Mass-Storage Structure

The file system can be viewed logically as consisting of three parts. In Chapter 10, we examine the user and programmer interface to the file system. In Chapter 11, we describe the internal data structures and algorithms used by the operating system to implement this interface. In this chapter, we begin discussion of file systems at the lowest level: the structure of secondary storage. We first describe the physical structure of magnetic disks and magnetic tapes. We then describe disk-scheduling algorithms, which schedule the order of disk I/Os to maximize performance. Next, we discuss disk formatting and management of boot blocks, damaged blocks, and swap space. We conclude with an examination of the structure of RAID systems.

Bibliographical Notes

[Services (2012)] provides an overview of data storage in a variety of modern computing environments. [Teorey and Pinkerton (1972)] present an early comparative analysis of disk-scheduling algorithms using simulations that model a disk for which seek time is linear in the number of cylinders crossed. Scheduling optimizations that exploit disk idle times are discussed in [Lumb et al. (2000)].[Kim et al. (2009)] discusses disk-scheduling algorithms for SSDs.

Discussions of redundant arrays of independent disks (RAIDs) are presented by [Patterson et al. (1988)].

[Russinovich and Solomon (2009)], [McDougall and Mauro (2007)], and [Love (2010)] discuss file system details in Windows, Solaris, and Linux respectively.

The I/O size and randomness of the workload has a considerable influence on disk performance. [Ousterhout et al. (1985)] and [Ruemmler and Wilkes (1993)] report numerous interesting workload characteristics, including that most files are small, most newly created files are deleted soon thereafter, most files that are opened for reading are read sequentially in their entirety, and most seeks are short.

The concept of a storage hierarchy has been studied for more than forty years. For instance, a 1970 paper by [Mattson et al. (1970)] describes a mathematical approach to predicting the performance of a storage hierarchy.

Bibliography

[Kim et al. (2009)] J. Kim, Y. Oh, E. Kim, J. C. D. Lee, and S. Noh, "Disk Schedulers for Solid State Drivers" (2009), pages 295–304.

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

[Lumb et al. (2000)] C. Lumb, J. Schindler, G. R. Ganger, D. F. Nagle, and E. Riedel, "Towards Higher Disk Head Utilization: Extracting Free Bandwidth From Busy Disk Drives", *Symposium on Operating Systems Design and Implementation* (2000).

[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).

[Ousterhout et al. (1985)] J. K. Ousterhout, H. D. Costa, D. Harrison, J. A. Kunze, M. Kupfer, and J. G. Thompson, "A Trace-Driven Analysis of the UNIX 4.2 BSD File System", *Proceedings of the ACM Symposium on Operating Systems Principles* (1985), pages 15–24.

[Patterson et al. (1988)] D. A. Patterson, G. Gibson, and R. H. Katz, "A Case for Redundant Arrays of Inexpensive Disks (RAID)", *Proceedings of the ACM SIGMOD International Conference on the Management of Data* (1988), pages 109–116.

[Ruemmler and Wilkes (1993)] C. Ruemmler and J. Wilkes, "Unix Disk Access Patterns", *Proceedings of the Winter USENIX Conference* (1993), pages 405–420.

[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).

[Services (2012)] E. E. Services, *Information Storage and Management: Storing, Managing, and Protecting Digital Information in Classic, Virtualized, and Cloud Environments*, Wiley (2012).

[Teorey and Pinkerton (1972)] T. J. Teorey and T. B. Pinkerton, "A Comparative Analysis of Disk Scheduling Policies", *Communications of the ACM*, Volume 15, Number 3 (1972), pages 177–184.