

# References

1. L. Abeni, G. Buttazo, Support for dynamic QoS in the HARTIK kernel. Proceeding of the IEEE Conference on Real-Time Computing Systems and Applications, Cheju Island, South Korea (2000)
2. L. Abeni, A. Goel, C. Krasic, J. Snow, J. Walpole, *A Measurement-Based Analysis of the Real-Time Performance of Linux*, Proceedings of the Eighth IEEE Real-Time and Embedded Technology and Applications Symposium (RTAS'02) (2002)
3. M. Accetta, R. Baron, W. Bolosky, D. Golub, R. Rashid, A. Tevanian, M. Young, "Mach: A new kernel foundation for UNIX development", 93–113, USENIX Association Conference Proceedings, USENIX Association, June 1986
4. G. Anzinger y N. Gamble (MontaVista Software), Design of a Fully Preemptable Linux Kernel. *Embedded Linux Journal* (2000)
5. F. Armand, I. Boule, M. Gien, M. Guillemont, F. Herrmann, C. Kaiser, S. Langlois, P. Leonard, and W. Neuhauser, Chorus distributed operating systems, M. Rozier, V. Abrossimov, 305–370, *USENIX Computing Systems*, 1, 4, Fall 1988
6. S.K. Baruah, L.E. Rosier, R.R. Howell, Algorithms and complexity concerning the preemptive scheduling of periodic, real-time tasks on one processor. *Journal of Real-Time Systems* **2** (1990)
7. M. Beck, H. Böhme, M. Dziadzka, U. Kunitz, R. Magnus, D. Verworner, *Linux Kernel Internals* (Addison-Wesley, Boston, 1998). ISBN 0-2011-33143-8
8. P. Brinch-Hansen, The nucleus of a multiprogramming system. *Communication of the ACM* **13**(4), 238–241 and 250 (1970)
9. P. Brinch-Hansen, *Operating System Principles* (Prentice Hall, New Delhi, 1973)
10. P. Brinch-Hansen, *Classic Operating Systems: From Batch Processing to Distributed Systems* (Springer, New York, 2000)
11. D. Brylow, N. Damgaard, J. Palsberg, Static checking of interrupt-driven software, in *Proceedings of ICSE'01, International Conference on Software Engineering*, (2001), pp. 47–56
12. A. Burns, A.J. Wellings, Implementing analyzable hard real-time sporadic tasks in Ada 9X. *ACM Ada Letters* **X/V**(1), 54 (1994)
13. G. C. Buttazzo. "Hartik: A real-time kernel for robotics applications". Proceedings of the IEEE Real-Time Systems Symposium, Diciembre 1993
14. G. C. Buttazzo y M. Di Natale. Hartik: A hard real-time kernel for programming robot tasks with explicit time constraints and guaranteed execution. Proceedings of the IEEE International Conference on Robotics and Automation, Mayo 1993

15. D. Comer, T.V. Fossum, *Operating System Design, Vol.1: The Xinu Approach (PC Edition)* (Prentice-Hall, 1988)
16. M. Coutinho, J. Rufino, C. Almeida, *Control of Event Handling Timeliness in RTEMS*, Proceedings of the 17th IASTED International Conference on Parallel and Distributed Computing and Systems (PDCS 2005), Phoenix, AZ, USA, November 14–16, 2005
17. H. Custer, *Inside Windows NT* (Microsoft Press, Redmond, 1992)
18. A. Damm, J. Reisinger, W. Schwabl, y H. Kopetz. “The real-time operating system of MARS”. *Operating System Review*, 23(3): 141–157, 1989
19. K. Dankwardt, Real Time and Linux,( Part 1, 2 and 3). January/February/March 2002 issue of *Embedded Linux Journal*. 2002
20. S.-T. Dietrich, D. Walker, The evolution of real-time linux. 7th Real-Time Linux Workshop: The Evolution of Real-Time Linux **17** (2005)
21. E.W. Dijkstra, THE structure of the THE multiprogramming system. *Commun. ACM* **11**(5), 341–346 (May 1968)
22. W. Dong, C. Chen, X. Liu, Y. Liu, J. Bu, K. Zheng, SenSpire OS: A predictable, flexible, and efficient operating system for wireless sensor networks. *IEEE Trans. Comput.* **60**(12), 1788–1801 (Dec. 2011)
23. J.R. Eykholt, S.R. Kleiman, S. Barton, R. Faulkner, A. Shivalingiah, M. Smith, D. Stein, J. Voll, M. Weeks, D. Williams, Beyond multiprocessing: Multithreading the SunOs kernel, in *Proceedings of the Summer 1992 USENIX Technical Conference*, (1992)
24. Y. Furukawa, T. Yamauchi, H. Taniguchi, Implementation and evaluation for sophisticated periodic execution control in embedded systems. *International Journal of Control and Automation* **4**(2), 59–78 (2011)
25. M. Hall, *Windows CE 5.0 for Real-Time Systems* (Embedded Computing Design, 2005)
26. D. Harel, A. Pnueli, On the development of reactive systems, in *Logics and Models of Concurrent Systems*, (1985), pp. 477–498
27. A.C. Heursch, A. Horstkotte, H. Rzehak, *Preemption concepts, Rheelstone Benchmark and scheduler analysis of Linux 2.4*, on the Real-Time & Embedded Computing Conference, Milan, November 27–28, 2001
28. A.C. Heursch, D. Grambow, D. Roedel, H. Rzehak, *Time-critical tasks in Linux 2.6, concepts to increase the preemptibility of Linux kernel*. Linux Automation Konferenz, University of Hannover, Germany, March 2004
29. D. Hildebrand, An architectural overview of QNX, in *Proc. of the USENIX Workshop on Micro-Kernels and Other Kernel Architectures*, (Seattle, 1992), pp. 113–126
30. W. Hofer, D. Danner, R. Müller, F. Scheler, W. Schröder-Preikschat, D. Lohmann, SLOTH ON TIME: Efficient hardware-based scheduling for time-triggered RTOS, in *2012 IEEE 33rd Real-Time Systems Symposium*, (2012), pp. 237–247
31. K. Jeffay, D.L. Stone, *Accounting for Interrupt Handling Cost in Dynamic Priority Task Systems*. Proceedings of the IEEE Real-Time Systems Symposium, Raleigh-Durham, NC, December 1993. pp. 212–221
32. Do-While Jones, *Interrupt-Free Design*. Circuit Cellar Magazine (1994), p. 36
33. L.J. Kenah y S.F. Bate, “VAX/VMS Internals and Data Structures”, Digital Press, Bedford, 1984
34. S. Kleiman, J. Eykholt, Interrupts as threads. *ACM SIGOPS Operating Systems Review* **29**(2), 21–26 (1995)
35. J.J. Labrose, *MicroC/OS-II: The Real-Time Kernel*, 2nd edn. (CMP Books, San Francisco, 2002)
36. E.L. Lamie, *Real-Time Embedded Multithreading: Using ThreadX and ARM* (CMP Books, San Francisco, 2005)
37. I. Lee, R. King, and R. Paul, *RK: A Real-Time Kernel for a Distributed System with Predictable Response*. Tech. Report MS-CIS-88-78/GRASP LAB 155 78, Dept. of Computer and Information Science, Univ. of Pennsylvania, Octubre 1988

38. J.P. Lehoczky, L. Sha, D. Ye, The rate-monotonic scheduling algorithm: Exact characterization and average case behaviour, in *Proceeding IEEE Real-Time System Symposium*, (1989), pp. 166–171
39. I. Leslie, D. McAuley, R. Black, T. Roscoe, P. Barham, D. Evers, R. Fairbairns, E. Hyden, The design and implementation of an operating system to support distributed multimedia applications. *IEEE Journal on Selected Areas in Communications* **14**(7), 1280–1297 (1996)
40. Q. Li, C. Yao, *Real Time Concepts for Embedded Systems* (CMP Books, San Francisco, 2003)
41. Ye Li, Richard West, Eric Missimer, The quest-V separation kernel for mixed criticality systems. *Proceedings of the 1st International Workshop on Mixed Criticality Systems (WMC) at the 34th IEEE Real-Time Systems Symposium (RTSS 2013)*, Vancouver, Canada, December 2013
42. L.L. del Foyo, P. Mejia-Alvarez, D. de Niz, Integrated task and interrupt Management for Real-Time Systems. *ACM Trans. Embed. Comput. Syst.* **11**(2), 4879 (2012)
43. J. Lion, *Commentary of Unix Version 6 Source Code: Lions' Commentary on Unix - with Source Code* (Peer-to-Peer Communications, San Jose, 1977)
44. D. Lohman, O. Spinczyk, W. Hofer, W. Schröder-Preikschat, The aspect-aware design and implementation of the CiAO operating-system family. *Transactions on Aspect-Oriented Software Development (TAOSD IX)* **7271**, 168–215 (2012)
45. D. Lohmann, O. Spinczyk, W. Schröder-Preikschat, On the configuration of non-functional properties in operating system product lines, in *Proceedings of the 4th AOSD Workshop on Aspects, Components, and Patterns for Infrastructure Software (AOSD-ACP4IS '05)*, (Northeastern University, Chicago, Boston (NU-CCIS-05-03), 2005), pp. 19–25
46. J. Mauro, R. McDougall, *Solaris Internals, Core Kernel Architecture* (Sun Microsystems Press, 2001)
47. M. Kirk McKusick, K. Bostic, M.J. Karels, J.S. Quarterman, *The Design and Implementation of the 4.4BSD Operating System* (Addison-Wesley, 1996)
48. A. Massa, *Embedded Software Development with eCos* (Prentice Hall, 2002)
49. J.C. Mogul, K.K. Ramakrishnan, Eliminating receive livelock in an interrupt-driven kernel. *ACM Transactions on Computer Systems (TOCS)* **15**(3), 217–252 (Aug. 1997)
50. R. Moore, *Link Service Routines for better interrupt handling*, 22/12/05
51. On-Line Applications Research Corporation (OAR), *RTEMS C User's Guide, edition 4.6.2*, for rtems 4.6.2 edition, August 2003
52. *OSEK/VDX Operating System, Versión 2.0 revision 1, 15.10, 1997.*, <http://www.osek-vdx.org>
53. J. Pagonis, *Overview of Symbian OS Hardware Interrupt Handling*, Revision 1.0, March 2004
54. John S. Quarterman, Avi Silberschatz y J. L. Peterson, “4.2 BSD and 4.3 BSD as examples of the UNIX system”, *ACM Computing Survey*, **17**, 4, 1985, 379–418
55. R. Rajkumar, *Synchronization in Real-Time Systems: A Priority Inheritance Approach* (Kluwer Academic Publishers, 1991). 208 pp, ISBN 0-7923-9211-6
56. J. Regehr, U. Duongsaa, Preventing interrupt overload. *ACM SIGPLAN Not.* **40**(7) (2005)
57. M. Rozier, V. Abrossimov, F. Armand, I. Boule, M. Gien, M. Guillemont, F. Herrmann, C. Kaiser, S. Langlois, P. L'eonard, W. Neuhauser, Chorus distributed operating systems. *Comput. Syst.* **1**(4), 305–367 (1988)
58. M. Saksena, *Linux as a Real-Time Operating System* (TymeSys White Paper, 2003)
59. K. Sandström, C. Eriksson, G. Fohler, *Handling Interrupts with Static Scheduling in an Automotive Vehicle Control System*, in *Proceedings of Fifth International Conference on Real-Time Computing Systems and Applications, Hiroshima, Japan, Oct, 1998*,
60. F. Scheler, W. Hofer, B. Oechslein, R. Pfister, W. Schröder-Preikschat, D. Lohmann, Parallel, hardware-supported interrupt handling in an event-triggered real-time operating system, in *ACM CASES '09 Proceedings of the 2009 International Conference on Compilers, Architecture, and Synthesis for Embedded Systems*, (Grenoble, 2009), pp. 167–174
61. B. Selic, *The Challenges of Real-Time Software Design*, *Embedded Systems Programming*, October 1996. (Article not available online)

62. W. Schröder-Preikschat, *The Logical Design of Parallel Operating Systems* (Prentice-Hall International, 1994). ISBN 0-13-183369-3
63. F. Schön, W. Schröder-Preikschat, O. Spinczyk, U. Spinczyk, On interrupt-transparent synchronization in an embedded object-oriented operating system, in *3rd International Symposium on Object-Oriented Real-Time Distributed Computing (ISORC 2000)*, (2000), pp. 270–277
64. D.C. Schmidt, C.D. Cranor, Half-sync/half-Async: A architectural pattern for efficient and well-structured concurrent I/O, in *Pattern Languages of Program Design*, ed. by J. O. Coplien, J. M. Vlissides, N. L. Kerth (Eds), (Addison-Wesley, Reading, 1996)
65. D.A. Solomon, *Inside Windows NT Second Edition* (Microsoft Press, Redmond, 1998)
66. D. Solomon, M. Russinovich, *Inside Windows*, 3rd edn. (Microsoft Press, Redmond, 2000)
67. B. Sprunt, *Aperiodic Task Scheduling for Real-Time Systems*. Ph.D. Thesis, Carnegie-Mellon University, August 1990
68. J.A. Stankovic y K. Ramamritham. The spring kernel. A new paradigm for real-time systems. *IEEE Softw.* (1991)
69. D.B. Stewart, Twenty-five-most commons mistakes with real-time software development, in *Proceedings of 1999 Embedded Systems Conference*, (San Jose, 2004)
70. M. Timmerman, *Is Windows CE 2.0 a real threat to the RTOS World ?*, *Real-Time Magazine* 3Q98, 1998
71. TimeSys Corporation, “A TimeSys Perspective on the Linux Preemptible Kernel”, 2003
72. K.W. Tindell, RTOS interrupt handling: Commom errors and how to avoid them. *Embedded Systems Programming Europe* (1999)
73. H. Tokuda y M. Kotera. “A real-time tool set for the arts kernel”. *Proceedings of the IEEE Real-Time Systems Symposium* 1988
74. H. Tokuda y C. W. Mercer. “Arts: A distributed real-time kernel”. *Operating System Review*, 23(3), 721989
75. M. Wilcox, *I’ll Do It Later: Softirqs, Tasklets, Bottom Halves, Task Queues, Work Queues and Timers*, Linux Conference, The University of Western Australia, Perth, Australia, 22nd to 25th January 2003
76. L. William, Pardon the interruption, p 58, *Information Quarterly* Vol. 3, Number 4, 2004
77. C. Williams, *Which is better -- the preempt patch, or the low-latency patch? Linux Scheduler Latency*. *Embedded Systems (Europe)* 2002
78. J. Yang, Y. Chen, H. Wang, B. Wang, A linux kernel with fixed interrupt latency for embedded real-time system, in *Proceedings of the Second International Conference on Embedded Software and Systems (ICESSE’05)*, (2005)
79. Y.-C. Wang, K.-J. Lin, Some discussion on the low latency patch for linux, in *Conf. Proc. of the Second Annual Real-Time Linux Workshop, Orlando, Florida, November 27–28*, (Florida, 2000)
80. T. Zwavashe, D. Vasumathi, Polling, interrupts &  $\mu$ COS-II: A comparative timing response simulation model for wireless processor-to-processor communication. *International Journal of Science and Research (IJSR)* 3(7), 116–122 (July 2014)
81. Intel Multiprocessor Specification. Version 1.4, 1997, <http://developer.intel.com/design/pentium/datashts/24201606.pdf>
82. <https://en.wikipedia.org/wiki/Interrupt>
83. R. Love, *Linux Kernel Development*, 3rd edn. (Addison-Wesley Professional, 2010). ISBN-10: 0672329468
84. Android Internals::Power User’s View. 2015, Jonathan Levin, ISBN-10: 0991055527
85. [https://en.wikipedia.org/wiki/Android\\_\(operating\\_system\)](https://en.wikipedia.org/wiki/Android_(operating_system))
86. C. Ji, Internal input event handling in the Linux kernel and the Android userspace. <https://seasonofcode.com/posts/internal-input-event-handling-in-the-linux-kernel-and-the-android-userspace.html>