ByteFrost File System

Version 1

File System Description

The initial design of the ByteFrost file system is intended for a rather simple usage of the disk for booting and to maintain a small number of files. In this sense, this initial filesystem is somewhat comparable to the Apple II DOS filesystem, in that the disk has no directories besides the root.

Since the ByteFrost has to handle all filesystem details, this initial implementation is kept simple to make easier the transition from the original architecture (where there are separate address spaces for the code, data RAM, and stack) to a more conventional architecture (one address space containing the code, stack, heap, etc.).

A few other changes may be necessary besides reading the program code from the disk into RAM in order to execute it, such as enlarging the address space slightly (perhaps initially a minor jump from 8-bit addressing to 9 or 10 bits, though eventually a 16-bit address space is desirable).

File System version 1 Specs

Spec	Value
ByteFrost `disk` file size	`512 KB`
Sector size (Arduino interface)	`256 bytes`
Block size (filesystem)	`256 bytes`
Disk block pointer size	`11 bits` (`8` bits block offset)
Maximum number of files	`16`
Directory structure	`root` only
Maximum file size	`31.5 KB`
Inode size	`256 bytes`

~~Disk file size is `16 MB` so that disk block pointers are `16 bits` wide (even though most of the 16 MB won't be used; if all 16 files are at max size, a little over half a megabyte will be used up). In later versions, disk space will be used more fully.~~

The disk layout looks like this:

Block Range	# blocks	Size	Use
`0 - 7`	`8`	`2 KB`	Boot code
`8`	`1`	`256 bytes`	Directory block
`9`	`1`	`256 bytes` (252 bytes used)	Data block freemap

Block Range	# blocks	Size	Use
`10`	`1`	`256 bytes` (2 bytes used)	Inode block freemap
`11 - 26`	`16`	`4 KB`	Inode blocks
`27 - 2042`	`2016`	`504 KB`	Data blocks
`2043 - 2047`	`5`	`1.25 KB`	Unused

`504 KB (data) / 512 KB (total) = 98.4%` of the disk can be used for data (`(2 KB + 0.75 KB + 4 KB) / 512 KB = 1.3%` overhead, and `1.25 KB / 512 KB = 0.2%` unused).

Each directory entry takes `16 bytes` of space (`14 bytes` for filename (14-char filenames) and `2 bytes` for inode block pointer (`11 bits` of which are used as disk block pointers are 11 bits long)).

To mark a directory entry invalid, set its inode block pointer to `0`.

Inode Structure

Field	Size
File size (in bytes)	`2 bytes`
File size (in blocks)	`1 byte`
File type	`1 byte`
126 direct data block pointers	`252 bytes`

Each file can at most be allocated `126 blocks` (hence file size in blocks can be represented using one byte), and as the max file size is thus `31.5 KB`, the file size in bytes can be represented using two bytes.