deadline in not necessary for all the task, that is, in case if a deadline is missed occasionally, it will not result in complete system failure. Example of such system are video and audio streaming [4].

As we see from Figure 1, the hard RTOS guarantees meeting deadline even in worst case scenario.

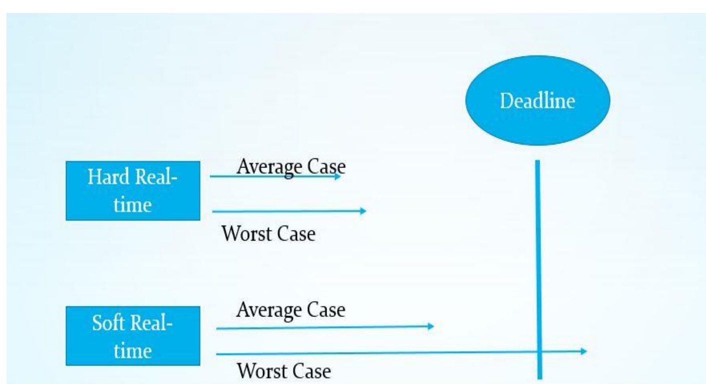

Figure 1: Types of RTOS [5].

C. Architecture of RTOS

Figure 2 below, provides an architectural view of RTOS.As we can see from the figure RTOS Kernel and Board Support Package (BSP) is sandwiched between the Application Layer and the Hardware Layer.

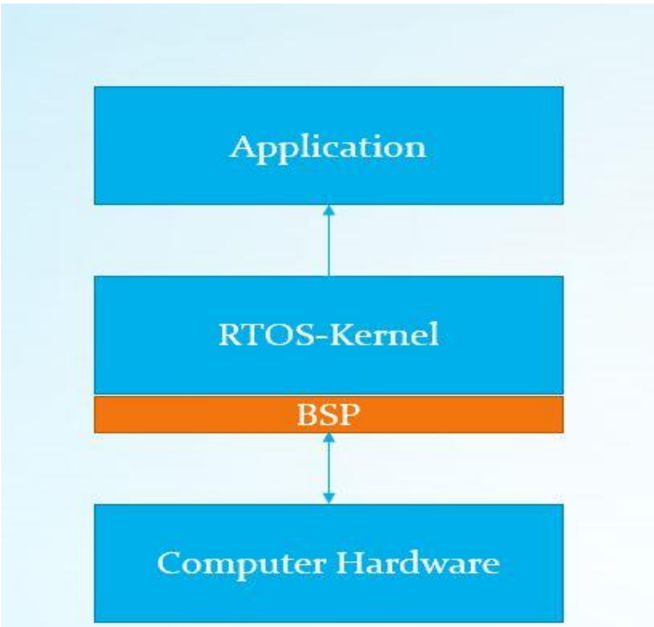

Figure 2: RTOS Architecture [6].

RTOS kernel can be of type Monolithic or Microkernel. In monolithic kernel (e.g. Unix, Linux) the entire kernel processes and activities run in the single address space; however, in microkernel (Mac OS, Windows NT) the kernel activities are split into servers executing in separate address space [7]. BSP provides the interface and APIs for ensuring the compatibility between the RTOS and native hardware [8].

The major functionalities associated with RTOS kernel are explained below:

1) Task Management: Tasks are light-weight threads in sequential execution and which can communicate with other tasks. RTOS task model is depicted below in Figure 3. As we can see every task has an initial phase followed by an execution phase. The task needs to be completed within the specified deadline [9].

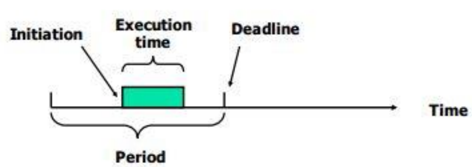

Figure 3: RTOS Task Execution [9].

2) Memory Management: Though dynamic memory allocation can be configured in RTOS; memory allocation is generally specified statically at compile time to avoid memory leaks or fragmentation issues and increase the context-switch or interrupt latency [10]. Virtual Memory is not supported in RTOS.

3) Scheduling: RTOS are designed with about 32 to 256 different priority levels for tasks/processes. At any time, the scheduler runs the task with the highest priority. This type of scheduling is referred to as priority based scheduling or preemptive scheduling [11].

4) Inter-process Communication (IPC): RTOS kernel handles IPC through various standard mechanism namely Semaphore, Shared variables, Message sharing and Mailboxes . The IPC resolves and prevents deadlock conditions.

5) Interrupt Handling: Interrupts have the highest priority in RTOS. Figure 4 below explains the working of Interrupt mechanism, as inferred as an interrupt occurs, first context-switch takes place (saving the task state of current running task) and an interrupt routine runs to execute and finally terminate the interrupt; thereafter normal execution is resumed [12].

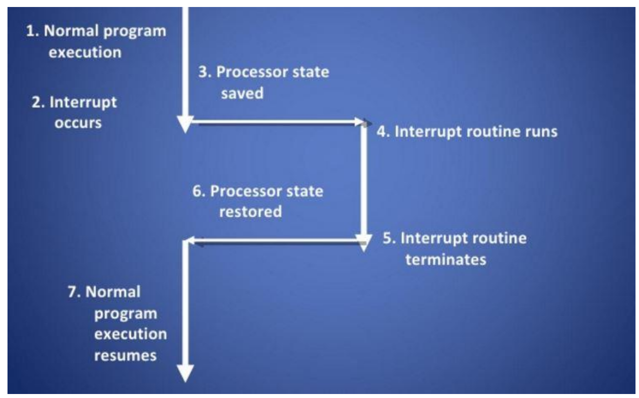

Figure 4: Interrupt handling in RTOS [13].

III. VXWORKS

In this section we explore the characteristics of Wind River's VxWorks; one of the pioneer and majorly used RTOS in Embedded Systems applications.

A. What is VxWorks?

VxWorks is a proprietary RTOS developed by Wind River Classification, CA. Figure 5 depicts the architecture of VxWorks RTOS.

The key features of VxWorks are listed below [14]:

1) VxWorks has a multitasking kernel and supports preemtive and round-robin scheduling that provided fast interrupt response.

2) The latest version VxWorks 7 released in March 2014, has microkernel with 20KB RTOS size.

3) Extremely secure, SHA-256 is used for password hashing algorithm.

4) Connectivity with L2CAP, SocketCAN, Bluetooth, IEEE Firewire 1394.

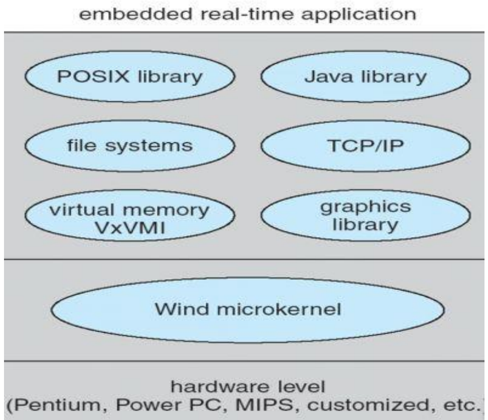

Figure 5: VxWorks Architecture [3].

C. Application in Embedded System

As per the 2014 Embedded Market Study, VxWorks is one of the leading RTOS for embedded applications [2]. There are numerous successful projects in the industry that have harnessed the capabilities of VxWorks. NASA's "Curiosity Rover" (Figure 6) also functions on the VxWorks for its underlying computing techniques. During Mars exploration, VxWorks not only ran tasks for Curiosity Rover navigation through space but also ensured its successful landing on the surface; after which the applications were modified for surface driving and other scientific experimental activities [15].

Figure 6: VxWorks in Embedded System (Mars Rover) [8].

IV. CONCLUSION

The rising amount of real-time data and applications with demand for their faster and reliable computation has increased the necessity for implementing Real Time Operating Systems. The real-time operating systems with its core functionality and management ensures the feasibility of complex system along with the dimension of predictability and reliability which cannot be achieved in a General Purpose Operating System. However, the RTOS are designed by keeping in mind the guarantee of task completion within a deadline (hard or soft as per demand) and not with the goals of increasing throughput.

V. REFERENCES

[1] Margaret Rouse, "*embedded system",* 2009. [Online]. Available:http://internetofthingsagenda.techtarget.com/definition/embedded-system. Accessed: June 24, 2016.

[2] Real-time operating system, in Wikipedia, Wikimedia Foundation, 2016. [Online]. Available: https://en.wikipedia.org/wiki/Real-time_operating_system. Accessed: June 28, 2016.

[3] Real-time Operating Systems (RTOS) 101, [Online], Available: https://www.nasa.gov/sites/default/files/482489main_4100_-_RTOS_101.pdf. Accessed: June 24, 2016.

[4] Real Time Operating System (RTOS), [Online], Available: http://www.electronicshub.org/real-time-operating-system-rtos/. Accessed: June 24, 2016.

[5] Do I need a Real-Time System, [Online], Available: http://www.ni.com/white-paper/14238/en/. Accessed: June 24, 2016.

[6] Real Time Operating System, [Online], Available: http://www.slideshare.net/HimanshuGhetia/my-ppt-14984168. Accessed: June 25, 2016.

[7] What is the difference between monolithic and micro kernel? , [Online], Available: http://stackoverflow.com/questions/4537850/what-is-difference-between-monolithic-and-micro-kernel. Accessed: June 25, 2016.

[8] FAQ : What is a Board Support Package, [Online], Available: http://www.windriver.com/products/bsp_web/what_is_a_bsp.pdf. Accessed: June 25, 2016.

[9] RTOS, [Online], Available: http://user.it.uu.se/~yi/courses/rts/dvp-rts-08/notes/RTOS.pdf . Accessed: June 25, 2016.

[10] Real Time Operating Systems , "*Chapter-5 Memory Management Functions",*[Online], Available : http://www.dauniv.ac.in/downloads/EmbsysRevEd_PPTs/Chap_8Lesson05EmsysNewMemoryMgt.pdf. Accessed: June 25, 2016.

[11] "How do real time operating systems work?" 2009. [Online]. Available: https://stackoverflow.com/questions/536506/how-do-real-time-operating-systems-work. Accessed: June 28, 2016.

[12] RTOS, [Online], Available : http://www.inf.ed.ac.uk/teaching/courses/es/PDFs/RTOS.pdf. Accessed: June 28, 2016.

[13] Real Time Operating System, "handling Interrrupts" 2009. [Online]. Available: http://www.slideshare.net/agarwalavanish/rtos-by-avanish-agarwal. Accessed: June 28, 2016.

[14] VxWorks, [Online], Available : https://en.wikipedia.org/wiki/VxWorks . Accessed: June 28,2016.

[15] Mars Science Laboratory, [Online], Available: https://en.wikipedia.org/wiki/Mars_Science_Laboratory Accessed: June 28,2016.