Review on Real-time operating system (RTOS)

Aman Gupta[1], Pulkit Jain[2]
[1]Student, BE
[2,]Assistant Professor,
Chandigarh University, Mohali, Punjab, India
gupta4327@gmail.com[1], pulkitjaindav@gmail.com[2]

Abstract

In almost every embedded system which works in the dynamic environment there is a need of RTOS (Real time operating system). In this type of environment to get an desired level output or performance we have to use RTOS. RTOS provides the time constrains output so as to complete the task before the deadline. As a little time delay in an event may lead to an undesirable effects and losses. This paper gives you a general over view on what RTOS is and what are the various parameters that effects RTOS functioning, improves the performance of RTOS and also briefly tells you about the RTOSs available in market.

Keyword: *RTOS; Real time embedded system (RTES); kernel; OS; time constraints.*

Introduction

A real time embedded approach are the one in which the result not only depends on logicality of the system but also on the time at which result have been generated basically a response time of a system. In RTES a little time error or delay in output may lead to the serious losses or may lead to any undesirable output from the system [8]. Considering the simple example of an RTES is the air bags control in cars, just a slight time hold up in opening an air bag may lead to an accident .Here when RTOS plays a major role by providing a time bounded outputs. An RTOS is just like a coherence that forms between an program application and an system hardware. The purpose of RTOS is to complete a specific task in a limited or bounded time and this also forms a key difference between a normalized operating system and RTOS [9]. In our regularly usable operating system there are no time constraints. Another considerable point is that meeting the deadline not completely depends on RTOS. It depends on overall design and structure as well.

RTOS basically help in task scheduling, managing resources, precise timing, communications, I/Os and synchronization. RTOSs have been evolved from using or building it for particularly specialized applications to the general purpose variants. RTOSs are available in many variants in market nowadays varying from commercial, open source to proprietary. Some of them are VXworks, Real time linux, Real time unix, pebble, salvo, EmbOs, etc.

Time management feature of RTOS allowed the designers to attain task delay, time triggering functions and many more time dependent processes without diving deep into their hardware functionalities like timers. For the small systems that don't use RTOS it becomes little tricky for the designers to attain time relying features as they need to have a complete understanding of peripherals (timers) and should also be able to link it with application code.

Literature Review

RTOS is used to generate likely response to an unlikely or unpredictable event. Let's look briefly about various variants of RTOS available nowadays from proprietary kernels to real time version of a commercially available operating system such as RT-Unix, RT-Linux, etc, research kernels, component root kernels, etc.

Tiny, Speedy, Proprietary kernels [6]

This type of RTOS comes in two ranges: one is homegrown and other is commercial. Homegrown kernels are unique to the application and have a swift and highly probable behavior. But in recent times due to the growth of commercial offerings as well as high cost and maintenance of homegrown kernels, there is minimization in growth of homegrown kernels. These type of kernels have many features such as multi-tasking, context switching, prior preemptive scheduling. They are suitable for small applications as in large systems its an hardship to met up with all time constraints by priority scheduling.

Real time versions of commercial operating systems [6]

They are basically an time optimized extensions of available operating systems such as of linux, unix, posix and named as RT-Linux, RT-Unix and RT-Posix. They provide us a better software development environment and have a various system APIs which makes it efficient and user friendly.

Research Kernels [6]

There are many universities that have developed research based kernels. They have been build for addressing research issues such as developing new paradigms, real time models, etc. Various kernels that comes under it are MARS, ARTS, SPRING, HARTOS, etc

Some other types of kernel are also available there including component based kernels and QoS kernels

Parameters that describe the real time implementation of RTOS

- **Time administration and its impact on performance [7]**
  Time administration can offer help for real time reaction of the system to guarantee it is running progressively and precision. In each OS, a clock is intended to create intermittent clock interupcy signals and at whatever point a clock interupcy arrives, the counter esteem is increased by 1. The time interim between two clock interupcy is the littlest planning unit of the OS clock, called the clock granularity. The granularity size of the clock decides the time exactness of the working framework and the reaction speed to occasions.

- **Task Scheduling and its effect [7]**
  OS performs various operations such as creation of tasks, its deletion, restoration, scheduling, suspension. Task scheduling is the process of finding when the task acquires system assets after allotting it a set of RT task and system assets. It is a need to manage system resources precisely in

order to acquire a better real time execution. It is must to execute scheduling functions at scheduling focuses such as when ISR is over, task is set, when task is on wait, etc

- **Interrupt latency and its effect on performance [7]**
  Interrupt is basically an asynchronous notification to the CPU. Whenever there is an interrupt CPU performs context switching and reaches to a subroutine address called as to be an Interrupt service routine (ISR). When an ISR task is completed it comes back to an interrupted task or say the task which have higher level of priority. Interrupt latency is when a CPU takes an time to read the interrupt and this is one of the main factors that effects the real time execution. Interrupt latency may occur due to the following reasons: 1) When system is processing a critical part of code interrupt needs to be disable. 2) When a CPU is dealing with an interrupt of higher priority it ignores the current low priority interrupt for a while. 3) It will take a plenty of time from reading an interrupt to reaching an ISR.

- **Communication as well as synchronization and their impacts [7]**
  When there is multi-tasking real time execution there may be a possibility or need of exchanging of data between two ongoing tasks or may be between interrupt and event (task). In order to have a secure access of information between ongoing tasks real time system provides various communication and synchronizations procedures such as semaphores, mail boxes, global variables, etc. There are many cases where shared data is allowed to be used by one task at a moment. Thus, sharing the resources for multi tasking system is one of the core parameter effecting real time execution. Semaphore is one of the mainly used share resource manager.

- **Memory administrations and its impact [7]**
  Memory holds code and information that the CPU can straightforwardly get to and are an imperative piece of RT systems. In request to guarantee ongoing and dependability of the working system, effective and complex memory administration ways are an essential piece of a real time OS. In this way, memory administration strategies utilized in the RT systems are exceptionally straightforward. Also, the memory administration is one of the major parts of RTS portion, yet not the main consideration influencing the execution of the system.

Conclusion

RTOS is gaining popularity with increase in real time applications such as smart watches. RTOS is different from a general OS in many ways such as for generating predicable response for unpredictable tasks [8], by its scheduling mechanism, in between tasks communications and many others. We have different variants of RTOS in the market and can choose any of them according to our need .There are various factors that optimize the real time performance of operating system from its scheduling capabilities to their memory managing ways.

References

1. Jerry Krasner, “RTOS Selection and Its Impact on Enhancing Time-To-Market and On-Time Design Outcomes”, **2007**

2. DONG Jia-liang, LI Yan-feng, YANG Qiu-song, “Real time performance evaluation of embedded operating system in aerospace”. Computer Engineering and Design, 34 (1), **2013**.

3. R. P. Kar, K. Porter. Rhealstone, “A Real Time Benchmarking Proposal [J]”. Dr. Dobb's Journal, **1989**.

4. WU Xun, MA Yuan, DONG Qin-peng, “Research on Real time Operating System Performance Measurement”. Journal of System Simulation. 25(2), **2013**.

5. Tran Nguyen Bao Anh, “A real-time operating system for 16-bit microcontroller”, MSc dissertation, School of Computer Engineering, Nanyang Technological University, Singapore, May **2008**.

6. John A. Stankovic and R. Rajkumar, “Real time operating system”, Kluwer Acedmic Publishers, **2004**.

7. S. Tan and Tran Nguyen Bao Anh, "Real-time operating system (RTOS) for small (16-bit) microcontroller," *2009 IEEE 13th International Symposium on Consumer Electronics*, Kyoto, 2009,pp, 1007-1011 date of issue: 10.11.09/ISCE.**2009**.

8. Ming Huang, shujie guo and Xu Liang, “Research on real time performance testing methods for robot operating system”, Journal of software, vol. 9, no. 10, october **2014**.

9. Nelson Weiderman. Hartstone: synthetic benchmark requirements for hard real-time applications. ACM SIGAda Ada Letters,(3): 126-136, **1990**.

10. Yang Heng, Qin Pinjian, Huang Liansheng, “Real- time Performance Test of Poloidal Field Power Supply Operation System. Computer Measurement &Control”,2730-2732, **2010.**