International Journal of Latest Trends in Engineering and Technology
Special Issue SACAIM 2016, pp. 54-58
e-ISSN:2278-621X

A STUDY OF OPERATING SYSTEM FOR EMBEDDED SYSTEMS

Aravinda Prabhu. S[1], Ganesh Prabhu[2] and Preethika .R[3]

Abstract – Every system in the present generation have an operating system build into it. Mobiles have android, IOs, windows etc. Computers have LINUX, Windows, MAC and various others. Similarly, the embedded systems have a OS built on it too, termed as RTOS(Real Time operating System). These operating systems are knows for their efficiency at resource usage and reliability. RTOS also have built-in features based on the functioning the embedded system provides.In this paper we are going to perform a study on the origin of RTOS in embedded systems and how they play a main role in the embedded systems.

Keywords –Operating System, Real time, Embedded system, RTOS.

I. INTRODUCTION

An embedded system is a microprocessor based system that is built to control a function and is not programmable by the end user unlike PC is. Children need embedded systems to play video games and operate automatic chocolate vending machines. Housewives need embedded systems for smart internet compliant home appliances (IoT) such as microwave, TV, sound systems and so on. Organizations need embedded systems for network management and products.

There are numerous other applications of these systems which have increasingly become an important part of our life.

II. ROLE OF OS IN EMBEDDED SYSTEM

Every technical system requires an operating system since it was a major interface which can user can use to interact with the computer and manage how programs functions within the computer system. Some of the popular operating systems are LINUX, WINDOWS series, MAC OS, UNIX etc each supporting their own set of uniquefeatures.

A RTOS is an OS that serves real-time application requests, with the ability to process data as it is input, without significant buffering delays.RTOS is usedin machinery control, for scientific instruments and industrial control systems.

There are a number of key potential properties of OS which include

1. **Reliability and Stability:**It is the ability of the OS to function with less glitch and crashes.
2. **Scalability:**It is the ability of the OS to improve as the time goes, with the help of new updates. This directly leads to better long time functioning of the OS.
3. **Availability:**It is the ability of the OS to actively process requests without crashing or giving out compatibility errors

[1] *Department of Computer Science & Bio Informatics AIMIT,St Aloysius College Mangalore, Karnataka, India*
[2] *St Aloysius College, AIMIT,Mangaluru, Karnataka, India*
[3] *St Aloysius College, AIMIT,Mangaluru, Karnataka, India*

4. **Usability:** The popularity of the OS in the markets and the amount of the systems that support it and use it.
5. **Security:**it is the ability of the OS to prevent the attack on its systems. Keeping the system sake and secure.
6. **Portability and clustering:** This is the ability of the OS to migrate and function among other systems.
7. **User-Interface:** The way with which the OSinteracts with the user.

A RTOS generally having limited user-interface has high rate of reliability, availability.

III. EVOLUTION OF OS IN EMBEDDED SYSTEM

There are various definitions for the Real time systems. One of the definitions for real time systemsproposed by James Martin, a UKIT consultant – A real time computer system may be defined as one which controls an environment by receiving data, processing them and taking action or returning results sufficiently quickly to affect the functioning of the environment at that time.There was for systems that can provide results with less time and more security. This was the motivationfor the development of the real time systems. The first system that produced results in real time was Whirlwind I, developed bythe U.S. Navy in the late 1940s as a simulator for flight training. Three years later, Air Force pickedup over the project. Thus this takeover led to the Whirlwind II design in 1957, which was usedby the American Air Force in their air defense systems. SAGE was effectivein generating a large image of the target location by stitching all the bird eye view pictures obtained by recon team.

As the time passed by, American Airlines started realizing the power of the real time systems. Earlier to the real time operating systems, airline ticket booking was a time consuming and a slow task. The ticket booking were done manually by eight operators used to sitting around a table. The operator would place a mark on a corresponding card whenever a ticket was booked, when the entire card was marked it was considered that the flight was fully booked and doesn’t hold any passenger capacity. This was a way for passengers to determine whether the flights were full or not visually. It was time consuming process based on the size of the airport and the manual work of marking each individual card, it was less secure and less reliable because it was prone to error due to the faulty booking, and it also required a huge amount of manpower. The airport authority decided to solve this problem and making the task easier by implementing the real time systems for ticket-booking and other small airport functions. This real time system was named as Semi-automated Business Research Environment (SABRE) which was implemented during 1959. It automated the American airline reservation system functions like ticket availability and ticket printing. In 1960s first commercial real time operating systems were developed for mainframe computers. In 1962, IBM developed Basic Executive which provided diverse real time scheduling. It was followed by Basic Executive II [4].

IV. CATEGORIZING OPERATING SYSTEM (OS)

OS can be categorized into distinct categories by the clustering OSs by the various properties we have listed above.OS is partitioned into 6 categories in order to estimate the number of devices under each OS so that some estimate of efficiency and usability of each OS. Linux OS comprised of publication related to disability that has been well known since an early age, RC products for children cards experiment using the internet and Bluetooth toy car control were found as examples of devices using Linux as well. There were special purpose OSs built specifically for particular applications that included, the core of certain low cost vehicles for simple reactive behavior. Robot C was the OS for a monoball robot based on LEGO mindstorm focused on educating children in elementary school. Strifeshadow Fantasy OS was used for multi-player online game. Another MP game used V-System distributed O.

It was observed that Windows OS was used for several devices as compared to others Oss. These raised questions about what could be the reason behind its popularity and hence journal articles known to discuss appropriate criteria for choosing an OS were reviewed to depth.

Some of the most commonly used parameters that were selected as forming the basis on which an OS should be chosen are reliability, scalability, availability, usability, security, portability, clustering and performance, stability and certification, alternate programs and interface. However, data were given in few literatures which presented the percentage of criteria with respect to percentage of limitation within RTOS.

These findings were compared with RTOS for other applications which included medicine, supercomputing, natural disaster recovery scheme, cloud computing, automobile, manufacturing industry and underwater devices. The objective was to compare the performance of OSs within toys with the performance of the same OS within other applications. [1]

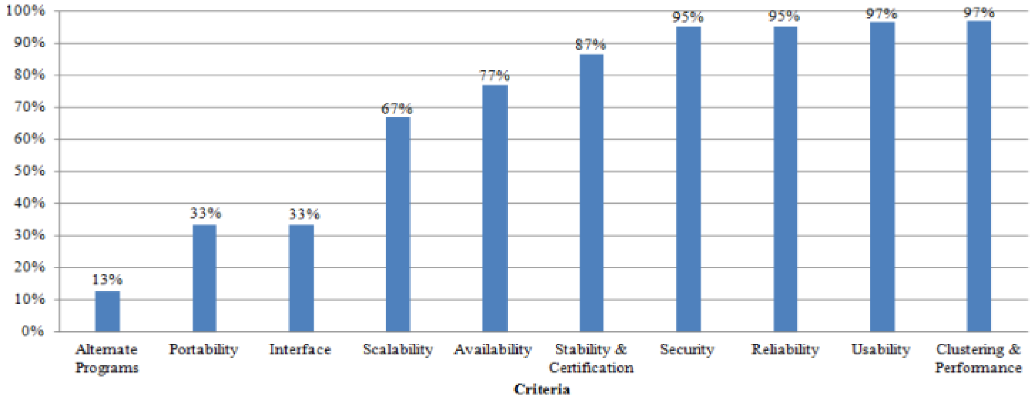

Figure 4. Criteria in RTOS used in supercomputing.

[1]

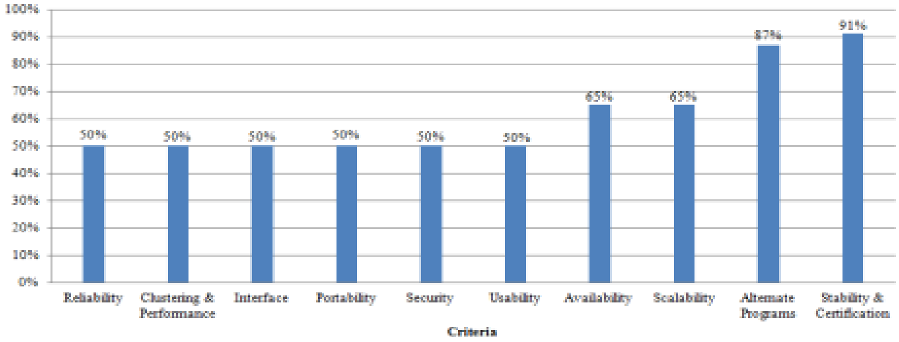

Figure 2. Criteria in RTOS used in manufacturing industry.

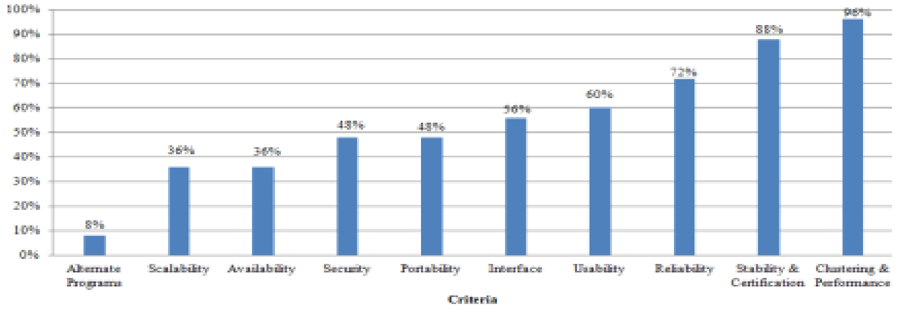

Figure 3. Criteria in RTOS used in other applications.

[1]

V. POPULAR RTOS

There are many types of RTOS [5] in current market but the popular ones are…..

i) **Windows CE:** It is an RTOS for handheld computers and mobile systems developed by Microsoft. CE stands for the properties i.e. compact, connectable, compatible, companion and efficient.

Features:

(1) It optimally functions with the devices that have minimal memory.

(2) Windows CE comply with standards of the definition of a real-time operating system, with deterministic interrupt latency. From Version 3 and onward, the system supports 256 priority levels and uses priority inheritance for dealing with priority inversion. The fundamental unit of execution is the thread. This helps to simplify the interface and improve execution time.

(3) A distinctive feature of Windows CE is that the source codeof the product is offered to the vendors so that can make their own adjustments and add more functionality.

ii) **OSEK:** It is an RTOS for automotive systems. In 1993, a German automotive company consortium (BMW, Robert Bosch GmbH, DaimlerChrysler, Opel, Siemens, and Volkswagen Group) and theUniversity of Karlsruhe found a RTOS named OSEK.. The French cars manufacturers Renault and PSA Peugeot Citroën, also had a similar project called VDX (Vehicle Distributed eXecutive), joined with the OSEK. Therefore, the official name is OSEK/VDX.

Features:

(1) The OSEK standard specifies interfaces to multitasking functions—generic I/O and peripheral access—and thus remains architecture dependent.

(2) OSEK systems can be run without memory protection on chips.OSEK implementation can be statistically configured during compile-time like the number of application tasks, stacks, mutexes etc. But it is not possible to create more at run time.

(3) OSEK supports two types of threads/tasks/compliance levels: basic tasks and enhanced tasks. There is no blocking in basic task; they "run to completion" (coroutine). There are sleep and block on event objects at Enhanced tasks. Certain tasks (basic and enhanced) or interrupt routines can trigger these events. Only static priorities are allowed for tasks. If there are tasks present with equal priority, the FIFO(First in First out) mechanism is used. There is also a mechanism to prevent deadlock and called priority ceiling.

iii) **LINUX 2.6.X and RTLINUS:**The embedded systems and real-time enhancements of that have LINUX 2.6.x as the popular RTOS, the enhancements provide high resolution timers, preemptive scheduling and preemptive interrupt service threads.It runs the entire Linux operating system as a fully preemptive process; hence it is termed as hard real time RTOS system which has the support of Micro Kernel. It is possible to control data acquisition systems,, robots, manufacturing plants, and other time-sensitive instruments and machines from RTLinux applications because of their hard real time property.

Features:

(1) It is a popular freeware like most of the popular LINUX OS.

(2) It has device driver features, has expandability of the kernel codes at Runtime and has provision of registering and deregistering device-driver modules.
(3) RT-Linux implements a POSIX API for a thread's manipulation

VI. Conclusion

This paper provides an insight about RTOS in embedded systems. This paper analyses the role of RTOS, evolution and compares different types of OS based on the criteria. This paper also covers the importance of OS in the embedded system.

References

[1] Real- time operating system(RTOS) with application to play models: HibaShahid, WadeeAlhalabi and John Reif

[2] Real-Time Operating Systems for Embedded: Yanbing Li, MiodragPotkonjak, and Wayne Wolf

[3] A Trusted, Scalable, Real-Time Operating System Environment :Douglas M. Wells

[4] Survey on RTOS: Evolution, Types and Current Research: Nandana V, Jithendran A, Shreelekshmi R

[5] Embedded systems – Architecture, programming and design (Textbook) – Raj Kamal – Tata McGraw-Hill