Alex Jones

Yeonseo Ju

Ryan McGill

Kegan Vanginkel

Yadi Zhong

Primary memory

File in memory entry - fmem

Fields

name	type(size in bytes)
contents	unsigned short*(8)
cur	unsigned short(2)
size	unsigned short(2)
bufSize	unsigned short(2)
flags	char(1)

contents

Type: unsigned short*

Stores a pointer to the file data buffer. File data buffer is allocated to be file size plus the sector free space to allow for a file to expand to it's maximum size. The data in this buffer is initialized to the current data on disk, both in the file, and in the space after it. This allows for proper setting of the flash flag. Deallocated when the file closed after the write to disk has complete

cur

Type: unsigned short

Stores the current writing/reading location in the file. Needed for seeking operation.

size

Type: unsigned short

The current file size. This indicates where the last valid data byte is in a file. This is incremented as the program writes past the end of the file. Cannot exceed bufsize.

bufsize

Type: unsigned short

Used to store the size of the contents memory buffer. This is set on file open, and is used to determine if additional writes would overflow a sector

flags

Type: char

First bit says if the file is open. This is used in combination with the flock table to lock programs from opening files if they are already open somewhere else.

The second bit is set if the data that has been written to the buffer would require the sector to be flashed. Allows for faster writes in the event that a write operation doesn't need to flash the sector.

The third bit specifies if the file was opened for writing or not. Needed to allow for reads that don't flush to disk when closed.

File in memory table - flock

Fields

name	type(size in bytes)
files	fmem(16)[NUM_SECTORS*NUM_FILES]

files

Type: array of fmem

Used to store the structure of every possible file on the disk. Uses minimal space because unless the file is open the memory buffer is unallocated. As the name implies this is used to prevent multiple openings of the same file. The index in the array specifies both the sector number and the file number within the sector.

There is only one of these structures in the entire memory.

Secondary memory

File allocation entry - fdisk

Fields

name	type(size in bytes)
name	char(1)[8]
size	unsigned short(2)

name

Type: array of char

Used to store the name of the file. Not necessary, as files can be addressed by their sector/file number, but a convenience feature. If directories are later desired, they can be added by increasing the length of this field and storing the path directly in this string. Not null terminated as this field is fixed size.

size

Type: unsigned short

Specifies the size of the file on disk. Needed to allow for multiple files in one block.

File allocation table - fat

Fields

name	type(size in bytes)
fsector	fdisk(10)[NUM_FILES]

fsector

One of these structures stored at the start of each sector, storing an explicit entry for each possible file in that sector. The number of entries in this table is set at compile time and dictates the number of files that can be stored in a sector.

Definitions

Name	Description
NUM_FILES	The number of files per sector
NUM_SECTORS	The number of sectors