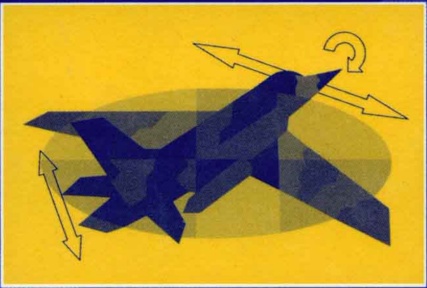
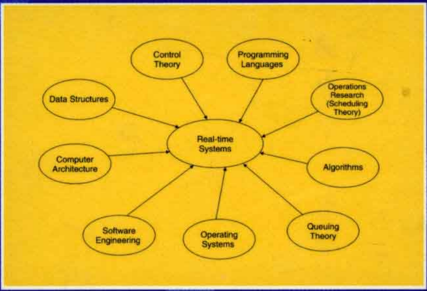
THIRD EDITION

REAL-TIME SYSTEMS DESIGN AND ANALYSIS





Phillip A. Laplante

REAL-TIME SYSTEMS DESIGN AND ANALYSIS

THIRD EDITION

Phillip A. Laplante

IEEE PRESS

WILEY-INTERSCIENCE

A JOHN WILEY & SONS, INC., PUBLICATION

Library of Congress Cataloging-in-Publication Data:

Laplante, Phillip A.
Real-time systems design and analysis : an engineer's handbook / Phillip A. Laplante.–3rd ed.
p. cm.
Includes bibliographical references and index.
ISBN 0-471-22855-9 (cloth)
1. Real-time data processing. 2. System design. I. Title.

QA76.54.L37 2004
004′.33–dc22

2003065702

Printed in the United States of America.

10 9 8 7 6 5 4 3 2 1

To my daughter, Charlotte

PREFACE TO THE THIRD EDITION

This book is an introduction to real-time systems. It is intended not as a cookbook, but, rather, as a stimulus for thinking about hardware and software in a different way. It is necessarily broader than deep. It is a survey book, designed to heighten the reader's awareness of real-time issues.

This book is the culmination of more than 20 years of building, studying, and teaching real-time systems. The author's travels have taken him to NASA, UPS, Lockheed Martin, the Canadian and Australian Defense Forces, MIT's Charles Stark Draper Labs, and many other places. These visits and interactions with literally hundreds of students from such places as Boeing, Motorola, and Siemens have resulted in a wider understanding of real-time systems and particularly their real application. This book is, in essence, a compendium of these experiences. The author's intent is to provide a practical framework for software engineers to design and implement real-time systems. This approach is somewhat different from that of other texts on the subject.

Because of the pragmatic approach, a few of the results and viewpoints presented book's may be controversial. The author has adapted many of the formal definitions from their traditional rigid form into words that are more compatible with practical design. In many places theoretical treatments have been omitted where they would have obscured applied results. In these cases, the reader is referred to additional reading. This author is a great believer in research in this area, and in many places has indicated where research needs to be done or is being done.

Although the book may appear simplistic, it is subtly complex. Consider the semaphore operators. They can be written with a minimum amount of code, yet they are fraught with danger for the real-time designer. In the same way, this book has a kind of Zen-like simplicity and complexity: a yin and a yang.

INTENDED AUDIENCE

This text is an introductory-level book intended for junior–senior level and graduate computer science and electrical engineering students, and practicing software engineers. It can be used as a graduate-level text if it is supplemented with an advanced reader, such as one by the author [Laplante00]. This book is especially useful in an industrial setting for new real-time systems designers who need to get "up to speed" very quickly. This author has used earlier editions of this book in this way to teach short courses for several clients.

The reader is assumed to have some experience in programming in one of the more popular languages, but other than this, the prerequisites for this text are minimal. Some familiarity with discrete mathematics is helpful in understanding some of the formalizations, but it is not essential. A background in basic calculus and probability theory will assist in the reading of Chapter 7.

PROGRAMMING LANGUAGES

Although there are certain "preferred" languages for real-time system design, such as C, C++, Ada 95, and increasingly Java, many real-time systems are still written in Fortran, assembly language, and even Visual BASIC. It would be unjust to focus this book on one language, say C, when the theory should be language independent. However, for uniformity of discussion, points are illustrated, as appropriate, in generic assembly language and C. While the C code is not intended to be ready-to-use, it can be easily adapted with a little tweaking for use in a real system.

ORGANIZATION OF THE BOOK

Real-time software designers must be familiar with computer architecture and organization, operating systems, software engineering, programming languages, and compiler theory. The text provides an overview of these subjects from the perspective of the real-time systems designer. Because this is a staggering task, depth is occasionally sacrificed for breadth. Again, suggestions for additional readings are provided where depth has been sacrificed.

The book is organized into chapters that are essentially self-contained. Thus, the material can be rearranged or omitted, depending on the background and interests of the audience or instructor. Each chapter contains both easy and challenging exercises that stimulate the reader to confront actual problems. The exercises, however, cannot serve as a substitute for practical experience.

The first chapter provides an overview of the nature of real-time systems. Much of the basic vocabulary relating to real-time systems is developed along with a discussion of the challenges facing the real-time system designer. Finally, a brief historical review is given. The purpose of this chapter is to foreshadow the rest of the book as well as quickly acquaint the reader with pertinent terminology.

The second chapter presents a more detailed review of basic computer architecture concepts from the perspective of the real-time systems designer and some basic concepts of electronics. Specifically, the impact of different architectural features on real-time performance is discussed. The remainder of the chapter discusses different memory technologies, input/output techniques, and peripheral support for real-time systems. The intent here is to increase the reader's awareness of the impact of the computer architecture on design considerations.

Chapter 3 provides the core elements of the text for those who are building practical real-time systems. This chapter describes the three critical real-time kernel services: scheduling/dispatching, intertask communication, and memory management. It also covers special problems inherent in these designs, such as deadlock and the priority inheritance problem. This chapter also highlights issues in POSIX compliance of real-time kernels.

In Chapter 4, the nature of requirements engineering is discussed. Next, structured analysis and object-oriented analysis are discussed as paradigms for requirements writing. An extensive design case study is provided.

Chapter 5 surveys several commonly used design specification techniques used in both structural and object-oriented design. Their applicability to real-time systems is emphasized throughout. No one technique is a silver bullet, and the reader is encouraged to adopt his or her own formulation of specification techniques for the given application. A design case study is also provided.

Chapter 6 begins with a discussion of the language features desirable in good software engineering practice in general and real-time systems design in particular. A review of several of the most widely used languages in real-time systems design, with respect to these features, follows. The intent is to provide criteria for rating a language's ability to support real-time systems and to alert the user to the possible drawbacks of using each language in real-time applications.

Chapter 7 discusses several techniques for improving the response time of real-time systems. Many of the ideas discussed in this chapter are well-known but unwritten laws of programming. Some are compiler optimization techniques that can be used to improve our code. Others are tricks that have been passed down by word of mouth. This chapter can help wring out that extra bit of performance from a critical system.

The final chapter discusses general software engineering considerations, including the use of metrics and techniques for improving the fault-tolerance and reliability of real-time systems. Later in the chapter, techniques for improving reliability through rigorous testing are discussed. Systems integration is also discussed. The chapter also reviews some special techniques that are needed in real-time systems.

While the difference between the first and second editions of this book is incremental, the third edition is essentially a new book. During the intervening eight years since the second edition, so many changes have taken place that more than a face-lift was needed. Approximately 50% of the material from the previous editions has been discarded and the remainder entirely rewritten. Hence, about 50% of the book is new material.

When this course is taught in a university setting, typically students are asked to build a real-time multitasking system of their choice. Usually, it is a game on a PC, but some students can be expected to build embedded hardware controllers of surprising complexity. The author's "assignment" to the reader would be to build such a game or simulation, using at least the coroutine model. The application should be useful or at least pleasing, so some sort of a game is a good choice. The project should take no more than 15 hours and cover all phases of the software life-cycle model discussed in the text. Hence, those readers who have never built a real-time system will have the benefit of the experience.

A NOTE ON REFERENCES

Real-Time Systems Engineering is based on more than 50 years of experience and work by many individuals. Rather than clutter the text with endless citations for the origin of each idea, the author chose to cite only the most key ideas where the reader would want to seek out the source for further reading. Some of the text is adapted from two other books written by the author on software engineering and computer architecture [Laplante03c] [Gilreath03]. Where this has been done, it is so noted. *Note:* In all cases where some sections of this text, particularly the author's own, appear as "adapted" or "paraphrased," it means that the work is being reprinted with both major and minor differences. However, rather than confuse the issue with intermittent quotation marks for verbatim text, the reader should attribute all ideas to cited authors from the point where the usage is noted to the ending reference. This author, however, retains responsibility for any errors. In all cases, permission to reprint this material has been obtained.

Many good theoretical treatments of real-time systems exist, and they are noted where applicable. However, these books are sometimes too theoretical for practicing software engineers and students who are often too impatient to wade through the derivations for the resultant payoff. These readers want results that they can use now in the trenches, and they want to see how they can be used, not just know that they exist. In this text, an attempt is made to distill the best of the theoretical results, combined with practical experience to provide a toolkit for the real-time designer.

This book contains an extensive bibliography. Where verbatim phrases were used or where a figure came from another source, the author tried to cite it appropriately. However, if any sources were inadvertently overlooked, the author wishes to correct the error. In addition, in a book of this magnitude and complexity, errors are bound to occur. Please notify the author if you find any errors of omission, commission, citation, and so on by email, at plaplante@psu.edu and they will be corrected at the next possible opportunity.

ACKNOWLEDGMENTS

The author wishes to acknowledge and thank the many individuals who assisted in the preparation of this book. Dr. Purnendu Sinha of Concordia University, wrote much of Chapter 3 and various parts of other chapters relating to scheduling theory, contributed many exercises, and classroom tested the material. Dr. Colin

Neill, a Penn State colleague, co-wrote Chapters 4 and 5, reviewed and contributed to other chapters, and single-handedly changed the author's mind concerning the value of object-oriented methods. Research collaborator William Gilreath reviewed parts of the manuscript, contributed to parts on computer architecture, and provided some of the more interesting exercises. Dr. David Russell of Penn State reviewed the manuscript and provided a most supportive environment at the Great Valley School of Graduate Professional Studies where the author works. Valuable reviews were also provided by Dr. Mike Hinchey, Dr. Dave Sinha, and Patricia Feingold. The acquisition and editorial team at IEEE Press/John Wiley, in particular Tony VenGratis and John Griffin, provided terrific support and encouragement. The author's wife, Nancy, typed and edited much of the material.

The author also wishes to thank the many students who, over the last 20 years, have contributed ideas to this book through discussions, projects, and classes. While an exhaustive list is impossible, for the third edition the author must single out Michael Barnes, David Cloutier, Jim Goldman, Dana Gryger, Michael Lutz, Dr. Jeff Nash, Mike Rapa, and Fred Woolsey. In particular, the author thanks Fred, Dana, and Mike for contributing the excellent case study on the traffic controller system found in Chapters 4 and 5, and David for contributing to the discussion on the use of object-oriented languages in Chapter 6.

The author is grateful for the success of the first editions of this book, with more than 15,000 copies sold to the college text and professional markets. The only thing more gratifying than its adoption at such prestigious universities as Carnegie Mellon University, University of Illinois at Urbana-Champaign, Princeton University, the United States Air Force Academy, Polytechnic University, and many others, has been the feedback received from individuals thankful for the influence that the book has had on them.

Finally, the author wishes to thank his wife Nancy, and his children, Christopher and Charlotte, for putting up with the seemingly endless work on this manuscript and too many other projects to mention. This book is dedicated to them with love.

PHILLIP A. LAPLANTE

Chester County, Pennsylvania
September, 2003

CONTENTS