Books

OS/2™ Features, Functions, and Applications, Standard Edition 1.0, Jeffrey I. Krantz, Ann M. Mizell, and Robert L. Williams, John Wiley & Sons, Inc., New York, NY, 1988. 282 pp. (ISBN 0-471-60709-6).

The authors state that they are "key members of the OS/2™ design team" and their objectives include sharing their detailed technical knowledge, selecting the most important features of Operating System/2™: Standard Edition 1.0, describing why the capabilities of OS/2 are important, and how these capabilities solve problems.

The book describes the OS/2 concepts for application developers and includes an overview of the 80286 microprocessor, describing the features which are made possible by this microprocessor (such as multitasking and virtual memory) and why they were not part of PC DOS. Operating systems concepts are presented intuitively as necessary. The book is also useful for those wanting an overview of the features of OS/2, to evaluate OS/2 as an environment for application development, or to understand the implementation issues related to an application's development and user services.

I found that knowledge of systems programming concepts and PC DOS concepts was helpful in understanding the book, even though the terminology was explained by the authors. The explanations at times occurred later in the book than their first uses, and, although they were well done and correct, were given without sufficient examples for someone without background.

The book describes concepts first in a general sense, then in terms of the applications programming interfaces, and finally offers some actual C program examples.

Chapter 1, "The OS/2 Perspective" provides background material necessary for the rest of the book. It describes advances which PC DOS does not provide: larger memory, multitasking, enhanced application programming interface, and ability to support multiple applications. It also describes the 80286 microprocessor and its use for providing access to larger real memory and virtual memory, protection of code and data from other applications, and protection of the system itself. The addressing, registers, and interrupt and protection mechanisms of the microprocessor are described as they relate to the features of OS/2, and some of their uses by OS/2 are given. This gives the reader an overview of key aspects of OS/2.

Chapter 2, "OS/2 Memory Capabilities" explains how OS/2 permits access to larger real memory than DOS and how it supports virtual memory. The segmentation techniques used to swap, discard, and move portions of memory are presented. The mechanisms used to support segmentation are also used to support memory protection and the ability of an application to load all of its segments (preload), or to permit them to be loaded only when needed (on demand). Applications can access system routines through interfaces provided. The functions of these interfaces are summarized and include the abilities to allocate, free, name, share, and lock memory segments.

Chapter 3, "Multitasking and Multiple Applications" presents the ability to do more than one thing at a time within an application (multitasking) and to run multiple applications simultaneously. Unlike DOS, OS/2 permits the overlapping of input/output and processing. There are services provided to manage asynchronous tasks in contrast to the synchronous environment of DOS. In the OS/2 environment, the terms *session*, *process*, and *thread* are used. Although these and the interfaces to support them are described, I would have preferred clearer definitions. Other aspects described are interprocess communication, timeslicing, and the scheduling algorithm.

The chapters which follow continue to describe features provided by the operating system, and they build on the knowledge presented in the early chap-

ters. They do not have to be read in sequence. Programming examples are provided to illustrate some of the concepts.

Chapter 4, "Illustrating Concepts in Programming Examples" describes the version of the C programming language used, so that the reader need not be a C programmer. Here, comments and text descriptions make up for places within the programs which are not easily readable. The examples were chosen to illustrate the types of considerations and techniques which must be understood and used by applications programmers wishing to use the features of OS/2 which are more powerful than those of DOS.

Chapter 5, "Application Input/Output Capabilities" and chapter 6, "Interrupt-Driven Device Management" describe using input/output devices through the interfaces provided, and writing device drivers using interrupts (instead of the polling logic of DOS). Chapter 7, "Advanced Programming Concepts" describes and illustrates combining modules and libraries through a program to build an executable application.

Each of the remaining chapters deals with a smaller, specific topic. Chapter 8 discusses capabilities for translating messages and character sets into different languages; chapter 9 describes support provided to permit the user to modify the system defaults; and chapter 10 introduces features to be provided by future versions of OS/2.

This book may not be the book of choice for people curious about the applicability of OS/2 to their problems, since it does not describe the command language interface and does require some technical background. However, it is useful for most of its intended audience, which includes systems analysts and designers and application developers.

Fran Goertzel Gustavson
Pace University
Pleasantville
New York

OS/2™ Programmer's Guide, Ed Iacobucci, Osborne McGraw-Hill, Berkeley, CA, 1988. 1100 pp. (ISBN 0-07-881300-X).

This book, written by the leader of the IBM OS/2™ design team, offers a vast amount of well-written material "for programmers who want to dive in and start using OS/2." All OS/2 versions are based on the architectural framework of OS/2 Standard Edition Release 1.0 described in the book.

Initially, I found the material overwhelming because details of the Intel architecture were new to me. I was absorbing them and old operating systems concepts presented in new ways, as well as an overview of a new operating system, all at once. Some of the terminology was defined later, but I could identify meanings from context, and the index is good. Once I passed "Part 1, Introduction," I began to admire the writing style, choice of sequencing, use of examples, and OS/2 itself. Each chapter begins by describing the chapter contents and reviewing earlier concepts as appropriate; each ends with a summary and a list of key terms and concepts ("important points").

The book should be useful for those with technical backgrounds wanting to evaluate OS/2 for application development or use, as well as for analysts and managers of programmers, and programmers. It describes the features of this system which are beyond those found in PC DOS, the advantages of each system, and how to write programs which take advantage of system features. The command language interface to the user is presented with good examples of command lines and of command files.

"Part 1, Introduction" introduces OS/2 as a new operating system built for Intel 80286/80386 protected-mode operation and describes the system and the 80286 microprocessor. The system was designed for future growth, so many of the system functions are replaceable and extendable. Large real memory and virtual memory are supported. Memory protection allows the system and applications to be protected from unplanned modification. Multiple applications can run simultaneously.

"Part 2, How OS/2 Works" describes the structure of the system as a whole and its components and how they interact with each other. These concepts are needed for designing programs and are applied in Part 3.

"The OS/2 Application Programming Interface" describes its characteristics and implementation and includes information about writing an application which will run under both OS/2's protected mode and PC DOS.

"Memory Management" includes a description of the memory manager and how it provides a multiprogramming environment with programs isolated from each other and with memory sharing, segmentation, and virtual memory.

"Multitasking and Dynamic Linking" describes program management and task control. The tasking model used by OS/2 involves threads, processes, and sessions, and these are carefully defined; their creation and use through the API (Application Programming Interface) are described. The OS/2 dispatching scheme and its four priority classes—time critical, foreground, regular, and idle—are explained. Other topics include: time slicing, the system timer, interprocess communications (pipes, queues, flags, shared memory and semaphores), and loading and linking (statically and dynamically).

"The File System" includes the structure of a disk, the file system model, file manipulation functions, the ability to lock and to share files, implementation of the file system, and input/output redirection.

"Resource Management" describes management of the hardware resources in a system in which more than one program is running concurrently.

"Device Drivers" are routines associated with each device and are the interface for the supported device. They isolate both applications and the OS/2 kernel from the PC hardware.

"I/O Subsystems" continues the discussion with input/output, including video, keyboard, and mouse functions. The OS/2 session is one or more processes which are running at the same time and share the keyboard, video display, and mouse. It generally corresponds to an application program, but the system permits a user to start up to 12 concurrent sessions.

"Session Management" first describes session management. It then goes on to describe the command interpreter, including the operators used for input/output redirection and other special operators. The power of pipes and filters is illustrated well. The command language, as well as the command line, is presented.

"System Topics" include supporting DOS applications, error processing, and support for other natural languages.

"OS/2 Commands and Utilities" includes aborting an application, creating and accessing subdirectories within the hierarchical file system, setting the date and time, and many more commands and services provided to the user to be accessed at the command line.

"Part 3, Using OS/2" is as comprehensive as Part 2. It uses the concepts of Part 2 in programs. These programs are written in assembler language using macros and comments to enhance readability. They illustrate the use of the programming interfaces and are useful examples. The appendices include more details of OS/2 function calls, and the sample programs, including the macros.

Fran Goertzel Gustavson
Pace University
Pleasantville
New York