THE ARCHITECTURE OF COMPUTER HARDWARE, SYSTEMS SOFTWARE, & NETWORKING

An information technology approach • fifth edition

Irv Englander

WILEY

The Self-Operating Napkin

The professor walks in his sleep, strolls through a cactus field in his bare feet, and screams out an idea for self-operating napkin.

As you raise spoon of soup (A) to your mouth it pulls string (B), thereby jerking ladle (C) which throws cracker (D) past parrot (E). Parrot jumps after cracker and perch (F) tilts, upsetting seeds (G) into pail (H). Extra weight in pail pulls cord (I) which opens and lights automatic cigar lighter (J), setting off sky-rocket (K) which causes sickle (L) to cut string (M) and allow pendulum with attached napkin to swing back and forth thereby wiping off your chin.

After the meal, substitute harmonica for the napkin and you'll be able to entertain the guests with a little music.

FIFTH EDITION

THE ARCHITECTURE OF COMPUTER HARDWARE, SYSTEMS SOFTWARE, & NETWORKING

AN INFORMATION TECHNOLOGY APPROACH

Irv Englander

Bentley University

WILEY

Publisher: *Don Fowley*
Editor: *Beth Lang Golub*
Project Manager: *Katie Singleton*
Editorial Assistant: *Jayne Ziemba*
Photo Editor: *Felicia Ruocco*
Cover Designer: *Maureen Eide*
Associate Production Manager: *Joyce Poh*
Senior Production Editor: *Jolene Ling*
Production Management Services: *Laserwords*
Cover Credit: *Norm Christiansen*

This book was set by Laserwords Private Limited.

Founded in 1807, John Wiley & Sons, Inc. has been a valued source of knowledge and understanding for more than 200 years, helping people around the world meet their needs and fulfill their aspirations. Our company is built on a foundation of principles that include responsibility to the communities we serve and where we live and work. In 2008, we launched a Corporate Citizenship Initiative, a global effort to address the environmental, social, economic, and ethical challenges we face in our business. Among the issues we are addressing are carbon impact, paper specifications and procurement, ethical conduct within our business and among our vendors, and community and charitable support. For more information, please visit our website: www.wiley.com/go/citizenship.

Library of Congress Cataloging-in-Publication Data

Englander, Irv.
The architecture of computer hardware, systems software, & networking : an information technology approach/ Irv Englander, Bentley University.—Fifth edition.
pages cm
Includes bibliographical references and index.
ISBN 978-1-118-32263-5 (paper)
1. Computers. 2. Systems software. I. Title.
QA76.5.E566 2014
005.4’3--dc23

2013027578

Printed in the United States of America

10 9 8 7 6 5 4 3 2 1

To four outstanding teachers and great human beings:

With your guidance, inspiration, and patience, you showed me that everything is possible.

Dr. Sidney H. Englander (1900–1980)
and Mildred K. Englander (1906–2008),
in memoriam my father and mother

Albert L. Daugherty, in memoriam
teacher of Science in Cleveland Heights, Ohio
from 1927 to 1970

Edith B. Malin, in memoriam
teacher of English in Cleveland Heights, Ohio
from 1924 to 1958

BRIEF CONTENTS

PART FOUR

NETWORKS AND DATA COMMUNICATIONS 358

PART FIVE

THE SOFTWARE COMPONENT 474

SUPPLEMENTARY CHAPTERS

On the Web at www.wiley.com/college/englander

CONTENTS

SUPPLEMENTARY CHAPTERS

On the Web at www.wiley.com/college/englander

PREFACE

The modern world offers a lot of readily available online resources for learning. Wikipedia, Google, news sources, millions of Web sites and blogs, even YouTube, offer access to information in nearly any subject that triggers your curiosity and interest. Nonetheless, I continue to believe that for deep understanding of something, nothing beats the integrated approach and focus of an old-fashioned printed-on-paper textbook. Well—maybe the e-book equivalent, but, still, a textbook.

When I open a new book, in *any* subject, the first thing I want to know is what the book has to offer that makes it worth my while to read it. I would like to try to help you answer that question for the book that you're holding in your hand or on your tablet.

The information systems and technology fields are wonderfully exciting places to be! It seems as though every day brings new developments that alter the ways we create and work with information. Of course, with this excitement comes a challenge. To be a successful player in IS or IT, we have to be adaptable and flexible.

Much of the change occurs around computer system technology. The computer is, after all, at the foundation of information systems. A deep understanding of computer systems is, therefore, an essential element of success. We must be able to understand each new development, assess its value, and place it in the context of our knowledge of computer systems.

The subject of this book is the architecture of computer systems. Computer architecture is about the structure and operation of digital computers and computer-based devices. Computer architecture is concerned with the operational methods of the hardware; with the services provided by operating system software; with the acquisition, processing, storage, and output of data; and with the interaction between computer-based devices.

There is a tendency for people in information systems and technology to neglect a study of computer architecture. After all, the technology changes so rapidly—is it really worth trying to understand something that may be out of date by the time I finish this book? There is no question that computer technology has evolved rapidly. The computer in a smartphone is far more powerful than the mainframe computer of twenty-five years ago, with memory, disk and flash storage capacity, display and multimedia capability, and ease of use that would have been unthinkable just a few years ago. Even more important, connecting systems to work together is now routine and simple.

Interestingly enough, however, as profound as advances in the technology have been, the concepts of computer architecture that really matter have changed only nominally over the last seventy years. The new technologies are based on a foundation of architectural concepts that were developed many years ago. The architecture of a modern computer system was developed in the 1940s. The instruction set in a modern personal computer or smartphone is nearly identical to that of computers built in the 1950s and 1960s. Modern operating system techniques were developed in the 1960s. The graphical user interface is based on a 1960s' project. The Internet is built from concepts developed more than forty years ago.

So you see that an understanding of computer architecture makes it possible to "ride the wave" of technological change, secure in the feeling that you are equipped to deal with new developments as they occur, and to have fun doing so. When you are done reading this book, you will have substantial knowledge about how a computer works and a good understanding of the operating concepts, the hardware, and system software that make up a computer. You will see the interaction between computers and between data and the computer. Plus, you will have learned lots of jargon that you can show off at parties and job interviews.

This textbook is designed for a wide range of readers, both undergraduate and graduate. The material is specifically directed toward IS and IT majors. There are no explicit prerequisites, although the book assumes that the student is familiar with a personal computer. It also assumes (but does not require) some basic programming skills: although there is no programming in the book, program code is occasionally used as an example to clarify an idea, and a knowledge of programming is helpful at understanding instruction set design and program execution concepts. The material in this textbook conforms to the criteria of the IT Infrastructure courses and core concepts, as described in the ACM and AIS standard IS2010 and IT2008 curricula. Although the material in this book may be useful as background for system design and implementation project courses, the course can be placed anywhere in the curriculum.

Most instructors will not cover the entire textbook in a single semester. The organization of this book is designed to allow an instructor to cover the major topic areas in different levels of depth, depending on the experience and needs of the students. On the other hand, it is my intention that this book will serve a student as a useful reference long after the formal course is completed. It is designed for use as a book where a professional can look up the basic concepts that clarify new developments as they occur.

This text is the outgrowth of courses that I have taught to CIS majors and minors at Bentley University at both the undergraduate and graduate levels for more than thirty years. Student responses to the material and the approach have generally been very enthusiastic. Many students have returned after graduation to tell me that their knowledge in this area has directly contributed to their career development. Along the way, student comments have also been extremely helpful to me in the book's continuing evolution.

Those familiar with previous editions will notice that the organization of recent editions has undergone substantial revision to reflect current technological practices and trends. In particular, it is no longer reasonable to discuss computers as individual units without also considering the networks that tie them together; computer networking is now covered thoroughly in its own section, and there is an increased emphasis on the integration and synergy of the various components of the computer system and on the system as a whole. Still, the basic philosophy, organization, and approach remain essentially similar to those of the first edition, reflecting the unchanging nature of the underlying principles.

ORGANIZATION OF THE FIFTH EDITION, NOTING CHANGES FROM PREVIOUS EDITIONS

The biggest challenge for me as the author of this book has been to preserve the guiding principles established in the first edition, while reflecting the major changes in the way computers are used, in the rapid deployment of new technology, and in the resulting evolution of IS/IT curriculum to reflect those changes. The fifth edition is a substantial, though incremental, update to previous editions. It reflects the rapid growth in the use of tablets and smartphones as important computing devices. The material on networking has been substantially reorganized. Still, users of previous editions will find that much of the material is familiar; after all, the way in which computers are used in IS/IT may have changed, but the basic guiding principles of computer architecture are essentially the same as they have been for many years.

This book is organized into five parts totaling eighteen chapters, plus four additional supplementary chapters that are posted on the Web. The first part serves as an introduction and overview of the role of the computer in information systems; it introduces the concept of a system and provides a brief introduction to each of the components that make up a modern computer system. Each of the remaining four parts deals with a single architectural aspect of the computer system.

Part Two discusses the role and representation of data in the computer. Here, we consider numbers, text, sound, images, video, and other data forms. Part Three presents the hardware architecture and operational concepts. It introduces the components of a computer and shows how they collaborate to execute computer instructions, discusses the nature of a computer instruction set, and explores the interaction between the CPU, memory, and I/O peripheral devices. Part Four presents a thorough introduction to the basics of computer networking. Part Five discusses the system software, the programs that function to make the resources of the computer system, and other interconnected computer systems and components, accessible to the user and to application programs.

The approach within each group of chapters is layered. Each new layer builds upon the previous material to add depth and understanding to the reader's knowledge. Each topic section consists of a short introduction that places the topic to be discussed into the context of the computer system as a whole and then lays out in detail the organization of the chapters within the section. Each topic area is introduced as gently as possible, using ideas and examples that are already familiar to students. Successive material is progressive and accumulative. In addition to the numerous examples that are used throughout the text, the supplementary chapters offer substantial case studies that show application of the text material to current examples of importance. Overall, the approach is gentle, progressive, and accumulative. As much as possible, each section is self-contained.

An overview of the organization of each part follows. A few comments about the differences between the fourth and fifth editions are included for those who are familiar with the previous edition. More details can be found in the introductions to each section.

Part One consists of two chapters that present a short overview of computing and placing architectural concepts into the context of information technology. Chapter 1 introduces the components of a computer system and shows the relationships among the components. In the new edition, I have assumed that today's students are more familiar with computer technology and terminology than previous generations, so there is less introduction of "IT101"-type material. Chapter 1 also presents a simple model of computing and discusses the importance of standards and protocols in the development of computer systems. This chapter concludes with a short history of computers from the architectural point of view. Chapter 2 focuses on the concepts of systems, models, and system architectures, using various types of computer systems as examples. There are several relatively small but important additions and updates in this chapter. In Section 2.2, there is an increased emphasis on *n*-tier architectures and distributed computing. Section 2.2 also contains a new section on cloud computing, which is presented as a variation of the client–server concept. A new Facebook application architecture example has also been added.

Chapters 3 through 5 comprise Part Two. Chapter 3 introduces number systems and basic number system operations; it then explores the relationships between numbers in different number bases and the conversion techniques between the different representations. Chapter 4 investigates different types of data formats, including alphanumeric, image, video, and audio formats. It considers the relationship between numerical and character-based representations. Previously, Chapter 4 also introduced various devices and data formats used for data input and output. Much of that material has been moved to Chapter 10 since it is more directly related to the devices themselves. Chapter 5 studies the various formats that are used to represent and to perform calculations on integer and floating point numbers.

Part Three discusses the hardware architecture and operational aspects of the computer. Chapter 6 begins the study with the introduction of the Little Man Computer, a simple model that provides a surprisingly accurate representation of the CPU and memory. The model is used to develop the concept of an instruction set and to explain the basic principles of the von Neumann architecture. Chapter 7 extends the discussion to a real computer. It introduces the components of the CPU and shows their relationship to the Little Man Computer model. It introduces the bus concept, explains the operation of memory, presents the instruction fetch–execute cycle, and discusses the instruction set. It identifies important classes of instructions and discusses the ways in which instructions can be categorized. The ARM instruction set is offered as an illustration of a typical current CPU model.

Chapter 8 expands the material in Chapter 7 to consider more advanced features of the CPU and memory. It offers an overview of various CPU architectures. It continues with a discussion of techniques for improving memory access, particularly cache memory, and an introduction to current CPU organization, design, and implementation techniques, including pipelining and superscalar processing. This chapter also introduces multiprocessing (or multicore, in current terminology) concepts. Chapter 8 received basic updating from the previous edition, including elimination of the VLIW and EPIC architectures that never caught on.

Chapter 9 presents the principles of I/O operation, and Chapter 10 illustrates how I/O is performed in various I/O devices. With the advent of powerful tablets and smartphones, Chapter 10 received a substantial overhaul. Solid-state storage is of increased importance, and Section 10.2 has been expanded to increase the coverage. Section 10.3, the display section, has a new discussion of graphical processing units. CRTs have been eliminated altogether. The discussions of text-mode display processing and different types of raster scans have been minimized. The discussion of printers in Section 10.4 now considers only laser and inkjet technologies. Section 10.8, user input devices, now includes the alternative sources of alphanumeric input, moved from Chapter 4, plus more on touch screens and voice input. There is an additional section discussing the sensors available on mobile devices, including GPS, accelerometers, gyroscopes, magnetic field sensors, and near-field communication sensors.

Chapter 11 discusses the computer system as a whole. It discusses the interconnection techniques and integration of the various hardware components. Chapter 11 has been updated to reflect modern systems, including mobile systems and radically changed bus architecture technology. The new, just-introduced Intel Haswell architecture and System on a Chip concepts are included. Firewire was removed; the Thunderbird port is introduced in its place. Chapter 11 also addresses the interconnection of computers to increase the performance and reliability of a computer with a specific focus on clustering and grid computing.

Three supplementary chapters on the Web provide additional resources to support the chapters in Part Three. Supplementary Chapter 1 (SC1) offers an introduction to Boolean algebra, combinatorial logic, and sequential logic for those readers that would like a deeper understanding of the computer in its simplest and most elegant form. Supplementary Chapter 2 (SC2) offers three detailed case studies of important architectures: the Intel x86 family, the Power computer, and the IBM zSystem. As of this writing, this supplement is currently being updated. Supplementary Chapter 3 (SC3) discusses alternative approaches to instruction addressing.

Part Four, Chapters 12–14, presents a thorough introduction to networking. For the fifth edition, Chapters 12 and 13 received a major reorganization; to a much lesser extent, Chapter 14 is also affected by the change. The goal was a more unified and systematic presentation of networking material.

Chapter 12 introduces the concept of a communication channel and explores its characteristics and configurations. This includes the communication channel model, the concept of links, packets, basic channel characteristics, network topology, types of networks (LAN, MAN, etc.), and basic network interconnection and routing. All of the material related to protocols and movement of data through the network has been moved to Chapter 13. The chapter is retitled as Networks and Data Communications—An Overview.

Chapter 13 now focuses on the passage of data packets through a network. The chapter is now titled Ethernet and TCP/IP Networking. Section 13.1 introduces TCP/IP and OSI and the concept of layered communication. Section 13.2 describes the differences between program applications and network applications. Following these two sections, the next three sections carefully describe the process of moving packets, one layer at a time, from the bottom-up. It should be noted that wireless networking has been moved from Chapter 14 and is now included as part of the Ethernet discussion. Sections 13.6 and 13.7 explain IPv4 and IPv6 addressing and DHCP, and DNS, respectively. Quality of Service and network security are briefly introduced in Sections 13.8 and 13.9. The chapter concludes with a discussion of alternative protocols, including a comparison of OSI and TCP/IP, as well as brief discussions of MPLS, cellular technology, and other protocol suites.

Chapter 14 focuses primarily on communication channel technology, including analog and digital signaling, modulation and data conversion techniques between analog and digital, and the characteristics of transmission media. Sections 14.1–14.3 are relatively unchanged. Section 14.4, however, is mostly new. Titled Alternative Technologies, it offers advanced introductions to the radio technologies of LTE cellular technology, Wi-Fi, and Bluetooth.

Part Five is dedicated to a discussion of system software. Chapter 15 provides an overview of the operating system. It explains the different roles played by the operating system and introduces the facilities and services provided. Chapter 16 presents the role of the operating system from the viewpoint of the user of a system. The fifth edition offers new screenshots of Windows 8 and recent versions of Linux. Chapter 17 discusses the all-important topic of file systems, including an introduction to Microsoft's new Resilient File System, intended to replace NTFS. Chapter 18 discusses the operating system as a resource manager, with an in-depth discussion of memory management, scheduling, process control, network services, and other basic operating system services. Chapter 18 includes a detailed introduction to virtual memory technique, rewritten for the fifth edition, with a new, detailed, worked-out example, carefully illustrating the different page replacement algorithms. The chapter also includes an introduction to virtual machines. In addition to its hardware discussions, Supplementary Chapter 2, when completed, will also provide current Windows, UNIX/Linux, and z/OS case studies.

A fourth supplementary chapter provides an introduction to the system development software that is used for the preparation and execution of programs.

This book has been a continuing labor of love. My primary goal has been to create and maintain a textbook that explains computer architecture in a way that conveys to you, the reader, the sense of excitement and fun that I believe makes a career in information systems and technology so satisfying. I hope that I have succeeded to some extent.

ADDITIONAL RESOURCES

Additional resources for students and instructors may be found at the textbook's Web site, www.wiley.com/college/englander. I can also be reached directly by e-mail at ienglander@bentley.edu. Although I am happy to communicate with students, I am unable to supply tutorial help or answers to review questions and exercises in this book.

ACKNOWLEDGMENTS

I've discovered that a major, ongoing textbook project is a formidable task. Many individuals have helped to make the task manageable—and kept me going when, from time to time, I became convinced that textbooks really *do* appear by magic and are *not* written by humans. It is impossible to thank people adequately for all their help and support. First and foremost, a special thank you to four friends who have helped me survive through all five editions, Wilson Wong, Ray Brackett, Luis Fernandez, and Rich Braun. Their continuing backup has been amazing! I couldn't have asked for a better support team. Dinner is ready and the champagne is on ice. *Yet* again!

My continuing thanks, too, to Stuart Madnick. Stuart, your technical inspiration and personal encouragement was invaluable to me when I struggled to get the first edition of this book going. You helped me to believe that this project was actually possible and worthwhile. That support has continued to inspire me through every subsequent edition.

Next, I thank the many colleagues at Bentley University who shared their ideas, experiences, and encouragement. In particular, colleagues Wilson Wong, David Yates, Doug Robertson, and Mary Ann Robbert have all offered contributions that have substantially improved the book over five editions. A special thank you, David, for your helpful technical discussions and reviews of the data communications material in the fourth and fifth editions, and to you, Wilson, for serving as a technical reviewer for the last three (!) editions, providing many comments, rewrites, examples, and suggestions for clarification, and for creating many of the ancillary materials for the book.

Thanks to the editors, production people, and marketing personnel at John Wiley & Sons and the editors and production people at SPi Global. Sometimes the process is difficult, but we always managed to get through it in a way that made the book better. I consider myself fortunate to have worked with such committed people. Particular thanks to Beth Lang Golub, Katie Singleton, and Felicia Ruocco for your ongoing efforts to make this book perfect, even though we all know it's impossible!

I would like to acknowledge the reviewers who have given of their time and effort over many editions to assure that this book was as good as it could be: Dr. Stu Westin, The University of Rhode Island; Alan Pinck, Algonquin College; Mark Jacobi, Programme Director for Undergrad Computing at Sheffield Hallam University; Dr. Dave Protheroe, South Bank University, London; Julius Ilinskas, Kaunas University of Technology; Anthony Richardson, United States Army Informations Systems Engineering Command; Renee A. Weather, Old Dominion University; Jack Claff, Southern Cross University; Jan L. Harrington, Marist College; YoungJoon Byun, California State University, Monterey Bay; William Myers, Belmont Abbey College; Barbara T. Grabowski, Benedictine College; G.E. Strouse, York College of Pennsylvania; Martin J. Doyle, Temple University; Richard Socash, Metropolitan State College of Denver; and Fred Cathers, Franklin University. Your comments, suggestions, and constructive criticism have made a real difference in the quality of this book. Thank you.

Thank you, too, to the reviewers of the current edition: Bob Brown, Southern Polytechnic State University; Ming-Hsing Chiu, Dillard University; Angela Clark, University of South Alabama; Chin-Tser Huang, University of South Carolina; Theresa Kraft, Capella University; Ng Shu Min, HELP University College; Carl Nehlig, Tulane University; Leasa Perkins, University of Maryland University College; and Mahbubur Syed, Minnesota State University, as well as to testbank author Ronald Munsee, University of Maryland University College, and to users from around the world who responded to the 2011 user survey.

Many colleagues offered corrections to previous editions that have had important impact on the quality of the current edition. To each and everyone, your assistance in eliminating errors has been much appreciated. Among these, I especially wish to acknowledge David Feinstein and his crew at the University of South Alabama, Gordon Grimsey of AIT in Auckland, New Zealand, and Stu Westin of University of Rhode Island for efforts well above and beyond the call of duty. Stu has also generously made his excellent Little Man Simulator publicly available, for which I am truly grateful. Thanks for everything, Stu.

Numerous students, too many to name you all, also offered corrections, made suggestions, and provided ideas. Please accept my deepest appreciation and thanks.

I hope that I have not forgotten anyone. If I have, I apologize.

I have strived to make this book as technically accurate as is humanly possible. Nonetheless, I know that errors have a way of creeping in when one least expects them. I would greatly appreciate hearing from readers who find errors that need correction. Your comments and suggestions about the book are also welcome.

Irv Englander
Boston, Massachusetts

ABOUT THE AUTHOR

Dr. Irv Englander has been involved in many different aspects of the computing field for more than fifty years. He has designed logic circuits, developed integrated circuits, developed computer architectures, designed computer-controlled systems, designed operating systems, developed application software, created the initial system design for a large water-monitoring system, performed software auditing and verification of critical control software, and developed and specified hardware components and application software as a consultant for business systems large and small.

As an educator, he has contributed papers and given workshops on end-user computing, e-commerce, and on various aspects of IS and IT pedagogy. He was an invited contributor and reviewer for the IS-97 and IS-2002 information systems curricula, and continues to publish and take an interest in the technical infrastructure components of the IS/IT curriculum. He is actively involved in the application of new technology to information systems. Most recently he served as a faculty advisor to a low-cost supercomputing student research project, which was presented at the ACM/IEEE Computer Society Supercomputing Conference SC13.

Dr. Englander has a Ph.D. from MIT in Computer Science. His doctoral thesis was based on the design of a large image-processing software laboratory. At MIT, he won the Supervised Investors Award for outstanding teaching. He holds the rank of Professor Emeritus of Computer Information Systems at Bentley University, where he taught full-time for thirty-three years.

PART ONE

A computer-based information system is made up of a number of different elements:

- The *data* element. Data is the fundamental representation of facts and observations. Data is processed by a computer system to provide the information that is the very reason for the computer's existence. As you know, data can take on a number of different forms: numbers, text, images, and sounds. But it's all numbers in the computer.
- The *hardware* element. Computer hardware processes the data by executing instructions, storing data, and moving data and information between the various input and output devices that make the system and the information accessible to the users.
- The *software* element. Software consists of the system and application programs that define the instructions that are executed by the hardware. The software determines the work to be performed and controls operation of the system.
- The *communication* element. Modern computer information systems depend on the ability to share processing operations and data among different computers and users, located both locally and remotely. Data communication provides this capability.

The combination of hardware, software, communication, and data make up the *architecture* of a computer system. The architecture of computer systems is remarkably similar whether the system is a playstation, a personal computer that sits on your lap while you work, an embedded computer that controls the functions in your cell phone or in your car, or a large mainframe system that is never actually seen by the hundreds of users who access it every day.

Amazingly, the changes in computer technology that you've seen in just the last few years are essentially superficial; the basic architecture of computer systems has changed surprisingly little over the last sixty years. The latest IBM mainframe computer executes essentially the same instruction set as the mainframe computer of 1965. The basic communication techniques used in today's systems were developed in the 1970s. As new as it might seem, the Internet celebrated its fortieth anniversary in 2010. All of this is surprising considering the growth of computing,

AN OVERVIEW OF COMPUTER SYSTEMS

the rapid change of technology, and the increased performance, functionality, and ease of use of today's systems. This makes the study of computer architecture extremely valuable as a foundation upon which to understand new developments in computing as they occur.

Computer system architecture is the subject of this textbook. Each element of the system is addressed in its own section of the text, always with an eye to the system as a whole.

Part I is made up of two chapters that present an overview of systems, and of the computer system in particular.

Chapter 1 addresses a number of issues, including:

- The ways in which a knowledge of computer architecture enhances our abilities as computer users and professionals
- A simplified view of typical computer system architectures
- The basic components that make up a computer system
- The fundamental operations that are performed by computer systems.

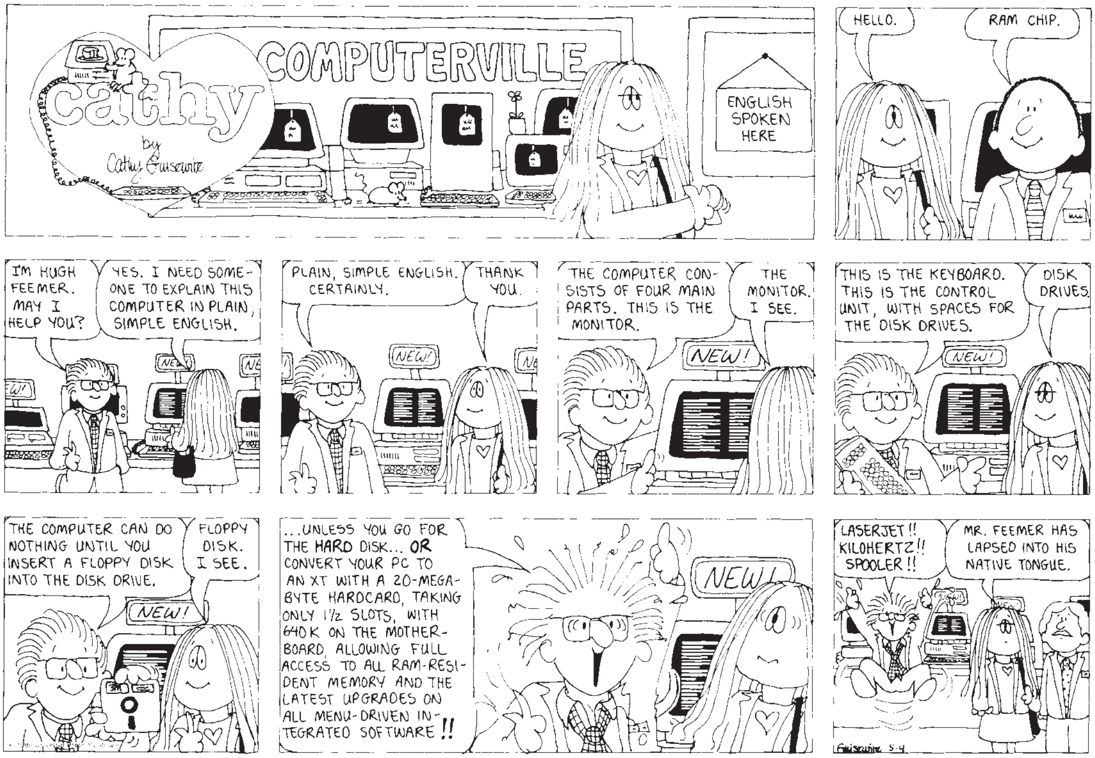
Chapter 1 concludes with a brief architectural history of the computer.

An encompassing theme throughout this text is that of systems and system architecture. The words "system" and "architecture" appear throughout this book: we talk about information systems, computer systems, operating systems, file systems, software architecture, I/O architecture, network architecture and more. You will probably take a course in *System Analysis and Design* sometime in your college career.

Although most people have an instinctive understanding of what a system is, it is more important for us as system professionals to understand the concepts of systems and system architecture at a deeper level than the average person. Chapter 2 offers careful definitions and examples of the concept of systems and system architecture, both generally and in the specific context of the computer systems that are the focus of this book.

CHAPTER 1

COMPUTERS AND SYSTEMS



1.0 INTRODUCTION

Welcome to the wonderful modern world of computers. Where the technology changes daily. Or does it?

It's true that the computer of today looks nothing like the computer of even five or ten years ago. It is nearly impossible today to escape the immediate and ubiquitous reach of computers and computer-based systems and devices. There is probably a smartphone in your pocket or on your desk. For many of you, your laptop or tablet is sitting nearby as you read this paragraph. (Or, maybe you're even *reading* this paragraph on your tablet or E-book.) Furthermore, your smartphone probably has more computing power than most of the computers on the market ten years ago. It fits easily in your pocket or purse. It weighs less than 1/100 of that old desktop computer and has at least ten times as much memory!

And that's not all. Your car has several embedded computers controlling various automotive functions, and perhaps a touch screen for hands-free telephone, navigation, the radio, Internet access, and more. Which is almost unnecessary, because you can probably tell it what you want verbally anyway. Even your microwave oven and the machine that launders your clothes depend on computers to function. As you are likely aware, most of these machines can talk to each other, using the Internet or some other networking technology. Just for fun, Figure 1.1 shows pictures typical of a 2005 laptop, a 2013 smartphone, and a current embedded computer that controls many functions in your car.

Although the focus of this book is on IT systems, our discussion of computer hardware, software, and networking applies equally well to workplace computers, tablets, smartphones, and, even, computers embedded in other equipment. In this figure, we have three seemingly very different looking pieces of equipment working on different types of applications. And yet, it's hopefully obvious to you that these three systems share a lot in common. They are all computer based. All contain at least one central processing unit (CPU, some contain more) and memory. All provide a facility for interacting with long-term storage and other devices and with users. What may be less obvious to you is that the programs that they run are also essentially similar, differing mostly in the details required by the different components of the particular system and by the nature of the applications. For example, systems may have different amounts of memory, different types of displays, different I/O devices, and different operating systems, as well as running different types of applications and serving different purposes.

In fact, a modern IT system may contain elements of many different types of systems, with networking that ties everything together.

When creating an IT system, our concerns are whether the various components provide the features and performance that the users require. To be an effective designer and user, you have to understand the specifications, their importance and their meaning; the terminology; and the jargon. Which features are important to the users? Is this the right combination of features that you need in your computer to have the computer perform the work that you wish to get done? Are there features missing that we need? Perhaps we are paying too much for the performance that we need. Or maybe we need more. What other information about this system would allow you to make a more informed decision?

FIGURE 1.1

Computer Devices, Old and New

© 2007 Hewlett-Packard Company

Photo by David Shepperd

Delphi Automotive

There is obviously no need to understand the inner workings of most modern computer-based systems to operate them adequately. Indeed, in many cases, the presence of the computer is hidden from us, or **embedded**, and its operation is invisible to us as users. We don't need to know how a computer works to read an E-book.

Even as experienced users, we can run standard software packages on a personal computer or apps on a smartphone without understanding exactly how they work; we can program a computer in a high-level or a scripting language without understanding the details of how the machine executes the individual instructions; we can design and implement Web pages without understanding how the Web browser gets its pages from a Web server or how the Web server creates those pages; we can purchase a tablet or a laptop computer from a salesperson without understanding the specifications of the system.

And yet, there is something missing. Perhaps the software doesn't do exactly what we want, and we don't understand the machine well enough to risk fooling around with the software's options. Perhaps if we understood the system better we might have written and configured the program to be faster and more efficient. Perhaps we could create Web pages that load faster and work better. Perhaps the salesperson did not sell us the optimum system for our job. Or perhaps it's nothing more than a sense of excitement that's missing. But that's important, too!

You are reading this book because you are a student studying to become a computer professional, or maybe you are simply a user wanting a deeper understanding of what the computer is all about. In either case, you know you'll be interacting with computer systems in some form or other for the rest of your life. It's nice (as well as useful) to know something about the tools of the trade. More important, understanding the computer system's operations has an immediate benefit: it will allow you to use the machine more effectively.

As a user, you will be more aware of the capabilities, strengths, and limitations of a computer system. You will have a better understanding of the commands that you use. You will understand what is taking place during the operation of the program applications that you use. You will be able to make better informed decisions about your computer equipment and application programs. You will understand more clearly what an operating system is, and how to use it effectively and to your advantage. You will know when it is preferable to do a job manually, and when the computer should be used. You will understand the most efficient way to "go online", and what benefits might be gained from your home network. You will improve your ability to communicate with system analysts, programmers, and other computer specialists.

As a programmer, it will allow you to write better programs. You will be able to use the characteristics of the machine to make your programs operate more effectively. For example, choosing the appropriate data type for a variable can result in significantly faster performance. Soon you will know why this is so, and how to make the appropriate choices.

You will discover that some computers will process nested loops much more quickly if the index variables are reversed. A rather surprising idea, perhaps, and you'll understand why this is true.

You will understand why programs written in a compiled language like C++ usually run much faster than those written in interpreted program languages like BASIC or scripting languages like JavaScript. Similarly, you'll see why the basic layout of a program can have a major impact on the program's run-time efficiency.

As a systems architect or system analyst, you will be responsible for the design and implementation of systems that meet an organization's information technology (IT) needs, recognizing that the differences in the cost and capabilities of the components that you select may have significant impact on the organization. With the knowledge gained here you will be in a better position to determine and justify the set of computer system components and the system architecture that are appropriate for a particular job and to determine the trade-offs with other possible system architectures.

You'll be able to assist management in making intelligent decisions about system strategy: should the company adopt a large mainframe/virtual machine system approach for its Web servers, or would a system consisting of a network of off-the-shelf blade servers provide better performance at lower cost? You'll be better prepared to analyze the best way to provide appropriate facilities to meet the needs of your users. In an era of fast-changing technology, you'll be more able to differentiate between simple technological obsolescence that does not affect the organization's requirements significantly and major advances that suggest a real need to replace older equipment. You will understand the trade-offs inherent in the use of cloud and other remote services.

When selecting computers, you would like to purchase the computers that best meet the needs of the organization's applications and the users. You must be able to read and understand the technical specifications in order to compare different alternatives and to match the system to the users' needs. This book will teach you what you need to know to specify and purchase a system intelligently. You'll know the differences between various CPU technologies and the advantages and disadvantages of each. You will learn what peripheral hardware is appropriate for your organization's files and the trade-offs between different file system formats, what is required to build an intranet, and what the speed, size, and performance limitations of a particular system are. You'll be able to compare the features of OS/X, Windows, and Linux knowledgeably and decide which ones are important to you. You'll be able to apply your basic understanding of computers to new technologies and concepts such as mobile IT, new network protocols, virtual machines and cloud services as they appear. You'll learn to understand the jargon used by computer salespeople and judge the validity of their sales claims.

As a networking professional, you are responsible for the design, maintenance, support, and management of the networks that connect your computer systems together and provide the required interfaces to the outside world. You must be able to specify network layouts that optimize your equipment and network resources. Your understanding of basic network configurations and protocols will allow you to control and provide sufficient and appropriate access to your users in an efficient manner. This text offers the basic tools as a starting point to prepare for a career in networking.

As a Web services designer, you must be able to make intelligent decisions to optimize your Web system configurations, page designs, data formatting and scripting language choices, and operating systems to optimize customer accessibility to your Web services.

As a system administrator or manager, your job is to maximize the availability and efficiency of your systems. You will need to understand the reports generated by your systems and be able to use the information in those reports to make changes to the systems that will optimize system performance. You will need to know when additional resources are required, and be able to specify appropriate choices. You will need to specify and configure operating system parameters, set up file systems, select cloud services, manage system and user PC upgrades in a fast-changing environment, reconfigure networks, provide and ensure the robustness of system security, and perform many other system management tasks. The configuration of large systems can be very challenging. This text will give you an understanding of operating system tools that is essential to the effective management of systems.

In brief, when you complete this book, you will understand what computer hardware and software are and how programs and data interact with the computer system. You will understand the computer hardware, software, and communication components that are required to make up a computer system and what the role of each component in the system is.

You will have a better understanding of what is happening inside the computer when you interact with the computer as a user. You will be able to write programs that are more efficient. You will be able to understand the function of the different components of the computer system and to specify the computer equipment and resources you need in a meaningful way. You will understand the options that you have as a system administrator or Web services or network designer.

In an era in which technology changes extremely rapidly, the architecture of the computer system rests on a solid foundation that has changed only slightly and gradually over the last sixty years. Understanding the foundations of computer system architecture makes it possible to flow comfortably with technological change and to understand changes in the context of the improvements that they make and the needs that they meet. In fact, interviews with former students and with IT executives and other IT professionals clearly indicate that a deep understanding of the basic concepts presented here is fundamental to long-term survival and growth in the field of information technology and IT management.

This type of understanding is at the very foundation of being a competent and successful system analyst, system architect, system administrator, or programmer. It may not be necessary to understand the workings of an automobile engine in order to drive a car, but you can bet that a top-notch race car driver knows his or her engine thoroughly and can use it to win races. Like the professional race car driver, it is our intention to help you to use your computer engine effectively to succeed in using your computer in a winning way. The typical end user might not care about how their computer system works, but you do.

These are the goals of this book. So let's get started!

1.1 THE STARTING POINT

Before we begin our detailed study of the architecture of computer systems, let us briefly review some of the fundamental principles, characteristics, and requirements that guide computer system design and operation. The fundamentals described here apply to computers in general, regardless of size or purpose, from the smallest embedded device to the largest mainframe computer.

In a simple scenario, you use your tablet, laptop, or desktop personal computer to word process a document. You probably use a pointing device such as a mouse or stylus or finger to move around the document and to control the features of the word processor software application, and you use a keyboard or touch screen to enter and modify the document text data. The word processor application program, together with your document, appears on a screen. Ultimately, you might print the document on a printer. You store the document on a disk or flash drive or some other storage device.

The fundamentals of a typical computer system are readily exposed in this simple example. Your pointing device movements and clicks and your text data entry represent input to the system. The computer processes the input and provides output to the screen, and, perhaps, to a printer. The computer system also provides a storage medium of some sort, usually flash memory or a hard disk, to store the text for future access. In simplest terms, your computer receives input from you, processes it, and outputs results to the screen. Your input takes the form of commands and data. The commands and programs tell the computer how to process the data.

Now consider a second, slightly more complex example. Your task in this example is to access a Web page on the Internet. Again, your input to the computer is via keyboard and pointer control device. When you type the Web page URL, however, your computer sends a message to another computer that contains Web server software. That computer, in turn, sends a Web page file that is interpreted by the browser on your computer and presented on your screen. You are probably already aware that HyperText Transfer Protocol (HTTP) is used as a standard for Web message exchanges.

The elements of this example differ only slightly from the first example. Your command inputs tell a Web browser software application on your computer what processing is to take place; in this case, your desire to access a Web page. The output from your computer is a message to a Web server on the remote computer requesting the data that represents the Web page. Your computer receives the data as input from the network; the Web browser processes the data and presents the Web page output on the screen. Figure 1.2 illustrates the layout for this example.