In Chapter 8, 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 user from physical memory. This technique frees programmers from the concerns of memory-storage limitations. Virtual memory also allows processes to share files easily 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 discuss virtual memory in the form of demand paging and examine its complexity and cost.

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.
Chapter 9  Virtual Memory

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 and Solomon (2005)] 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 and Neville-Neil (2005)], respectively. [Ganapathy and Schimmel (1998)] and [Navarro et al. (2002)] discussed operating system support for multiple page sizes.

Bibliography


on Software Engineering, Volume SE-6, Number 1 (1980), pages 64–84.

Purpose Operating System Support for Multiple Page Sizes”, Proceedings of the

[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 Organiza-

of the ACM, Volume 8, Number 10 (1965), pages 623–624.

Library (2010).

[Mattson et al. (1970)]  R. L. Mattson, J. Gecsei, D. R. Slutz, and I. L. Traiger,
9, Number 2 (1970), pages 78–117.

[McDougall and Mauro (2007)]  R. McDougall and J. Mauro, Solaris Internals,

[McKusick and Neville-Neil (2005)]  M. K. McKusick and G. V. Neville-Neil,
The Design and Implementation of the FreeBSD UNIX Operating System, Addison
Wesley (2005).

[Navarro et al. (2002)]  J. Navarro, S. Lyer, P. Druschel, and A. Cox, “Practical,
Transparent Operating System Support for Superpages”, Proceedings of the

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

[Peterson and Norman (1977)]  J. L. Peterson and T. A. Norman, “Buddy Sys-

Variable Space Page-Replacement Algorithm”, Communications of the ACM,
Volume 19, Number 5 (1976), pages 295–297.

Properties of the Buddy System”, J. ACM, Volume 17, Number 4 (1970), pages
683–697.

[Russinovich and Solomon (2005)]  M. E. Russinovich and D. A. Solomon,

[Solomon and Russinovich (2000)]  D. A. Solomon and M. E. Russinovich, Inside

Storage Allocation”, Proceedings of the Ninth Symposium on Operating Systems
Chapter 9  Virtual Memory
