PERFORMANCE ANALYSIS OF REAL TIME SCHEDULING ALGORITHMS USING CHEDDAR

Deepika Saxena[1]
SGI ,Sikar, Rajasthan,India

Abstract:

Nothing has changed the world around us the way digital technology and computers have. Computers have entered every aspect of our life and the environment around us. Real time systems are playing vital role in daily life. In just a few decades ago, real time systems are reshaping the way people live, work, and play. Real time systems come in an endless variety of types, each exhibiting unique characteristics. For example, most vehicles driven today embed intelligent computer chips that perform value-added tasks, which make the vehicles easier, cleaner, and more fun to drive. Even private homes are being filled with intelligent appliances, which facilitate and enhance everyday life. Real- Time systems span several domains of computer science.They are defense and space systems networked multimedia systems, embedded automotive electronics etc. The purpose of this chapter is to give an overview of the broad area of real-time systems. This task daunting because real time systems are everywhere and yet no generally accepted definition differentiates real-time systems from non–real-time systems.

A system is defined as being real-time if it is required to respond to input stimuli within a finite and specified time interval. The stimuli being either an event at the interface to the system or some internal clock tick that is, at least notionally, coordinated with the passage of time in the system's environment. Real-time systems are found in a wide range of applications areas, from simple domestic appliances to multi-media systems, large scale process control and safety critical avionics.

In some systems the required response times aremeasured in milliseconds, in others it is seconds or even minutes

Introduction

Real-time systems originated with the need to solve two main types of applications: event response, and closed loop control systems. Event response applications require a response to a stimulus in a determined amount of time, an example of such a system is an automotive airbag system. Closed loop control systems continuously process feedback inorder to adjust an output; an automotive cruise control system is an example of a closed-loop control system. Both of these types of systems require the completion of an operation within a specific deadline. Definition of "real time system" is "Any information processing activity or system which has to respond to externally generated input stimuli within a finite and specified delay".

Now, we can say that a **real-time system** is a system whose specification includes both logical and temporal correctness requirements.

Logical Correctness: Produces correctoutputs.

Temporal Correctness: Produces output at the right time.

It is mandatory to define embedded system with real time system. To most people, embedded systems are not recognizable as computers. Instead, they are hidden inside everyday objects that surround us and help us inour lives. Embedded systems typically do not interface with the outside world through familiar personal computer interface devices such as a mouse, keyboard and graphic user interface. Instead, they interface with the outside world through unusual interfaces such as sensors, actuators and specialized communication links. Realtime and embedded systems operate in constrained environments in which computer memory and processing power are limited. They often need to provide their services within strict time deadlines to their users and to the surrounding world. It is these memory, speed and timing constraints that dictate the use of real-time operating systems in embedded software. Here are some definitions of embedded system.

"An embedded system is a special-purpose computer system designed to perform one or a few dedicated functions, usually with real-time computing constrain".

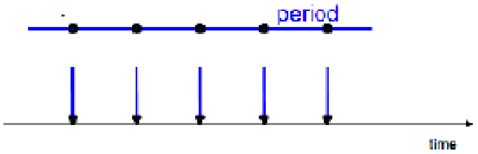
Classification of Real Time Systems:

The first classifications, hard real-time versus soft real-time, and fail-safe versus fail-operational, depend on the characteristics of the application, i.e., on factors outside the computer system. Third class of real time system is event-triggered versustime-triggered, depend on the design and implementation, i.e., on factors inside the computer system.

Types of Scheduling Jobs

Periodic Job: *In periodic tasks activity occurs repeatedly. For example, to monitor environment values, temperature, etc.*

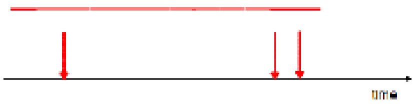
Figure I



Periodic Task

Aperiodic Job: *Aperiodic Tasks can occur any time. In aperiodic tasks no arrival time or pattern is given.*

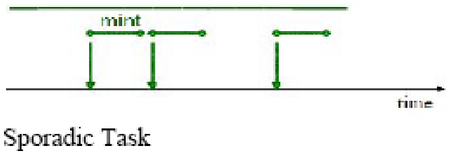
Figure II



Aperiodic Task

Sporadic Job: *Sporadic task can occur any time, but minimum time between arrivals of tasks is given.*

Figure III



Sporadic Task

Timing Constraints

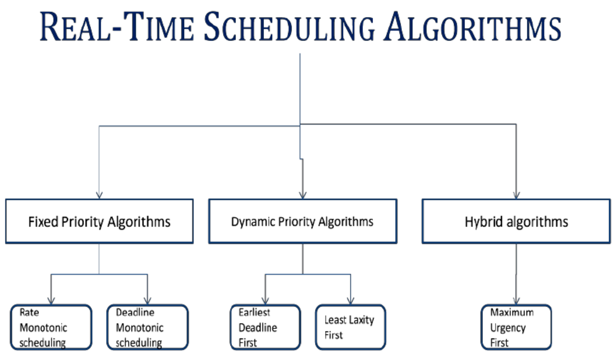
Real Time system is typically reactive and real-time in nature because it continuously has to react to the stimuli coming from its environment and it also has to perform this interaction u nd e restricted timing constraints.

These timing constraints are called the system's external timing constraints as they are imposed on the system's externalbehavior.In a typical embedded system design flow, the requirements specification phase describes what the system's external behavior must be without describing how the system works internally. The latter is expressed in the architectural design phase by means of a task graph, which describes a decomposition of the system into manageable components or tasks.

Since the system has many activities occurring in parallel, these tasks are usually concurrent.

The system's external timing constraints can be classified into two groups: those on the system a nd those on the environment. The former are called the required timing constraints or the timing requirements because the system is required to meet them, whereas the latter are called the given timing constraints because the environment rather than the system is required to meet them, that is, they are committed to the system.

Figure IV - Types of scheduling algorithms


REAL-TIME SCHEDULING ALGORITHMS
Fixed Priority Algorithms
Dynamic Priority Algorithms
Hybrid algorithms
Rate Monotonic scheduling
Deadline Monotonic scheduling
Earliest Deadline First
Least Laxity First
Maximum Urgency First

Real-Time System Application Domains

Potential uses for real-time systems include but are not limited to:

- Telecommunication systems
- Automotive control
- Multimedia servers and workstations
- Signal processing systems
- Radar systems
- Consumer electronics
- Process control
- Automated manufacturing systems
- Supervisory control and data acquisition (SCADA) systems.
- Electrical utilities
- Semiconductor fabrication systems
- Defense systems
- Avionics
- Air traffic control
- Autonomous navigation systems
- Vehicle control systems
- Transportation and traffic control systems
- Satellite systems
- Nuclear power control systems

Glossary of Terms and Concepts

Action The smallest decomposition of a segment of a response that cannot change resource allocation.

A periodic event
An event sequence whose arrival pattern is not periodic.

Resource
A physical entity such as a processor, a backplane bus, network link, or a network router which can be used by one or more actions. A resource may have allocation policy (such as rate-monotonic scheduling) and a data-sharing policy.

Reference

[1] Audsley and A. Burns, Department of computer Science, University of York, UK
[2] Abeni, L., and Buttazzo, G. 1998. Integrating multimedia applications in hard real-time systems.
[3] Abeni, L., and Buttazzo, G. 2004. Dynamic real-time systems.
[4] A. Aldarmi and Alan Burns, Real-Time Systems, Group Department of Computer Science,The University of York, YO10 5DD,U.K. November 1998
[5] Arezou Mohammadi and Selim G. Akl, Scheduling Algorithms for Real-Time System Technical Report No.2005-499, School of Computing, Queen's University
[6] Buttazzo,G.C.2003. Rate monotonic vs. EDF: Judgment day.
[7] C.L .Liu, J.W.Layland, Algorithms in Multiprogramming in a Hard Real Time Environment.
[8] D. Stewert, Introduction to Real Time Scheduling Theory, Spring 2000.
[9] Ed. Overton, A Foray into uniprocessor Real Time Scheduling Algorithms and Intractibility December 3,,1997