IJCRT.ORG ISSN : 2320-2882

INTERNATIONAL JOURNAL OF CREATIVE RESEARCH THOUGHTS (IJCRT)

An International Open Access, Peer-reviewed, Refereed Journal

A COMPREHENSIVE REVIEW ON FUNCTIONAL ANALYSIS OF REAL - TIME OPERATING SYSTEMS

[1]S Ayyappan, [2]M Ameena Banu, [3]S M Vijayarajan and [4]A Gopi Saminathan

[1]Assistant Professor, [2]Professor, [3]Associate Professor, [4]Professor

[1] Department of ECE,

[1]NPR College of Engineering and Technology, Dindigul, Tamilnadu, India

Abstract: A real-time operating system (RTOS) is a system made to help us in real-time using the real-time applications which provide no delay. A RTOS is time-bound, which means that the time is the main part of the system, as in the system will fail if it goes beyond the prescribed time. Processing the time requirements in this system are measured in tenths of seconds. The integrity of a RTOS system established not only takes the logical effects of computation taken into account, but also the amount of time required to get such results. It must prioritise vital jobs while keeping context switching time to a minimal.

Index Terms **- Real-Time Operating Systems, Hard RTOS, Soft RTOS, GPOS.**

I. INTRODUCTION

A Real-Time Operating System (RTOS) is an operating system in which its main function is to complete a given job within those specified time limits. It is employed in systems when a process's ongoing operation is affected by a program's results. When an incident occurs that is not related to the computer, a sensor that is used to monitor the event sends the information to the computer. The operating system receives a signal from the sensor and interprets it as an interrupt. In order to handle an interrupt, the operating system will launch a certain process or group of processes. This process runs entirely uninterrupted unless a higher-priority interrupt occurs. The interruptions must therefore be prioritized in a particular manner. The process will only allow the highest-priority interrupt to begin, while lower-priority interrupts will be buffered and will be taken care afterwards. In such an operating system, interrupt management is critical. Students, freelancers, contractors, and anybody else who uses a phone, computer, or tablet should get familiar with the various operating systems so that they may select the machine and operating system that best meets their needs. If you want to study IT in school and pursue a career in IT, you'll need a thorough understanding of the operating system to provide you with all the skills you'll need to succeed. Special-purpose operating systems are employed because ordinary operating systems cannot provide the performance needed for real-time operating systems [1]. The RTOS MTS, Lynx, QNX, VxWorks etc., are some examples. If we look at RTOS, we can see that it is also an operating system that serves as an interface between the system's hardware and the user. File management, process management, memory management, and other operations are also handled by RTOS.

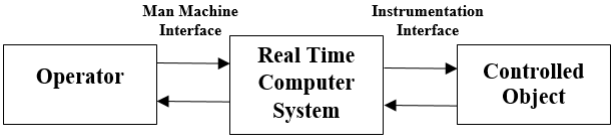

Figure 1: Real-Time System

A system is nothing more than a collection of peripherals linked together to process and output data. Real-time systems must be implemented quickly to process input data and provide output in a specified amount of time. Three different platforms for running applications in real-time are Hard, Soft and Firm Real-time Systems [2].

II. EASE OF USE

ERTOS will help us decrease the response time for many application where time is very important and this will make the applications more efficient.

III.TERMS USED IN RTOS

Task is a group of connected nodes that communicate together to run the system. A Job is a short task that can be assigned to a processor and which may or may not require resources. A job's release time is the time at which we can say that the task is ready to be executed. Job execution time is a point at which work finishes executing. A job's deadline is the deadline by which the job should be done. A job's response time is the interval between the release of a task and its completion. Absolute deadline is the deadline that should never be missed.

IV. APPLICATIONS OF RTOS

Applications that operate in real-time and adhere to dead- lines use RTOS. Some of the most common uses for real-time operating systems are listed below. The Radar device makes advantage of real-time operating system. Missile guidance employs real-time The Radar device makes advantage of real- time operating system. Missile guidance employs RTOS. RTOS are utilised in online stock trading. The switching device for mobile phones uses RTOS [3] .RTOS is used by air site visitors to change the structures. Real-time OS is used by medical imaging systems. The Fuel injection device employs real-time OS. The Traffic Control Device makes advantage of real-time OS [4].

V. COMPONENTS OF RTOS

The Scheduler: This RTOS component decides the priority in which the tasks should be done.
Symmetric Multiprocessing (SMP): This is used to do parallel processing i.e., the RTOS oversees and helps all the jobs to simultaneously process.
The Function Library: It is an interface of kernel and application program. The program uses function library to send requests to the Kernel, allowing the application to receive the desired results.
Memory Management: It allocates memory to each pro- gram in the RTOS which is an important job.
Latency-free dispatch: Latency-free dispatch is the amount of time a system takes to respond to a request for a process to begin operation without any delay.
User-defined data items and classes: The RTOS system uses programming languages like C and C++, which you will modify and execute depending on the desired function.

VI. TYPES OF RTOS

There are three types RTOS are listed below [5],

A. Hard RTOS

Hard RTOS requires that all crucial actions be completed within the allotted time limit or by the deadline. Failure to meet the deadline might have disastrous consequences, including damage to the equipment or even sacrifice of life. This type of operating systems make sure that key tasks are accomplished within a certain time frame with no delays allowed. A robot, for example, is used to observe a patient and inform us if there is any problem, but if it makes any error, the patient may be in danger. Therefore it is a RTOS that should be very specific in its time. They are used in critical applications where there is absolute deadline. The error management is done by itself in hard RTOS. In a hard RTOS system, timelines are seen as a deadline, and the deadline should never be neglected. Due to the lack of permanent memory in hard RTOS, processes must be done correctly the first time. In the time provided, the hard RTOS should give correct responses to events. A hard RTOS is a completely accurate and time-constrained system. If customers anticipate a result in 5 seconds for a given input, the system should process the data and provide the outcome in exactly 5 seconds. It should not provide the output before the 6th or 4th second. The deadline for completing the procedure for the provided data is over 5 seconds. Meeting the deadline is critical in a hard real-time system; if the deadline is not reached, the system's performance suffers.

B. Soft RTOS

In a soft RTOS, it is not required to finish the job before the deadlines every time but it is not recommended to miss the deadline. It doesn't have strict time limitations like the hard RTOS, and is acceptable to have a small delay but it has to complete its process and give the result. Even soft real- time systems should not miss the deadline for every job or process. The system should not consistently miss the deadline as the system's performance will suffer, and if the system's performance goes down then the users will not be able to use it. In the soft RTOS, the size of the data file can be large as opposite to the hard real-time system. They are more adaptable than the hard real -time systems. They are used in non-critical applications. The error management is assisted by the user. Audio and video systems are also a part of the soft RTOS. Some of the examples of soft RTOS are PC, multimedia systems, web browsing, electronic games, DVD players, etc.

C. Firm RTOS

Firm RTOS must adhere to deadlines as well. However, while the absence of a deadline will not result to system failure, but it can cause the quality of the results to be downgraded. When a firm real-time system does not complete its task by the deadline, the whole system does not fail like in the hard real-time systems but its results are considered as failure if it passes the deadline. It is mostly not used in practical applications like hard and soft RTOS. Some of the instances of the firm real-time system are video conferencing and satellite based tracking.

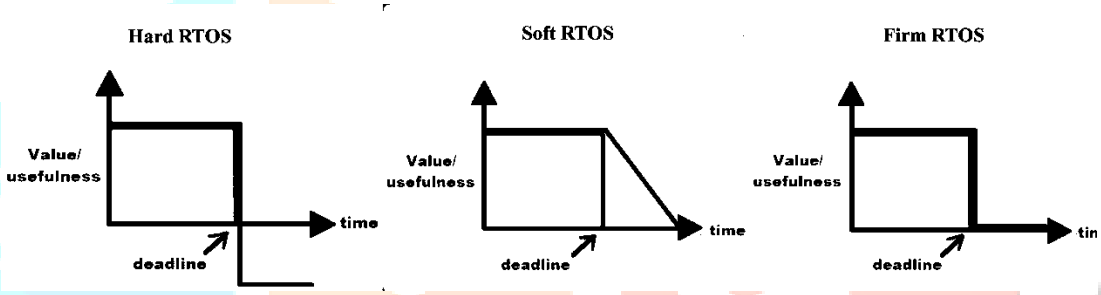

Figure 2: Comparison of Hard, Soft and Firm RTOS

VII. VARIATION BETW EEN HARD AND SOFT RTOS

How to tell between hard and soft real-time systems is that in a hard real-time system, time is the deciding factor and not meeting the specific time will result in the total failure of the system [6]. A soft real-time system, on the other hand, does not result in the failure of the system even if a deadline is missed, but its performance suffers if the deadline is consistently missed. In the soft real-time, the time limitations are not as severe as hard real-time [7, 8].

VIII. COMPARISON BETWEEN HARD RTOS AND SOFT RTOS

Real-Time Operating Systems (RTOS) can be broadly categorized into hard and soft types, each tailored to specific application needs. The Comparison based on different characteristics of RTOs are listed in Table 1 [9].

Table1: Comparison of Hard RTOS and Soft RTOS

Terms	Hard Real-Time System	Soft Real-Time System
Definition	A hard-real-time system is a system in which a failure to meet even a single deadline may lead to complete or appalling system failure.	A soft real-time system is a system in which one or more failures to meet the deadline are not considered complete system failure, but that performance is considered to be degraded.
File size	In a hard real-time system, the size of a data file is small or medium.	In a soft real-time system, the size of the data file is large.
Response time	In this system, response time is predefined that is in a millisecond.	In this system, response time is higher.
Utility	A hard-real-time system has more utility.	A soft real-time system has less utility.
Database	A hard real-time system has short databases.	A soft real-time system has enlarged databases.
Performance	Peak load performance should be predictable.	In a soft real-time system, peak load can be tolerated.
Safety	In this system, safety is critical.	In this system, safety is not critical.
Integrity	Hard real-time systems have short term data integrity.	Soft real-time systems have long term data integrity.
Restrictive nature	A hard real-time system is very restrictive.	A Soft real-time system is less restrictive.
Computation	In case of an error in a hard real-time system, the computation is rolled back.	In a soft real-time system, computation is rolled back to a previously established checkpoint to initiate a recovery action.
Flexibility and laxity	Hard real-time systems are not flexible, and they have less laxity and generally provide full deadline compliance.	Soft real-time systems are more flexible. They have greater laxity and can tolerate certain amounts of deadline misses.
Validation	All users of hard real-time systems get validation when needed.	All users of soft real-time systems do not get validation.
Examples	Satellite launch, Railway signalling systems, and Safety-critical systems are good examples of a hard real-time system [10].	DVD player, telephone switches, electronic games, Linux, and many other OS provide a soft real-time system [11,12].

IX. ADVANTAGES OF RTOS:

Real-time operating systems are offered a variety of advantages, including the ones listed below,

- Maximum consumption refers to the most efficient use of devices and systems. As a result, more production from all
  resources.
- In these systems, there is relatively little time allotted for task shifting. It takes roughly 10 microseconds with earlier systems, for example. It takes 3 microseconds to switch from one task to another on today's systems.
- RTOS give importance to applications that are already executing compared to those which are in the queue.
- Because the data volume is lower than that of soft real-time operating systems, real-time operating systems are frequently utilised in embedded systems.
- RTOS does not have any errors.
- Memory allocation is not a problem in RTOS.

X. DISADVANTAGES OF RTOS

Real-time operating systems come in a variety of limitations, including the ones listed below:

- RTOS focuses on tasks that are already running, the other tasks wait in the ready queue. It cannot do multitasking.
- The algorithms are extremely complicated and tough to write.
- An OS should have different memory domains for different processes but there is no such function in RTOS.
- In RTOS, it is not easy to change the thread priority which is given before.
- The RTOS conducts the bare minimum of task switching.

XI. RTOS CHARACTERISTI CS

The following are some of the most important characteristics for a RTOS:

- Memory use should be quite low.
- The resource consumption is quite high for a RTOS.
- Time response should be minimum.
- The Kernel is the one in RTOS that will decide task to be performed.
- The result should be the same no matter how many times you run it.
- It is a time-dependent system.

XII. FACTORS FOR SELECTING AN RTOS:

Some important things to consider while choosing a RTOS are,

- Performance: We should check the performance of the specific RTOS as it is an important feature for the system.
- Middleware: RTOS should have a proper middleware to function.
- Time: Time is a main factor for deciding a RTOS as it has severe time deadlines to follow.
- Code: The code should be reusable which makes the system more efficient.
- Maximum Consumption: it should utilize all the resources to the maximum.
- Task shifting: There should be no delay time to switch between two tasks.

XIII. DIFFERENCE BETWEEN GPOS AND RTOS

Some key distinctions between GPOS and RTOS are listed in Table 2.

Table 2: Difference between GPOS and RTOS

S. No.	General Purpose Operating System (GPOS)	Real-time Operating System (RTOS)
1	GNU/Linux is a general-purpose operating system (GPOS) [13].	It is mainly used in embedded systems.
2	It is used in both laptops and computers.	Round-robin scheduling is a type of time-based scheduling.
3	Scheduling based on processes.	Interrupt latency is not important compared to GPOS
4	Interrupt latency is extremely low, measured in microseconds	There is priority inversion mechanism in the system , opposite to the GPOS
5	There is no priority inversion mechanism in the system.	The operation of the kernel can be paused.
6	Kernel's operation might be pre-empted or not.	The system is continuously running without any gap
7	Priority reversal goes overlooked. IX. It has large memory.	It does not have large memory

XIV. EXISTING AND PROPOSING APPLICATIONS

Existing Applications

1) **Airlines reservation system:** As the fares keep on changing for the airlines, real-time operating system is important.
2) **RADAR:** The RTOS is used in radar in the defense to spot any incoming missiles [14, 15].
3) **Camera:** RTOS is needed in camera as taking a photo is a time-bound process, like if u set a timer to take the photo.
4) **Robotics:** RTOS is also used in the field of robotics where there are certain robots which need to finish their work within a specific time [16].
5) **Medical Sector:** There are many medical devices which use RTOS to shorten the time for any medical cases which is very beneficial.

Proposing Applications

1) **Airbag System in cars:** If the RTOS is implemented in airbag system it can make the process faster and it will be very helpful.
2) **Home security systems:** RTOS can be implemented in the home security systems where it can lock the doors immediately if there is an attempt to intrusion.
3) **Fire-alarm systems:** RTOS should be implemented in the fire-alarm systems as it can increase the response in time of the fire-alarm models operating in commercial properties and industrial settings.
4) **ATMs:** We can implement an RTOS in an ATM to improve the time response of the system.
5) **Anti-braking systems**: Anti-braking systems are crucial in the automobiles so implementing a RTOS which can provide a better response time is crucial.
6) **IoT:** Free RTOS supports low power consumption, crucial for battery-operated devices, and is scalable to accommodate IoT deployments of different sizes. With a strong community of developers, it benefits from continuous improvement, bug fixes, and a wealth of resources, contributing to its reliability and security features in the ever-evolving landscape of IoT technologies [17].

XV. SUMMARY

Computer operating systems fall under the category of real-time operating systems in which its main function is to complete a given job within those specified time limits. The main components of the RTOS include the Scheduler, Symmetric Multi Processing (SMP), Function library, Memory management, Latency free-dispatch, User-defined data items and classes.

There are three types of real-time operating systems:
1) **Hard Time:** tasks in Hard RTOS needs to be finished by the deadline or by the end of specified time range.
2) **Soft Time:** achieving deadlines isn't required for every activity all the time, but the process must be completed, and the product delivered.
3) **Firm Time:** Firm RTOS must adhere to deadlines as well.

However, while the absence of a deadline will make the result a failure. The RTOS system uses extremely little memory and consumes very few resources. When it comes to choosing an RTOS, performance is the one of the most important thing to consider. GPOS (General-Purpose Operating System) is utilised on desktop PCs and laptops, but RTOS (Real-Time Operating System) is exclusively tested on embedded systems. Some of the embedded systems in which RTOS is used are multimedia systems, robots, airline reservation system etc.

REFERENCES

[1] Singh, Sanjay; Tripathi, Nishant; Chaudhary, Anil Kumar; and Singh, Mahesh Kumar (2012) "An Introduction to RTOS," International Journal of Computer Science and Informatics: Vol. 1: ISSN. 4, Article 12.

[2] Karnakanti, Suresh, "A Survey on Different Real-time Operating Systems" (June 17, 2021). International Journal of Engineering and Advanced Technology (IJEAT), Volume-10 Issue-5, June 2021, Page No: 221-223, Available at SSRN: https://ssrn.com/abstract=3922073 or http://dx.DOI.org/10.2139/ssrn.3922073

[3] Karunakar Pothuganti, Aredo Haile and Swathi Pothuganti "A Comparative Study of Real Time Operating Systems for Embedded Systems", June 2016, International Journal of Innovative Research in Computer and Communication Engineering, Vol. 4, Issue 6, June 2016.

[4] Sarah Khan, "Real-Time Operating System (RTOS) with Different Application: A Systematic Mapping", EJERS, European Journal of Engineering and Technology Research Vol. 6, No. 1, January 2021.

[5] Navneet Tiwari, Jatan Saraf, M. Agrawal, Siddarth Jain, "A User Friendly Classification & Analysis of Real Time Operating System", 2011, International Symposium on Computing, Communication, and Control (ISCCC 2009) Proc .of CSIT vol.1 IACSIT Press, Singapore.

[6] Walter Cedeno and Phillip A. Laplante, "An Overview of Real-Time Operating Systems", 2007, The Association for Laboratory Automation, DOI:10.1016/j.jala.2006.10.016

[7] Durgesh Raghuvanshi , "Study on Real-time operating System and its Scheduling Procedures", International Journal of Trend in Scientific Research and Development (IJTSRD), ISSN: 2456-6470, Volume-2 | Issue-2, February 2018, pp.1368-1370.

[8] A. Sha, K. Anvesh, G. N. Naidu, R. Dasari and R. V. R, "Recent Trends and Opportunities in Domain Specific Operating Systems," 2022 3rd International Conference for Emerging Technology (INCET), Belgaum, India, 2022, pp. 1-5, DOI: 10.1109/INCET54531.2022.9824237.

[9] Santhosh M S and Nagaraja G S, "A Survey on Different Real Time Operating Systems", International Journal of Engineering and Advanced Technology, Vol.10, Issue: 5, June 2021.

[10] Priyanka Kumar and Raghul M, "Location Based Parental Control-Child Tracking App Using Android Mobile Operating System," 2018 4th International Conference on Computing Communication and Automation (ICCCA), 2018, pp. 1-4, DOI: 10.1109/CCAA.2018.8777612.

[11] Radhakrishnan, Vivek, Hari Narayanan, and Shiju Sathyadevan, "System Call Authorization in Linux by a Secure DAEMON", ARPN Journal of Engineering and Applied Sciences, VOL. 12, NO. 12, JUNE 2017 ISSN 1819-6608

[12] H. Narayanan, Radhakrishnan, Shiju-Sathyadevan and J. Poroor, "Architectural Design for a Secure Linux Operating System," 2017 International Conference on Wireless Communications, Signal Processing and Networking (WiSPNET)", 2017, pp. 949-953, DOI: 10.1109/WiSP- NET.2017.8299902.

[13] Domenico Mundo, Marco Gubitosa, and Alessandro Toso, "Real-Time and Real-Fast Performance of General-Purpose and Real-Time Operating Systems in Multithreaded Physical Simulation of Complex Mechanical Systems", Hindawi Publishing Corporation Mathematical Problems in Engineering Volume 2014, Article ID 945850, 14 pages http://dx.DOI.org/10.1155/2014/945850.

[14] Felix G. Hamza-Lup, Hongjun Su and Coby Wissing, "Web-enabled Software for Real-time Autonomous Wireless Sensors Data Visualization", Conference: Proceedings of the International Conference on Industrial Control Network and System Engineering Research - ICNSER2019, Year: 2019, Page 70.DOI: 10.1145/3333581.3333594

[15] R. Guntha, N. B. Sai Shibu and S. N. Rao, "Resource Constrained Onboard Controller System for Real-time Fishing Boat Tracking in High Seas," 2019 9th International Symposium on Embedded Computing and System Design (ISED), 2019, pp. 1-5, DOI:0.109/ISED48680.2019.9096236.

[16] A. S. Pillai and T. B. Isha, "ERTSim: An embedded Real-Time Task Simulator for Scheduling," 2013 IEEE International Conference on Computational Intelligence and Computing Research, 2013, pp. 1-4, DOI:10.1109/ICCIC.2013.6724195.

[17] Rafael Raymundo Belleza, Edison de Freitas Pignaton, "Performance Study of Real-Time Operating Systems for Internet of Things Devices", IET Softw., 2018, Vol. 12 Iss. 3, pp. 176-182 DOI: 10.1049/iet-sen.2017.0048