2009 International Symposium on Computing, Communication, and Control (ISCCC 2009)
Proc .of CSIT vol.1 (2011) © (2011) IACSIT Press, Singapore

A User Friendly Classification & Analysis of Real Time Operating System

Navneet Tiwari[1 +], Jatan Saraf[2], Mohit Agrawal[3] and Siddarth Jain[4]

Medicaps Institute of Technology and Management, Indore, India

Abstract: The Idea behind the operating system is not very new, it's many years old. Evolution of operating system causes significant changes in task solving methodology. They stay's responsible for the overall system requirement, performance, and task solving methodology. A system who works on the aspect of time determination is generally known as the real time system. Advancement of embedded based real time operating system guarantees the time constraint capability and predictability of an application. In this paper, we will discuss the enhancement of opera ting system era in real time environment. Our approach tends to introduce the emerging trends in this field and provide a user friendly classification, which can cover more than one professional group. It begins with the conceptual enhancement of the current technology at fundamental level for better technical understanding. The classification provides choices to system designer, student and researchers.

Keywords: CBR, FTP, LINUX, NS2, Queue.

1. Introduction

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 preemptive 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.[5] *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. [11] 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.[15]. *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.. [16] 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 preemptive and real time kernel, which have multiple threading. This

[+] Corresponding author. Tel.: + 919826079288
E-mail address: navneet1231@gmail.com

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.

2. 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.[14] 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 non-technical 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.

2.1. Selection Process 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. [9] 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 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.

3. 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. [12]

3.1. 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. [7] 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. [10]

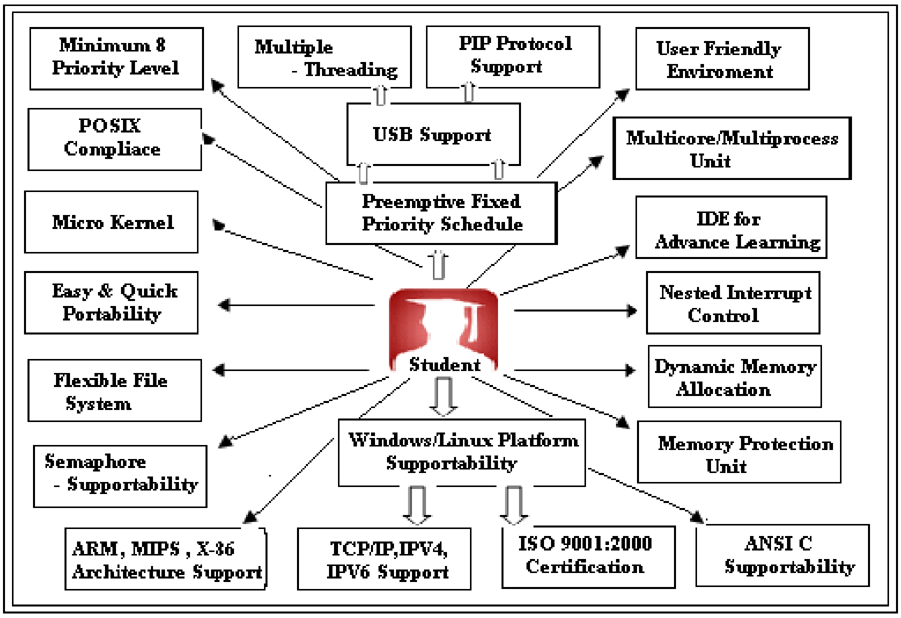

Figure 1 Demonstration of the Considerable Points Platform requirements

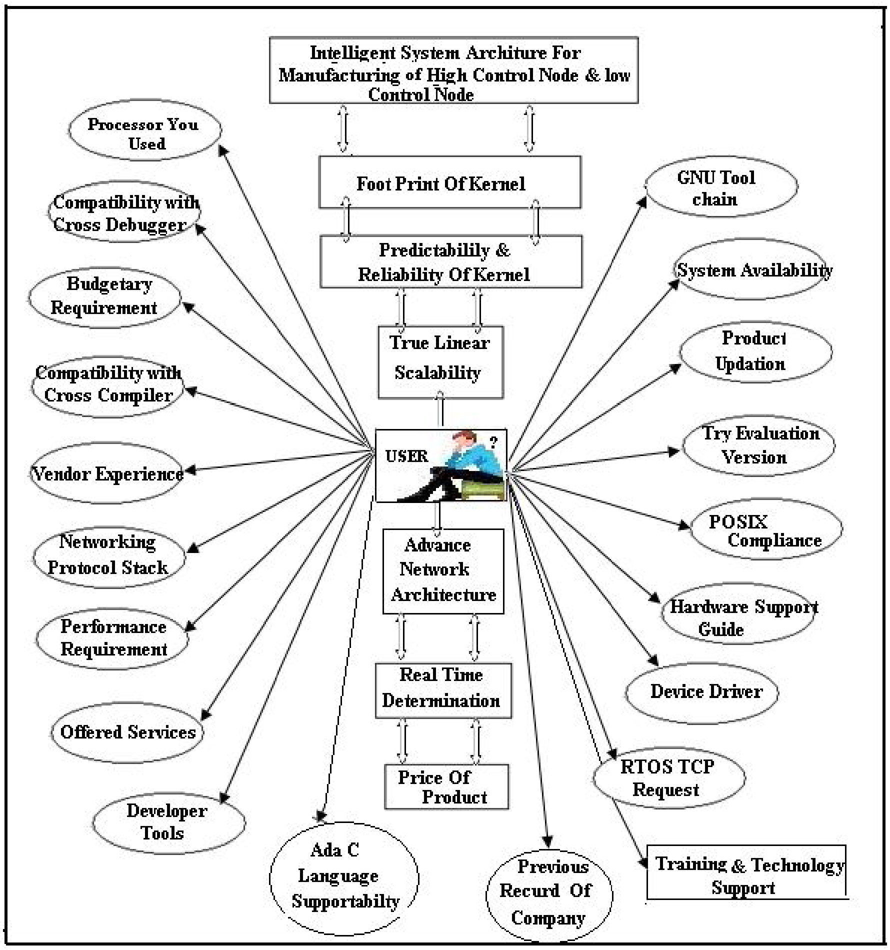

Figure 2 Demonstration of the General Purpose RTOS before the Selection of Student Purpose

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	ISO 9001:2000	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

3.2. 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[12].

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.

3.3. 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 technique which enforces the highest priority task to execute first[4]. 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

3.4. Technology Specification

Technology specification is the description of the development tools, network protocol stack technology and semaphore supportability. Different manufacturer have introduced various advanced feature in their RTOS. This advanced feature is applicable and supportable in the different workspaces like the Windows CE consist Visual Studio and.NET compact framework. It is more suitable for the minimalistic computers and hardware system. [13]

TABLE 4 DEMONSTRATION OF CLASSIFICATION OVER TECHNOLOGY SPECIFICATIONS

RTOS	QNX NEUTRINO OS	Lynx OS	VxWorks	Windows CE	Nucleus RTOS	MicroC/OS-2
*Development Tools	Productivity Analysis tool	GNU C/C++ compiler	Memory Analyzer	.Net Compact frame work	Data Analyzer	GNU C/C++ compiler
*Technology Used	Patent Pending Adaptive Partitioning	Application Binary Interface Compatibility	Tornado Development Environment	Platform Builder IDE	OSEK/VDX Abstraction Layer	Accelerated Technology Product
*Network Protocol Tech.	IPV4,IPV6, TCP/IP	IPV4,IPV6, TCP/IP	IPV4, IPV6, IEEE 802.11 Driver	Server Authentication Feature	TCP/IP	TCP/IP, IPV4, IPV6,

3.5. Advanced Feature

The continuous, interactive and the integrative research effort in the embedded system technology increase the requirement of the advanced feature. Mission critical approach, file system, multicore /multiprocessor supportability and news group for technical discussions are the few examples of the RTOS advanced feature.

3.6. Leading Edge

The leading edge is defined in terms of the specific feature of the RTOS. The portability, scalability, consultancy & training Services are the few signatures of the leading edge. The Clients/partners names are also introduced up in this segment. Because, we think that a company and his product is always being appreciate in terms of his associates.

TABLE 5 DEMONSTRATION OF THE CLASSIFICATION OVER ADVANCE FEATURE

RTOS	QNX NEUTRINO OS	Lynx OS	VxWorks	Windows CE	Nucleus RTOS	MicroC/OS-2
*Multicore / Multiprocessor support	Consist Aware Tool-Chain	Symmetrical Multiprocess support	Symmetrical Multiprocess support	Symmetrical Multiprocess support	Symmetrical Multi –process support	Symmetrical Multi –process support
*File System Supportability	Disk, Flash, CIFS	ELF File System	Flexible File System	EXFAT File Allocation.	FAT Embedded System	FAT,ISO960& Flash Format

TABLE 6 DEMONSTRATION OF THE CLASSIFICATION OVER LEADING EDGE

RTOS =>	QNX NEUTRINO OS	Lynx OS	VxWorks	Windows CE	Nucleus RTOS	MicroC/OS-2
*IDE	Eclipsed based IDE tools	Code warrior & Luminosity IDE.	Eclipsed Platform.	Windows Powered Solution.	IDE supportable	Edge mentor's tools.
*Clients / Partner's name	AMD,Aonix,ARM,EMRT, Genesi,IBM	Xilinx,AIM,4front Technology.	Nasa Jet Propulsion Lab	Thin client work station	Intel IBM,Atmel,MIPS sharp Samsung	Access Co. Ltd.
*Special Feature	Inbuilt self Healing System.	Application Binary Interface Compatibility	Memory protection Unit.	Multilingual language Support.	Multimedia frame work	C written kernel System

4. 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 7 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

5. 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. [4] 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.

6. 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/OS-II-Datasheet- 2008.pdf.

[11] QNX Neutrino,"Real Time Operating System User Manual Guide'.

[12] Ramesh Yerraballi,"Real-Time Operating System: An Ongoing Review" Department of Computer Science and Engineering, University of Texas Arlington.

[13] Shonil Vijay, "A study of Real-Time Embedded Software System and Real Time Operating Systems",Master Of Technology Seminar Report ",IIT Powai.

[14] Express Logic," Survey of embedded developers shows effect of RTOS on project success", Embedded Systems Conference Silicon Valley, San Jose, CA (April 2, 2007) http:// www.rtos.Com /news/detail/?prid=122

[15] VxWorks Programmers Guide",http:// www.wrs,com /pdf vxworksguide.pdf

[16] Windows CE, "Corporate Sync1.1.3foe WindowsCE user's guide",http://www.stellar.com/notes/corpsync/WinCE/1-1- 3/CSyncWCE.pdf.