PREV
E. Further Reading

Bibliography

[biblio01_01] [1] James H. Anderson. Lamport on mutual exclusion: 27 years of planting seeds. In *Twentieth Annual ACM Symposium on Principles of Distributed Computing*, pages 3–12, 2001.

[biblio01_02] [2] Gregory R. Andrews. *Foundations of Multithreaded, Parallel, and Distributed Programming*. Addison-Wesley, Reading, MA, <year>2000</year>.

[biblio01_03] [3] Krzysztof R. Apt and Ernst-Rüdiger Olderog. *Verification of Sequential and Concurrent Programs*. Springer-Verlag, Berlin, <year>1991</year>.

[biblio01_04] [4] Hagit Attiya and Jennifer Welch. *Distributed Computing*. McGraw-Hill, London, <year>1998</year>.

[biblio01_05] [5] John Barnes. *High Integrity Software: The SPARK Approach to Safety and Security*. Addison-Wesley, Harlow, <year>2003</year>.

[biblio01_06] [6] Hans W. Barz. Implemementing semaphores by binary semaphores. *ACM SIGPLAN Notices*, 18(2):39–45, 1983.

[biblio01_07] [7] Mordechai Ben-Ari. *Ada for Software Engineers*. John Wiley, Chichester, <year>1998</year>. Out of print. The book can be freely downloaded for use in education and research from my website http://stwww.weizmann.ac.il/g-cs/benari/books/ (http://stwww.weizmann.ac.il/g-cs/benari/books/).

[biblio01_08] [8] Mordechai Ben-Ari. Interactive execution of distributed algorithms. *ACM Journal on Educational Resources in Computing*, 1(2), 2001.

[biblio01_09] [9] Mordechai Ben-Ari. Mathematical Logic for [biblio01_10] [10] Mordechi Ben-Ari and Alan Burns. Extreme interleavings. *IEEE Concurrency*, 6(3):90, 1998.

[biblio01_11] [11] Piotr Berman and Juan A. Garay. Cloture votes: $n/4$-resilient distributed consensus in $t + 1$ rounds. *Mathematical Systems Theory*, 26(1):3–19, 1993.

[biblio01_12] [12] Nikolaj S. Bj∅rner, Anca Browne, Michael A. Colón, Bernd Finkbeiner, Zohar Manna, Henny B. Sipma, and Tomás E. Uriber. Verifying temporal properties of reactive systems: A STeP tutorial. *Formal Methods in System Design*, 16(3):227–270, 2000.

[biblio01_13] [13] Per Brinch Hansen. *The Search for Simplicity: Essays in Parallel Programming*. IEEE Computer Society Press, Los Alamitos, CA, <year>1996</year>.

[biblio01_14] [14] Per BrinchHansen, editor. *Classic Operating Systems: From Batch Processing to Distributed Systems*. Springer-Verlag, New York, <year>2001</year>.

[biblio01_15] [15] Peter A. Buhr, Michel Fortier, and Michael H. Coffin. Monitor classification. *ACM Computing Surveys*, 27(1):63–107, 1995.

[biblio01_16] [16] Peter A. Buhr and Ashif S. Harji. Concurrent urban legends. *Concurrency and Computation: Practice and Experience*, 17:1133–1172, 2005.

[biblio01_17] [17] Alan Burns, Brian Dobbing, and Tullio Vardanega. Guide for the use of the Ada Ravenscar Profile in high integrity systems. Technical Report YCS-2003-348, University of York, 2003. http://www.cs.york.ac.uk/ftpdir/reports/YCS-2003-348.pdf (http://www.cs.york.ac.uk/ftpdir/reports/YCS-2003-348.pdf).

[biblio01_18] [18] Alan Burns and Andy Wellings. *Concurrency in Ada (Second Edition)*. Cambridge University Press, Cambridge, <year>1998</year>.

[biblio01_19] [19] Alan Burns and Andy Wellings. *Real Time Systems and Programming Languages: Ada 95, Real-Time Java and Real-Time C/POSIX (3rd Edition)*. Addison-Wesley, Reading, MA, <year>2001</year>.

[biblio01_20] [20] Bill Bynum and Tracy Camp. After you, Alfonse: A mutual exclusion toolkit. *ACM SIGCSE Bulletin*, 28(1):170–174, 1996.

[biblio01_21] [21] O. S.F. Carvalho and G. Roucairol. On mutual exclusion in computer networks. *Communications of the ACM*, 26(2):146–148, 1983.

[biblio01_22] [22] K. Mani Chandy and Leslie Lamport. Distributed snapshots: Determining global states of distributed systems. *ACM Transactions on Computer Systems*, 3(1):63–75, 1985.

[biblio01_23] [23] K. Mani Chandy and Jayadev Misra. *Parallel Program Design: A Foundation*. Addison-Wesley, Reading, MA, <year>1988</year>.

[biblio01_24] [24] Edmund M. Clarke,, Jr. Orna Grumberg, and Doron A. Peled. *Model Checking*. MIT Press, Cambridge, MA, <year>1999</year>.

[biblio01_25] [25] Edsger W. Dijkstra and Carel S. Scholten. Termination detection for diffusing computations. *Information Processing Letters*, 11(1):1–4, 1980.

[biblio01_26] [26] R. W. Doran and L. K. Thomas. Variants of the software solution to mutual exclusion. *Information Processing Letters*, 10(4/5):206–208, 1980.

[biblio01_27] [27] Al Geist, Adam Beguelin, Jack Dongarra, Weicheng Jiang, Robert Manchek, and Vaidy Sunderam. *PVM: Parallel Virtual Machine: A User's Guide and Tutorial for Networked Parallel Computing*. MIT Press, Cambridge, MA, <year>1994</year>. http://www.netlib.org/pvm3/index.html (http://www.netlib.org/pvm3/index.html).

[biblio01_28] [28] David Gelernter and Nicholas Carriero. *How to Write Parallel Programs—A First Course*. MIT Press, Cambridge, MA, <year>1990</year>.

[biblio01_29] [29] William Gropp, Ewing Lusk, and Anthony Skjellum. *Using MPI: Portable Parallel Programming with the Message-Passing Interface (Second Edition)*. MIT Press, Cambridge, MA, <year>1999</year>.

[biblio01_30] [30] Dan Gusfield and Robert W. Irving. *The Stable Marriage Problem: Structure and Algorithms*. MIT Press, Cambridge, MA, <year>1989</year>.

[biblio01_31] [31] Stephen J. Hartley. *Concurrent Programming: The Java Programming Language*. Oxford University Press, Oxford, <year>1998</year>.

[biblio01_32] [32] C. A. R. Hoare. *Communicating Sequential Processes*. Prentice Hall International, Hemel Hempstead, UK, 1985–2004. http://www.usingcsp.com/cspbook.pdf (http://www.usingcsp.com/cspbook.pdf).

[biblio01_33] [33] Gerard J. Holzmann. *The Spin Model Checker: Primer and Reference Manual*. Addison-Wesley, Reading, MA, <year>2004</year>.

[biblio01_34] [34]Shing-Tsaan Huang. Detecting termination of distributed computations by external agents. In *IEEE 9th International Conference on Distributed Computer Systems*, pages 79–84, 1989.

[biblio01_35] [35] Mike Jones. What really happened on Mars Rover Pathfinder. *The Risks Digest*, 19(49), 1997. http://catless.ncl.ac.uk/Risks/19.49.html (http://catless.ncl.ac.uk/Risks/19.49.html).

[biblio01_36] [36] Leslie Lamport. A new solution of Dijkstra's concurrent programming problem. *Communications of the ACM*, 17(8):453–455, 1974.

[biblio01_37] [37] Leslie Lamport. Time, clocks, and the ordering of events in a distributed system. *Communications of the ACM*, 21(7):558–565, 1978.

[biblio01_38] [38] Leslie Lamport. The mutual exclusion problem: Part I—a theory of interprocess communication. *Journal of the ACM*, 33(2):313–326, 1986.

[biblio01_39] [39] Leslie Lamport. The mutual exclusion problem: Part II—statement and solutions. *Journal of the ACM*, 33(2):327–348, 1986.

[biblio01_40] [40] Leslie Lamport. A fast mutual exclusion algorithm. *ACM Transactions on Computer Systems*, 5(1):1–11, 1987.

[biblio01_41] [41] Leslie Lamport. *Specifying Systems: The TLA+ Language and Tools for Hardware and Software Engineers*. Addison-Wesley, Reading, MA, <year>2002</year>. http://research.microsoft.com/users/lamport/tla/tla.html (http://research.microsoft.com/users/lamport/tla/tla.html).

[biblio01_42] [42] Leslie Lamport, Robert Shostak, and Marshall Pease. The Byzantine Generals problem. *ACM Transactions on Programming Languages and Systems*, 4(3):382–401, 1982.

[biblio01_43] [43] Kim G. Larsen, Paul Pettersson, and Wang Yi. UPPAAL in a nutshell. *International Journal on Software Tools for Technology Transfer*, 1(1–2):134–152, 1998. http://www.uppaal.com/documentation.shtml (http://www.uppaal.com/documentation.shtml).

[biblio01_44] [44] Doug Lea. *Concurrent Programming in Java*. Addison-Wesley, Reading, MA, <year>1997</year>.

[biblio01_45] [45] J. L. Lions. Ariane 5 flight 501 failure: Report by the inquiry board. http://ravel.esrin.esa.it/docs/esa-x-1819eng.pdf (http://ravel.esrin.esa.it/docs/esa-x-1819eng.pdf), 1996.

[biblio01_46] [46] Richard J. Lipton. Reduction: A method of proving properties of parallel programs. *Communications of the ACM*, 18(12):717–721, 1975.

[biblio01_47] [47] Jane W. S. Liu. *Real-Time Systems*. Prentice Hall, Upper Saddle River, NJ, <year>2000</year>.

[biblio01_48] [48] Nancy A. Lynch. *Distributed Algorithms*. Morgan Kaufman, San Francisco, CA, <year>1996</year>.

[biblio01_49] [49] Jeff Magee and Jeff Kramer. *Concurrency: State Models & Java Programs*. John Wiley, Chichester, <year>1999</year>.

[biblio01_50] [50] Zohar Manna and Amir Pnueli. *The Temporal Logic of Reactive and Concurrent Systems. Volume I: Specification*. Springer-Verlag, New York, <year>1992</year>.

[biblio01_51] [51] Zohar Manna and Amir Pnueli. *The Temporal Verification of Reactive Systems. Volume II: Safety*. Springer-Verlag, New York, <year>1995</year>.

[biblio01_52] [52] Jeff Matocha and Tracy Camp. A taxonomy of distributed termination algorithms. *The Journal of Systems and Software*, 43:207–221, 1998.

[biblio01_53] [53] Friedemann Mattern. Global quiescence detection based on credit distribution and recovery. *Information Processing Letters*, 30(4):195–200, 1989.

[biblio01_54] [54] Mitchell L. Neilsen and Masaaki Mizuno. A dag-based algorithm for distributed mutual exclusion. In *IEEE 11th International Conference on Distributed Computing Systems*, pages 354–360, 1991.

[biblio01_55] [55] A. W. Roscoe. *Theory and Practice of Concurrency*. Prentice Hall, Hemel Hempstead, UK, <year>1998</year>.

[biblio01_56] [56] Theo C. Ruys. *Towards Effective Model Checking*. PhD thesis, University of Twente, <year>2001</year>.

http://wwwhome.cs.utwente.nl/~ruys/ (http://wwwhome.cs.utwente.nl/~ruys/).

[biblio01_57] [57] Fred B. Schneider. *On Concurrent Programming*. Springer-Verlag, New York, <year>1997</year>.

[biblio01_58] [58] Steve Schneider. *Concurrent and Real-time Systems: The CSP Approach*. John Wiley, Chichester, <year>2000</year>.

[biblio01_59] [59] Liu Sha, Ragunathan Rajkumar, and John P. Lehoczky. Priority inheritance protocols: An approach to real-time synchronization. *IEEE Transactions on Computers*, 39:1175–1185, 1990.

[biblio01_60] [60] Abraham Silberschatz, Greg Gagne, and Peter Baer Galvin. *Operating Systems Concepts*. John Wiley, Hoboken, NJ, <year>2002</year>.

[biblio01_61] [61] H. R. Simpson. Four-slot fully asynchronous communication mechanism. *IEE Proceedings*, 137 Part E (1):17–30, 1990.

[biblio01_62] [62] Alfred Spector and David Gifford. The Space Shuttle primary computer system. *Communications of the ACM*, 27(9):874–900, 1984.

[biblio01_63] [63] William Stallings. *Operating Systems: Internals and Design Principles*. Prentice Hall, Upper Saddle River, NJ, <year>2000</year>.

[biblio01_64] [64] Ichiro Suzuki and Tadao Kasami. A distributed mutual exclusion algorithm. *ACM Transactions on Computer Systems*, 3(4):344–349, 1985.

[biblio01_65] [65] Ahuva Tikvati, Mordechai Ben-Ari, and Yifat Ben-David Kolikant. Virtual trees for the Byzantine Generals algorithm. *ACM SIGCSE Bulletin*, 36(1):392–396, 2004.

[biblio01_66] [66] James E. Tomayko. Computers in spaceflight: The NASA experience. http://www.hq.nasa.gov/office/pao/History/computers/Compspace.html (http://www.hq.nasa.gov/office/pao/History/computers/Compspace.html), 1988.

[biblio01_67] [67] John A. Trono and William E. Taylor. Further comments on “a correct and unrestrictive implementation of general semaphores”. *ACM SIGOPS Operating Systems Review*, 34(3):5–10, 2000.

[biblio01_68] [68] Jan Tijmen Udding. Absence of individual starvation using weak semaphores. *Information Processing Letters*, 23:159–162, 1986.

[biblio01_69] [69] Willem Visser, Klaus Havelund, Guillaume Brat, Seungjoon Park, and Flavio Lerda. Model checking programs. *Automated Software Engineering*, 10(2):203–232, 2003.

PREV
E. Further Reading

Sign Out