
HARD REAL-TIME COMPUTING SYSTEMS

*Predictable Scheduling
Algorithms and Applications*

THE KLUWER INTERNATIONAL SERIES IN ENGINEERING AND COMPUTER SCIENCE

REAL-TIME SYSTEMS

Consulting Editor

John A. Stankovic

REAL-TIME SYSTEMS: *Design Principles for Distributed Embedded Applications*, by Hermann Kopetz, ISBN: 0-7923-9894-7

REAL-TIME DATABASE SYSTEMS: *Issues and Applications*, edited by Azer Bestavros, Kwei-Jay Lin and Sang Hyuk Son, ISBN: 0-7923-9897-1

FAULT-TOLERANT REAL-TIME SYSTEMS: *The Problem of Replica Determinism*, by Stefan Poledna, ISBN: 0-7923-9657-X

RESPONSIVE COMPUTER SYSTEMS: *Steps Toward Fault-Tolerant Real-Time Systems*, by Donald Fussell and Mirosław Malek, ISBN: 0-7923-9563-8

IMPRECISE AND APPROXIMATE COMPUTATION, by Swaminathan Natarajan, ISBN: 0-7923-9579-4

FOUNDATIONS OF DEPENDABLE COMPUTING: *System Implementation*, edited by Gary M. Koob and Clifford G. Lau, ISBN: 0-7923-9486-0

FOUNDATIONS OF DEPENDABLE COMPUTING: *Paradigms for Dependable Applications*, edited by Gary M. Koob and Clifford G. Lau, ISBN: 0-7923-9485-2

FOUNDATIONS OF DEPENDABLE COMPUTING: *Models and Frameworks for Dependable Systems*, edited by Gary M. Koob and Clifford G. Lau, ISBN: 0-7923-9484-4

THE TESTABILITY OF DISTRIBUTED REAL-TIME SYSTEMS,
Werner Schütz, ISBN: 0-7923-9386-4

A PRACTITIONER'S HANDBOOK FOR REAL-TIME ANALYSIS: *Guide to Rate Monotonic Analysis for Real-Time Systems*, Carnegie Mellon University (Mark Klein, Thomas Ralya, Bill Pollak, Ray Obenza, Michale González Harbour); ISBN: 0-7923-9361-9

FORMAL TECHNIQUES IN REAL-TIME FAULT-TOLERANT SYSTEMS, J. Vytopil, ISBN: 0-7923-9332-5

SYNCHRONOUS PROGRAMMING OF REACTIVE SYSTEMS, N. Halbwachs, ISBN: 0-7923-9311-2

REAL-TIME SYSTEMS ENGINEERING AND APPLICATIONS, M. Schiebe, S. Pferrer, ISBN: 0-7923-9196-9

SYNCHRONIZATION IN REAL-TIME SYSTEMS: *A Priority Inheritance Approach*, R. Rajkumar, ISBN: 0-7923-9211-6

CONSTRUCTING PREDICTABLE REAL TIME SYSTEMS, W. A. Halang, A. D. Stoyenko, ISBN: 0-7923-9202-7

FOUNDATIONS OF REAL-TIME COMPUTING: *Formal Specifications and Methods*, A. M. van Tilborg, G. M. Koob, ISBN: 0-7923-9167-5

FOUNDATIONS OF REAL-TIME COMPUTING: *Scheduling and Resource Management*, A. M. van Tilborg, G. M. Koob, ISBN: 0-7923-9166-7

REAL-TIME UNIX SYSTEMS: *Design and Application Guide*, B. Furht, D. Grostick, D. Gluch, G. Rabbat, J. Parker, M. McRoberts, ISBN: 0-7923-9099-7

**HARD REAL-TIME
COMPUTING SYSTEMS**
*Predictable Scheduling
Algorithms and Applications*

by

Giorgio C. Buttazzo
Scuola Superiore S. Anna
Pisa, Italy



KLUWER ACADEMIC PUBLISHERS
Boston / Dordrecht / London

Distributors for North America:

Kluwer Academic Publishers
101 Philip Drive
Assinippi Park
Norwell, Massachusetts 02061 USA

Distributors for all other countries:

Kluwer Academic Publishers Group
Distribution Centre
Post Office Box 322
3300 AH Dordrecht, THE NETHERLANDS

Library of Congress Cataloging-in-Publication Data

A C.I.P. Catalogue record for this book is available
from the Library of Congress.

Copyright © 1997 by Kluwer Academic Publishers

All rights reserved. No part of this publication may be reproduced, stored in a retrieval system or transmitted in any form or by any means, mechanical, photocopying, recording, or otherwise, without the prior written permission of the publisher, Kluwer Academic Publishers, 101 Philip Drive, Assinippi Park, Norwell, Massachusetts 02061

Printed on acid-free paper.

Printed in the United States of America

CONTENTS

Preface	ix
1 A GENERAL VIEW	1
1.1 Introduction	1
1.2 What does real time mean?	4
1.3 Achieving predictability	12
2 BASIC CONCEPTS	23
2.1 Introduction	23
2.2 Types of task constraints	25
2.3 Definition of scheduling problems	34
2.4 Scheduling anomalies	44
3 APERIODIC TASK SCHEDULING	51
3.1 Introduction	51
3.2 Jackson's algorithm	52
3.3 Horn's algorithm	56
3.4 Non-preemptive scheduling	61
3.5 Scheduling with precedence constraints	68
3.6 Summary	74
4 PERIODIC TASK SCHEDULING	77
4.1 Introduction	77
4.2 Rate Monotonic scheduling	82
4.3 Earliest Deadline First	93
4.4 Deadline Monotonic	96
4.5 EDF with deadlines less than periods	102
4.6 Summary	107

5	FIXED-PRIORITY SERVERS	109
5.1	Introduction	109
5.2	Background scheduling	110
5.3	Polling Server	111
5.4	Deferrable Server	116
5.5	Priority Exchange	125
5.6	Sporadic Server	132
5.7	Slack stealing	138
5.8	Non-existence of optimal servers	142
5.9	Performance evaluation	145
5.10	Summary	146
6	DYNAMIC PRIORITY SERVERS	149
6.1	Introduction	149
6.2	Dynamic Priority Exchange Server	150
6.3	Dynamic Sporadic Server	155
6.4	Total Bandwidth Server	159
6.5	Earliest Deadline Late Server	163
6.6	Improved Priority Exchange Server	167
6.7	Improving TBS	171
6.8	Performance evaluation	175
6.9	Summary	178
7	RESOURCE ACCESS PROTOCOLS	181
7.1	Introduction	181
7.2	The priority inversion phenomenon	182
7.3	Priority Inheritance Protocol	186
7.4	Priority Ceiling Protocol	201
7.5	Stack Resource Policy	208
7.6	Summary	221
8	HANDLING OVERLOAD CONDITIONS	225
8.1	Introduction	225
8.2	Load definitions	228
8.3	Performance metrics	230
8.4	Scheduling schemes for overload	243
8.5	Performance evaluation	249

9	KERNEL DESIGN ISSUES	253
9.1	Structure of a real-time kernel	253
9.2	Process states	256
9.3	Data structures	261
9.4	Miscellaneous	265
9.5	Kernel primitives	271
9.6	Intertask communication mechanisms	289
9.7	System overhead	296
10	APPLICATION DESIGN ISSUES	301
10.1	Introduction	302
10.2	Time constraints definition	306
10.3	Hierarchical design	313
10.4	A robot control example	318
11	EXAMPLES OF REAL-TIME SYSTEMS	323
11.1	Introduction	323
11.2	MARS	325
11.3	Spring	331
11.4	RK	336
11.5	ARTS	340
11.6	HARTIK	345
	Glossary	353
	REFERENCES	363
	INDEX	373

PREFACE

Real-time computing plays a crucial role in our society since an increasing number of complex systems rely, in part or completely, on processor control. Examples of applications that require real-time computing include nuclear power plants, railway switching systems, automotive electronics, air traffic control, telecommunications, robotics, and military systems.

In spite of this large application domain, most of the current real-time systems are still designed and implemented using low-level programming and empirical techniques, without the support of a scientific methodology. This approach results in a lack of reliability, which in critical applications may cause serious environmental damage or even loss of life.

This book is a basic treatise on real-time computing, with particular emphasis on predictable scheduling algorithms. The main objectives of the book are to introduce the basic concepts of real-time computing, illustrate the most significant results in the field, and provide the basic methodologies for designing predictable computing systems useful in supporting critical control applications.

The book is written for instructional use and is organized to enable readers without a strong knowledge of the subject matter to quickly grasp the material. Technical concepts are clearly defined at the beginning of each chapter, and algorithm descriptions are reinforced through concrete examples, illustrations, and tables.

Contents of the chapters

Chapter 1 presents a general introduction to real-time computing and real-time operating systems. It introduces the basic terminology and concepts used in the book and clearly illustrates the main characteristics that distinguish real-time processing from other types of computing.

Chapter 2 treats the general issue of scheduling tasks on a single processor system. Objectives, performance metrics, and hypotheses are clearly presented, and the scheduling problem is precisely formalized. The different algorithms proposed in the literature are then classified in a taxonomy, which provides a useful reference framework for understanding the different approaches. At the end of the chapter, a number of multiprocessor scheduling anomalies are illustrated to show that real-time computing is not equivalent to fast computing.

The rest of the book is dedicated to specific scheduling algorithms, which are presented as a function of the task characteristics.

Chapter 3 introduces a number of real-time scheduling algorithms for handling aperiodic tasks with explicit deadlines. Each algorithm is examined in regard to the task set assumptions, formal properties, performance, and implementation complexity.

Chapter 4 treats the problem of scheduling a set of real-time tasks with periodic activation requirements. In particular, three classical algorithms are presented in detail: Rate Monotonic, Earliest Deadline First, and Deadline Monotonic. A schedulability test is derived for each algorithm.

Chapter 5 deals with the problem of scheduling hybrid sets consisting of hard periodic and soft aperiodic tasks, in the context of fixed-priority assignments. Several algorithms proposed in the literature are analyzed in detail. Each algorithm is compared with respect to the assumptions made on the task set, its formal properties, its performance, and its implementation complexity.

Chapter 6 considers the same problem addressed in Chapter 5, but in the context of a dynamic priority assignment. Performance results and comparisons are presented at the end of the chapter.

Chapter 7 introduces the problem of scheduling a set of real-time tasks that may interact through shared resources and hence have both time and resource constraints. Three important resource access protocols are described in detail: the Priority Inheritance Protocol, the Priority Ceiling Protocol, and the Stack Resource Policy. These protocols are essential for achieving predictable behavior, since they bound the maximum blocking time of a process when accessing shared resources. The latter two protocols also prevent deadlocks and chained blocking.

Chapter 8 deals with the problem of real-time scheduling during transient overload conditions; that is, those situations in which the total task demand exceeds

the available processing time. These conditions are critical for real-time systems, since not all tasks can complete within their timing constraints. This chapter introduces new metrics for evaluating the performance of a system and presents a new class of scheduling algorithms capable of achieving graceful degradation in overload conditions.

Chapter 9 describes some basic guidelines that should be considered during the design and the development of a hard real-time kernel for critical control applications. An example of a small real-time kernel is presented. The problem of time predictable intertask communication is also discussed, and a particular communication mechanism for exchanging asynchronous messages among periodic tasks is illustrated. The final section shows how the runtime overhead of the kernel can be evaluated and taken into account in the guarantee tests.

Chapter 10 discusses some important issues related to the design of real-time applications. A robot control system is considered as a specific example for illustrating why control applications need real-time computing and how time constraints can be derived from the application requirements, even though they are not explicitly specified by the user. Finally, the basic set of kernel primitives presented in Chapter 9 is used to illustrate a concrete programming example of real-time tasks for sensory processing and control activities.

Chapter 11 concludes the book by presenting a number of hard real-time operating systems proposed in the literature. The systems examined include MARS, Spring, RK, ARTS, and HARTIK. Each system is considered in terms of supported architecture, scheduling algorithm, communication mechanism, and interrupt handling.

Acknowledgments

This work is the result of seven years of research and teaching activity in the field of real-time control systems. The majority of the material presented in this book is based on class notes for an operating systems course taught at the University of Pisa.

Though this book carries the name of a single author, it has been positively influenced by a number of people to whom I am indebted. Foremost, I would like to thank my students at the University of Pisa, who have directly and indirectly contributed to its readability and clarity.

A personal note of appreciation goes to Paolo Ancilotti, who gave me the opportunity to teach these topics. Moreover, I would like to acknowledge the contributions of Jack Stankovic, Krithi Ramamritham, Herman Kopetz, John Lehoczky, and Gerard Le Lann. Their input enhanced the overall quality of this work. I would also like to thank the Kluwer editorial staff, and especially Bob Holland, for the support I received during the preparation of the manuscript.

Special appreciation goes to Marco Spuri, who gave a substantial contribution to the development of dynamic scheduling algorithms for aperiodic service, Benedetto Allotta, who worked with me in approaching some problems related to control theory and robotics applications, and Gerhard Fohler, for the interesting discussions on leading scheduling issues.

I also wish to thank Antonino Gambuzza, Marco Di Natale, Giacomo Borlizzi, Stefano Petrucci, Enrico Rebaudo, Fabrizio Sensini, Gerardo Lamastra, Giuseppe Lipari, Antonino Casile, Fabio Conticelli, Paolo Della Capanna, and Marco Caccamo, who gave a valuable contribution to the development of the HARTIK system.

Finally, I express my appreciation to my wife, Maria Grazia, and my daughter, Rossella, for their patience and understanding during the preparation of this book.