Chapter 15
Real-Time Attributes in Operating Systems

Tommaso Cucinotta
Scuola Superiore Sant’Anna, Italy

Spyridon V. Gogouvitis
National Technical University of Athens, Greece

ABSTRACT
General-Purpose Operating Systems (GPOSes) are being used more and more extensively to support interactive, real-time, and distributed applications, as found in the multimedia domain. In fact, the wide availability of supported multimedia devices and protocols, together with the wide availability of libraries and tools for handling multimedia contents, make them an almost ideal platform for the development of this kind of complex applications. However, contrarily to Real-Time Operating Systems, General-Purpose ones used to lack some important functionality needed for providing proper scheduling guarantees to application processes. Recently, the increasing use of GPOSes for multimedia applications is gradually pushing OS developers towards enriching the kernel of a GPOS so as to provide more and more real-time functionality, thus enhancing the performance and responsiveness of hosted time-sensitive applications. In this chapter, an overview is performed over the efforts done in the direction of enriching GPOSes with real-time capabilities, with a particular focus on the Linux OS. Due to its open-source nature and wide diffusion and availability, Linux is one of the most widely used OSes for such experimentations.

INTRODUCTION
New generation interactive distributed applications have significant demands on processing, storage and networking capabilities, as well as stringent timing requirements, such as various multimedia, virtual collaboration and e-learning applications. These are, among others, types of applications that have been undergoing in the last few years a very steep demand on the side of end-to-end interactivity level, response time, and throughput. This is also due to the growing availability of affordable broadband Internet connections. This allowed people to realize that most of the activities they used to carry on through off-line workflows (especially when relatively high
data volumes were involved) now could be done within on-line collaborative systems, in a much more interactive way.

For a long time, these applications have been made available to end-users in the form of costly professional solutions running on dedicated hardware. However, the growth in computation capabilities of computing systems has led to a growing interest in providing such applications in the form of applications that may be run on General-Purpose Operating Systems (GPOSes). This allows for sharing a set of physical resources among different applications or application instances, thus scaling down the costs needed to deploy and run them.

Unfortunately, GPOSes are not designed to provide the run-time support that is necessary for meeting timing requirements of individual applications. In fact, in the traditional hard real-time domain, a Real-Time Operating System (RTOS) is used for such purpose. A RTOS usually provides those features that allow for a well-known, predictable and analysable timing behaviour of hosted applications: all kernel segments are characterized through well-known worst-case durations, scheduling latencies and interrupt latencies which may be controlled through the appropriate tuning of the interrupt and process schedulers. A set of real-time scheduling policies is available for the system designer and programmers. Time may be measured (and timers may fire) with a high precision (typically sub-millisecond) while there are tools available for Worst-Case Execution Time (WCET) estimation and (off-line) analysis, as well as for schedulability analysis. Unfortunately, such OSes are designed for embedded control applications, thus they imply serious constraints on the supported hardware and available high-level software infrastructures.

Therefore, in recent years, various efforts have been done towards the integration of real-time technologies within GPOSes, particularly within Linux, for the availability of its kernel as open-source and its worldwide diffusion.

Various drawbacks make the Linux kernel, as well as most of the GPOSes, particularly unsuitable for running real-time applications: the monolithic structure of the kernel and the wide variety of drivers that may be loaded within; the impossibility to keep under control all the non-preemptable sections possibly added by such drivers; the general structure of the interrupt management core framework that privileges portability with respect to latencies; and others.

As a result, the latency experienced by time-sensitive activities can be as large as tens or hundreds of milliseconds. This makes GPOSes non-suitable for hard real-time applications with tight timing constraints. Though, soft real-time applications may run quite well within such an environment, especially when the original kernel is modified so as to integrate the necessary real-time capabilities.

Objective of this chapter is to provide an overview of the approaches appeared in the research literature aiming to integrate real-time capabilities within General-Purpose Operating Systems, with a particular focus on Linux.

**BACKGROUND**

Although not being a real-time system, the 2.6 Linux kernel includes a set of features making it particularly suitable for soft real-time applications. First, it is a fully preemtable kernel, like most of the existing real-time operating systems, and a lot of effort has been spent on reducing the length of non-preemptable sections (the major source of kernel latencies). It is noteworthy that the 2.6 kernel series introduced a new scheduler with a bounded execution time, resulting in a highly decreased scheduling latency. Also, in the latest kernel series, a modular framework has been introduced that allows for an easier integration of other scheduling policies. Second, although being a general-purpose time-sharing kernel, it includes such real-time extensions to the POSIX
11 more pages are available in the full version of this document, which may be purchased using the "Add to Cart" button on the product’s webpage: 
www.igi-global.com/chapter/real-time-attributes-operating-systems/55253?camid=4v1

www.igi-global.com/e-resources/library-recommendation/?id=1

Related Content

Extending Dynamic Scheduling Policies in WorkflowSim by Using Variance based Approach
www.igi-global.com/article/extending-dynamic-scheduling-policies-in-workflowsim-by-using-variance-based-approach/153971?camid=4v1a

High Performance Computing Design by Code Migration for Distributed Desktop Computing Grids
www.igi-global.com/chapter/high-performance-computing-design-code/69035?camid=4v1a

An Intelligent Sensor Placement Method to Reach a High Coverage in Wireless Sensor Networks
www.igi-global.com/article/intelligent-sensor-placement-method-reach/56356?camid=4v1a

Grid Access Control Models and Architectures
www.igi-global.com/chapter/grid-access-control-models-architectures/58746?camid=4v1a