

# Main Memory



In Chapter 6, we showed how the CPU can be shared by a set of processes. As a result of CPU scheduling, we can improve both the utilization of the CPU and the speed of the computer's response to its users. To realize this increase in performance, however, we must keep several processes in memory—that is, we must share memory.

In this chapter, we discuss various ways to manage memory. The memory-management algorithms vary from a primitive bare-machine approach to paging and segmentation strategies. Each approach has its own advantages and disadvantages. Selection of a memory-management method for a specific system depends on many factors, especially on the hardware design of the system. As we shall see, many algorithms require hardware support, leading many systems to have closely integrated hardware and operating-system memory management.

## Bibliographical Notes

Dynamic storage allocation was discussed by [Knuth (1973)] (Section 2.5), who found through simulation results that first fit is generally superior to best fit. [Knuth (1973)] also discussed the 50-percent rule.

The concept of paging can be credited to the designers of the Atlas system, which has been described by [Kilburn et al. (1961)] and by [Howarth et al. (1961)]. The concept of segmentation was first discussed by [Dennis (1965)]. Paged segmentation was first supported in the GE 645, on which MULTICS was originally implemented ([Organick (1972)] and [Daley and Dennis (1967)]).

Inverted page tables are discussed in an article about the IBM RT storage manager by [Chang and Mergen (1988)].

[Hennessy and Patterson (2012)] explains the hardware aspects of TLBs, caches, and MMUs. [Talluri et al. (1995)] discusses page tables for 64-bit address spaces. [Jacob and Mudge (2001)] describes techniques for managing the TLB. [Fang et al. (2001)] evaluates support for large pages.

<http://msdn.microsoft.com/en-us/library/windows/hardware/gg487512.aspx> discusses PAE support for Windows systems.

<http://www.intel.com/content/www/us/en/processors/architectures-software-developer-manuals.html> provides various manuals for Intel 64 and IA-32 architectures.

<http://www.arm.com/products/processors/cortex-a/cortex-a9.php> provides an overview of the ARM architecture.

## Bibliography

- [**Bershad et al. (1995)**] B. N. Bershad, S. Savage, P. Pardyak, E. G. Sirer, M. Fiuczynski, D. Becker, S. Eggers, and C. Chambers, “Extensibility, Safety and Performance in the SPIN Operating System”, *Proceedings of the ACM Symposium on Operating Systems Principles* (1995), pages 267–284.
- [**Bovet and Cesati (2002)**] D. P. Bovet and M. Cesati, *Understanding the Linux Kernel, Second Edition*, O’Reilly & Associates (2002).
- [**Chang and Mergen (1988)**] A. Chang and M. F. Mergen, “801 Storage: Architecture and Programming”, *ACM Transactions on Computer Systems*, Volume 6, Number 1 (1988), pages 28–50.
- [**Chase et al. (1994)**] J. S. Chase, H. M. Levy, M. J. Feeley, and E. D. Lazowska, “Sharing and Protection in a Single-Address-Space Operating System”, *ACM Transactions on Computer Systems*, Volume 12, Number 4 (1994), pages 271–307.
- [**Daley and Dennis (1967)**] R. C. Daley and J. B. Dennis, “Virtual Memory, Processes, and Sharing in Multics”, *Proceedings of the ACM Symposium on Operating Systems Principles* (1967), pages 121–128.
- [**Dennis (1965)**] J. B. Dennis, “Segmentation and the Design of Multiprogrammed Computer Systems”, *Communications of the ACM*, Volume 8, Number 4 (1965), pages 589–602.
- [**Dougan et al. (1999)**] C. Dougan, P. Mackerras, and V. Yodaiken, “Optimizing the Idle Task and Other MMU Tricks”, *Proceedings of the Symposium on Operating System Design and Implementation* (1999).
- [**Fang et al. (2001)**] Z. Fang, L. Zhang, J. B. Carter, W. C. Hsieh, and S. A. McKee, “Reevaluating Online Superpage Promotion with Hardware Support”, *Proceedings of the International Symposium on High-Performance Computer Architecture*, Volume 50, Number 5 (2001).
- [**Hennessy and Patterson (2012)**] J. Hennessy and D. Patterson, *Computer Architecture: A Quantitative Approach, Fifth Edition*, Morgan Kaufmann (2012).
- [**Howarth et al. (1961)**] D. J. Howarth, R. B. Payne, and F. H. Sumner, “The Manchester University Atlas Operating System, Part II: User’s Description”, *Computer Journal*, Volume 4, Number 3 (1961), pages 226–229.
- [**Jacob and Mudge (1997)**] B. Jacob and T. Mudge, “Software-Managed Address Translation”, *Proceedings of the International Symposium on High Performance Computer Architecture and Implementation* (1997).
- [**Jacob and Mudge (1998)**] B. Jacob and T. Mudge, “Virtual Memory in Contemporary Microprocessors”, *IEEE Micro Magazine*, Volume 18, (1998), pages 60–75.

- [**Jacob and Mudge (2001)**] B. Jacob and T. Mudge, “Uniprocessor Virtual Memory Without TLBs”, *IEEE Transactions on Computers*, Volume 50, Number 5 (2001).
- [**Kilburn et al. (1961)**] T. Kilburn, D. J. Howarth, R. B. Payne, and F. H. Sumner, “The Manchester University Atlas Operating System, Part I: Internal Organization”, *Computer Journal*, Volume 4, Number 3 (1961), pages 222–225.
- [**Knuth (1973)**] D. E. Knuth, *The Art of Computer Programming, Volume 1: Fundamental Algorithms, Second Edition*, Addison-Wesley (1973).
- [**Organick (1972)**] E. I. Organick, *The Multics System: An Examination of Its Structure*, MIT Press (1972).
- [**Talluri et al. (1995)**] M. Talluri, M. D. Hill, and Y. A. Khalidi, “A New Page Table for 64-bit Address Spaces”, *Proceedings of the ACM Symposium on Operating Systems Principles* (1995), pages 184–200.
- [**Tanenbaum (2001)**] A. S. Tanenbaum, *Modern Operating Systems*, Prentice Hall (2001).
- [**Thorn (1997)**] T. Thorn, “Programming Languages for Mobile Code”, *ACM Computing Surveys*, Volume 29, Number 3 (1997), pages 213–239.
- [**Wahbe et al. (1993)**] R. Wahbe, S. Lucco, T. E. Anderson, and S. L. Graham, “Efficient Software-Based Fault Isolation”, *ACM SIGOPS Operating Systems Review*, Volume 27, Number 5 (1993), pages 203–216.

