CHAPTER 1

Fixed Priority Scheduling Theory for Hard Real-Time Systems

John P. Lehoczky², Lui Sha³, J. K. Strosnider⁴ and Hide Tokuda⁵
Carnegie Mellon University
Pittsburgh, PA 15213-3890

Abstract

This paper summarizes recent research results on the theory of fixed priority scheduling of hard real-time systems developed by the Advanced Real-Time Technology (ART) Project of Carnegie Mellon University. The Liu and Layland theory of rate monotonic scheduling is presented along with many generalizations including an exact characterization of task set schedulability, average case behavior and allowance for arbitrary task deadlines. Recent research results including the priority ceiling protocol which provides predictable scheduling when tasks synchronize and the deferrable and sporadic server algorithms which provide fast response times for aperiodic tasks while preserving periodic task deadlines are also presented.

1 Introduction

Real-time computer systems are used for the monitoring and control of physical processes. Unlike general purpose computer systems, the dynamics of the underlying physical process place explicit timing requirements on individual tasks which must be met in order to insure the

---

¹Sponsored in part by the Office of Naval Research under contract N00014-84-K-0734, in part by the Naval Ocean Systems Center under contract N66001-87-C-0155, and in part by the Systems Integration Division of IBM Corporation under University Agreement Y-278067.
²Department of Statistics
³Software Engineering Institute and School of Computer Science
⁴Department of Electrical and Computer Engineering
⁵School of Computer Science
correctness and safety of the real-time system. Historically, hand-crafted techniques were used to insure the timing correctness by statically binding task executions to fixed slots via timelines. This *ad hoc* approach tended to result in systems which were not only expensive to develop, but also extremely difficult and costly to upgrade and maintain. The Advanced Real-Time Technology (ART) Project at Carnegie Mellon University has been developing algorithmic-based scheduling solutions for real-time computing that guarantee individual task execution times in multi-tasking, interrupt driven environments. Unlike earlier timeline scheduling approaches, the scheduling theory developed by the ART Project ensures the timing correctness of real-time tasks without the costly handcrafting and exhaustive testing associated with the use of timelines. Furthermore, this algorithmic-based scheduling approach is designed to support dynamic, process-level reconfiguration, an important approach to achieving high system dependability with limited hardware resources.

Embedded real-time systems must schedule diverse activities to meet the timing requirements imposed by the physical environments. While this may be a difficult problem, the real-time system developer works in a controlled environment, and typically has the advantage of knowing the entire set of tasks that are to be processed by the system. An algorithmic-based scheduling methodology allows the developer to combine task characterization data with its associated timing requirements to determine through the use of formulas whether or not the task set is schedulable. This capability readily lends itself to automation allowing the system developer not only to quickly determine the timing correctness of current processing requirements, but also to be able to assess the timing impact of future system upgrades and modifications. The scheduling theory makes the timing properties of the system *predictable*. This means that one is able to determine analytically whether the timing requirements of a task set will be met, and if not, which task timing requirements will fail.

The Advanced Real-Time Technology Project of Carnegie Mellon University has been developing and testing a theory of predictable hard real-time scheduling based on fixed priority scheduling algorithms. An important aspect of the project is the development of a real-time operating system, ARTS, and a real-time tool set, both developed by H. Tokuda