

# Log-structured file system

From Wikipedia, the free encyclopedia  
(Redirected from LogFS)

*This is about the general concept of log-structured file systems. For the NetBSD file system, see Log-structured File System.*

A **log-structured filesystem** is a file system design first proposed by John K. Ousterhout and Fred Douglass.

## Contents

- 1 Rationale
- 2 Implementations
- 3 Disadvantages
- 4 References
- 5 See also

## Rationale

Conventional file systems tend to lay out files with great care for spatial locality and make in-place changes to their data structures in order to perform well on magnetic disks, which tend to seek relatively slowly.

The design of log-structured file systems is based on the hypothesis that this will no longer be effective because ever-increasing memory sizes on modern computers would lead to I/O becoming write-heavy because reads would be almost always satisfied from memory cache. A log-structured file system thus treats its storage as a circular log and writes sequentially to the head of the log. This maximizes write throughput on magnetic media by avoiding costly seeks.

Keeping a log has several important side effects:

- Writes create multiple, chronologically-advancing versions of both file data and meta-data. Some implementations make these old file versions nameable and accessible, a feature sometimes called time-travel or snapshotting. This is very similar to a versioning file system.
- Recovery from crashes is simpler. Upon its next mount, the file system does not need to walk all its data structures to fix any inconsistencies, but can reconstruct its state from the last consistent point in the log.
- Free space must be constantly reclaimed from the tail of the log to prevent the file system from becoming full when the head of the log wraps around to meet it. The tail itself can skip forward over data for which newer versions exist farther ahead in the log; the remainder is simply moved out of the way by appending it back to the head. To minimize the overhead incurred by this garbage collection, most implementations avoid purely circular logs and divide up their storage into segments or extents. The head of the log can then advance onto non-adjacent segments which are already free or which are less full than at the tail, thus reducing the amount of garbage collection I/O needed to reclaim space.

## Implementations

- John K. Ousterhout and Mendel Rosenblum implemented the first log-structured file system for the Sprite operating system in 1992.<sup>[1][2]</sup>
- BSD-LFS, an implementation by Margo Seltzer was added to 4.4BSD, and was later ported to 386BSD. It lacked support for snapshots. It was removed from FreeBSD and OpenBSD, but still lives on in NetBSD.
- Plan 9's Fossil file system is also log-structured and supports snapshots.
- NILFS is a log-structured file system implementation for Linux by NTT/Verio which supports snapshots. As of May 2007, it is still in alpha and not ready for production use.
- LinLogFS (formerly dtfs) and LFS (<http://logfs.sourceforge.net/>) are log-structured file system implementations for Linux. The latter was part of Google Summer of Code 2005. Both projects have been abandoned.
- ZFS from Sun is a log-structured file system which supports snapshotting. However, it uses reference counting instead of garbage collection to manage free space.
- LFS (<http://aiya.ms.mff.cuni.cz/lfs>) is another log-structured file system for Linux developed Charles University, Prague. It was to include support for snapshots and indexed directories, but development has since ceased.

- LogFS (<http://logfs.org/logfs/>) is a scalable flash filesystem, intended to replace JFFS2. Early in development.

Some kinds of storage media, such as flash memory and CD-RW, degrade slowly as they are written to and have a limited number of erase/write cycles at any one location. Log-structured file systems are sometimes used on these media because they make fewer in-place writes and thus prolong the life of the device by wear levelling. The more common such file systems include:

- UDF is a file system commonly used on optical discs.
- JFFS and its successor JFFS2 are simple Linux file systems intended for flash-based devices.
- YAFFS is a NAND flash-specific file system for Linux.

## Disadvantages

- The design rationale for log-structured file systems assume that most reads will be optimized away by ever-enlarging memory caches. This assumption is not true for all workloads: on magnetic media (where seeks are relatively expensive) the log structure may actually make reads much slower, since it fragments files that conventional file systems normally keep contiguous with in-place writes.

## References

1. ^ Rosenblum, Mendel and Ousterhout, John K. (June 1990) - "The LFS Storage Manager (<http://citeseer.ist.psu.edu/rosenblum90lfs.html>) ". *Proceedings of the 1990 Summer Usenix*. pp315-324.
2. ^ Rosenblum, Mendel and Ousterhout, John K. (February 1992) - "The Design and Implementation of a Log-Structured File System (<http://citeseer.ist.psu.edu/rosenblum91design.html>) ". *ACM Transactions on Computer Systems, Vol. 10 Issue 1*. pp26-52.

## See also

- Comparison of file systems

Retrieved from "[http://en.wikipedia.org/wiki/Log-structured\\_file\\_system](http://en.wikipedia.org/wiki/Log-structured_file_system)"

Categories: Computer file systems | Computer storage | Unix | Linux | Sun Microsystems | Bell Labs

- 
- This page was last modified 06:13, 16 July 2007.
  - All text is available under the terms of the GNU Free Documentation License. (See **Copyrights** for details.)
- Wikipedia® is a registered trademark of the Wikimedia Foundation, Inc., a US-registered 501(c)(3) tax-deductible nonprofit charity.