

Book Reviews

S. Somogy

The PowerPC Macintosh Book

Addison-Wesley Publishing Company, Reading, Massachusetts, 1994, pp. xix, 227, ISBN 0-201-62650-0.

The book describes new generation of RISC-based Macintosh computers. It provides extensive coverage of the Power Macintosh hardware and software. The goal of the book is to provide crucial information that every user, manager or organization should know in order to make "...educated decisions about when and how" to migrate to new PowerPC-based Macintosh computers.

The book consists of nine chapters. In the introductory chapter "How We Got Here From There", the author gives the brief history and development of the Apple/IBM/Motorola alliance. Namely, the new Macintosh computers are based on a RISC PowerPC chip — a product of the above mentioned alliance formed in 1991.

Chapter 2 provides an overview of the features of Apple's Power Macintosh hardware. The heart of Mac computers — PowerPC 601 RISC chip. Level 2 cache, direct memory access (DMA), NuBus, Power Macintosh I/O subsystem and other hardware components are described.

Chapter 3 entitled "Power Macintosh Software Overview" provides a brief description of the Power Macintosh software. It introduces and discusses features such as a new version of the System 7, emulator, toolbox acceleration, mixed mode, native QuickDraw, native QuickTime, Virtual Memory, and networking software.

Chapter 4 is an introduction to microprocessors. Fundamental microprocessor concepts such as cycle, address, register, instruction branch, bus, transistor, die, etc. are described. Also, the concepts of pipelining, superpipelining and superscalar implementation are introduced. The characteristics of an instruction set as an architectural issue, RISC versus CISC and other architectural features are explained. Although the book can be used as a reference book, i.e. it does not necessarily have to be read sequentially its organization can be improved if this chapter is transferred to the beginning.

Chapter 5 introduces the PowerPC family. Basic features of the first Power member of the PowerPC family — the Power 601 is described: a program model of this 32-bit processor, 32 K byte unified cache, multiprocessing support, three execution units (BPU, IU, FPU), memory management unit, and bus-interface unit. The 603 as the second member of the family and the PowerPC 604 are also described in this chapter.

Chapter 6 "Emulators on Power Macintosh", is the first of the three PowerMac in-depth chapters. It describes different emulators for the Power Mac and explains how they work.

Chapter 7 provides an in-depth look at the Power Macintosh system's hardware components. The structure of the motherboard of the PowerPC is described with many technical details: the 601 CPU bus, level 2 cache, high-speed memory controller, processor direct slot, NuBus controller, I/O bus, Apple memory mapped I/O controller, etc.

Chapter 8 offers an in-depth look at Power Mac software. Topics like mixed mode, mixed mode switch 64 K to PowerPC and vice versa, code fragment manager, traps and emulated traps are discussed in this chapter.

Chapter 9 looks into the near future of technologies relevant to Power Macs or PowerPC based personal computers.

The book can be recommended as a starting text on a user's way into the world of Power Macintosh computers, yet the users of x86-based PCs may also find it useful for understanding of the new Macintosh computers. Finally, it can be used as a basis for comparison between these two architectures.

Slobodan Ribarić
Faculty of Electrical Engineering
and Computing
University of Zagreb
Zagreb, Croatia

Craig Partridge

Gigabit Networking

Addison-Wesley Publishing Company, Reading, Massachusetts, 1994, pp. xiv, 396, ISBN 0-201-56333-9.

Craig Partridge is a senior scientist at BBN (Bolt, Beranek and Newman), the networking company of the ARPAnet fame that had also an important role in developing the Internet. He is editor-in-chief of *IEEE Network Magazine*, a professor at Stanford University, instructor for INTEROP and a world-wide recognized authority in gigabit networking. This alone, not to mention his ample research work, habilitates him for writing a book on one of the hottest issues in contemporary computer communications.

While reading *Gigabit Networking* a thought came to my mind bringing back to memory the sentence written by Werner Heisenberg in his *Physics and Philosophy*: "Even for a physicist, the description, in plain language, will be a criterion of the degree of understanding that has been reached". This book is an example in support of this maxim. Although designed as professional reference, it is written with such style, that it provides not only referential data

but a pleasure of reading by itself. This is probably due to the fact that most of the structure of this book is based on a course its author had taught at the INTEROP conference for a number of years. The book has clear pedagogical substance, so it can be used as a textbook, at least for graduate courses on gigabit networking. For an undergraduate course in networking it can be used as a supplemental reading, because this text assumes that the reader already has some networking background. The text is well balanced between the needs of professional and academic readers and less technical readers interested in becoming acquainted with the field. Very often technical issues are introduced using nontechnical terms and analogies, and every technical dissemination is ensued with examples in abundance.

The book is structured in sixteen chapters, of which Chapter 1 gives the introduction to gigabit networking and serves as refresher on the terminology in data communication and telecommunication. Chapter 2 is a general introduction to fiber optics, intended to give the reader a good understanding of how optical fiber technology influences network design. Chapters 3 through 6 are a detailed investigation of cell networking, including a full chapter (Chapter 4) on ATM which is the form of cell networking being deployed inside the telephone networks. Cell networking is a technology that seems to have many advantages for gigabit networking, and it has been widely studied and implemented at gigabit speeds. At the same time, cell networking has some disadvantages; so, to try to correct the balance, Chapter 7 looks at the modest but very interesting work done so far on noncell gigabit networks. This first half of the book shows that there is a vast variety of technologies for gigabit networking, all of which are still in development. It seems little probable that any one technology will gain market domination.

As an introduction and motivation to the second half of the book, Chapter 8 looks at gigabit applications and their requirements. Part of the challenge of gigabit networking is to support new types of applications. Chapter 8 introduces the applications and their requirements in preparation for remaining chapters. Chapter 9 looks at problems of interfacing computers to gigabit networks. The problems are not just in making the computers move data swiftly

to and from the network, but also in enhancing the computers to support new applications, like multimedia conferencing, that gigabit networks are enabling. Chapter 10 examines the problem of making today's protocols ready for gigabit networks. Then Chapters 11, 12, and 13 study how to enhance protocols to provide network support for time constrained applications. Chapter 14 examines the surprising ways in which the performance of distributed systems will be affected by gigabit networks and some of the possible solutions to various performance problems. Chapters 15 and 16 conclude with a discussion of open problems in gigabit networking and suggestions for ways to keep track of ongoing work.

Gigabit Networking is a very professionally written book and highly recommendable to a wide group of readers, ranging from undergraduate students to professional specialists. Every chapter of the book is concluded with advice and references for further reading, for readers wanting to delve deeper into the topic discussed in the chapter. At the end of the book there is a very extensive bibliography containing almost three hundred entries. The index is well composed and useful although, for a professional reference book, someone would like it to contain a little more entries.

Dragutin Vuković
MicroLAB Computer Consulting &
Engineering
Zagreb, Croatia

Abraham Silberschatz, Peter B. Galvin

Operating System Concepts,
Fourth edition

Addison-Wesley Publishing Company, Inc.,
pp. ix, 704, Reading, Massachusetts, 1994,
ISBN 0-201-50480-4.

The fourth edition of Silberschatz's and Galvin's textbook is one of the best introductions to operating systems. In the preface of the first edition the authors pointed out that "this book does not concentrate on any particular operating systems

or hardware" and that "it discusses fundamental concepts that are applicable to a variety of systems". This approach was successfully applied throughout all editions. It does not mean that the authors ignored the evolution of real operating systems. On the contrary, every edition was updated with appropriate new information, examples and case studies. The fourth edition was also rewritten and reorganised. It consists of six major parts: overview (chapters 1 to 3), process management (chapters 4 to 7), storage management (chapters 8 to 12), protection and security (chapters 13 to 14), distributed systems (chapters 15 to 18), and case studies (chapters 19 to 21). In all the chapters the older material was brought up-to-date and new material was included.

The major revisions in the fourth edition deal with process management and with distributed systems. The concepts of processes and threads are thoroughly discussed together with their synchronisation and co-ordination mechanisms. The examples are related to Solaris 2 and Mach systems. In the chapters 15 and 16 the authors included the coverage of network protocols and functionality, remote services, thread-management, and the Open Software Foundation's Distributed Computing Environment (DCE) thread package.

The new Appendix, written by Professor Thomas Anderson from UC Berkeley, provides a brief tutorial introduction to Nachos, an instructional operating system intended for use as course project for an undergraduate or first-year graduate course in operating systems. The Nachos is freely available in the public domain and the Appendix concludes with instructions for retrieving it from Internet via ftp.

The book is a very good text for courses in operating systems. With variations that can be made by choice of additional readings, by selection of examples, and by appropriate course project tailoring based on the Nachos operating system, the text should be the base for almost all course profiles in computing and even non-computing curricula.

Leo Budin
University of Zagreb
Faculty of Electrical Engineering
and Computing
Zagreb, Croatia

Woody Leonhard, Vincent Chen, Scott Krueger

Hacker's Guide to Word for Windows, Second Edition

Addison-Wesley Publishing Company, Reading, Massachusetts, 1995, pp. xvi, 910, ISBN 0-201-40763-9.

The large acceptance of MS Windows 3.x made the Word for Windows one of the most popular text processors, the main reason for this being that the tools it offers are not limited just to those available in all Windows applications. Moreover, it includes a powerful WordBasic language which provides a user with almost unlimited variety of tools which he or she can create himself, by writing macros. Unfortunately, WordBasic is not adequately covered in the documentation supplied with WinWord package — the user is left to explore it on his own, cruising through on-line help. The main aim of *Hacker's Guide to Word for Windows* is to help the user, who has mastered the basics, in extending the power of Word for Windows. It also contains an extensive list of previously undocumented information about WinWord, including bugs, gaffes and workarounds. The second edition of the *Hacker's Guide* covers both versions 2.0 and 6.0 of WinWord.

The authors are recognized experts in WinWord. Introducing themselves, the authors claim that they “aren't skills for Microsoft; don't wear tie-dyed T-shirts or designer sneakers; don't bow to Redmond five times a day...”. Woody Leonhard is a contributing editor for *PC/Computing magazine*. He is also the author of *The Underground Guide to Word for Windows*. Vincent Chen is a co-author of the *PC/Computing* “Toolkit” column. Scott Krueger was a product support engineer for Microsoft. Currently, he creates software for and writes about Word for Windows. He is also a contributor to *PC Magazine* and *Windows Tech Journal*.

The book is divided into five sections, of which the fourth section *WordBasic Reference*, is the largest (extends on more than 650 pages!) and most valuable. What is more, there is a companion disk included with Hacker6 Library routines and a shareware version of WOPR. Hacker6 is a collection of valuable macros that help to overcome some of WinWord bugs. WOPR

(for “Woody's Office Power Pack”) is an add-on which includes dozens of features that Microsoft missed.

Section 1, *Concepts, Cabbages, Kings*, is a very useful overview of basic WinWord features. Although some experienced users may find this section not altogether necessary, I, who like to think of myself as of an experienced user, found quite a few useful tips. The section includes valuable subsections on most frequently asked questions, bugs and on writing macros.

The WordBasic Overview section is an introduction intended mainly for those who have not programmed in (any) Basic and/or written WordBasic macros yet. Despite the fact that most of the material is covered in the User's Guide, even an advanced user will find some useful hint in this section, e.g. how to speed up the macro execution by preventing screen updating. This section ends with a concise and systematic preview of built-in macro commands.

Section 3 contains a brief discussion on creating and using Word's new dynamic dialogs. This is a new powerful feature added to Word 6.0. It enables the display of a dialog to change after pushing a button or entering some data, like, for example, Format Font dialog shows the samples of font you have selected. The section includes examples of how to use dynamic dialogs and circumvent the limitations.

As already stated, the fourth section (*WordBasic Reference*) is the principal part of the whole book. It contains the reference list of the WordBasic, WOPR Library and Hacker6 Library commands. The commands are listed in alphabetical order, each being extensively examined. An example is often included with each command. If a command has parameter(s), they are listed in a tabular form. The commands new to version 6.0 are annotated, which makes this reference useful even to users of the previous WinWord version. But the most valuable are the signs which will warn you of a bug in a command. Usually, a workaround tip is given or suggested.

The last section, *Fields, Bookmarks, INI*, contains a descriptive list of WinWord fields, description of the built-in bookmarks and explanation of settings in WINWORD6.INI.

The appendix consists of two parts: the first describes the content of the companion disk, i.e. Hacker6 library and shareware version of

WOPR. The second one gives a list of printable ANSI Windows characters, and a list of all characters sorted in the same manner as WinWord does. Namely, WinWord has an odd feature: it does not use language-specific information from Windows language DLL file, but sorts all characters according to their internal sequence. Sorting in any Western European language will work properly, but if you use any other language (like the author of this review does), items with language specific characters may appear in an unexpected position (unless there is a version of WinWord adapted for that language). Therefore, the user must rely on his or her programming skills to overcome this shortcoming. This is where the given sort sequence becomes very useful!

When I first caught sight of this book in a bookshop, I was repelled by its title (I am somehow skeptical about hackers). I must admit, I was wrong! The book bristles with valuable information, especially about WinWord bugs and advices how to overcome them. These bits of information are noticeably marked by visual gimmicks, i.e. small drawings of bug(s), wizard, frantic person etc. The amusing and popular style of the book will keep attention of every reader. The heart of the book, *WordBasic Reference* section, is the most helpful and elaborated list of macro commands which will force the user to keep this book always at hand (an edition on CD-ROM would be a great idea!).

In closing, let me say that this book is definitely the most extensive collection of valuable information on a product I have ever seen. Even without add-on libraries on the companion disk, the book is sufficient to spare a lot of your time and nerves. Therefore, this is a must-book for every serious user of WinWord. Moreover, I believe that having read this book, most users will share my impression that there are just two books that an experienced user of WinWord should have. The first is *User's Guide* which tells us what WinWord command was meant to do. The second one is *Hacker's Guide* which explains what that command really does and how to force it to do what it should be doing.

Julijan Šribar
Faculty of Electrical Engineering
and Computing
University of Zagreb
Zagreb, Croatia

Mary E. S. Loomis

Object Databases: The Essentials

Addison-Wesley Publishing Company, Reading, Massachusetts, 1994, pp. xv, 230, ISBN 0-201-56341-X.

One of the motivations for the development of object database management systems (ODBMSs) can be tied to the recently growing popularity of multimedia data (such as graphics, sound and video). This trend has boosted the search for data management options alternatives, since present tools offer insufficient functionality for complex data manipulation. Another reason for introducing ODBMSs can be located in the expectation that the combination of object oriented (OO) analysis and design, OO programming language and OO database could offer benefits of a synergistic development tool. This is especially important in the case of large application domains based on simple alphanumeric data that involve associations too complex for today's relational database management systems (RDBMSs). As a consequence, application developers who have decided to use an OO programming language often later decide to use an ODBMS because it fits well with the programming environment.

Mary E. S. Loomis is a highly qualified person to author a book on ODBMSs because of her extensive present and previous expertise in the field of ODBMSs. A vice president of technology with Versant Object Technology, leading vendor of ODBMS products, she is currently heading the Software Technology Lab at Hewlett Packard Laboratories. She also writes a regular column in the *Journal of Object-Oriented Programming*, "ODBMS", and is a featured speaker at database and object-oriented conferences.

The book *Object Databases: The Essentials* deals with fundamental aspects of object databases. According to the author's statement in the preface, this is a book "about databases that are closely coupled with one or more object programming languages, usually C++ and/or Smalltalk".

The book consists of 11 not too much related chapters, a bibliography and an index. It begins with a chapter that explains the role of object database management systems (ODBMSs) and also contains overviews of common terminology and basic principles of object technology.

Chapter 2 discusses the properties which object programmers typically expect from object databases. For object programmers a basic feature of an object database is a persistent object storage; *invisibility* object databases is another desired property meaning the programmers can stay within the same language paradigm without having to switch models, modes of thinking or language syntax when dealing with the persistent objects. Chapter 3 elaborates on object databases from a database management perspective. It deals with ODBMSs functions related to concurrency control, transaction management, schema management, recoverability, and access control. Chapter 4 confronts the technical areas where programming languages and the relational and extended-relational DBMS perspectives clash. Chapter 5 describes what has to be done to make a relational database appear to contain objects. Since RDBMSs do not manage object models, the programmer has to write a large amount of additional code to deal with semantic structures like relationships and inheritance.

Chapters 6 through 9 are predominantly technically oriented. Chapter 6 explores in more detail the object model developed by the Object Database Management Group (ODMG) consortium of ODBMS vendors. Chapter 7 describes techniques used by ODBMSs to store and find objects like addressing modes, caching, clustering, indexing and replication. Chapter 8 explains how ODBMSs enable object sharing by addressing the issues of locking, logging, transaction management and versioning, which are fundamental functions for database management. Query processing and optimization capabilities are described in Chapter 9. The conclusion of this chapter is that ODBMS products still do not offer query facilities as some relational DBMSs do. Chapter 10 shortly elaborates on different parameters and related factors that can be used to evaluate the ODBMS products. Chapter 11 concludes the book with the author's opinion about the future development direction of the ODBMS market. It would be erroneous

to conclude that it is doomed to a stall, in spite of the fact that it hasn't shown development trends as predicted earlier. Moreover, it is a widely accepted belief that the ODBMS market still has the potential to boom in the near future.

ODBMSs will coexist with other kinds of database management for a long time. Therefore it is important to understand the characteristics of each generation and the kind of application best suited to each of them. This book provides sources and criteria to be considered when evaluating databases. It is also written in a well structured style. The structure of each chapter is of the overview-detail-summary type. Additionally, some chapters offer sidebars where the author gives more details about the specific issues of the subject matter. It shows an excellent balance of readability and technical depth, so it can be recommended both for readers with minimum previous knowledge about object technology as well as readers with experience in these areas.

Dubravka Pukljak
Faculty of Electrical Engineering
and Computing
University of Zagreb
Zagreb, Croatia

William R. Ogden, Richard Schreiber

Distributed CICS, An In-Depth Assessment for Downsizing Applications

John Wiley & Sons, New York, 1994, pp. xiii, 267, ISBN 0-471-06055-0.

Four years ago my team had a contract to solve the automatic coding of optically read free text from the forms of the 1991 Census in Croatia and in Bosnia-Herzegovina [1]. The expected most challenging aspect was the text recognition. The processing on an IBM 4381 mainframe was so slow, that two PCs with i860 cards had to be used instead. Thus, this most demanding task was successfully accomplished. However, an unexpected problem for my team arose: we had also to satisfy the contract requirement, that in cases of uncertainty the coding should

not proceed automatically, but offer provision for human arbitration. This was a conceptually simple multiuser application that should have displayed up to nine best matches on the screen and recorded the operator's choice. It was absolutely necessary that this application has run under the Customer Information Control System (CICS) to be viable on the mainframe used. As we, people from the UNIX and MS-DOS world, encountered the CICS environment, that was quite a shock. I immediately felt 25 years younger. It reminded me of arcane job control statements for the IBM System/360 I had to deal with at the beginning of my professional career. I wondered, how it was possible that there still were apparently normal people around who used and even praised such stuff.

The reviewed book gives some answers: "CICS has been in wide commercial use for over twenty years. It is a proven, industrial-strength product, used for mission-critical applications by . . . 490 of IBM's top 500 customers". The product stability and reliability are points easily neglected by computing professionals from academic community, where progress and modernity are sometimes uncritically embraced. The book states that "over 300.000 programmers have some experience writing CICS programs". For a scholar it may be too easy to state that "they should learn modern tools". For some of those people it is definitely true, but for some others it can be too late. They might never gain the efficiency and self-confidence on a new platform as they have achieved by programming in CICS for decades. Taking into account the chronic applications backlog and the shortage of competent and reliable programmers, the picture gets clearer. Much acclaimed bright hackers and neighbor's kids are completely unusable for the production of serious commercial applications. The problem gets worse when the companies enchainned by CICS are forced to renew or enlarge their hardware. They have no other choice but to buy or lease conceptually obsolete, complicated and expensive computing equipment. My personal experience is, that no arguments can help against it and people in charge can be perceived as CICS addicts.

The appearance of such products as CICS/6000 for the IBM RISC System/6000 AIX platform is more than welcome. It shows the best of IBM. It unites the world of renown IBM reliability and stability with the processing speed and

modern computing technology. While introducing modern equipment, it ensures the decision maker's peace of mind in the way that probably only the acquisition of IBM products can provide. It enables the exploitation of existing software and available programming skills. At the same time the hardware and system software platforms offer the development of new applications with modern tools. CICS, maybe like COBOL, is a good survivor, but I strongly believe that its days are counted. The book authors believe otherwise. A book that addresses CICS fans and persuades them to switch to UNIX and other non-mainframe platforms, without abandoning their beloved environment, can do much good and spare us from huge investments in inappropriate computing platforms. The reviewed book can help in accomplishing this task.

The book consists of the following chapters: Preface, Introduction, Transaction Processing, CICS Fundamentals, CICS Application Programming Interface, Distributed Computing Environment and ENCINA Set of Products, CICS/6000, CICS OS/2, CICS/400, UniKix, Micro Focus Transaction System, HP CICS/9000, Selecting CICS Products, two Appendices and an Index.

The book appears to be competently written, it is highly readable and full of practical considerations, in-depth analyses, hints and information. I do not see any need to go into chapter details in this review. For the connoisseurs the titles would be sufficient. For those who are not, the book makes no sense because, at least in my opinion, they should not start learning now about a more than twenty years old software product.

I can highly recommend the book to many a member of the not so small CICS community. It is intended for technical managers, experienced CICS programmers and designers.

Damir Kalpić
Faculty of Electrical Engineering
and Computing
University of Zagreb
Zagreb, Croatia

References

- [1] D. Kalpić: "Automated Coding of Census Data", *Journal of Official Statistics*, Vol. 10, No. 4, 1994, pp. 449–463, Stockholm

