Knight File System (KFS) Specs

KnightOS

The KFS is split into two primary entities: FAT and Data. These cannot co-exist on the same flash page. The FAT pages are further split into Directory and File segments.

FAT Segments

The FAT, or File Allocation Table, describes how the files and directories are organized, and contains information such as file size, names, and flash pages. These pages are prefixed with a FAT page ID.

Directories

Directory entries start from offset 0000h and continue up to 0FFFh, and File entries start from 1000h and continue to 3FFFh. Directory entries are composed as follows:

Type ID (11111110b for Directory) Byte

Directory ID Word ; The unique ID for this directory

Residing Directory ID Word ; The directory containing this directory

Name (0 delimited)

Files

File entries describe actual files, and are composed like this:

Type ID (11111110b for File) Byte

Directory ID Word ; The directory containing this file

Name (0 delimited)

Offset Word

Multi-Page Allocation

The FAT can extend over several pages if need be. A type ID of 11111100b represents a page swap, found at the end of a FAT page. It consists of the following:

Type ID (11111100b for page swap) byte

Next Page byte

Type IDs

Type IDs represent the type of entry, and there are four (in binary):

11111111b represents the end of a file or directory table

11111110b represents a file or directory entry

11111100b represents a page swap entry (See Multi-Page Allocation for more information)

11111000b represents a FAT page

11110000b represents a data page

00000000b represents a garbage entry, or deleted entry (See Garbage Collection for more information)

Special Entries

Garbage

A garbage entry is an entry that is set to be removed on the next garbage collection. All bits in the type ID are reset.

Page Swap

Page swap entries are found at the end of a table, and represent that the table continues on another page.

Data Segments

Data segments contain the actual files. They are back to back on data pages. Files have a simple header: 3 bytes mark the file size. When edited, old versions are abandoned, and the new version is added onto the end of the data segment. When the data segment is out of space, a garbage collection is triggered.

Garbage Collection

When either the FAT or Data sections run out of space, a garbage collection is triggered. It should follow this process:

1. Update the FAT.
    a. Copy it to the swap sector
    b. Erase the original sector
    c. Write it back, leaving out garbage entries
    d. Erase the swap sector
2. Update the data
    a. Copy the first page to the swap sector
    b. Erase the original page
    c. Re-write it back, leaving out entries that do not have a corresponding FAT entry