

CEM Embedded File system

This document describes the implementation of an embedded file system for solid state hard disks. It is loosely based on the pSOS pHILE system.

Volume blocks

The media surface is divided up to blocks. The size of these blocks vary, depending on the media. Normally each block is 512 bytes.

Block number	Item	Comment
0	Boot load	Not used
1	Boot load	Not used
2	Root block	Defines volume
3	Root Directory	
4	Usage bit map	Defines used blocks on media
4 + X	File description list	Description of files
4 + X + Y	Data area	

Root Block

Element	Item	Comment
0	Bit map address	Location block of usage bit map
1	File list address	Location block of file lists
2	Data address	Location block of data.
3	Initialisation time	Date and time when created
4	Volume name	
7	Volume size	Number of blocks
8	File list size	Number of file list descriptors
9	Validation key	

Root directory

The system will consist of files, directories and path names.

File names and directory names can have a maximum length of 12 characters.

Directory record structure.

This has a size of 16 bytes, so each block of 512 bytes, will hold 32 file names.

Item	Size (bytes)	Comment
File number	4	Points to record in FLD blocks
Name	12	

Volume bit map

This table is a binary representation of the blocks on the media surface. When a block is defined as used, it's state will be changed from a 0 to a 1.

This has a file name of BITMAP.SYS and it's size is fixed when the media is initialised, based on the size of the media device in use.

File list descriptor

Every file will have a file list descriptor record associated with it. This has a file name of FLIST.SYS and it's size is fixed when the media is initialised to the maximum number of files and directories that can be held on the media.

Each entry is 128 bytes long.

Element	Item	Description
0	Next fcb	
1	File size (bytes)	
2	File size (blocks)	System, data, directory file
3	File type	
4	Last modification time	Minimum size increment
5	File expansion size	
6	No of open occurrences	
7	File list descriptor	
8	Extent start 0	Block address first data block
9	Extent size 0	Number of blocks used
10	Extent start 1	
11	Extent size 1	
12	Extent start 2	
13	Extent size 2	
14	Extent start 3	
15	Extent size 3	
16	Extent start 4	
17	Extent size 4	
18	Extent start 5	
19	Extent size 5	
20	Extent start 6	
21	Extent size 6	
22	Extent start 7	
23	Extent size 7	
24	Extent start 8	
25	Extent size 8	
26	Extent start 9	
27	Extent size 9	
28	Indirect block	
29	Not used	
30	Index indirect	
31	Not used	

Disk functions

These are a list of functions that are likely to be required.

Function	Operation	Parameters in	returned
Create_f	Create new file	Name, len, size, expand	P/F
Create_d	Create new directory	Name, len	P/F
Ch-dir	Change to specified directory	Name, len	P/F
Delete_f	Delete file or directory	Name, len	P/F
Delete-files	Delete files	Name, len	P/F
Open_f	Open required file	Name, len	File ID
Close_f	Close required file	File ID	P/F
Read_f	Read data from file	File ID, length, buffer	Number read
Write_f	Write data to file	File ID, length, buffer	Number written
Lseek_f	Move data pointer in file	File ID, pointer	T/f
Annex-file	Add extra blocks to file	File ID, size	T/f
Get_f#	Get file number	Name, len	File number in FLD
Open_f#	Open file with file number	File number in FLD	File ID
Read_vol	Read media directly	Start byte, length, buffer	Length read
Write_vol	Write to media directly	Start byte, length, buffer	Length written
Init_vol	Format media	Name(char)	Size
Ldir	List directory files	Void	Void

Open file table

Every task has a series of Open file tables, one for every open file it is using. There is a limit to the number of files that a task may have open at the same time.

Element	Item
0	Read, write pointer
1	File control block pointer
2	File FLD number

File control block

Every open file has a File Control Block associated with it and more than one task can have access to the same FCB.

The system has a fixed number of FCBs, created at startup of the system. These contain a copy of the part of the FLD for the particular open file to allow fast access to the data

Element	Item	Description
0	File size (bytes)	
1	File size (blocks)	
2	File type	System, data, directory file
3	Last modification time	
4	File expansion size	Minimum size increment
5		
6	Task number	Task ID of current access.
7	File FLD number	Pointer to FLD
8	Extent start 0	Block address first data block
9	Extent size 0	Number of blocks used
10	Extent start 1	
11	Extent size 1	
12	Extent start 2	
13	Extent size 2	
14	Extent start 3	
15	Extent size 3	
16	Extent start 4	
17	Extent size 4	
18	Extent start 5	
19	Extent size 5	
20	Extent start 6	
21	Extent size 6	
22	Extent start 7	
23	Extent size 7	
24	Extent start 8	
25	Extent size 8	
26	Extent start 9	
27	Extent size 9	
28	Indirect block	
29	Not used	
30	Index indirect	
31	Not used	

