[74] S. Macguire. *Writing Solid Code*. Microsoft Press, 1993.

[75] S. A. Mahlke, W. Y. Chen, J. C. Gyllenhal, and W. W. Hwu. Compiler code transformations for superscalar-based high-performance systems. In *Proceedings of the 1992 ACM/IEEE Conference on Supercomputing*, pages 808–817. ACM, 1992.

[76] E. Marshall. Fatal error: How Patriot overlooked a Scud. *Science*, page 1347, March 13, 1992.

[77] M. Matz, J. Hubička, A. Jaeger, and M. Mitchell. System V application binary interface AMD64 architecture processor supplement. Technical Report, x86-64.org, 2013. Available at http://www.x86-64.org/documentation_folder/abi-0.99.pdf.

[78] J. Morris, M. Satyanarayanan, M. Conner, J. Howard, D. Rosenthal, and F. Smith. Andrew: A distributed personal computing environment. *Communications of the ACM*, pages 184–201, March 1986.

[79] T. Mowry, M. Lam, and A. Gupta. Design and evaluation of a compiler algorithm for prefetching. In *Proceedings of the 5th International Conference on Architectural Support for Programming Languages and Operating Systems (ASPLOS)*, pages 62–73. ACM, October 1992.

[80] S. S. Muchnick. *Advanced Compiler Design and Implementation*. Morgan Kaufmann, 1997.

[81] S. Nath and P. Gibbons. Online maintenance of very large random samples on flash storage. In *Proceedings of VLDB*, pages 970–983. VLDB Endowment, August 2008.

[82] M. Overton. *Numerical Computing with IEEE Floating Point Arithmetic*. SIAM, 2001.

[83] D. Patterson, G. Gibson, and R. Katz. A case for redundant arrays of inexpensive disks (RAID). In *Proceedings of the 1998 ACM SIGMOD International Conference on Management of Data*, pages 109–116. ACM, June 1988.

[84] L. Peterson and B. Davie. *Computer Networks: A Systems Approach, Fifth Edition*. Morgan Kaufmann, 2011.

[85] J. Pincus and B. Baker. Beyond stack smashing: Recent advances in exploiting buffer overruns. *IEEE Security and Privacy* 2(4):20–27, 2004.

[86] S. Przybylski. *Cache and Memory Hierarchy Design: A Performance-Directed Approach*. Morgan Kaufmann, 1990.

[87] W. Pugh. The Omega test: A fast and practical integer programming algorithm for dependence analysis. *Communications of the ACM* 35(8):102–114, August 1992.

[88] W. Pugh. Fixing the Java memory model. In *Proceedings of the ACM Conference on Java Grande*, pages 89–98. ACM, June 1999.

[89] J. Rabaey, A. Chandrakasan, and B. Nikolic. *Digital Integrated Circuits: A Design Perspective, Second Edition*. Prentice Hall, 2003.

[90] J. Reinders. *Intel Threading Building Blocks*. O'Reilly, 2007.

[91] D. Ritchie. The evolution of the Unix time-sharing system. *AT&T Bell Laboratories Technical Journal* 63(6 Part 2):1577–1593, October 1984.

[92] D. Ritchie. The development of the C language. In *Proceedings of the 2nd ACM SIGPLAN Conference on History of Programming Languages*, pages 201–208. ACM, April 1993.

[93] D. Ritchie and K. Thompson. The Unix time-sharing system. *Communications of the ACM* 17(7):365–367, July 1974.

[94] M. Satyanarayanan, J. Kistler, P. Kumar, M. Okasaki, E. Siegel, and D. Steere. Coda: A highly available file system for a distributed workstation environment. *IEEE Transactions on Computers* 39(4):447–459, April 1990.

[95] J. Schindler and G. Ganger. Automated disk drive characterization. Technical Report CMU-CS-99-176, School of Computer Science, Carnegie Mellon University, 1999.

[96] F. B. Schneider and K. P. Birman. The monoculture risk put into context. *IEEE Security and Privacy* 7(1):14–17, January 2009.

[97] R. C. Seacord. *Secure Coding in C and C++, Second Edition*. Addison-Wesley, 2013.

[98] R. Sedgewick and K. Wayne. *Algorithms, Fourth Edition*. Addison-Wesley, 2011.

[99] H. Shacham, M. Page, B. Pfaff, E.-J. Goh, N. Modadugu, and D. Boneh. On the effectiveness of address-space randomization. In *Proceedings of the 11th ACM Conference on Computer and Communications Security (CCS)*, pages 298–307. ACM, 2004.

[100] J. P. Shen and M. Lipasti. *Modern Processor Design: Fundamentals of Superscalar Processors*. McGraw Hill, 2005.