Understanding Real Time OS Concepts

NisarAhmed
Case Islamabad Pakistan.
Email: lalanisarahmed@hotmail.com

Abstract— Real-time operating systems have conquered the world of computing in a very short span. It is detailed system software. The word 'real time' does not refer that system reacts rapidly, it denotes that there are solid and set time requirements that must be conformed at the requisite situation. If these time requirements are not completed, the results will be mistaken or untrustworthy. The basics of Real Time Operating Systems (RTOS) will be argued in this analysis. What a real-time operating system (RTOS) is, how RTOS are used for computation and manage applications, and how they changed from general idea operating systems like Windows.

Keywords: Real Time Operating Systems (RTOS), General Purpose Operating System (GPOS)

1 INTRODUCTION

Usually standard operating system is designed for controlling and managing hardware resources of computer and application that run on the computer. On the other hand a real time operating system is designed to run and process applications with exact and absolute timing with high level of reliability. This is extremely important in computing and for automation System where response time is very much costly and the program delay could cause a severe damage to the life. Real-time operating systems respond to input quickly. *Real time* can also refer to events simulated by a computer at the same speed that they would occur in real life. Operating system that guaranteed maximum measured calculated output and have maximum time for each of the critical operations called hard real time and the operating System that slightly deviates its specific time is referred as Soft real time. Most general operating systems are not considered as real-time because there reaction time is varied from seconds to minutes. Real time operating systems are designed for real-time applications which produce exact, strict and rigid .

conclusion within the time frame the target result is achieved. The fundamental structure is quiet similar to general operating system with inclusion of preemptive scheduling. Real time operating systems cannot increase or decrease the execution speed. They produce only specific and confined timing features.

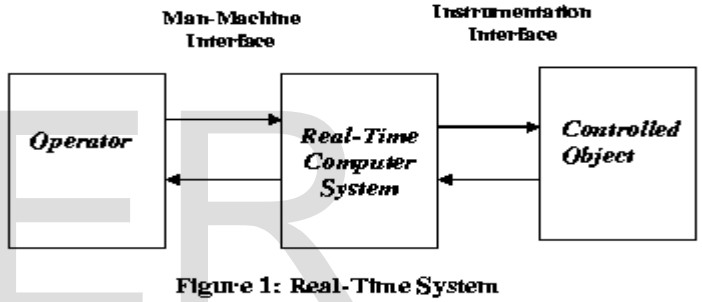


Figure 1: Real-Time System

1.1. Hard Real Time System

To understand these concepts, let us consider an example .Suppose we are designing an air bag system for a car. In this case a small delusion in timing causing the airbag to open too early or too late could be disastrous and cause harmful injury therefore a hard real-time system is necessary. So as a designer you need to assure that not a single operation will deviate from a certain time limit.

1.2. Soft real time system

AT machine (Auto Teller machine) is an electronic device whose goal is to operate in real time. When, you start a transaction it might wait to respond from seconds to minutes except from fraction of a second normally, nothing catastrophic will happen. You may be irritated but you'll consider something important is going on back of transaction. This type of device is an example of a "soft real-time" embedded device. The term "soft" indicates that there is some flexibility in the real-time requirement. RTOS can assure that a program will run with very absolute timing. Make sure that important deadlines are met.

2. Distinction between Hard Real-Time and Soft Real-Time

2.1. Hard real time system

- *Nisar Ahmed (BCS,MCS,MBA,Mphil Management)*
- *Lecturer TIPS college Faisalabad Pakistan.*
- CASE Islamabad Pakistan.
- *Email=lalanisarahmed@hotmail.com*
- *00923008666983*

- Reaction time in milliseconds.
- Predictable performance
- Predefined deadlines
- Result will be havoc if requirement not met.
- Accurate with any type of hard cases.
- Minor database with small data files.
- Error recovery limited.

2.2. Soft real time system

- Reaction time is higher
- Result will not catastrophic
- Degraded performance may occur.
- Peak load time can be neglected.
- Accuracy varies with high load.
- Larger database and required long time integrity.
- Roll back to previous phase is possible.

3. Features and Terminology

3.1. Reliability

Rtos are more reliable than any other operating system. The word reliability explains that the system will run continuously and consistently provide service and does not fail. The downtime per year is negligible.

3.2. Predictability

Predictability stand for that the completion of task should be done within the known timeframes and ambiguity in the response time should be smaller and negligible.

3.3. Compactness

By limiting the size of application within the circle of system and user requirement .the designers must be well aware of static and dynamic memory utilization of the RTOS and the application that will run on it small in size handle very easily and portable in nature.

3.4. Scalability

Real time operating system used in wide variety. There scope is wide any module can be deleted and added to any specific real time system. As result they save money and cost.

3.5. Performance

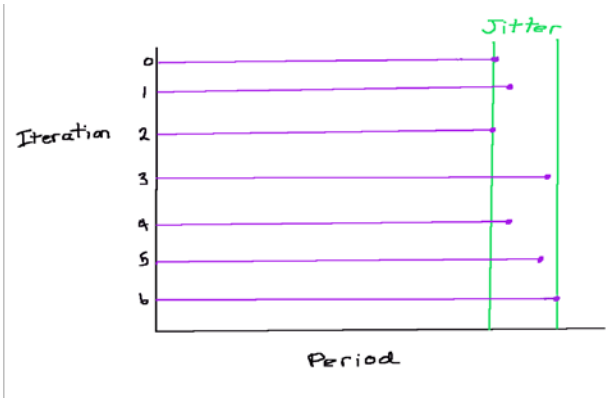
Rtos performed better than general operating system. It is fast enough to fulfill the demand.

3.6. Determinism

The main contradiction between standard operating systems and real-time operating systems is the calculation. A real-time Operating System is referred to as deterministic if its timing can be assured within a described margin of error. Operating System consumes known and calculated degree of time.

3.7. Jitter

Small uncertainty between subsequent time periods for an assigned task real time system must be designed to reduce the level of uncertainty or jitter so application will be fluent and acceptable. Increase the speed can reduce the jitter. Jitter is a variable deviation from proposed timing. Scheduling jitter is the delay between the times when task shall be started, for example consider a task should start after 10ms, but some reason, in started after 15ms. So the jitter is 5ms!



4. Comparison between RTOS and General Purpose OS

The basic distinction between gpos and rtos based upon the whether the system is time bounded or not. Mainly gpos used in those system which are not time punctual like windows,linux,unix etc.rtos are able to operate without any control from another system they are designed for limited boundary and ultimate state at terminal point. on the other hand gpos are economical, prudent use of resources, operating on a limited scale, little in size, objectively measurable aspect. A GPOS is made for high end, general purpose systems like a personal computer, a work station, a server system. Gpos are designed for superior end, they are exclusive and up market price like personnel computer, workstation and servers. The low end and high end differentiated on the basis of hardware configuration, functional arrangements of parts and components. Mainly rtos depend upon the low hardware specification and gpos consume much hardware utilizations and a very high upward extension. Another dilemma in gpos is uncontrolled latency. The speed to carry, execute and finishing task is varied. All are categorized as unbounded latency. On the other hand rtos has no such problem because all the tasks are bounded within a specified time limit.

5. Issues in real time systems

5.1. Real time acknowledgement

The real time system must response to the external interference with in proposed time limit. Successful completion of task depends upon the exact and reliable operation of the system.

5.2. Architecture design

Architecture must be designed within the scope of given requirement. Every requirement must be fulfilled with in design. Design should be compatible with all Processes, requirements and the architecture. It must be reliable in any physical condition.

5.3. Speed optimization

The speed of the system must full filled general criteria. Speed must constant within all performed tasks. Latency in speed cause disastrous results.

5.4. Processing components

The hardware must be fast and efficient but should not be crossing its execution power. Sometimes it happens that high priority task overcome the low priority task and stopping low priority task to execute it may lead to fatal result.

5.5. Operating system

All above issues directly respond to operating system. Because operating system is a core part of this whole scenario. All hardware management, processing control and speed optimization

depends upon the operating systems. We have to select an operating system with such features, perceptible measurement, interrupt latency low and scheduling is high.

5.6. Failure recovery

Real time system must be bound to recover from any physical and hardware failure. Many software contradicted conditions lead to a task that directly punch a processor exception. In this critical scenario the task must rollback to its previous physical state.

5.7. Hardware failure

Sometimes hardware cannot bear the outer pressure and environmental effects. Suppose when a processor fails, other processors should alter about the failure. So the processors will then abort any interactions with the failed processor node. Real-time systems must be able to recover from hardware failures. The system should be able to detect and recover from board failures. The system must inform the operator about it and the system should be able to switch to extra board**.** Usually communication in real systems done with the help of nodes if communicated link failed so the traffic must deflected to other link node without any dispersion to make communication uninterrupted**.**

5.8. Distributed environment

Some real time systems work on a distributed terminology's the load is distributed among several processor. One of the biggest headaches in Real-time systems is defining and maintaining message interfaces. Designing of interfaces is complex because every day technology is updated. It must be essential that our interface must be supported by any backward or older interface. It also important that one protocol must be supported by other protocol on the other hand load distribution is another problem the load of the task must be distributed equally among all the processors.

5.9. Non parallelism

Real time operating system communication based on asynchronous terminology. Usually Real-time systems support state machine based design .so the message passing or rpc (remote procedural call) limited.

6. Advantages of real time operating system

Real time operating system utilized its maximum resources as a result peak output is taken by keeping all its components active phase. Switching one task to other consumes a lot time but with advancement and up gradation of the system it is been reduced and adjusted to fixed proportion. Real times operating systems emphasize on current running process neglecting all waiting task to produce exact results. Real time operating system are less prone to error for performing task, they are bug free real time operating system incomparable with other general system in memory management they most supreme and ideal in memory management.

7. Disadvantages of real time operating system

Real time operating system is narrow. They are restricted to particular task. They are tightly coupled to perform specific assignments. They utilize massive resources. They lack ability to do more than one task at a time as result they not support multitasking environment real time operating system users multiplex and complicated algorithm which not easy to analyze and understand. RTOS require peculiar device drivers and interrupt signals to react quickly to interrupts. Transfer of task to the other usually rare so switching is less supported. Low priority task does may not get time because concentration fixed on a running task. Effective program is required to accomplish the task.

8. Real time application

- **Automatic Electric Iron**: heating components depends upon outer temperature.
- **Servo Voltage Stabilizer**: based upon output voltage
- **Water Level Controller**: consumption is regulated by the level of water in the tank
- **Missile Launched & Auto Tracked by Radar**: position of the target controlled by the directions.
- **An Air Conditioner:** Functioning and performance depends upon the room temperature.
- **Cooling System in Car:** Based on the inner temperature of a car.
- **A pacemaker:** It is used check electrical impulses at regular intervals.
- **The Anti-lock Braking System (ABS) controller in a car:** Correctly managing timing interval of the break. Minor flaw lead to seize the break.

Acknowledgement

Foremost, I would like to express my sincere gratitude to my advisor Sir Zahid Javed for the continuous support of my study and research, for his patience, motivation, enthusiasm, and immense knowledge. His guidance helped me in all the time of research and writing of this paper. I could not have imagined having a better advisor and mentor for my study. Last but not the least; I would like to thank my family: my parents, for giving birth to me at the first place and supporting me spiritually throughout my life.

Result

The fast expansion of real time system led to development of plan of time sharing systems. The development in this meticulous area led the researcher to enlarge and broader their mind. The progress has made researcher to decrease bugs and forgery in real time operating system and them more trustworthy, competent and better than any other system to discover different real-time scheduling algorithms, and to plan competent resource provision mechanisms for different system resources. The objective of this research is to present the basic design problems in supporting real-time abilities in a common purpose operating system. We review representative works in this area and present an integrated perspective for competent multi-resource provision and scheduling.

Referrences

1.Author(s).“Articletitle”.*Journaltitle*,vol.,pp,date.

Example:

2.Tanenbaum, Andrew (2008). *Modern Operating Systems*. Upper Saddle River, NJ: Pearson/Prentice Hall. p. 160. ISBN 978-0-13-600663-3.

3.József Kopják, Dr. János Kovács,New Trends in Real Time Operating Systems”, Kandó Kálmán Faculty of Electrical Engineering Óbuda University, Budapest, Hungary

4.Ke Yu,” Real-Time Operating System, Modelling and Simulation Using System “Submitted for the degree of Doctor of Philosophy Department of Computer Science ,June 2010

5.Andreas Gerstlauer Haobo Yu Daniel D. Gajski,”RTOS Modeling for System Level Design Center for Embedded Computer Systems University of California, Irvine Irvine, CA 92697, USA

6.Ramesh Yerraballi,,”Real-Time Operating Systems: An Ongoing Reviewed. of Computer Science and Engineering University of Texas at Arlington

7.Michael B. Jones, Paul J. Leach,Richard P. Draves, Joseph S. Barrera,”III Modular Real-Time Resource Management in the Rialto Operating System” May, 1995Technical Report

8.M.V. Panduranaga RaoA,”Research in Real Time Scheduling Policy for Embedded System Domain” *research Scholar, Department of Computer Engineering, National Institute of Technology Karnataka, Surathkal, India-575 025.*