YAFFS

From Wikipedia, the free encyclopedia

YAFFS (**Yet Another Flash File System**) was designed and written by Charles Manning, of Whitecliffs, New Zealand, for the company Aleph One (http://www.aleph1.co.uk/) . YAFFS is the first file system that was designed specifically for NAND flash.

Yaffs1 is the first version of this file system and works on NAND chips that have 512 byte pages + 16 byte spare (OOB) areas. These older chips also generally allow 2 or 3 write cycles per page, which YAFFS takes advantage of - i.e. dirty pages are marked by writing to a specific spare area byte.

Newer NAND flash chips have larger pages, 2048 bytes + 64 bytes spare areas, and stricter write requirements. Each page within a block must be written to in sequential order, and each page must be written only once. YAFFS2 was designed to accommodate these newer chips. YAFFS2 is based on the YAFFS1 source code, with the major difference being that internal structures are not fixed to assume 512 byte sizing, and a block sequence number is placed on each written page. In this way older pages can be logically overwritten without violating the "write once" rule.

YAFFS is a robust log-structured file system that holds data integrity as a high priority. A secondary YAFFS goal is high performance. YAFFS will typically outperform most alternatives. It is also designed to be portable and has been used on Linux, WinCE, pSOS, eCOS, ThreadX and various special-purpose OSes. A variant 'YAFFS/Direct' is used in situations where there is no OS, embedded OSes and bootloaders: it has the same core filesystem but simpler interfacing to the OS and NAND flash hardware.

The filesystem is licensed both under the GPL and under per-product licences available from Aleph One.

Contents

- 1 YAFFS1
- 2 YAFFS2
- 3 See also
- 4 External links

YAFFS1

YAFFS has no inherent formatting, an erased flash chip is formatted. It follows the smart media scheme of marking the 5th byte of the spare area for bad blocks, and ignores any blocks where the spare area byte 5 is not 0xFF.

To write file data, YAFFS initially writes a whole page (chunk in YAFFS terminology) that describes the file metadata, such as timestamps, name, path, etc. The new file is assigned a unique object ID number; every data chunk within the file will contain this unique object ID within the spare area. YAFFS maintains a tree structure in RAM memory of the physical location of these chunks. When a chunk is no longer valid (the file is deleted, or parts of the file are overwritten), YAFFS marks a particular byte in the spare area of the chunk as ‘dirty’. When an entire block (32 pages) is marked as dirty, YAFFS can erase the block and reclaim the space. If free space on the device is low, YAFFS may need to choose a block that has some number of dirty pages and some number of good pages, move the good pages to a new block, mark the old pages as dirty and erase the block. The process of moving good pages & erasing blocks is called Garbage Collection.

When a YAFFS system mounts a NAND flash device, it must scan the spare areas of every block to check for valid data, whereby it can then reconstitute the tree data structures.

YAFFS2

YAFFS2 is similar in concept to YAFFS1, and shares much the same code; and the YAFFS2 code base supports YAFFS1 data formats through backward compatibility. The main difference is that YAFFS2 needs to jump through significant hoops to meet the "write once" requirement of modern NAND flash.

YAFFS2 marks every newly written block with a sequence number that is monotonically increasing. The sequence of the chunks can be inferred from the block sequence number and the chunk offset within the block. Thereby when YAFFS2 scans the flash and detects multiple chunks that have identical ObjectIDs and ChunkNumbers, it can choose which to use by taking the greatest sequence number. For efficiency reasons YAFFS2 also introduces the concept of shrink headers. For example when a file is resized to a smaller size, YAFFS1 will mark all of the affected chunks as dirty - YAFFS2 cannot do this due to the "write once" rule. YAFFS2 instead writes a "shrink header", which indicates that a certain number of pages before that point are invalid. This lets YAFFS2 reconstruct the final state of the filesystem when the system reboots.

YAFFS2 uses a more abstract definition of the NAND flash allowing it to be used with a wider variety of flash parts with different geometries, bad block handling rules etc.

YAFFS2 now supports "checkpointing" which bypasses normal mount scanning, allowing very fast mount times. Mileage will vary, but mount times of ~3 seconds for 2Gbytes have been reported.

See also

- List of file systems
- JFFS
- JFFS2
- Open NAND Flash Interface Working Group

External links

- Introducing YAFFS, the first NAND-specific flash file system (http://www.linuxdevices.com/articles/AT9680239525.html)
- YAFFS - Yet Another Flash Filing System, is a filing system optimised for NAND Flash chips (http://www.aleph1.co.uk/yaffs/)
- YAFFS main website (http://www.yaffs.net)

Retrieved from "http://en.wikipedia.org/wiki/YAFFS"

Category: Flash file systems
