Scheduling Requirement For Better Performance In Real Time Embedded System

Anukriti kulshrestha
M.Tech. (VLSI), Suresh Gyan Vihar University Jaipur
Email: anukriti856@gmail.com

Abstract- In present times, it is very difficult to give a surity of the merging of application time constraints at the executing time, whenever we design and integrate an embedded software. The whole above examines the performance of some certain real time system. During the choice of OS for an embedded real time specific application, testing of OS services is required, which are characterized through a performance matrices using concerned parameters. The most important constraints included in the correction of real time performance of Linux comprises of scheduling latency & interrupt latency. This research paper gives an idea on the correct choice of scheme for application specific embedded system using scheduling.

1. INTRODUCTION:

The real time embedded system are utilized in number of fields such as, industries, defence, medical and many other fields.

Real time parameters are the most important parameters which are necessary to define the performance surity of these systems, it is the most particular characteristic having its effect over the entire system. The increasing technologies in the field of computer and electronics giving rise to a number of systems handled by computer systems, in present times. In today's environment, most of the computers are utilizing- Microsoft's windows/Linux, which are open source as OS(operating system) and not the real time OS. That's why they do not have sufficient real time parameters. Today most of the real time systems are embedded and on the other side constructed by embedded processors having embedded OS. The following paragraph are mentioning the "real time systems" which are embedded system in particular.

2. REAL TIME SYSTEMS (EMBEDDED SYSTEMS)

With the combination of real time and time sharing categories of the systems, there is a formation of hybrid OS, which gives the services of both the expectable real time tasks execution and non-real time operations with understandable platforms and a number of current applications. To get a comparatively low design and maintenance expenses, time sharing category in hybrid systems are according to accommodate OS as Linux.

The expenditure of energy is also a critical and most important design parameter for mobile embedded system, which are energized by battery and thermally constraints. In mobile embedded systems, the maximum power expenses is through displays LCD gives a very less compatability with energy consumption, as their panel utilizes a fixed power approximately, excluding the display expenses, whereas the outer illumination masters the power expenses of the system.

Rapidly growing technologies in the field of embeddes system applications made Linux a very important OS gradually. The requirement of the real time embedded system is to sense and respond or giving acknowledgement during the system task in the time limitation which was defined previously. A number of events in parallel tasks are executing at the similar time cycle, in parallel, in multitasking real time systems .To achieve the above requirements, it is necessary that each task get completed in time frame limitation.

3. MULTITASK MANAGEMENT:

Managing the multitasking process in various embedded system application, we engineers have to construct a program code for particular applications practically. So these specific software technologies provides less flexibility for code reusability, upgrades and maintenance, concluding a great problem.

In the real time system designs, the real time OS acts as a base for the reason of multitasking, real time kernel, fast responding machamism, scheduling and upto so on. These real time parameters minimize the developer's work load and corrects efficiency of the system upto a great extent, appropriate conveniency for upgrade and maintenance. Every system that utilizes real time OS is not a real time system in general. It just allow a platform for the construction of real time systems and is one of the most important part in real time systems fulfilling the crictical time task limitation needs.

4. REAL TIME OS(COMPONENTS AND PARAMETERS):

OS is sub-categorized as RTOS, it comprises a number of features showing similarity to general OS in various aspects. It acts as the major body in controlling and managing various hardware resources to get the hardware system active or available and allows higher application.

It controls the system resources, scheduling, execution and maintains a consistency for application code development.

Components – In general, many RTOS kernels comprises of following components:

Scheduler- Every kernel is having its heart existing in the scheduler. The main function of scheduler is to determine the task execution in various time cycles, it is encapsulated in the form of algorithms in the scheduler.

Objects- The most general and necessary objects of RTOS kernel are message queues, tasks and semaphores.

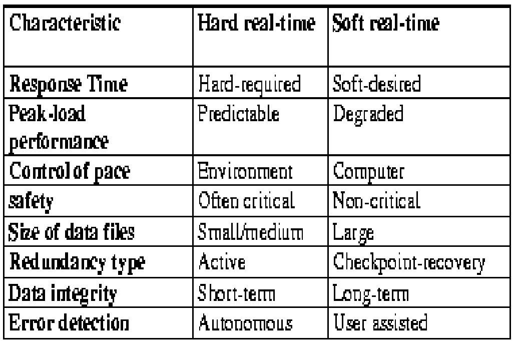
Services- In general, many kernel give the services which assist the developers in creating real time embedded applications. The services includes a set of API calls which can be utilized in performing tasks on kernel objects and also gives the following services:

(a)Time/counter management

(b)Interrupt handler

(c)Device input-output

(d)Memory organization



Characteristic	Hard real-time	Soft real-time
Response Time	Hard-required	Soft-desired
Peak-load performance	Predictable	Degraded
Control of pace	Environment	Computer
safety	Often critical	Non-critical
Size of data files	Small/medium	Large
Redundancy type	Active	Checkpoint-recovery
Data integrity	Short-term	Long-term
Error detection	Autonomous	User assisted

Table 1.1: comparison of hard and soft real time OS.

5. PARAMETERS

The parameters of real time systems characterize the following- In time limitation, responding to the events and completion of the task (event occurring) in concerned time limitation. Scheduling policy prioritizes the various tasks as high priority and low priority and allowing the higher priority task to execute first.

6. EFFECTING FACTORS OF RTOS CHARACTERISTICS:

A number of factors have their impact on real time systems. These factors include OS and their individual factors have critical roles comprising of multitask process management, context switching time, scheduling, interrupt handler time etc.

7. SCHEDULING POLICY AND PERFORMANCE DEPENDENCY:

As we know that, scheduling is procedure of giving priorities to various tasks during execution. The most common type of scheduling is preemptive scheduling which sets a great base for real time system, as it must have real time characteristics, thus increases the efficiency of the system. The most common algorithm for real time scheduling are rate-monotonic and earliest deadline priority.

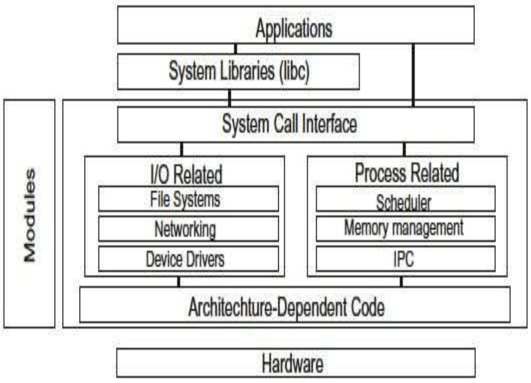


Fig 1.1: embedded operating system

8. CONCLUSION

There is a great need of correct choice for various particular applications, having a considerable effect on real time performance. The critical correlation among the events and tasks can create great system consumption during interprocess communication. Scheduling and task operations provide a flexible environment for the application of various algorithms for task queues. The real time system performance will be decreased by the task synchronization. Lastly the utilization of RTOS for implementation of correct application is the necessary ingredient for all developers of embedded system.

REFERENCES:

[1] Miao Liu et.all, “On Improving Real Time Interrupt Latencies of Hybrid Operating Systems with Two-Level Hardware Interrupts”,IEEE Transactions On Computers, Vol. 60, No 7, July 2011, pp. 978-991.

[2] Mian Dong and Lin Zhong, “Power Modeling and Optimisation for OLED Displays”, IEEE Transactions on Computers,Vol. 11, No 9, September 2012, pp. 1587-1599.

[3] L.I. Bing and L.I. Zhong-wen, “Analysis of Linux Real-time Mechanism”, Computer Technology and Development, vol. 17(09), Sep. 2007, pp. 41-44.

[4] B. J. Wang, M. S. Li and Z. G. Wang , “Uniprocessor static priority scheduling with limited priority levels”, Journal of Software, vol. 17(03), March 2006, pp. 602-610.

[5] S. Andrew, Tanenbaum, S. Albert and Woodhull, “Operating Systems Design and Implementation”（Third Edition）, Prentice Hall, January 04,2006.