2007

Real-time operating system in Java

Qinghua Lu
University of Wollongong

Real-time Operating System in Java

A thesis submitted in fulfilment of the requirements for the award of the degree

Master of Computer Science -Research

from

UNIVERSITY OF WOLLONGONG

by

Qinghua Lu

School of Computer Science & Software Engineering

August, 2007

Dedicated to

My Parents,

Lu Changyou and Luo Xiue!

The following papers were written as part of this research.

1. M[c]Kerrow, P.J., Lu, Q., Zhou, Z.Q. and Chen, L. (2007), Developing real-time systems in Java on Macintosh, ***Submitted to AUC'07***, Apple University Consortium, Gold Coast, September, 23-26, 2007.

2. M[c]Kerrow, P.J., Lu, Q., Zhou, Z.Q. and Chen, L. (2007), Software development of embedded systems on Macintosh, ***Submitted to AUC'07***, Apple University Consortium, Gold Coast, September, 23-26, 2007.

Declaration

I, Qinghua Lu, declare that this thesis, submitted in fulfilment of the requirements for the award of Master of Computer Science -Research, in the School of Computer Science & Software Engineering, University of Wollongong, is wholly my own work unless otherwise referenced or acknowledged. The document has not been submitted for qualifications at any other academic institution.

Qinghua Lu
31 August 2007

Table of Contents

List of Abbreviations

API	Application Programming Interface
CPU	Central Processing Unit
EDF	Earliest-deadline-first
IDE	Integrated Development Environment
I/O	Input/Output
JVM	Java Virtual Machine
Mac OS	The Macintosh Operating System
Mac OSX	the latest version of the Mac OS
MMU	Memory Management Unit
msec	millisecond
msg.	message
no.	number
OS	Operating System
proc.	process
PC	Program Counter
rti	return from interrupt
RM	Rate-Monotonic
RTOS	Real-time Operating System
RTOSes	Real-time Operating Systems
Sun SPOT	Sun Small Programmable Object Technology
TCB	Task Control Block
TCP/IP	Transmission Control Protocol/Internet Protocol
TINI	Tiny InterNet Interface
TNI	TINI Native Interface
UAV	Unmanned Aerial Vehicle
UML	Unified Modeling Language

Abstract

Real-time operating systems (RTOSes) are required to run for years, and never fail, without human intervention. Safety is the primary concern for RTOSes because they usually control physical equipment. One strand of real-time operating system (RTOS) research is looking at the question: can developing an RTOS in a safe language result in a system that an errant process can't crash? Choosing a good programming language can significantly improve the safety of the RTOS. In this thesis, we examine the advantages and associated problems of writing RTOSes in a safe language, namely Java.

We design an RTOS named JARTOS that schedules processes on a micro-controller called TINI. The code of the JARTOS system is mainly written in Java, since Java provides both static and dynamic safety. The Java compiler handles potentially unsafe operations rather than the programmer. Also, Java includes run-time support to catch and handle run-time errors.

JARTOS is designed to be a time-sharing system, where cooperative multiprocessing is used to schedule real-time processes. JARTOS switches processes on a timer interrupt. Each process is required to execute quickly and then give up the processor. Otherwise it will be timed out. To implement a timeout, JARTOS supports a timer interrupt that regularly updates a clock and checks for timeouts. To keep the number of interrupts to a minimum, input/output is done using polling where possible. Also, interrupts code is designed to be transparent to the processes. An interrupt handler sets flags and values, and then returns to the process it interrupted.

In the context of achieving real-time performance, we look at the issues of implementing our system design in Java. We introduce how we used Java constructs to implement the design of JARTOS, and how we solved the low-level issues.

RTOSes have to guarantee that real-time processes execute within specified time deadlines. Loss of synchronization can occur when deadlines are not met. Timing problems are often very difficult to find. In JARTOS, we designed a set of performance measurements to investigate timing problems. These performance measurements are carefully designed to provide the right information at minimal cost in performance. Performance of TINI and JARTOS are measured and discussed.

Acknowledgments

First, I would like to express my deepest appreciation to my supervisor, A./Prof. Phillip McKerrow, for his guidance, patience, support and enthusiasm during the process of this thesis.

I would like to thank my co-supervisor, Dr. Zhi Quan Zhou, for guidance, patience and help during my study.

Thanks also to my colleagues Sherine Antoun and Li Chen for their help and friendship.

From the bottom of my heart, I would like to express my special thanks to my parents, Lu Changyou and Luo Xiue, for their endless love, encouragement and support during my study and all of my life.

Finally, special thanks to my fiancé, Qiao Shuaichang, for his endless love, helpful advice. Without his support, the work in this thesis would not have been possible.