ISSN(Online): 2320-9801
ISSN (Print) : 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

*(An ISO 3297: 2007 Certified Organization)*

**Vol. 4, Issue 6, June 2016**

# A Comparative Study of Real Time Operating Systems for Embedded Systems

Karunakar Pothuganti, Aredo Haile, Swathi Pothuganti

Assistant Professor, Department of Electrical and Computer Engineering, Madawalabu University, Bale Robe, Ethiopia.

Head of the Department, Department of Electrical and Computer Engineering, Madawalabu University, Bale Robe, Ethiopia.

Assistant Professor, Department of Computer Science Engineering, Sree Chaitanya Institute of Technological sciences, Karimnagar, India.

**ABSTRACT**: This paper gives quantitative and qualitative results obtained from the analysis of real time operating systems (RTOS). In this paper studied systems were Windows CE, QNX Neutrino, Vx Works, Linux and RTAI-Linux, which are mostly used in industrial and academic environments. Windows XP was also analysed, as a reference for conventional non-real-time operating system, since such systems are also commonly and inadvertently used for instrumentation and control purposes. The evaluations include worst case response times for latency, latency jitter and response time.

**KEYWORDS:** Real time operating systems, Windows CE, QNX Neutrino, Vx Works, Linux and RTAI-Linux

## 1. INTRODUCTION

Real Time Operating Systems (RTOS) are specially designed to meet rigorous time constraints. In several situations RTOS are present in embedded systems, and most of the time they are not noticed by the users. A good example of this situation may be observed in the automobile industry, where it is estimated that 33% of the semiconductors used in a car are microcontrollers, being common for a car to have dozens of microcontrollers. As a consequence, the manufacturing costs of vehicles are reaching proportions that existed only in the aerospace industry, where 1/3 of the total cost of a vehicle is spent in the chassis, 1/3 in the power train and 1/3 in electronics. Seeking for improvements on its products and development time reduction, the car manufacturers have been adopting RTOS to control the software that runs in the vehicles. A good example, is the electronic injection of fuel into the car engine, which must be made with rigorous time constraints. At each motor cycle, sensors need to measure and analyse the output gases generated by the combustion, and then compute the next mixture combination before the next ignition happens.

Additionally, it is known that nowadays even simple motors, such as the ones used in motorbikes, already uses RTOS in their software. Further examples are the recently developed avionics control systems, which use a single computer to cope with several aircraft subsystems, thus requiring an operating system with temporal and spatial partitioning systems. Spatial partitioning refers to tasks isolation in the computer memory, while temporal partitioning refers to tasks scheduling, dividing the processor time properly. These partitions allow a single processor to execute several tasks simultaneously, without the risk of one task causing interference in the temporal requirements of other tasks. This approach allows reduction of computers required to fly a plane, making it lighter. In the international market, there are more than a hundred options of RTOS to choose from, while additionally there are free similar options as the Linux operating system. In this way, the decision on which system to use may be a key factor to the success or failure of a project, and the analysis must be made with impartiality and adequate criteria.

Real time operating systems are the multitasking operating systems, which not only depend upon the logical correctness but also depend upon the application delivery time. These valuable RTOS works on the philosophy of the round robin algorithm and pre-emptive priority scheduling method. The RTOS requires very less amount of space around 10 KB to 100 KB in memory. There are several advantages of RTOS like simple implementation, low overhead and predictability. The specific real time operating systems that are: *Nucleus RTOS: This is developed by the embedded system division of the Mentor Graphics. It consist out Full-featured toolsets. Nucleus OS is part of a complete embedded solution with a full complement of embedded middleware. *Lynx OS: Current enhanced Lynx OS version 5.0 added microkernel design, which replaces the monolithic architecture of older Lynx OS. Lynx OS 5.0 has consist serial ATA attachment support, symmetric multiprocessing unit, executable and linkable file format and RAM support increases up to 2 GB.*QNX Neutrino: QNX is a microkernel based commercial UNIX like RTOS. The QNX consist a operation known as 'Msgsend', which inter process the communication between all thread according to priority order scheduling property. It also implements POSIX message queues beside the kernel.*VxWorks: From Phoenix Mars Lander to Deep Impact Space Probe and from Spirit to Mars Reconnaissance are the few examples of significant spacecraft, which uses the VxWorks as a sole of their embedded system. The micro kernel of the VxWorks support the scheduling, multitasking and memory management. *Windows CE:. Windows CE 6.0 is the latest launched RTOS of Microsoft. This latest version of the RTOS uses the Microsoft Visual Studio and.NET compact framework Platform. The deletion and creation of the thread is a periodic process, which occurs for 1 millisecond after every periodic one second. *µC/OS-II: This operating system is written in the C language and specifically designed for the embedded systems. It consists out a pre-emptive and real time kernel, which have multiple threading. This operating system is designed with the vision to provide superior quality software component for industry solution. The µC/OS-II is freely available software for the educational purposes.

## II. MOTIVATION OF THE STUDY

One of the recent survey that is conducted by the EMF, which is a premier market intelligence and advisory firm in embedded technology industry.The survey showed that the various OS vendors have launched campaigns that suggest that various stringent standards need to be required for medical device applications. But the question arises that, are this certified RTOS truly necessary for the medical device applications? If yes, than which one is the suitable platform for my applications?

The objective of preparing this paper is to enlighten the person who belongs to more than one professional group and especially nontechnical group. It is our best possible effort to provide a comparison chart among various popular RTOS. The various selection parameters of the RTOS have been discussed in the following points.

## III. SELECTION CRITERIA

A user before purchasing the RTOS for their industry always goes through with lengthy discussions, various meetings and consulting with the technical staff, which often creates the state of confusion. Generally, the user has to consider the following four important parameters before selection of RTOS.These four parameters consist the whole series of selection.

A. *Functionality*: The functionality of the RTOS is totally related to basic architecture of the OS. In major consideration the functionality is related with the lower end of spectrum, which offers basic preemptive scheduler and key system call. These RTOS is less expensive and come with the source code, which can be easily modify according to the applications. To increase the functionality of the RTOS, we have to go through beyond the basic scheduler. The advance scheduler consist out a variety of tools, which can be used for the advance development environment. This advance development environment, also known by the IDTs.

B.*Performance*: The performance of the RTOS is the cornerstone of the quality assurance and reliability. The choice of platform and processor also affects the performance of RTOS. If the application requires limited features than at that time lower and basic version of the RTOS should be used. Otherwise It increases your expenditure.

C..*Price*: The RTOS comes into wide range and different variety. Price is one of the foremost issues of selection criteria. The RTOS you actually needed should be purchased according to technical specification, so that the price will adjust according to the application requirement.

D.*Advanced Feature*: Apart from the common attribute, there must be some requirement of advanced feature. Some of the advanced feature can be described in the following points like. i) The kernel of operating system must have multimedia supportability. ii) A complete management and control over display must be available.The GUI can provide

ISSN(Online): 2320-9801
ISSN (Print) : 2320-9798

**International Journal of Innovative Research in Computer and Communication Engineering**

*(An ISO 3297: 2007 Certified Organization)*

**Vol. 4, Issue 6, June 2016**

effective services and management. Iii) Micro kernel architecture of the RTOS is preferable due to availability of good cache memory, kernel memory and high CPU cycle utilization.

## IV. RTOS FEATURE COMPARISON

The kernel is one of the most important part of the real-time system. Scheduling, task management, timer handling, IPC mechanism are the basic services, which is provided by the kernel. A RTOS should have the following features: [a].Virtual and dynamic memory management. [b]Task synchronization.[c]Various platform and USB supportability. [d]Scheduling, creation and dispatching of task.[e]Suitability for the mission critical approach. [f]Multicore/Multiprocessor supportability.[g].Interrupt handling capability [h] Semaphore supportability. The following sections will describe the functionality and the comparison of various available RTOS.

## V. STANDARD AND PROTOCOL COMPLIANCE

The standard and protocol provides a prototype, which work as a reference model for individual manufacturer. They are essential for maintaining an open and competitive market for advancement and processing of real-time operating system technology. Some of the RTOS and their standard compliance are shown out below. Most of the RTOS is Ada C language supportable. The μC/OS-II is written in highly portable ANSI C, with target microprocessor specific code written in assembly language.

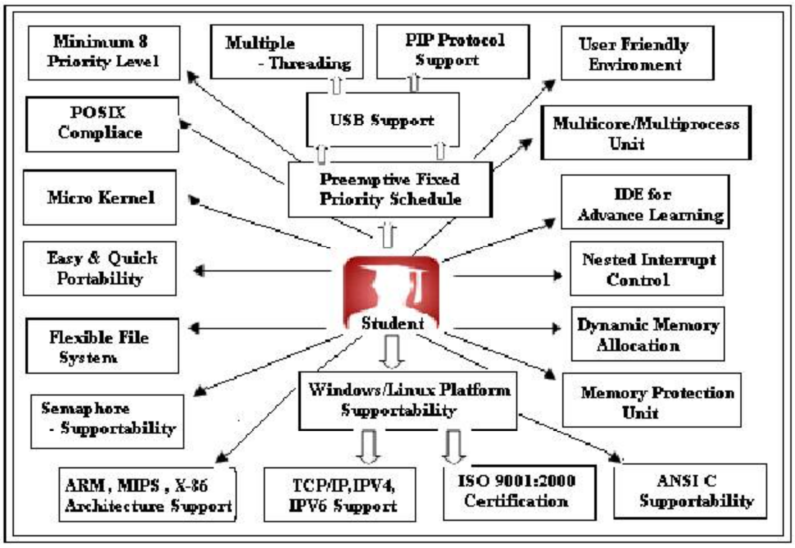


Fig. 1 Demonstration of the Considerable Points Platform requirements

TABLE 1 DEMONSTRATION OF CLASSIFICATION OVER STANDARD & PROTOCOL COMPLIANCE

| **RTOS** | **QNX NEUTRINO OS** | **Lynx OS** | **VxWorks** | **Windows CE** | **Nucleus RTOS** | **MicroC/OS-2** |
|---|---|---|---|---|---|---|
| *Standard | Open GLES,IEC Integrity level | ISO9001:2000 | Service Capability Support | WSD Supportable | MAX IL Certify | RTCA, IAR's Spy Debugger, |
| *POSIX Compliance | YES | YES | Queued POSIX | -- | YES | YES |
| *Protocol | PIP | PIP | HLP | PIP | PIP | PIP |

ISSN(Online): 2320-9801
ISSN (Print) : 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

*(An ISO 3297: 2007 Certified Organization)*

Vol. 4, Issue 6, June 2016

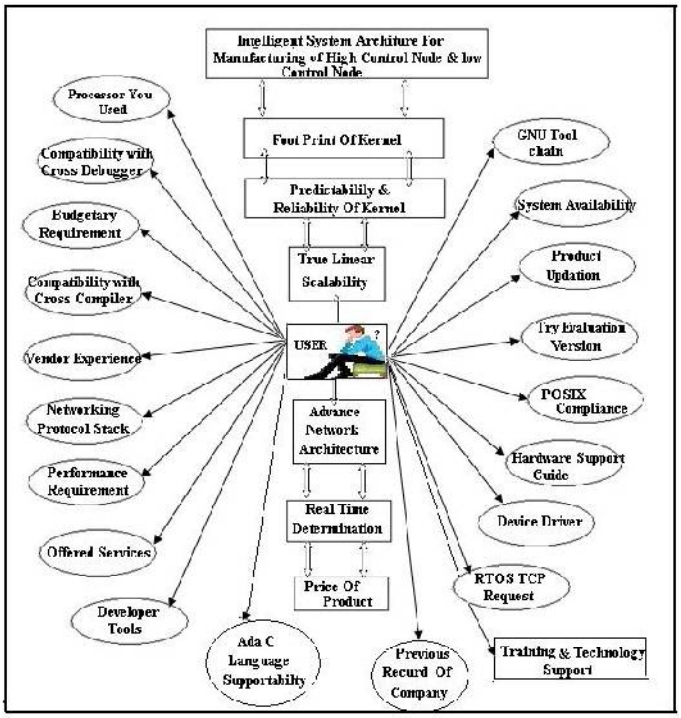


Fig 2 Demonstration of the General Purpose RTOS before the Selection of Student Purpose

## VI. MEMORY MANAGEMENT

The emergence of MMU [Memory Management Unit] brings a better supportability for the virtual memory management. In RTOS, there are two types of memory management available. Static memory mechanism works on the principle of pools division. The pools mechanism allows application software to allocate the chunk of memory into 4 to 8 different buffer size per pools. It permits a buffer to put down the buffer list in available size for reuse of original buffer in future. In another type of memory management mechanism the task multi programming phenomenon is used in which, we uses the concept of demand paging.

TABLE 2 DEMONSTRATION OF CLASSIFICATION OVER MEMORY MANAGEMENT

| RTOS | QNX NEUTRINO OS | Lynx OS | VxWorks | Windows CE | Nucleus RTOS | MicroC/OS-2 |
|---|---|---|---|---|---|---|
| *Type of Kernel | Micro Kernel | Dynami Kernel | Micro Kernel | Monolithic and Hybrid | Real Time | Preemptive/ Real Time |
| *MMU & Virtual Memory Paging | Strict Memory Protection by MMU | MMU with Virtual Addressing | Best Fit Algorithm | Flexible Memory Model | MMU Available | Fixed Size Memory Block |

## VII. PROCESS SPECIFICATION & TASK SYNCHRONIZATION

The process is defined as an instance of a program running on the computer. The process consist out three components, i.e. [a] An executable program [b] Associated data with that program. [c] Execution context of the program. Synchronization is needed to share the mutual exclusive resources of a real-time system. The priority inversion is the

ISSN(Online): 2320-9801
ISSN (Print) : 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 4, Issue 6, June 2016

technique which enforces the highest priority task to execute first. The one of the most traditional approach for this problem is the priority inversion protocol [PIP] and the highest locker protocol [HLP].

TABLE3 DEMONSTRATION OF CLASSIFICATION OVER PROCESS SPECIFICATION & TASK SYNCHRONIZATION

| **RTOS** | **QNX NEUTRINO OS** | **Lynx OS** | **VxWorks** | **Windows CE** | **Nucleus RTOS** | **MicroC/OS-2** |
|---|---|---|---|---|---|---|
| *Threading | Single | Multiple | Single | Single | Multiple | Multiple |
| *PriorityLevel | 32 level | 256 level | 256 level | 8 level | 64 level | -- |
| *Nested Interrupt | Yes | Yes | Yes | No | Yes | Yes |

## VIII. RESULTS AND SUGGESTED APPLICATIONS

The selection of the RTOS is a complex job. According to the application they may be considerably chosen. Right selection provides a cost effective solution and can be able to produce excellent result within the deadline. As the memories of the integrated circuits are getting denser, they were considerably scaled down for the general purpose operating system versions. In end after considering the various parameters, here we are providing some suggested application for the RTOS.

TABLE 4 DEMONSTRATION OF SUGGESTED APPLICATIONS FOR VARIOUS AVAILABLE RTOS

| **RTOS** | **Overall Architecture and Performance** | **Suggested Application** |
|---|---|---|
| Vxworks | Excellent | Complex real time and embedded application,space- craft. |
| MicroC/OS-2 | Excellent | Educational & Embedded Based Applications. |
| QNX NEUTRINO OS | Very good | Server, Embedded & Workstations. |
| Windows CE | good | Minimalist computer and embedded application |
| Lynx OS | Excellent | Military, avionics, Industry, control, telecommunication |
| Nucleus RTOS | Very good | Setup Boxes, Cellular Phones &Consumer Electronics |

## IX. FUTURE WORKS & CONCLUSION

In this paper, we had shown out the various applications, requirements and selection criteria's of six widely used RTOS. In present scenario, there is a need of accurate time determination. A mathematical model, with an approach of precise time determination can brings a notable change in the field of real-time operating system. We are currently working on development of this mathematical model. In the future work we will create a benchmark by installing and testing the various RTOS., along with a detailed performance study of the various RTOS. We are also broadening our approach by studying the Enterprise RTOS.

ISSN(Online): 2320-9801
ISSN (Print) : 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

*(An ISO 3297: 2007 Certified Organization)*

Vol. 4, Issue 6, June 2016

## REFERENCES

[1] C. Walls,"RTOS for Microcontroller Applications", Electronic Engineering, Volume 68, no.831, page number 57-61, 1996.
[2] Chun-chi Jim Liu,"Introduction to Nucleus RTOS "from http://netlab.cs.nchu.edu.tw.
[3] D.Hart, J.Stultz and T.Ts'o,"Real Time Linux in real time ", IBM System Journal Volume 47, Number 2, 2008.
[4] Douglas,C.Schimidit, Mayur Deshpandey and Carlos O' Ryan,"Operating System Performance in Support of Real Time Middle Ware", Department of Electrical and Computer Engineering,University of California,Irvine.
[5] Embedded Linux OS,"The scalable,reliable and highly deterministic operating system for real time embedded application "http://www.lynx.com/pat's/lynxOS.pdf.
[6] Dam O'Dowd's, "Embedded Linux: with friends like this, who needed enemies?"http://www.embe dded.com/columns/guest/20740 2542-46k.
[7] David Kalinsky, "Basic Concept of Real-Time operating system" November18, 2003. http://www.linuxdevices.com/articles/ AT 462795573.html.
[8] J.Corbet, "The developer's certificate of origin", V1.1, Linux weekly news [June 2005] http://lwn.net/Articles/139916.
[9] Michel Barr,"How to Choose a Real-Time operating system ", January 28.2003.
[10] Micro C/OS-2,The Real-Time kernel "user manual,guide http://www.micrium.com/downloads/datasheets/ µC/OSII- Datasheet- 2008.pdf.

## BIOGRAPHY

**Mr.Karunakar Pothuganti** received B.tech degree in Electronics and communication Engineering from JNTU in 2006, and he received M.tech in Embedded systems from JNTUH. He completed PGDES from pune university, He is having more than 9years of Teaching and industrial experience, he published so many articles in international journals. Presently working as Assistant professor in Madawalabu University, Ethiopia. His research areas are Embedded systems and medical image processing.

**Mr. Aredo Haile** is working as Head of Electrical and computer Engineering department, Madawalabu university, Ethiopia. He received his degree from Dire dawa university, His research areas are control systems and embedded systems. He published so many articles in international journals.

**Mrs Swathi Pothuganti** received B.tech and M.tech degree in Computer science Engineering from JNTUH, India in 2013 and 2015 respectively. Her research areas are real time operating systems and cloud computing.