

Threads



The process model introduced in Chapter 3 assumed that a process was an executing program with a single thread of control. Virtually all modern operating systems, however, provide features enabling a process to contain multiple threads of control. In this chapter, we introduce many concepts associated with multithreaded computer systems, including a discussion of the APIs for the Pthreads, Windows, and Java thread libraries. We look at a number of issues related to multithreaded programming and its effect on the design of operating systems. Finally, we explore how the Windows and Linux operating systems support threads at the kernel level.

Bibliographical Notes

Threads have had a long evolution, starting as “cheap concurrency” in programming languages and moving to “lightweight processes,” with early examples that included the Thoth system ([Cheriton et al. (1979)]) and the Pilot system ([Redell et al. (1980)]). [Binding (1985)] described moving threads into the UNIX kernel. Mach ([Accetta et al. (1986)], [Tevanian et al. (1987)]), and V ([Cheriton (1988)]) made extensive use of threads, and eventually almost all major operating systems implemented them in some form or another.

[Vahalia (1996)] covers threading in several versions of UNIX. [McDougall and Mauro (2007)] describes developments in threading the Solaris kernel. [Russinovich and Solomon (2009)] discuss threading in the Windows operating system family. [Mauerer (2008)] and [Love (2010)] explain how Linux handles threading, and [Singh (2007)] covers threads in Mac OS X.

Information on Pthreads programming is given in [Lewis and Berg (1998)] and [Butenhof (1997)]. [Oaks and Wong (1999)] and [Lewis and Berg (2000)] discuss multithreading in Java. [Goetz et al. (2006)] present a detailed discussion of concurrent programming in Java. [Hart (2005)] describes multithreading using Windows. Details on using OpenMP can be found at <http://openmp.org>.

An analysis of an optimal thread-pool size can be found in [Ling et al. (2000)]. Scheduler activations were first presented in [Anderson et al. (1991)], and [Williams (2002)] discusses scheduler activations in the NetBSD system.

[Breshears (2009)] and [Pacheco (2011)] cover parallel programming in detail. [Hill and Marty (2008)] examines Amdahl’s Law with respect to multi-core systems. The Monte Carlo technique for estimating π is further discussed in <http://math.fullerton.edu/mathews/n2003/montecarlopimod.html>.

Bibliography

- [Accetta et al. (1986)] M. Accetta, R. Baron, W. Bolosky, D. B. Golub, R. Rashid, A. Tevanian, and M. Young, “Mach: A New Kernel Foundation for UNIX Development”, *Proceedings of the Summer USENIX Conference* (1986), pages 93–112.
- [Anderson et al. (1991)] T. E. Anderson, B. N. Bershad, E. D. Lazowska, and H. M. Levy, “Scheduler Activations: Effective Kernel Support for the User-Level Management of Parallelism”, *Proceedings of the ACM Symposium on Operating Systems Principles* (1991), pages 95–109.
- [Binding (1985)] C. Binding, “Cheap Concurrency in C”, *SIGPLAN Notices*, Volume 20, Number 9 (1985), pages 21–27.
- [Breshears (2009)] C. Breshears, *The Art of Concurrency*, O’Reilly & Associates (2009).
- [Butenhof (1997)] D. Butenhof, *Programming with POSIX Threads*, Addison-Wesley (1997).
- [Cheriton (1988)] D. Cheriton, “The V Distributed System”, *Communications of the ACM*, Volume 31, Number 3 (1988), pages 314–333.
- [Cheriton et al. (1979)] D. R. Cheriton, M. A. Malcolm, L. S. Melen, and G. R. Sager, “Thoth, a Portable Real-Time Operating System”, *Communications of the ACM*, Volume 22, Number 2 (1979), pages 105–115.
- [Goetz et al. (2006)] B. Goetz, T. Peirls, J. Bloch, J. Bowbeer, D. Holmes, and D. Lea, *Java Concurrency in Practice*, Addison-Wesley (2006).
- [Hart (2005)] J. M. Hart, *Windows System Programming*, Third Edition, Addison-Wesley (2005).
- [Hill and Marty (2008)] M. Hill and M. Marty, “Amdahl’s Law in the Multicore Era”, *IEEE Computer*, Volume 41, Number 7 (2008), pages 33–38.
- [Lewis and Berg (1998)] B. Lewis and D. Berg, *Multithreaded Programming with Pthreads*, Sun Microsystems Press (1998).
- [Lewis and Berg (2000)] B. Lewis and D. Berg, *Multithreaded Programming with Java Technology*, Sun Microsystems Press (2000).
- [Ling et al. (2000)] Y. Ling, T. Mullen, and X. Lin, “Analysis of Optimal Thread Pool Size”, *Operating System Review*, Volume 34, Number 2 (2000), pages 42–55.
- [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).
- [**Oaks and Wong (1999)**] S. Oaks and H. Wong, *Java Threads*, Second Edition, O'Reilly & Associates (1999).
- [**Pacheco (2011)**] P. S. Pacheco, *An Introduction to Parallel Programming*, Morgan Kaufmann (2011).
- [**Redell et al. (1980)**] D. D. Redell, Y. K. Dalal, T. R. Horsley, H. C. Lauer, W. C. Lynch, P. R. McJones, H. G. Murray, and S. P. Purcell, "Pilot: An Operating System for a Personal Computer", *Communications of the ACM*, Volume 23, Number 2 (1980), pages 81–92.
- [**Russinovich and Solomon (2009)**] M. E. Russinovich and D. A. Solomon, *Windows Internals: Including Windows Server 2008 and Windows Vista*, Fifth Edition, Microsoft Press (2009).
- [**Singh (2007)**] A. Singh, *Mac OS X Internals: A Systems Approach*, Addison-Wesley (2007).
- [**Tevanian et al. (1987)**] A. Tevanian, Jr., R. F. Rashid, D. B. Golub, D. L. Black, E. Cooper, and M. W. Young, "Mach Threads and the Unix Kernel: The Battle for Control", *Proceedings of the Summer USENIX Conference* (1987).
- [**Vahalia (1996)**] U. Vahalia, *Unix Internals: The New Frontiers*, Prentice Hall (1996).
- [**Williams (2002)**] N. Williams, "An Implementation of Scheduler Activations on the NetBSD Operating System", *2002 USENIX Annual Technical Conference, FREENIX Track* (2002).

