Operating systems are large and complex, and yet must function with near-absolute reliability—that's why they're a class unto themselves in the field of software development. Since its first release 20 years ago, the dinosaur book—Operating System Concepts by Abraham Silberschatz, Peter Baer Galvin, and Greg Gagne—has been a valuable reference for designers and implementers of operating systems. The newly released sixth edition of this book maintains the volume's authority with new sections on thread management, distributed processes, and the Java Virtual Machine (JVM). There's also information on the workings of the latest crop of operating systems, including Microsoft Windows 2000, Linux, FreeBSD, and compact operating systems for handheld devices. This book is concerned with the design of operating systems, which is to say it enumerates the problems that pop up in the creation of efficient systems and explores alternative ways of dealing with them, detailing the advantages and shortcomings of each. For example, in their chapter on scheduling CPU activity, the authors explain several algorithms (first-come, first-served, and round-robin scheduling, among others) for allocating the capacity of single and multiple processors among jobs. They highlight the relative advantages of each, and explain how several real-life operating systems solve the problem. They then present the reader with exercises (this book is essentially a university textbook) that inspire thought and discussion.

My Personal Review:
This popular book was written as an introductory course to operating systems but systematically provides an extensive description of operating system concepts. The 1st half of the book is typically used for undergraduate computer science classes although the book as a whole is often required for graduate level classes.

It is assumed that readers will have some knowledge of high-level languages and general computer organization. The book does not spotlight any one particular operating system but rather presents concepts and algorithms that are common to many of the Oss that are commonly used today, including MS-DOS, Windows 2000 & NT, Linux, Sun Microsystems Solaris 2, IBM OS/2, Apple Macintosh, and DEC VMS.

The book has 7 major parts:
1) Overview: What Operating Systems are, what they do, how they are designed, and where they came from. General history and explanations. Some discussion on hardware.
2) Process Management: How information is processed. Methods for process scheduling, interprocess communication, process synchronization, deadlock handling, and threads.
3) Storage Management: How main memory functions and executes. The mechanisms for storage of and access to data is covered. The classic internal algorithms and structures of storage management is discussed and the advantages and disadvantages of each.
4) I/O Systems: The types of devices that attach to a computer. How the devices are accessed and controlled. Performance issues and examined thoroughly.
5) Distributed systems: The collection of processors that do not share a clock or memory. How distributed file systems are shared, synchronized, communicate, and deal with deadlocks.
6) Protection and Security: How mechanisms ensure that only certain processes that have obtained proper authorization can use certain files, memory segments, CPU, etc.
7) Case Studies: This is where individual real operating systems are discussed in depth. These systems are Linux, Windows 2000, FreeBSD, Mach, and Nachos.

Of course this is a very general list and omits many other aspects of Operating Systems that are included in the book. This 887 page book does not include formal proofs but it does contain (though it would be better to have more) figures, diagrams, examples, and notes to help explain concepts.

For More 5 Star Customer Reviews and Lowest Price: