

Operating- System Structures



An operating system provides the environment within which programs are executed. Internally, operating systems vary greatly in their makeup, since they are organized along many different lines. The design of a new operating system is a major task. It is important that the goals of the system be well defined before the design begins. These goals form the basis for choices among various algorithms and strategies.

We can view an operating system from several vantage points. One view focuses on the services that the system provides; another, on the interface that it makes available to users and programmers; a third, on its components and their interconnections. In this chapter, we explore all three aspects of operating systems, showing the viewpoints of users, programmers, and operating system designers. We consider what services an operating system provides, how they are provided, how they are debugged, and what the various methodologies are for designing such systems. Finally, we describe how operating systems are created and how a computer starts its operating system.

Bibliographical Notes

[Brinch-Hansen (1970)] was an early proponent of constructing an operating system as a kernel (or nucleus) on which more complete systems could be built. [Bryant and O'Hallaron (2015)] provide an overview of computer systems, including the role of the linker and loader. [Levin (2013)] covers the internals of both macOS and iOS, and [Levin (2015)] covers details of the Android system. Windows internals are covered in [Rusinovich et al. (2017)]. BSD UNIX is described in [McKusick et al. (2015)]. [Love (2010)] and [Mauerer (2008)] thoroughly discuss the Linux kernel. In particular, [Love (2010)] covers Linux kernel modules and kernel data structures. Several UNIX systems—including Mach—are treated in detail in [Vahalia (1996)]. Solaris is fully described in [McDougall and Mauro (2007)].

DTrace is discussed in [Gregg and Mauro (2011)]. The DTrace source code is available at <http://src.opensolaris.org/source/>. Comprehensive coverage of Linux kernel modules can be found at <http://www.tldp.org/LDP/lkmpg/2.6/lkmpg.pdf>. [Ward (2015)] and <http://www>

.ibm.com/developerworks/linux/library/l-linuxboot/ describe the Linux boot process using GRUB. Performance tuning—with a focus on Linux and Solaris systems—is covered in [Gregg (2014)].

Bibliography

- [Brinch-Hansen (1970)] P. Brinch-Hansen, “The Nucleus of a Multiprogramming System”, *Communications of the ACM*, Volume 13, Number 4 (1970), pages 238–241 and 250.
- [Bryant and O’Hallaron (2015)] R. Bryant and D. O’Hallaron, *Computer Systems: A Programmer’s Perspective*, Third Edition (2015).
- [Gregg (2014)] B. Gregg, *Systems Performance—Enterprise and the Cloud*, Pearson (2014).
- [Gregg and Mauro (2011)] B. Gregg and J. Mauro, *DTrace—Dynamic Tracing in Oracle Solaris, Mac OS X, and FreeBSD*, Prentice Hall (2011).
- [Levin (2013)] J. Levin, *Mac OS X and iOS Internals to the Apple’s Core*, Wiley (2013).
- [Levin (2015)] J. Levin, *Android Internals—A Confectioner’s Cookbook. Volume I* (2015).
- [Love (2010)] R. Love, *Linux Kernel Development*, Third Edition, Developer’s Library (2010).
- [Mauerer (2008)] W. Mauerer, *Professional Linux Kernel Architecture*, John Wiley and Sons (2008).
- [McDougall and Mauro (2007)] R. McDougall and J. Mauro, *Solaris Internals*, Second Edition, Prentice Hall (2007).
- [McKusick et al. (2015)] M. K. McKusick, G. V. Neville-Neil, and R. N. M. Watson, *The Design and Implementation of the FreeBSD UNIX Operating System—Second Edition*, Pearson (2015).
- [Rusinovich et al. (2017)] M. Rusinovich, D. A. Solomon, and A. Ionescu, *Windows Internals—Part 1*, Seventh Edition, Microsoft Press (2017).
- [Vahalia (1996)] U. Vahalia, *Unix Internals: The New Frontiers*, Prentice Hall (1996).
- [Ward (2015)] B. Ward, *How LINUX Works—What Every Superuser Should Know*, Second Edition, No Starch Press (2015).