NAME

cpm − CP/M disk and file system format

DESCRIPTION

Characteristic sizes

Each CP/M disk format is described by the following specific sizes:

Sector size in bytes
Number of tracks
Number of sectors
Block size
Number of directory entries
Logical sector skew
Number of reserved system tracks (optional)
Offset to start of volume (optional)

A block is the smallest allocatable storage unit. CP/M supports block sizes of 1024, 2048, 4096, 8192 and 16384 bytes. Unfortunately, this format specification is not stored on the disk and there are lots of formats. Accessing a block is performed by accessing its sectors, which are stored with the given software skew.

Device areas

A CP/M disk contains three areas:

Volume offset (optional)
System tracks (optional)
Directory
Data

The system tracks store the boot loader and CP/M itself. In order to save disk space, there are non-bootable formats which omit those system tracks. The term *disk capacity* always excludes the space for system tracks. Note that there is no bitmap or list for free blocks. When accessing a drive for the first time, CP/M builds this bitmap in core from the directory.

A hard disk can have the additional notion of a *volume offset* to locate the start of the drive image (which may or may not have system tracks associated with it). The base unit for volume offset is byte count from the beginning of the physical disk, but specifiers of *K*, *M*, *T* or *S* may be appended to denote kilobytes, megabytes, tracks or sectors. If provided, a specifier must immediately follow the numeric value with no whitespace. For convenience upper and lower case are both accepted and only the first letter is significant, thus 2KB, 8MB, 1000trk and 16sec are valid values. Offset must appear subsequent to track, sector and sector length values.

Directory entries

The directory is a sequence of directory entries (also called extents), which contain 32 bytes of the following structure:

St	F0	F1	F2	F3	F4	F5	F6	F7	E0	E1	E2	Xl	Bc	Xh	Rc
Al	Al	Al	Al	Al	Al	Al	Al	Al	Al	Al	Al	Al	Al	Al	Al

St is the status; possible values are:

0–15: used for file, status is the user number
16–31: used for file, status is the user number (P2DOS) or used for password extent (CP/M 3 or higher)
32: disc label
33: time stamp (P2DOS)
0xE5: unused

F0–E2 are the file name and its extension. They may consist of any printable 7 bit ASCII character but: **< > . , ; : = ? * [ ]**. The file name must not be empty, the extension may be empty. Both are padded with blanks. The highest bit of each character of the file name and extension is used as attribute. The attributes have the following meaning:

F0: requires set wheel byte (Backgrounder II)
F1: public file (P2DOS, ZSDOS), forground-only command (Backgrounder II)
F2: date stamp (ZSDOS), background-only commands (Backgrounder II)
F7: wheel protect (ZSDOS)
E0: read-only
E1: system file
E2: archived

Public files (visible under each user number) are not supported by CP/M 2.2, but there is a patch and some free CP/M clones support them without any patches.

The wheel byte is (by default) the memory location at 0x4b. If it is zero, only non-privileged commands may be executed.

Xl and **Xh** store the extent number. A file may use more than one directory entry, if it contains more blocks than an extent can hold. In this case, more extents are allocated and each of them is numbered sequentially with an extent number. If a physical extent stores more than 16k, it is considered to contain multiple logical extents, each pointing to 16k data, and the extent number of the last used logical extent is stored. Note: Some formats decided to always store only one logical extent in a physical extent, thus wasting extent space. CP/M 2.2 allows 512 extents per file, CP/M 3 and higher allow up to 2048. Bit 5–7 of Xl are 0, bit 0–4 store the lower bits of the extent number. Bit 6 and 7 of Xh are 0, bit 0–5 store the higher bits of the extent number.

Rc and **Bc** determine the length of the data used by this extent. The physical extent is divided into logical extents, each of them being 16k in size (a physical extent must hold at least one logical extent, e.g. a blocksize of 1024 byte with two-byte block pointers is not allowed). Rc stores the number of 128 byte records of the last used logical extent. Bc stores the number of bytes in the last used record. The value 0 means 128 for backward compatibility with CP/M 2.2, which did not support Bc. ISX records the number of unused instead of used bytes in Bc.

Al stores block pointers. If the disk capacity is less than 256 blocks, Al is interpreted as 16 byte-values, otherwise as 8 double-byte-values. A block pointer of 0 marks a hole in the file. If a hole covers the range of a full extent, the extent will not be allocated. In particular, the first extent of a file does not neccessarily have extent number 0. A file may not share blocks with other files, as its blocks would be freed if the other files is erased without a following disk system reset. CP/M returns EOF when it reaches a hole, whereas UNIX returns zero-value bytes, which makes holes invisible.

Native time stamps

P2DOS and CP/M Plus support time stamps, which are stored in each fourth directory entry. This entry contains the time stamps for the extents using the previous three directory entries. Note that you really have time stamps for each extent, no matter if it is the first extent of a file or not. The structure of time stamp entries is:

1 byte status 0x21
8 bytes time stamp for third-last directory entry
2 bytes unused
8 bytes time stamp for second-last directory entry
2 bytes unused
8 bytes time stamp for last directory entry

A time stamp consists of two dates: Creation and modification date (the latter being recorded when the file is closed). CP/M Plus further allows optionally to record the access instead of creation date as first time stamp.

2 bytes (little-endian) days starting with 1 at 01-01-1978
1 byte hour in BCD format
1 byte minute in BCD format

DateStamper time stamps

The DateStamper software added functions to the BDOS to manage time stamps by allocating a read only file with the name "!!!TIME&.DAT" in the very first directory entry, covering the very first data blocks. It contains one entry per directory entry with the following structure of 16 bytes:

5 bytes create datefield
5 bytes access datefield
5 bytes modify datefield
1 byte checksum

The checksum is only used on every 8th entry (last entry in 128-byte record) and is the sum of the first 127 bytes of the record. Each datefield has this structure:

1 byte BCD coded year (no century, so it is sane assuming any year < 70 means 21st century)
1 byte BCD coded month
1 byte BCD coded day
1 byte BCD coded hour or, if the high bit is set, the high byte of a counter for systems without real time clock
1 byte BCD coded minute, or the low byte of the counter

Disc labels

CP/M Plus support disc labels, which are stored in an arbitrary directory entry. The structure of disc labels is:

1 byte status 0x20
F0–E2 are the disc label
1 byte mode: bit 7 activates password protection, bit 6 causes time stamps on access, but 5 causes time stamps on modifications, bit 4 causes time stamps on creation and bit 0 is set when a label exists. Bit 4 and 6 are exclusively set.
1 byte password decode byte: To decode the password, xor this byte with the password bytes in reverse order. To encode a password, add its characters to get the decode byte.
2 reserved bytes
8 password bytes
4 bytes label creation time stamp
4 bytes label modification time stamp

Passwords

CP/M Plus supports passwords, which are stored in an arbitrary directory entry. The structure of these entries is:

1 byte status (user number plus 16)
F0–E2 are the file name and its extension.
1 byte password mode: bit 7 means password required for reading, bit 6 for writing and bit 5 for deleting.
1 byte password decode byte: To decode the password, xor this byte with the password bytes in reverse order. To encode a password, add its characters to get the decode byte.
2 reserved bytes

8 password bytes

SEE ALSO

mkfs.cpm(1), *fsck.cpm*(1), *fsed.cpm*(1), *cpmls*(1)