

- Free and open-source operating systems are available in source-code format. Free software is licensed to allow no-cost use, redistribution, and modification. GNU/Linux, FreeBSD, and Solaris are examples of popular open-source systems.

Practice Exercises

- 1.1 What are the three main purposes of an operating system?
- 1.2 We have stressed the need for an operating system to make efficient use of the computing hardware. When is it appropriate for the operating system to forsake this principle and to “waste” resources? Why is such a system not really wasteful?
- 1.3 What is the main difficulty that a programmer must overcome in writing an operating system for a real-time environment?
- 1.4 Keeping in mind the various definitions of *operating system*, consider whether the operating system should include applications such as web browsers and mail programs. Argue both that it should and that it should not, and support your answers.
- 1.5 How does the distinction between kernel mode and user mode function as a rudimentary form of protection (security)?
- 1.6 Which of the following instructions should be privileged?
 - a. Set value of timer.
 - b. Read the clock.
 - c. Clear memory.
 - d. Issue a trap instruction.
 - e. Turn off interrupts.
 - f. Modify entries in device-status table.
 - g. Switch from user to kernel mode.
 - h. Access I/O device.
- 1.7 Some early computers protected the operating system by placing it in a memory partition that could not be modified by either the user job or the operating system itself. Describe two difficulties that you think could arise with such a scheme.
- 1.8 Some CPUs provide for more than two modes of operation. What are two possible uses of these multiple modes?
- 1.9 Timers could be used to compute the current time. Provide a short description of how this could be accomplished.
- 1.10 Give two reasons why caches are useful. What problems do they solve? What problems do they cause? If a cache can be made as large as the

device for which it is caching (for instance, a cache as large as a disk), why not make it that large and eliminate the device?

- 1.11 Distinguish between the client–server and peer-to-peer models of distributed systems.

Further Reading

Many general textbooks cover operating systems, including [Stallings (2017)] and [Tanenbaum (2014)]. [Hennessy and Patterson (2012)] provide coverage of I/O systems and buses and of system architecture in general. [Kurose and Ross (2017)] provides a general overview of computer networks.

[Rusinovich et al. (2017)] give an overview of Microsoft Windows and covers considerable technical detail about the system internals and components. [McDougall and Mauro (2007)] cover the internals of the Solaris operating system. The macOS and iOS internals are discussed in [Levin (2013)]. [Levin (2015)] covers the internals of Android. [Love (2010)] provides an overview of the Linux operating system and great detail about data structures used in the Linux kernel. The Free Software Foundation has published its philosophy at <http://www.gnu.org/philosophy/free-software-for-freedom.html>.

Bibliography

- [Hennessy and Patterson (2012)] J. Hennessy and D. Patterson, *Computer Architecture: A Quantitative Approach*, Fifth Edition, Morgan Kaufmann (2012).
- [Kurose and Ross (2017)] J. Kurose and K. Ross, *Computer Networking—A Top-Down Approach*, Seventh Edition, Addison-Wesley (2017).
- [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).
- [McDougall and Mauro (2007)] R. McDougall and J. Mauro, *Solaris Internals*, Second Edition, Prentice Hall (2007).
- [Rusinovich et al. (2017)] M. Rusinovich, D. A. Solomon, and A. Ionescu, *Windows Internals—Part 1*, Seventh Edition, Microsoft Press (2017).
- [Stallings (2017)] W. Stallings, *Operating Systems, Internals and Design Principles (9th Edition)* Ninth Edition, Prentice Hall (2017).
- [Tanenbaum (2014)] A. S. Tanenbaum, *Modern Operating Systems*, Prentice Hall (2014).