DORITOS (Distributed Object-based Real-time InsTructional Operating System): A complete package for teaching principles and practices of real-time operating system

Jae C. Oh and Daniel Mossé
Department of Computer Science, University of Pittsburgh, Pittsburgh, PA 15260.
{joh, mosse}@cs.pitt.edu

I. Introduction

We are developing a teaching package that can be used in a college course that would fill a gap among current science majors and teach senior-level undergraduate students theory and practices of real-time operating systems (RTOS); these include requirements, characteristics, internals, and specification of such systems. This course has two components: (1) a theoretical part, and (2) a practical hands-on implementation component achieved with DORITOS (Distributed Object-Based Real-time InsTructional Operating System) as the implementation environment. DORITOS' design is based on UC-Berkeley's NACHOS [1]. DORITOS package comes with DORITOS basic system which allows students to run simple threads. Throughout the term, assignments will require students to use and modify DORITOS to implement real-time elements as well as to analyze performances of implemented algorithms. The theoretical component will be a text book and a laboratory manual that accompany DORITOS.

II. Developing DORITOS

We chose Java as language for developing DORITOS. Therefore, students are to use Java to do their programming projects. There are pedagogical as well as practical reasons for choosing Java as the language to be used:

- *Portability of DORIOTS:* Since Java is platform independent. Therefore, students in various institutions will have less problems in installing and using DORITOS in fast changing computing environment.
- *Accessibility:* There exist free Java Virtual Machines (JVM) – for example, Kaffe by T. J. Wilkins & Assoc. [2] – with source code that can be modified. Since we need precise timing mechanism to emulate real-time systems in great detail, it was necessary to modify JVM to support mechanisms of DORITOS. Kaffe comes with "Berkeley-type license" so that it can be modified and can be distributed freely.
- *Object-Oriented Nature of Java:* Java supports strongly-typed language and make it easy to reuse software components. Therefore, students using DORITOS will utilize their time more wisely, with less debugging of unnecessary mistakes and, therefore, spending more time in learning principles of real-time systems [3].
- *Better Network Support of Java:* Network classes and libraries are well supported in Java and easier to use than in other conventional languages. DORITOS assignments that implement a RT network will take advantage of the network capabilities of Java.
- *Popularity and widespread use of Java.*

III. Project Status and Future Work

Currently, the alpha version of DORITOS system has been completed. It supports threading, synchronization, execution of multiple user threads, memory management including a primitive form of paging, system call handling, and timer device. Also modified JVM (called DKaffe – DORITOS Kaffe) has been completed. We still need to implement the RT file system and RT network capabilities for DORITOS. We expect to be able to take advantage of relatively simple-to-use Java socket libraries in developing the network capabilities.

The programming assignments that accompany DORITOS are to implement various real-time system components including scheduling, file system, and networking. Three assignments that teach the following concepts are completed: real-time scheduling algorithms (deadline, and memory), and measuring execution times of user programs.

So far, DORITOS has been tested only on Linux-Intel machines. However, since DORITOS runs on DKaffe, a Java virtual machine, it should run on other platforms once DKaffe has been compiled for the target platform.

DORITOS has not been field tested in classes yet as of May 1998; however, we plan to use DORITOS for an undergraduate operating system course in Fall term 1998 in the Computer Science Department at the University of Pittsburgh. In Spring term 1999, we plan to use DORITOS for a real-time operating systems course. Lessons learned from these courses will be used to improve DORITOS and to write a laboratory manual. We plan to release a beta test version for external testing sometime early 1999.

References

[1] W. A. Christopher, S. J. Procter, and T. E. Anderson, *The Nachos Instructional Operating System*, available via ftp `sprite.berkeley.edu`, file `nachos/nachos-3.0.tar`.
[2] T. J. Wilkins & Associates, *Kaffe: A free virtual machine to run Java code*, available via `http://www.kaffe.org`
[3] W. F. Tichy, *Programming-in-the-large: Past, Present, and Future*, Proceedings of the 14th International Conference on Software Engineering, 362–367, 1992.

This research is being funded by National Science Foundation