Partition tables on PCs

by **Hale Landis** (landis@sugs.tware.com)

PARTITION SECTOR/RECORD/TABLE BASICS

FDISK creates all partition records (sectors). The primary purpose of a partition record is to hold a partition table. The rules for how FDISK works are unwritten but so far most FDISK programs (DOS, OS/2, WinNT, etc) seem to follow the same basic idea.

First, all partition table records (sectors) have the same format. This includes the partition table record at cylinder 0, head 0, sector 1 -- what is known as the Master Boot Record (MBR). The last 66 bytes of a partition table record contain a partition table and a 2 byte signature. The first 446 bytes of these sectors usually contain a program but only the program in the MBR is ever executed (so extended partition table records could contain something other than a program in the first 466 bytes). See "How It Works -- The Master Boot Record".

Second, extended partitions are "nested" inside one another and extended partition table records form a "linked list". I will attempt to show this in a diagram below.

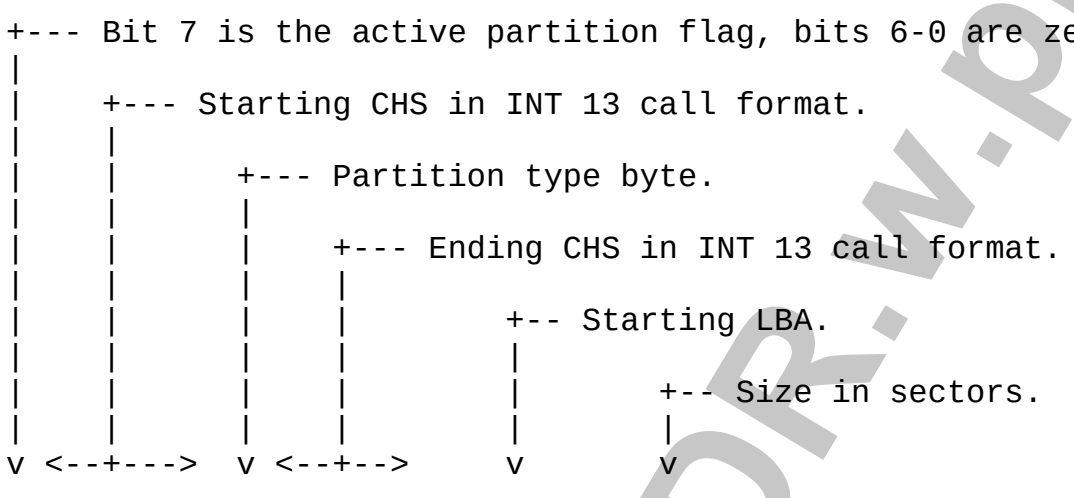
PARTITION TABLE ENTRY FORMAT

Each partition table entry is 16 bytes and contains things like the start and end location of a partition in CHS, the start in LBA, the size in sectors, the partition "type" and the "active" flag. Warning: older versions of FDISK may compute incorrect LBA or size values. And note: When your computer boots itself, only the CHS fields of the partition table entries are used (another reason LBA doesn't solve the >528MB problem). The CHS fields in the partition tables are in L-CHS format -- see "How It Works -- CHS Translation".

There is no central clearing house to assign the codes used in the one byte "type" field. But codes are assigned (or used) to define most every type of file system that anyone has ever implemented on the x86 PC: 12-bit FAT, 16-bit FAT, HPFS, NTFS, etc. Plus, an extended partition also has a unique type code.

Note: I know of no complete list of all the type codes that have been used to date. However, I try to include such a list in a future version of this document.

The 16 bytes of a partition table entry are used as follows:



0	1	2	3	4	5	6	7	8 9 A B	C D E F
DH	DL	CH	CL	TB	DL	CH	CL	LBA.....	SIZE....
80	01	01	00	06	0e	be	94	3e000000	0c610900
00	00	81	95	05	0e	fe	7d	4a610900	724e0300
00	00	00	00	00	00	00	00	00000000	00000000
00	00	00	00	00	00	00	00	00000000	00000000

1st entry

2nd entry

3rd entry

4th entry

Bytes 0-3 are used by the small program in the Master Boot Record to read the first sector of an active partition into memory. The DH, DL, CH and CL above show which x86 register is loaded when the MBR program calls INT 13H AH=02H to read the active partition's boot sector. See "How It Works -- Master Boot Record".

These entries define the following partitions:

1) The first partition, a primary partition DOS FAT, starts at CHS 0H,1H,1H (LBA 3EH) and ends at CHS 294H,EH,3EH with a size of 9610CH sectors.

2) The second partition, an extended partition, starts at CHS 295H,0H,1H (LBA 9614AH) and ends at CHS 37DH,EH,3EH with a size of 34E72H sectors.

3)The third and fourth table entries are unused.

PARTITION TABLE RULES

Keep in mind that there are NO written rules and NO industry standards on how FDISK should work but here are some basic rules that seem to be followed by most versions of FDISK:

1) In the MBR there can be 0-4 "primary" partitions, OR, 0-3 primary partitions and 0-1 extended partition entry.

2) In an extended partition there can be 0-1 "secondary" partition entries and 0-1 extended partition entries.

3) Only 1 primary partition in the MBR can be marked "active" at any given time.

4) In most versions of FDISK, the first sector of a partition will be aligned such that it is at head 0, sector 1 of a cylinder. This means that there may be unused sectors on the track(s) prior to the first sector of a partition and that there may be unused sectors following a partition table sector.

   For example, most new versions of FDISK start the first partition (primary or extended) at cylinder 0, head 1, sector 0. This leaves the sectors at cylinder 0, head 0, sectors 2...n as unused sectors. This same layout may be seen on the first track of an extended partition. See example 2 below.

   Also note that software drivers like Ontrack's Disk Manager depend on these unused sectors because these drivers will "hide" their code there (in cylinder 0, head 0, sectors 2...n). This is also a good place for boot sector virus programs to hang out.

5) The partition table entries (slots) can be used in any order. Some versions of FDISK fill the table from the bottom up and some versions of FDISK fill the table from the top down. Deleting a partition can leave an unused entry (slot) in the middle of a table.

6) And then there is the "hack" that some newer OS's (OS/2 and Linux) use in order to place a partition spanning or passed cylinder 1024 on a system that does not have a CHS translating BIOS. These systems create a partition table entry with the partition's starting and ending CHS information set to all FFH. The starting and ending LBA information is used to describe the location of the partition. The LBA can be converted back to a CHS -- most likely a CHS with more than 1024 cylinders. Since such a CHS can't be used by the system BIOS, these partitions can not be booted or accessed until the OS's kernel and hard disk device drivers are loaded. It is not known if the systems using this "hack" follow the same rules for the creation of these type of partitions.

There are NO written rules as to how an OS scans the partition table entries so each OS can have a different method. For DOS, this means that different versions could assign different drive letters to the same FAT file system partitions.

PARTITION NESTING

What do I mean when I say the partitions are "nested" within each other? Lets look at this example:

```
     M = Master Boot Record (and any unused sectors
         on the same track)
     E = Extended partition record (and any unused sectors
         on the same track)
   pri = a primary partition (first sector is a "boot" sector)
   sec = a secondary partition (first sector is a "boot" sector)


  |<----------------the entire disk-------------->|
  |                                               |
  |M<pri>                                         |
  |                                               |
  |      E<sec><---rest of 1st ext part---------->|
  |                                               |
  |            E<sec><---rest of 2nd ext part---->|
```

The first extended partition is described in the MBR and it occupies the entire disk following the primary partition. The second extended partition is described in the first extended partition record and it occupies the entire disk following the first secondary partition.

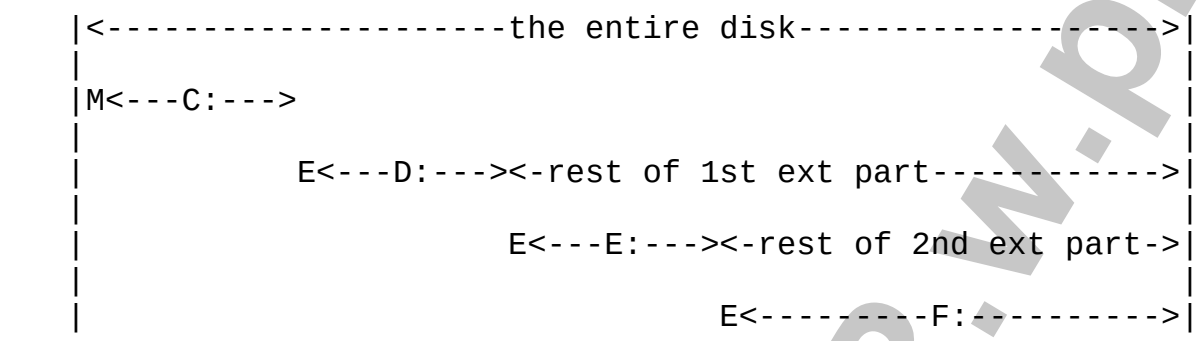
PARTITION TABLE LINKING

What do I mean when I say the partition records (tables) form a "linked" list? This means that the MBR has an entry that describes (points to) the first extended partition, the first extended partition table has an entry that describes (points to) the second extended partition table, and so on. There is, in theory, no limited to out long this linked list is. When you ask FDISK to show the DOS "logical drives" it scans the linked list looking for all of the DOS FAT type partitions that may exist. Remember that in an extended partition table, only two entries of the four can be used (rule 2 above).

And one more thing... Within a partition, the layout of the file system data varies greatly. However, the first sector of a partition is expected to be a "boot" sector. A DOS FAT file system has: a boot sector, first FAT sectors, second FAT sectors, root directory sectors and finally the file data area. See "How It Works -- OS2 Boot Sector".

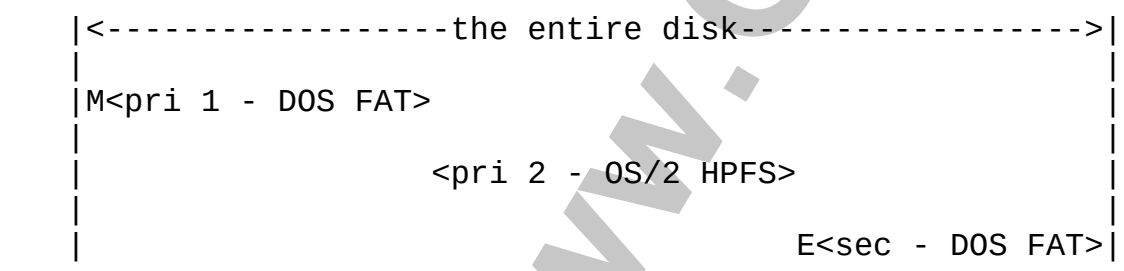
EXAMPLE 1

A disk containing four DOS FAT partitions (C, D, E and F):

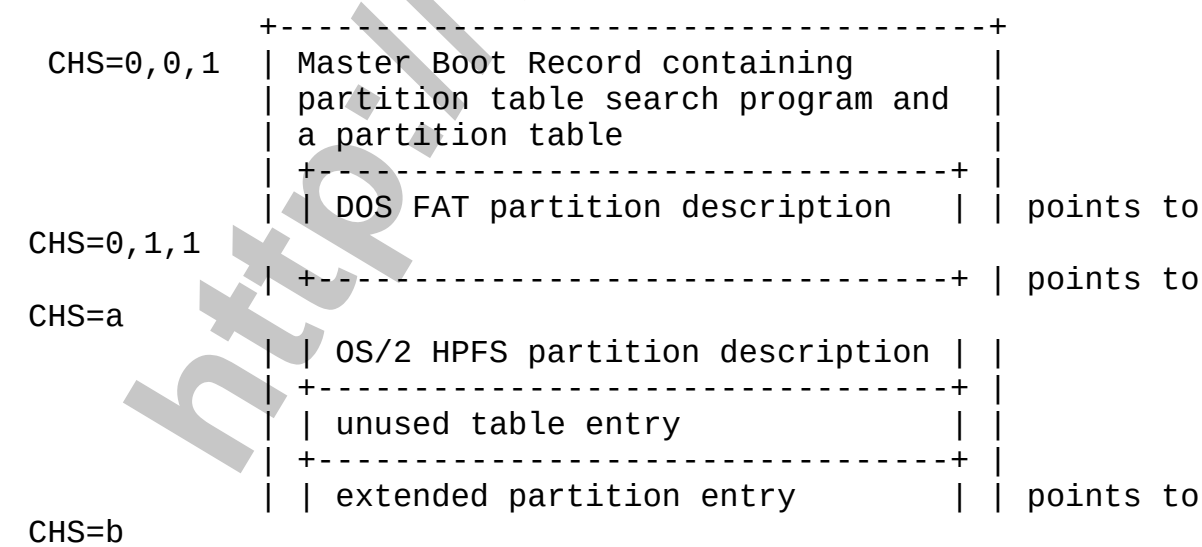


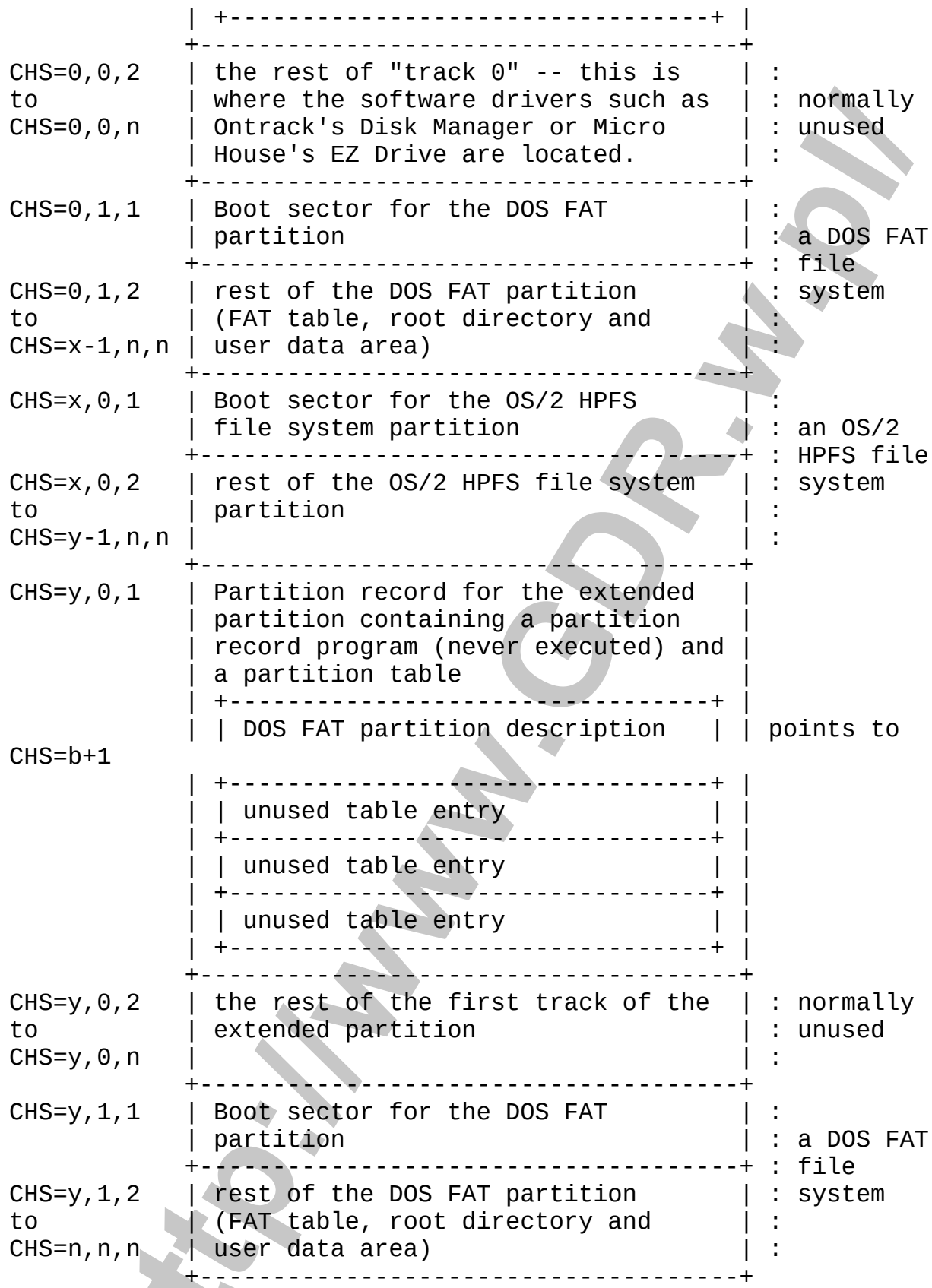
EXAMPLE 2

So here is an example of a disk with two primary partitions, one DOS FAT and one OS/2 HPFS, plus an extended partition with another DOS FAT:



Or in more detail ('n' is the highest cylinder, head or sector number number allowed in the indicated field of the CHS)...





EXAMPLE 3

Here is a partition record from an extended partition (the first sector of an extended partition). Note that it contains no program code. It contains only the partition table and the signature data.

```
OFFSET 0 1 2 3  4 5 6 7  8 9 A B  C D E F  *0123456789ABCDEF*
000000 00000000 00000000 00000000 00000000 *................*
000010 TO 0001af SAME AS ABOVE
0001b0 00000000 00000000 00000000 00000001 *................*
0001c0 8195060e fe7d3e00 0000344e 03000000 *.....}>...4N....*
0001d0 00000000 00000000 00000000 00000000 *................*
0001e0 00000000 00000000 00000000 00000000 *................*
0001f0 00000000 00000000 00000000 000055aa *..............U.*
```

NOTES

Thanks to yue@heron.Stanford.EDU (Kenneth C. Yue) for pointing out that in V0 of this document I did not properly describe the unused sectors normally found around the partition table sectors.