Errata[1]

OPERATING SYSTEM CONCEPTS, TENTH EDITION (print edition)
Silberschatz, Galvin, and Gagne

June 5, 2024

We list below errors, clarifications, and recent updates. NOTE: If you own an international edition note that these editions follow a different correction schedule, so your copy may still have errata that have been fixed in the US edition. Check your copy for the errata noted here, and ignore those that have been fixed in your copy.

Preface:

Page viii, second bullet: replace with ➔ **E-text with print companion bundle**. For a nominal additional cost, the e-text also is available with an abridged print companion that includes the main chapters text, end-of-chapter "Practice Exercises", and "Further Reading" sections.

Page xvii, line 2: Rick ➔ Rik

Contents:

Page xxii: Chapter A ➔ Appendix A
Page xxii: Chapter B ➔ Appendix B
Page xxii: Chapter C ➔ Appendix C
Page xxii: Chapter D ➔ Appendix D

Chapter 1:

Page 8, line -6: assicated ➔ associated
Page 13 line 19: blu-ray ➔ Blu-ray
Page 25 line 17: The instruction to switch to kernel mode is an example of a privileged instruction. Some other examples include ➔ The instruction to switch to kernel mode is is handled specially (sometimes via a trap, sometimes as a unique instruction). Examples of privileged instructions include
Page 30 line -16: CD DVD ➔ CD, DVD
Page 31 figure 1.14 SRAM ➔ DRAM
Page 35 line -14: ESXand ➔ ESX and

Chapter 2:

Page 76 line -15: most systems allow a program to dynamically link libraries as the program is loaded ➔ most systems allow a program to dynamically link libraries as the program is loaded or even when it is executed
Page 83 figure 2.13: glibc standard c library ➔ glibc standard C library
Page 86 line -8: We cover creating LKMs in Linux in several programming exercises at the end of this chapter ➔
Page 94 line -16: CPUand ➔ CPU and
Page 97 line 8: In both this chapter and Chapter 3, we provide programming projects where you will create and

---

[1] Errors reported by: Peter Galvin, Greg Gagne, John Trono, Zdzislaw Ploski, Sinan Hanay, Bond James, Zhiguang Yin, Muhammad Wasif

access the /proc file system. ➔

Chapter 3:

Page 108 example box line 10 agrv ➔ argv
Page 108 example box line -4 The data field refers to unitialized data, and the bss refers to initialized data ➔ The data field refers to initialized data, and the bss refers to unitialized data
Page 112 line 11: CPU's core This ➔ CPU's core. This
Page 112 line 12: header contains pointers ➔ header contains a pointer
Page 115 line 2: of special instructions (such as a single instruction to load or store all registers). A typical speed is a several microseconds. ➔ of special instructions (such as a single instruction to load or store all registers), and typical takes hundreds to thousands of nanoseconds.
Page 120 line 3: VOID ➔ void
Page 133 Figure 3.16: line 19: obect ➔ object
Page 134 Figure 3.17: line 15: obect ➔ object
Page 134 Figure 3.17: line 21:
ptr = (char *) mmap(0, SIZE, PROT_READ | PROT_WRITE, MAP_SHARED, fd, 0); ➔
ptr = (char *) mmap(0, SIZE, PROT_READ, MAP_SHARED, fd, 0);
Page 137 Figure 3.18: line -7: MACH_RCV_MSG, // sending a message ➔ MACH_RCV_MSG, // receiving a message
Page 143 Figure 3.23: line 5: VOID ➔ void
Page 144 Figure 3.24: line 10: START_INFO ➔ STARTUPINFO
Page 144 line -10: pipe. ➔ pipe
Page 145 Figure 3.25: line 3: VOID ➔ void
Page 145 Figure 3.25: line 6: Readhandle ➔ ReadHandle

Chapter 4:

Page 181 line 11: problem) ➔ problem))
Page 181 line 12: problem) ➔ problem))
Page 186 line 18: block ➔ block.
Page 187 line -13: The second parameter is a C++ lambda ➔ The third parameter is a C++ Lambda
Page 196 line 14: parent process ➔ parent task
Page 196 line -17: Finally, the flexibility of the clone() system call can be extended to the concept of containers, a virtualization topic which was introduced in Chapter 1. Recall from that chapter that a container ➔ Finally, the flexibility of the clone() system call can be extended to the concept of containers, a virtualization topic which is covered in chapter 18. A container

Chapter 5:

Page 242 line 6: most optimal ➔ optimal
Page 250 line -3: real-time tasks, ➔ real-time tasks.
Page 252 line -4: listed below ➔ listed above

Chapter 6:

Page 261 line 18: variable kernel variable ➔ kernel variable
Page 266 line -8: Hardware Instructions ➔ Further Hardware Instructions
Page 269 line -5: Atomic variables can be used in to ensure ➔ Atomic variables can be used to ensure
Page 275 line 6: is negative, its magnitude ➔ is negative, its absolute Value
Page 287 line -10: Section 6.4 ➔ Section 6.2
Page 288 line -4: Mckenney ➔ McKenney

Chapter 7:

Errata - page 3

Page 290 line -15: semaphore full ➔ semaphore full;
Page 299 line 9: thread-info ➔ thread_info
Page 300 line -17: POSIX specifies ➔ POSIX SEM specifies
Page 303 line 13: We provide several programming problems and projects at the end of this chapter that use Pthreadsmutex locks and condition variables, as well as POSIX semaphores. ➔

Page 312 line -13: In Section 4.5.2 ➔ In Section 4.5.3

Chapter 8:

Page 329 line -3: 25.0) ➔ 25.0);
Page 329 line -1: 50.0) ➔ 50.0);
Page 332 line 18: 9 2 ➔ 9 7
Page 338 line -10: the possibility of deadlock ➔ the possibility of a deadlock.
Page 343 line -12: indicates deadlock. ➔ indicates the possibility of a deadlock.

Chapter 9:

Page 351 line 10: new Figure 9.1
Page 356 line 10: the DLL in main memory ➔ the DLL is in main memory
Page 363 line -6: Pthreds ➔ Pthreads
Page 366 line -14: LRU entry replacement ➔ TLB entry replacement
Page 380 line -10: used to generate a linear address ➔ used to generate the address
Page 385 line 14: that contains the frame ➔ that contains the frame number
Page 385 line 21: if the frame for the page is in the TLB. If it is, the frame is obtained from the TLB. If the frame is not present in the TLB, it must be retrieved from the page table. ➔ if the frame number for the page is in the TLB. If it is, the frame number is obtained from the TLB. If the frame number is not present in the TLB, it must be retrieved from the page table.
Page 385 line -9: ARMv9 ➔ ARMv8
Page 387 line 8: PAE support for Windows systems.is ➔ PAE support for Windows systems is
Page 387 line 9: .aspx An overview ➔ .aspx. An overview

Chapter 10:

Page 399 line 20: these systems demand-page ➔ these systems demand page
Page 399 line 22: data can be demand-paged ➔ data can be demand paged
Page 435 Figure 10.28 title: why frames used ➔ why pages used

Chapter 11:

Page 456 line 20: between the cache host DRAM ➔ between the cache and host DRAM
Page 465 line 7: DIRECT ➔ <code font>DIRECT</code font>
Page 485 line 3: hard drives and nonvolatile ➔ hard drives and other nonvolatile

Chapter 12:

Page 495 figure 12.4 title: Mac OS X ➔ macOS
Page 497 line 17: handing ➔ handling
Page 498 line 19: Windows10 ➔ Windows 10
Page 506 line 6: issues a blocking system ➔ issues a blocking (synchronous) system
Page 506 line 9: run queue ➔ ready queue
Page 506 line 10: run queue ➔ ready queue
Page 507 line -3: readv ➔ readv()
Page 510 line -1: physical memory ➔ main memory
Page 512 line -13: monitor mode➔ kernel mode

Errata - page 4

Page 514 line -6: CPUsand ➔ CPUs and
Page 517 line 6: and that table entry tells ➔ and that table entry (in a nutshell) tells
Page 518 line 14: run queue ➔ ready queue

Chapter 13:

Page 532 line 3: create, write, read ➔ create, open, write, read
Page 533 line 18: create() and delete() are system calls that work with closed ➔ create() is a system call that creates files and delete() is a system call that works with closed
Page 538 line 15: ASCII characters➔ text
Page 544 line -13: version number➔ version number of the file
Page 544 line -12: directory name➔ path name
Page 554 line 13: on Windows 7 NTFS file system. In this example, user "guest" is specifically denied access to the file ListPanel.java ➔ on Windows 10.

Chapter 14:

Page 579 line 16: which is the location➔ whose position counted from the beginning of the bitmap is the location
Page 581 line -14: do not allow overwrite➔ do not allow immediate overwrite
Page 595 line 7: http://src.opensolaris.org/source/xref/onnv/onnv-gate/usr/src/uts/common/ fs/zfs/space_map.c ➔ http://open-zfs.org/wiki/Documentation
Page 595 line 8: http://www.opensolaris.org/os/community/ZFS/docs ➔ https://github.com/openzfs/
Page 595 line 10: Ext3 ➔ Ext4

Chapter 16:

Page 624 Figure 16.1: logic bugs ➔ logic bombs
Page 628 Figure 16.2: return 0 ; } ➔ return 0 ;
Page 638 line -24: sender can encode ➔ sender can encrypt
Page 638 line -23: can decode ➔ can decrypt
Page 641 line 7: .edu /hellman ➔ .edu/helman
Page 641 line 8: must a key ➔ must a public key
Page 645 Figure 16.9: should have vertical lines connecting message m to encryption algorithm E, etc
Page 649 line -17: four-character ➔ four-decimal
Page 649 line -14: four-character ➔ four-decimal
Page 665 line -17: security-center/research ➔ security-center/research. See also https://www.us-cert.gov.

Chapter 17:

Page 685 line 12: processes). When ➔ processes), when
Page 685 line 3: FreeBSD made DAC ➔ FreeBSD made MAC
Page 685 line 6: security features of MAC ➔ security features of macOS
Page 689 line 14: fork system call ➔ fork() system call

Chapter 18:

Page 721 line -10: the operating system➔ the guest operating system
Page 724 line -2: discernable ➔ discernible
Page 727 Figure 18.9: free BSD ➔ FreeBSD

Chapter 19:

Page 742 line -15: data-network layer ➔ data-link layer
Page 744 Figure 19.7: presentation layer ➔ presentation layer header

Errata - page 5

Page 744 Figure 19.7: application layer ➔ application layer header
Page 747 Figure 19.10: inital ➔ initial
Page 749 Figure 19.11: inital ➔ initial
Page 770 line 3: K. Shvachko ➔ S. Shvachko
Page 770 line -19: http://standards.ieee.org/about/get/802/802.11.html➔
https://standards.ieee.org/standard/802_11-2016.html
Page 771 line 9: K. Shvachko ➔ S. Shvachko
Page 771 line -6: Approacm ➔ Approach

Chapter 20:

Page 791 line -26: process runs for ➔ thread runs for
Page 791 line -26: process runs for ➔ thread runs for
Page 791 line -25: process runs for ➔ thread runs for
Page 791 line -12: switch costs are maximized ➔ switching costs are optimized
Page 813 line -16: memory-mapped memory ➔ memory-mapped file

Chapter 21:

Page 834 line 4: Figure Figure ➔ Figure
Page 845 line 19: compute the exact number ➔ compute the number
Page 850 line -14: HarddiskVolumeN ➔ HarddiskVolume2
Page 861 line -15: WINXXIII ➔ Win32
Page 865 Figure 21.6: VM manager ➔ MM manager
Page 871 line 10: UWPModern/Metro ➔ UWP Modern/Metro
Page 872 line -5: storage manager ➔ compression store manager
Page 881 line -6: Uniform Naming Convention ➔ Universal Naming Convention
Page 887 line 14: CreateProcess ➔ CreateProcess()
Page 892 line 18: ChangeWindowMessageFilterEx ➔ ChangeWindowMessageFilterEx()
Page 893 Figure 21.12: 8MB ➔ 8 MB
Page 895 Figure 21.13: T1s ➔ Tls (4 times)
Page 895 line -13: Move the following fragment to the end of section 21.7.5.4:
To use a thread-local static variable, the application declares the variable as follows to ensure that every thread has its own private copy:

```
_declspec(thread) DWORD cur pos = 0;
```

Credits:

Page 963: line 4: Sebree ➔ Sebre

Appendix A:

Page A.12: Mac OS ➔ MacOS
Page A.18: MacOSand ➔ MacOS and
Page A.18: IOSvariants ➔ IOS variants
Page A.21: [Frah (2001)]➔ [IFrah (2001)]
Page A.22: Frah➔ IFrah

Appendix B:

Page B.6: SNM➔ SNMP
Page B.43: Uniform Naming Convention ➔ Universal Naming Convention
Page B.54: `T1s ➔ tls`

Appendix C:

Errata - page 6

Page C.1: line 6, UnixBSD ➔ FreeBSD
All pages: <X.Y> BSD ➔ <X.Y>BSD
Page C.3: line -4, see Chapter 11 ➔ See Chapter 14
Page C.4: line -10, and is replacing ➔ and replaced
Page C.13: line -9, bs character➔ fs character
Page C.16: line xxx, % %➔ %
Page C.18: line -9, is rapidly becoming➔ rapidly became
Page C.31: line 16, synchronized➔ consistent
Page C.36: line 17, system phase➔ system mode

Appendix D:

Page D.5: include a copy of the message➔ include a pointer to a copy of the message
Page D.18: multicomputers➔ multiple computers
Page D.22: multicomputers➔ multiprocessor computers