



A Real-Time Linux for Multicore Platforms

Department of Computer Science

University of North Carolina at Chapel Hill

August 2010

The Challenge

Existing operating systems that support real-time applications are rooted in resource-allocation policies designed for uniprocessors. Such policies are no longer sufficient in a multicore world. This project is part of a continuing effort (funded to date by ARO) to analyze different real-time policy choices within a real working operating system (OS). This effort has resulted in the development of a testbed system called LITMUS^{RT}, which is an extension of Linux that allows different (multiprocessor) real-time scheduling and synchronization policies to be specified as plugin components. As its name suggests, LITMUS^{RT} was originally designed as a *testbed* for evaluating different real-time-related policy choices. The challenge being addressed in this project is to extend LITMUS^{RT} to obtain a fully-featured OS capable of supporting actual real-time workloads.

The Approach

LITMUS^{RT} currently provides plugins for several real-time scheduling and synchronization algorithms. The scheduling algorithms that are provided can be used to host either hard or soft real-time task systems. In a hard real-time application, task deadlines cannot be missed. In a soft real-time application, some deadline tardiness is permissible. In the definition of “soft real-time” that we have focused on, deadline tardiness is required to be (provably) bounded. While the same scheduling algorithm can be used to schedule both types of applications (hard and soft), different analysis is required to prove real-time correctness for hard and soft real-time applications, as zero tardiness is required in the former case, but merely bounded tardiness in the latter. In our past LITMUS^{RT}-based research, various real-time scheduling and synchronization have been prototyped and evaluated using *synthetic* task systems (not real application workloads).

In order to evolve LITMUS^{RT} so that it can host real applications, a number of extensions are required. Planned extensions include: new adaptive real-time scheduling policies for dealing with dynamic workloads and system overloads; modifications to support

systems in which hard real-time, soft real-time, and non-real-time components are co-present; and extensions for supporting I/O and enforcing precedence constraints in real-time tasks. The resulting LITMUS^{RT} system will differ from existing commercial OSs in that its design will be rooted in real-time resource-allocation mechanisms that *explicitly* target multicore platforms. The efficacy of the resulting system will be demonstrated via two case-study experimental efforts, one involving avionics workloads, and another involving real-time stream warehouses (joint work with AT&T Research).

Significance

Multicore platforms provide significant processing power that could potentially be harnessed to support complex, resource-demanding real-time applications. However, for this to happen, appropriate real-time scheduling and synchronization algorithms must be devised, analyzed, and implemented within a working OS. This is precisely the objective of this project.

Project Members

James Anderson, professor (PI)
Sanjoy Baruah, professor (Co-PI)

Research Sponsor

U.S. Army Research Office

For More Information

Dr. James Anderson
Department of Computer Science
University of North Carolina at Chapel Hill
CB#3175, Frederick P. Brooks, Jr. Building
Chapel Hill, NC 27599-3175
Phone: (919) 962-1757
Fax: (919) 962-1799
E-mail: anderson@cs.unc.edu

<http://www.cs.unc.edu/~anderson/projects/rtmlinux.html>