CP/M 1.4 disc formats

CP/M 1.4 was designed to work with 8" 250k discs. Thus a CP/M 1.4 disc will be laid out in the following way:

```
 77 tracks in total;
 26 128-byte sectors per track, software skewed;
  2 reserved tracks;
  2 1k directory blocks, giving 64 directory entries;
240 1k data blocks, numbered 2-241.
```

The skew table reads:

```
1,7,13,19,25,5,11,17,23,3,9,15,21,2,8,14,20,26,6,12,18,24,4,10,16,22
```

The reserved tracks will contain an image of CP/M 1.4, used when the system is rebooted. It can therefore be deduced that CP/M 1.4 fits in 6.5k.

CP/M 1.4 directory

The CP/M 1.4 directory only has one type of entry:

```
SS F1 F2 F3 F4 F5 F6 F7 F8 T1 T2 T3 EX S1 S2 RC   .FILENAMETYP....
AL AL AL AL AL AL AL AL AL AL AL AL AL AL AL AL   ................

SS = Status.   0  => File exists
             0E5h => File deleted
              80h => File exists and is hidden. This feature was undocumented
                     and does not exist in later versions of CP/M.
Fn - filename
Tn - filetype. The characters used for these are 7-bit ASCII.
EX = Extent counter. If a file grows above 16k, then it will have multiple
    directory entries. The first entry has EX=0, the second has EX=1 etc.
    EX ranges from 0 to 31, thus allowing files up to 512k. CP/M 1.4 only
    allows 256k discs anyway.
S1 - reserved, set to 0.
S2 - reserved, set to 0.
RC - Number of records (1 record=128 bytes) used in this extent. If it is
    80h, this extent is full and there may be another one on the disc. File
    lengths are only saved to the nearest 128 bytes.
AL - Allocation. Each AL is the number of a 1k block on the disc. If an AL
    number is zero, that section of the file has no storage allocated to it
    (ie it does not exist). For example, a 3k file might have allocation
    5,6,8,0,0.... - the first 1k is in block 5, the second in block 6, the
    third in block 8.
```

Back to the formats listing

Back to the CP/M archive list

CP/M 2.2 disc formats

CP/M 2.2 works with a much larger range of discs than CP/M 1.4. The disc statistics are stored in a parameter block (the DPB), which contains the following information:

```
        DEFW    spt     ;Number of 128-byte records per track
        DEFB    bsh     ;Block shift. 3 => 1k, 4 => 2k, 5 => 4k....
        DEFB    blm     ;Block mask. 7 => 1k, 0Fh => 2k, 1Fh => 4k...
        DEFB    exm     ;Extent mask, see later
        DEFW    dsm     ;(no. of blocks on the disc)-1
        DEFW    drm     ;(no. of directory entries)-1
        DEFB    al0     ;Directory allocation bitmap, first byte
        DEFB    al1     ;Directory allocation bitmap, second byte
        DEFW    cks     ;Checksum vector size, 0 for a fixed disc
                        ;No. directory entries/4, rounded up.
        DEFW    off     ;Offset, number of reserved tracks
```

```
The directory allocation bitmap is interpreted as:

       al0              al1
b7b6b5b4b3b2b1b0 b7b6b5b4b3b2b1b0
 1 1 1 1 0 0 0 0  0 0 0 0 0 0 0 0

 - ie, in this example, the first 4 blocks of the disc contain the directory.
```

The DPB is not stored on disc. It is either hardwired into the BIOS or generated on the fly when a disc is logged in.

The reserved tracks will contain an image of CP/M 2.2, used when the system is rebooted. Discs can be formatted as data only discs, in which case they have no system tracks and cannot be used to reboot the system.

CP/M 2.2 directory

The CP/M 2.2 directory has only one type of entry:

```
UU F1 F2 F3 F4 F5 F6 F7 F8 T1 T2 T3 EX S1 S2 RC   .FILENAMETYP....
AL AL AL AL AL AL AL AL AL AL AL AL AL AL AL AL   ................

UU = User number. 0-15 (on some systems, 0-31). The user number allows multiple
    files of the same name to coexist on the disc.
     User number = 0E5h => File deleted
Fn - filename
Tn - filetype. The characters used for these are 7-bit ASCII.
       The top bit of T1 (often referred to as T1') is set if the file is
     read-only.
       T2' is set if the file is a system file (this corresponds to "hidden" on
     other systems).
EX = Extent counter, low byte - takes values from 0-31
S2 = Extent counter, high byte.
```

```
      An extent is the portion of a file controlled by one directory entry.
    If a file takes up more blocks than can be listed in one directory entry,
    it is given multiple entries, distinguished by their EX and S2 bytes. The
    formula is: Entry number = ((32*S2)+EX) / (exm+1) where exm is the
    extent mask value from the Disc Parameter Block.

S1 - reserved, set to 0.
RC - Number of records (1 record=128 bytes) used in this extent, low byte.
    The total number of records used in this extent is

    (EX & exm) * 128 + RC

    If RC is 80h, this extent is full and there may be another one on the disc.
    File lengths are only saved to the nearest 128 bytes.

AL - Allocation. Each AL is the number of a block on the disc. If an AL
    number is zero, that section of the file has no storage allocated to it
    (ie it does not exist). For example, a 3k file might have allocation
    5,6,8,0,0.... - the first 1k is in block 5, the second in block 6, the
    third in block 8.
     AL numbers can either be 8-bit (if there are fewer than 256 blocks on the
    disc) or 16-bit (stored low byte first).
```

Date stamps

Some compatible 3rd-party BDOSes (such as Z80DOS and DOS+) implement date stamping. Unfortunately the date stamp format they use is different from that used by CP/M 3.

Every fourth entry of a date-stamped directory will contain stamps for the preceding three entries:

```
21 00 C1 C1 M1 M1 M1 M1 A1 A1 A1 A1 C2 C2 M2 M2    !...............
M2 M2 A2 A2 A2 A2 C3 C3 M3 M3 M3 M3 A3 A3 A3 A3    ................

C1 = File 1 Create date
M1 = File 1 Modify date/time
A1 = File 1 Access date/time
C2 = File 2 Create date
M2 = File 2 Modify date/time
A2 = File 2 Access date/time
C3 = File 3 Create date
M3 = File 3 Modify date/time
A3 = File 3 Access date/time

The format of a date/time entry is:

        DW      day     ;Julian day number, stored low byte first.
                        ;Day 1 = 1 Jan 1978.
        DB      hour    ;BCD hour, eg 13h => 13:xx
        DB      min     ;BCD minute
```

Back to the formats listing

Back to the CP/M archive list

CP/M 3.1 disc formats

CP/M 3.1 uses a very similar system to CP/M 2.2, but with even more formats supported. The disc statistics are stored in a parameter block (the DPB), which contains the following information:

```
        DEFW    spt     ;Number of 128-byte records per track
        DEFB    bsh     ;Block shift. 3 => 1k, 4 => 2k, 5 => 4k....
        DEFB    blm     ;Block mask. 7 => 1k, 0Fh => 2k, 1Fh => 4k...
        DEFB    exm     ;Extent mask, see later
        DEFW    dsm     ;(no. of blocks on the disc)-1
        DEFW    drm     ;(no. of directory entries)-1
        DEFB    al0     ;Directory allocation bitmap, first byte
        DEFB    al1     ;Directory allocation bitmap, second byte
        DEFW    cks     ;Checksum vector size, 0 or 8000h for a fixed disc.
                        ;No. directory entries/4, rounded up.
        DEFW    off     ;Offset, number of reserved tracks
        DEFB    psh     ;Physical sector shift, 0 => 128-byte sectors
                        ;1 => 256-byte sectors  2 => 512-byte sectors...
        DEFB    phm     ;Physical sector mask,  0 => 128-byte sectors
                        ;1 => 256-byte sectors, 3 => 512-byte sectors...

The directory allocation bitmap is interpreted as:

       al0             al1
b7b6b5b4b3b2b1b0 b7b6b5b4b3b2b1b0
 1 1 1 1 0 0 0 0  0 0 0 0 0 0 0 0

 - ie, in this example, the first 4 blocks of the disc contain the directory.
```

The DPB is not usually stored on disc. It is either hardwired into the BIOS, or generated on the fly.

CP/M 3.1 directory

The CP/M 3.1 directory has four types of entry:

Files:

```
0U F1 F2 F3 F4 F5 F6 F7 F8 T1 T2 T3 EX S1 S2 RC   .FILENAMETYP....
AL AL AL AL AL AL AL AL AL AL AL AL AL AL AL AL   ................

0U = User number. 0-15. The user number allows multiple files of the same name
    to coexist on the disc.
     User number = 0E5h => File deleted
Fn - filename
Tn - filetype. The characters used for these are 7-bit ASCII.
       The top bit of T1 (often referred to as T1') is set if the file is
     read-only.
       T2' is set if the file is a system file (this corresponds to "hidden" on
```

```
     other systems). System files with user number 0 can be read from any user
     number.
       T3' is set if the file has been backed up.
EX = Extent counter, low byte - takes values from 0-31
S2 = Extent counter, high byte.

      An extent is the portion of a file controlled by one directory entry.
    If a file takes up more blocks than can be listed in one directory entry,
    it is given multiple entries, distinguished by their EX and S2 bytes. The
    formula is: Entry number = ((32*S2)+EX) / (exm+1) where exm is the
    extent mask value from the Disc Parameter Block.

S1 - Last Record Byte Count
RC - Number of records (1 record=128 bytes) used in this extent, low byte.
    The total number of records used in this extent is

    (EX & exm) * 128 + RC

    If RC is 80h, this extent is full and there may be another one on the disc.
    File lengths are optionally saved exactly (using the S1 byte) but this
    system is hardly ever used.
AL - Allocation. Each AL is the number of a block on the disc. If an AL
    number is zero, that section of the file has no storage allocated to it
    (ie it does not exist). For example, a 3k file might have allocation
    5,6,8,0,0.... - the first 1k is in block 5, the second in block 6, the
    third in block 8.
     AL numbers can either be 8-bit (if there are fewer than 256 blocks on the
    disc) or 16-bit (stored low byte first).
```

Disc label

```
20 F1 F2 F3 F4 F5 F6 F7 F8 T1 T2 T3 LB PB RR RR     LABENAMETYP....
P1 P2 P3 P4 P5 P6 P7 P8 D1 D1 D1 D1 D2 D2 D2 D2    ................

20h - Characteristic number of a disc label
F1-F8, T1-T3 - Label name, 7-bit ASCII
LB - Label byte. Bit 0 set => Label exists
                 Bit 4 set => Time stamp on create --+
                 Bit 5 set => Time stamp on update   +--These 2 are mutually
                 Bit 6 set => Time stamp on access --+   exclusive
                 Bit 7 set => Password protection enabled
PB - Used to decode the label password
RR - Reserved, set to zero.
P1-P8 - password, rather feebly encrypted.
D1 - Label create datestamp
D2 - Label update datestamp
```

Date stamps

If date stamps are in use, then every fourth directory entry will be a date stamp entry, containing stamps for the preceding three entries.

```
21 D1 D1 D1 D1 D2 D2 D2 D2 M1 00 D3 D3 D3 D3 D4    !...............
D4 D4 D4 M2 00 D5 D5 D5 D5 D6 D6 D6 D6 M3 00 00    ................

21h - Characteristic number of a date stamp.
D1  - File 1 create OR access date
D2  - File 1 update date
D3  - File 2 create OR access date
```

```
D4  - File 2 update date
D5  - File 3 create OR access date
D6  - File 3 update date
M1  - File 1 password mode
M2  - File 2 password mode
M3  - File 3 password mode
00  - Reserved.

The format of a date stamp is:

        DW      day     ;Julian day number, stored low byte first.
                        ;Day 1 = 1 Jan 1978.
        DB      hour    ;BCD hour, eg 13h => 13:xx
        DB      min     ;BCD minute
```

Password control

```
1U F1 F2 F3 F4 F5 F6 F7 F8 T1 T2 T3 PM PB RR RR   .FILENAMETYP....
P1 P2 P3 P4 P5 P6 P7 P8 RR RR RR RR RR RR RR RR   ................

1U = 16+User number (ie 16-31). The user number will be the number of
    the file to which the password belongs.
F1-F8 - Filename of the file to which the password belongs
T1-T3 - Filetype of the file to which the password belongs
PM    - Password mode byte
         Bit 7 set => Password required to read from file
         Bit 6 set => Password required to write to file
         Bit 5 set => Password required to delete file
PB    - Used to decode the password
P1-P8 - The password, rather feebly encrypted.
RR    - Reserved, set to 0.
```

Password encryption system

This system is extremely simple:

- When making the password, add all 8 bytes together (packing with spaces if necessary). This becomes PB (the decode byte). XOR each byte with PB and store them backwards in the directory (ie the last byte becomes P1).
- To decode the password, XOR PB with the 8 bytes of the password and read it off backwards.

[Back to the formats listing]

[Back to the CP/M archive list]