Virtual Memory

In Chapter 9, we discussed various memory-management strategies used in computer systems. All these strategies have the same goal: to keep many processes in memory simultaneously to allow multiprogramming. However, they tend to require that an entire process be in memory before it can execute.

Virtual memory is a technique that allows the execution of processes that are not completely in memory. One major advantage of this scheme is that programs can be larger than physical memory. Further, virtual memory abstracts main memory into an extremely large, uniform array of storage, separating logical memory as viewed by the programmer from physical memory. This technique frees programmers from the concerns of memory-storage limitations. Virtual memory also allows processes to share files and libraries, and to implement shared memory. In addition, it provides an efficient mechanism for process creation. Virtual memory is not easy to implement, however, and may substantially decrease performance if it is used carelessly. In this chapter, we provide a detailed overview of virtual memory, examine how it is implemented, and explore its complexity and benefits.

Bibliographical Notes

Demand paging was first used in the Atlas system, implemented on the Manchester University MUSE computer around 1960 ([Kilburn et al. (1961)]). Another early demand-paging system was MULTICS, implemented on the GE 645 system ([Organick (1972)]). Virtual memory was added to Unix in 1979 [Babaoglu and Joy (1981)]

[Belady et al. (1969)] were the first researchers to observe that the FIFO replacement strategy may produce the anomaly that bears Belady's name. [Mattson et al. (1970)] demonstrated that stack algorithms are not subject to Belady's anomaly.

The optimal replacement algorithm was presented by [Belady (1966)] and was proved to be optimal by [Mattson et al. (1970)]. Belady's optimal algorithm is for a fixed allocation; [Prieve and Fabry (1976)] presented an optimal algorithm for situations in which the allocation can vary.

The enhanced clock algorithm was discussed by [Carr and Hennessy (1981)].

The working-set model was developed by [Denning (1968)]. Discussions concerning the working-set model were presented by [Denning (1980)].

The scheme for monitoring the page-fault rate was developed by [Wulf (1969)], who successfully applied this technique to the Burroughs B5500 computer system.

Buddy system memory allocators were described in [Knowlton (1965)], [Peterson and Norman (1977)], and [Purdom, Jr. and Stigler (1970)]. [Bonwick (1994)] discussed the slab allocator, and [Bonwick and Adams (2001)] extended the discussion to multiple processors. Other memory-fitting algorithms can be found in [Stephenson (1983)], [Bays (1977)], and [Brent (1989)]. A survey of memory-allocation strategies can be found in [Wilson et al. (1995)].

[Solomon and Russinovich (2000)] and [Russinovich et al. (2017)] described how Windows implements virtual memory. [McDougall and Mauro (2007)] discussed virtual memory in Solaris. Virtual memory techniques in Linux and FreeBSD were described by [Love (2010)] and [McKusick et al. (2015)], respectively. [Ganapathy and Schimmel (1998)] and [Navarro et al. (2002)] discussed operating system support for multiple page sizes.

Bibliography

[Babaoglu and Joy (1981)] O. Babaoglu and W. Joy, "Converting a Swap-Based System to Do Paging in an Architecture Lacking Page-Reference Bits", *Proceedings of the ACM Symposium on Operating Systems Principles* (1981), pages 78–86.

[Bays (1977)] C. Bays, "A Comparison of Next-Fit, First-Fit and Best-Fit", *Communications of the ACM*, Volume 20, Number 3 (1977), pages 191–192.

[Belady (1966)] L. A. Belady, "A Study of Replacement Algorithms for a Virtual-Storage Computer", *IBM Systems Journal*, Volume 5, Number 2 (1966), pages 78–101.

[Belady et al. (1969)] L. A. Belady, R. A. Nelson, and G. S. Shedler, "An Anomaly in Space-Time Characteristics of Certain Programs Running in a Paging Machine", *Communications of the ACM*, Volume 12, Number 6 (1969), pages 349–353.

[Bonwick (1994)] J. Bonwick, "The Slab Allocator: An Object-Caching Kernel Memory Allocator", *USENIX Summer* (1994), pages 87–98.

[Bonwick and Adams (2001)] J. Bonwick and J. Adams, "Magazines and Vmem: Extending the Slab Allocator to Many CPUs and Arbitrary Resources", *Proceedings of the 2001 USENIX Annual Technical Conference* (2001).

[Brent (1989)] R. Brent, "Efficient Implementation of the First-Fit Strategy for Dynamic Storage Allocation", *ACM Transactions on Programming Languages and Systems*, Volume 11, Number 3 (1989), pages 388–403.

[Carr and Hennessy (1981)] W. R. Carr and J. L. Hennessy, "WSClock—A Simple and Effective Algorithm for Virtual Memory Management", *Proceedings of the ACM Symposium on Operating Systems Principles* (1981), pages 87–95.

[Denning (1968)] P. J. Denning, "The Working Set Model for Program Behavior", *Communications of the ACM*, Volume 11, Number 5 (1968), pages 323–333.

[Denning (1980)] P. J. Denning, "Working Sets Past and Present", *IEEE Transactions on Software Engineering*, Volume SE-6, Number 1 (1980), pages 64–84.

[Ganapathy and Schimmel (1998)] N. Ganapathy and C. Schimmel, "General Purpose Operating System Support for Multiple Page Sizes", *Proceedings of the USENIX Technical Conference* (1998).

[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.

[Knowlton (1965)] K. C. Knowlton, "A Fast Storage Allocator", *Communications of the ACM*, Volume 8, Number 10 (1965), pages 623–624.

[Love (2010)] R. Love, *Linux Kernel Development*, Third Edition, Developer's Library (2010).

[Mattson et al. (1970)] R. L. Mattson, J. Gecsei, D. R. Slutz, and I. L. Traiger, "Evaluation Techniques for Storage Hierarchies", *IBM Systems Journal*, Volume 9, Number 2 (1970), pages 78–117.

[McDougall and Mauro (2007)] R. McDougall and J. Mauro, *Solaris Internals*, Second Edition, Prentice Hall (2007).

[McKusick et al. (2015)] M. K. McKusick, G. V. Neville-Neil, and R. N. M. Watson, *The Design and Implementation of the FreeBSD UNIX Operating System–Second Edition*, Pearson (2015).

[Navarro et al. (2002)] J. Navarro, S. Lyer, P. Druschel, and A. Cox, "Practical, Transparent Operating System Support for Superpages", *Proceedings of the USENIX Symposium on Operating Systems Design and Implementation* (2002).

[Organick (1972)] E. I. Organick, *The Multics System: An Examination of Its Structure*, MIT Press (1972).

[Peterson and Norman (1977)] J. L. Peterson and T. A. Norman, "Buddy Systems", *Communications of the ACM*, Volume 20, Number 6 (1977), pages 421–431.

[Prieve and Fabry (1976)] B. G. Prieve and R. S. Fabry, "VMIN—An Optimal Variable Space Page-Replacement Algorithm", *Communications of the ACM*, Volume 19, Number 5 (1976), pages 295–297.

[Purdom, Jr. and Stigler (1970)] P. W. Purdom, Jr. and S. M. Stigler, "Statistical Properties of the Buddy System", *J. ACM*, Volume 17, Number 4 (1970), pages 683–697.

[Russinovich et al. (2017)] M. Russinovich, D. A. Solomon, and A. Ionescu, *Windows Internals–Part 1*, Seventh Edition, Microsoft Press (2017).

[Solomon and Russinovich (2000)] D. A. Solomon and M. E. Russinovich, *Inside Microsoft Windows 2000*, Third Edition, Microsoft Press (2000).

[Stephenson (1983)] C. J. Stephenson, "Fast Fits: A New Method for Dynamic Storage Allocation", *Proceedings of the Ninth Symposium on Operating Systems Principles* (1983), pages 30–32.

[Wilson et al. (1995)] P. R. Wilson, M. S. Johnstone, M. Neely, and D. Boles, "Dynamic Storage Allocation: A Survey and Critical Review", *Proceedings of the International Workshop on Memory Management* (1995), pages 1–116.

[Wulf (1969)] W. A. Wulf, "Performance Monitors for Multiprogramming Systems", *Proceedings of the ACM Symposium on Operating Systems Principles* (1969), pages 175–181.