

# Partition Notes

## Svend Olaf Mikkelsen

### Harddisk

- A harddisk is accessed in units of sectors. One sector is 512 bytes. The sectors are addressed using Logical Block Addresses (LBA), or using Cylinder, Head, Sector (CHS) addresses.
- LBA addresses are numbered from 0 up to the number of sectors on the disk minus 1.
- For CHS addresses, cylinder is the most significant unit, and sector the least significant. Cylinders and heads are numbered from 0, and sectors from 1.
- To use CHS addresses a disk geometry has to be assumed. For current harddisks, the geometry is not related to the actual disk design, but is used for convenience only. Most often the geometry is supplied by the BIOS.
- The first sector of a harddisk is called the Master Boot Record (MBR). It has the LBA address 0, and the CHS address 0/0/1, i.e. cylinder 0, head 0, sector 1.

### Partitions

- A partition is a contiguous area of the disk. The partition locations are defined in partition tables. The main partition table is located in the MBR after a tiny boot program. It consists of up to four entries.
- Partitions defined in the MBR partition table are called primary partitions.
- To provide room for more than 4 partitions on a disk, one of the primary partitions can be an extended partition. This is a container partition, which can contain a number of logical partitions.
- The locations of the logical partitions are defined in extended partition tables. The first extended partition table is in the very first sector of the extended partition.
- Each extended partition table can have two entries.
- The first entry defines the location of the belonging logical partition, and the second entry points to the next extended partition table.
- If there is no logical partition at the beginning of the extended partition, the first logical partition table will contain only the link to next extended partition table.
- The partition table entries contain the locations of the partitions in two ways: The CHS address of the first and last sector in the partition, and using LBA addresses. Since only 10 bits are available for the cylinder number, the highest cylinder number that can be entered is 1023. For this reason only the LBA addresses are used, but the CHS numbers should be filled with some standard numbers, as described in the following.
- By convention, a partition, primary or logical, should always end at a cylinder boundary. For a 255 heads, 63 sectors disk, this as example can be cylinder 399, head 254, sector 63. If this rule is followed, the extended partition tables will be in the first sector of a cylinder. Well, can be, since some partitioning tools have some ideas of their own.

A partition table entry contains:

## Partition Notes

Svend Olaf Mikkelsen

- The ID (type). This is a number between 1 and 255, which defines the type of the partition. Conventionally the number is written in hexadecimal. For a FAT32 partition the ID is 0B or 0C hexadecimal, which will be 11 or 12 in decimal.
- The active flag. This is 0 for not active, and hex 80 (decimal 128) for active. Only one primary partition should have the active flag set.
- Begin CHS address. If the cylinder number is higher than 1023, 1023 is used. Some partitioning tools will use CHS 1023/(heads - 1)/sectors. For a disk with a 255 heads, 63 sectors geometry, this will be CHS 1023/254/63.
- End CHS address. As for begin CHS address.
- Relative sectors. For primary partitions, and logical partitions (not for links to next extended partition table), this is the offset from the partition table sector, which is the number of sectors from the partition table to the beginning of the partition. For link to next extended partition table, the relative entry is relative to the beginning of the entire extended partition. (This is not difficult at all, but the most difficult part to understand in partition tables, which are simple).
- Number of sectors. This is the number of sectors in the partition.

Some often used partition ID's:

06 Primary FAT16 partition ending cylinder 1023 or lower, or logical FAT16 partition.

0E Primary FAT16 partition ending cylinder 1024 or later.

0B Primary FAT32 partition ending cylinder 1023 or lower, or logical FAT32 partition.

0C Primary FAT32 partition ending cylinder 1024 or later.

07 NTFS or HPFS.

83 Linux partition, ext2, ext3 or reiserfs.

82 Linux swap partition.

0F Extended partition ending cylinder 1024 or later.

05 Extended partition ending cylinder 1023 or below. Also used for link to next extended partition table.

85 Extended partition, known by Linux only.

Let us look at an example partition table:

```
Findpart, version 4.32.  
Copyright Svend Olaf Mikkelsen, 2003.
```

```
OS: DOS 8.00
```

```
Disk: 1 Cylinders: 2489 Heads: 255 Sectors: 63 MB: 19524
```

```
Partitions according to partition tables on first harddisk:
```

```
-PCyl N ID -----Rel -----Num ---MB -Start CHS- --End CHS-- BS CHS  
  0 1*0C      63 23615487 11530   0   1   1 1469*254 63 OK  OK  
  0 2 0F 23615550 16370235  7993 1470*  0   1   1 2488*254 63      OK  
  
1470 1 82      63  530082   258 1470   1   1 1502 254 63      OK  
1470 2 05  530145 3550365 1733 1503   0   1 1723 254 63      OK  
  
 1503 1 83      63 3550302 1733 1503#  1   1 1723 254 63 OK  OK
```

## Partition Notes

Svend Olaf Mikkelsen

```
1503 2 05 4080510 12289725 6000 1724 0 1 2488 254 63 OK
1724 1 0B 63 12289662 6000 1724 1 1 2488 254 63 OK OK
```

- Each line is a partition table entry. The lines with Pcyl (partition table cylinder) 0 are from the MBR partition table.
- The first line defines a primary, active (the star before ID) FAT32 partition from CHS 0/1/1 to 1469/254/63. The relative entry is 63, which is the distance in sectors from the MBR partition table in sector 0 (included) to the beginning of the partition. The number of sectors in the partition is 23615487.
- The second line defines an extended partition from CHS 1470/0/1 to 2488/254/63. Since cylinders and heads are numbered from 0, this is the very last sector of the disk.
- The lines with 1470 in the Pcyl field are from the extended partition table in CHS 1470/0/1.
- The first entry at cylinder 1470 defines a Linux swap partition. The 63 in the relative field is the number of sectors from CHS 1470/0/1 to the beginning of the swap partition at CHS 1470/1/1.
- The second entry at cylinder 1470 links to the next extended partition table in CHS 1503/0/1. The relative field in the link contains the number of sectors between CHS 1470/0/1 and 1503/0/1. Note that 530145 is  $(1503 - 1470) * 255 * 63$ . The cylinder difference multiplied with the number of heads and number of sectors from the disk geometry.
- The first entry at cylinder 1503 defines a Linux partition.
- In the second entry at cylinder 1503, the relative field in the link to next extended partition table in CHS 1724/0/1 is calculated relative to the beginning of the extended partition, 4080510 is  $(1724 - 1470) * 255 * 63$ .
- The last extended partition table at CHS 1724 defines a FAT32 partition.
- If the example partition tables were lost, but the partition contents OK, and the partition locations known, the Editpart commands to write the partition tables to the disk would be:

```
editpart 1 0 1 * 0C 0 1 1 1469 254 63 0 2489 255 63 26
editpart 1 0 2 - 0F 1470 0 1 2488 254 63 0 2489 255 63 26

editpart 1 1470 1 - 82 1470 1 1 1502 254 63 1470 2489 255 63 26
editpart 1 1470 2 - 05 1503 0 1 1723 254 63 1470 2489 255 63 26

editpart 1 1503 1 - 83 1503 1 1 1723 254 63 1503 2489 255 63 26
editpart 1 1503 2 - 05 1724 0 1 2488 254 63 1470 2489 255 63 26

editpart 1 1724 1 - 0B 1724 1 1 2488 254 63 1724 2489 255 63 26
```

The last four numbers in each line are for check only. It is the disk geometry, and Editpart version number.

No relative or number of sectors values are entered. They are calculated by Editpart. Also cylinder numbers above 1023 is converted to the correct field values by Editpart.

Note how the "offset cylinder" parameter after the End CHS address is the same as the partition table cylinder for primary and logical partitions, and 1470 for link to next extended partition table.

## Partition Notes

Svend Olaf Mikkelsen

This was a simple example. If the sector that contains the partition table does not contain a partition table signature in the last two bytes, the "force" parameter may be needed. The force parameter should be used with care, since if the partition table location is wrong, partitions may be damaged.

- Also if invalid entries exist in not used locations, it may be needed to delete them. ID 00 in the Editpart command line deletes the entry.
- Different standards exist for cylinder entries above 1023. In the example partition table, the \* after 1469 indicates that the CHS field values are 1023/0/1. The # after 1503 indicates that the field values are CHS 1023/254/63. Since the CHS addresses inside extended partitions type 0F are not used, it does not matter.
- The first sector in each partition is called the boot sector, no matter if the sector is used during boot or not.

### DOS/Windows 95/98/ME Partition Issues

- If an extended partition ends cylinder 1024 or later, the type of the extended partition must be type 0F.
- If it is not, data damage will occur. Links to next extended partition tables should still be type 05.
- If the last partition in any extended partition is not a FAT partition, each disk cannot contain more than one primary FAT partition. If this rule is not followed, data damage can occur.
- If the last partition in any extended partition is not a FAT partition, all extended partitions should be of the same type. If one disk has an extended partition type 0F, all extended partition on all disks in the system must be type 0F. Damage can occur if this rule is not followed.
- If the extended partition is type 85 (Linux Extended), the extended partition does not count as an extended partition in these matters.
- Disks larger than 32 GB and 128 GB cannot be considered reliable in Windows 95/98/ME, due to the Windows (or Windows disk driver) 32 GB and 128 GB problems. The 32 GB problem can be reliably detected using the GB32 program. The only situation I know of, where disks larger than 128 GB should be in the system, is if the chipset is Intel, and a recent version of Intel Application Accelerator is installed.

### The Windows 2000/XP 128 GB Problem

- Disks with partitions ending more than 128 GB into the disk should not be in a Windows 2000 system without service pack 4 installed \*and\* this registry setting made:

**Windows Registry Editor Version 5.00**

```
[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\atapi\Parameters]
"EnableBigLba"=dword:00000001
```

- You can save the lines in a file with extension .reg and double click the file.
- Disks larger than 128 GB should not be in a Windows XP system without service pack 1.

## Partition Notes

Svend Olaf Mikkelsen

- The 128 GB problem is present if a disk larger than 128 GB is listed as 131069 MB by Findpart.
- If these rules are not followed, all data on the disk will be lost.

### Hidden Partitions

- Hidden partitions are partitions that are hidden for the operating system by changing the ID in the partition tables to other values, not known by the operating system. Note that if the last logical partition in an extended partition is a hidden FAT partition, it counts as a non-FAT partition in the DOS issues mentioned.
- Windows 2000 does not respect hidden partition types, while Windows XP seems to do. For Linux the ID of the partition does not matter, so if a partition should not be accessed, it should not be mounted.
- FAT and NTFS partitions are hidden by changing the first 0 in the hexadecimal ID to 1. As example ID 1C is hidden primary FAT32 ending cylinder 1024 or later.
- If the purpose is to protect a partition from further damage, it often can be safer to use another start location for the partition, if possible. As example the start CHS address for the first primary partition can be CHS 0/0/2 instead of CHS 0/1/1.

### Fdisk Warnings

- If partitions are lost, the DOS/Windows 95/98/ME Fdisk create partition screen should not be entered.
- Just entering the screen will write sectors containing ascii 246 (hex F6) to large areas in the beginning of free partition space. If this happens, most data can be recovered anyway, but it will be more difficult.
- Attempting to recover a partition by creating a new using standard partitioning tools may cause more damage than Fdisk, since FAT or other data structures may be created at the same time as the partition table entry.
- Do not use DOS Fdisk if non FAT logical partitions are present.

### Findpart Output

The output for each disk is divided into 3 parts. The search result, the FAT search result, and the partition tables for the disk.

Abbreviations used in the output:

B Based on boot sector.

BU Based on backup boot sector (meaning that the boot sector may be missing or damaged, or the partition is obsolete).

00 The boot sector is filled with ascii 0 characters.

F6 The boot sector is filled with ascii 246 characters.

R The Reiser file system.

3 Ext3 file system

H HPFS (OS/2 file system).

## Partition Notes

Svend Olaf Mikkelsen

B5 A Linux boot sector with boot signature.  
NR Boot sector not readable. Bad sector, or outside disk area.  
NB Nota Bene.  
R0 The boot sector relative field is relative to sector 0. This is OK  
OK Okay.

- !! An extended partition ending cylinder 1024 or later is type 05.
- \* In front of ID: Active flag set.  
After cylinder number: The actual entry is 1023.  
After Cluster KB in FAT output: Based on boot sector.  
After Root size/Cluster number in FAT output: Based on boot sector.
- # After cylinder number: The actual entry is 1023/(heads - 1)/sectors.  
After Cluster KB in FAT output: Not confirmed in boot sector.  
After Root size/Cluster number in FAT output: Not confirmed in boot sector.
- ? In front of ID: Active flag not hexadecimal 00 or 80.  
After partition line: Ending after end of disk.

If you look in the "N" field of the search output, you may notice that some lines has a number in this field, while other are empty.

If there is a number, an extended partition table is present, and the finding is based on this entry. If the same line contains an OK or R0 in the BS (boot sector) field, the presence of the partition was confirmed. If there is not, the partition may be damaged, obsolete, or in some way differ from the partition table entry.

If there is a difference between the extended partition table entry, and the belonging logical partition, you may see another line for the same partition location. One explanation can be that the actual partition space is smaller than according to the partition table entry. This usually is as it should be, since a partition table entry always should define the partition as ending at a cylinder boundary.

In the FAT part of the output, you may see some "Second FAT not found" lines. These are most often from obsolete or false positive FAT findings. The message also can be present for a FAT that is OK, since Findpart by default does only detect FAT smaller than a certain size. To verify if a FAT partition is OK, the Chsdir program can be used with the FAT1 and FAT2 options.

### Linux

- Errors in Linux partitioning tools is a common cause of data loss. It is not possible to install RedHat Linux without damaging good partition tables, although in many cases the partition tables will still be valid. An exception may be that the tables will not be damaged if the operating system is installed to primary partitions only.
- In some cases, if the partition tables are not standard, but valid, the damage will not occur until another partitioning tool attempts to interpret the tables.
- This may work: Install Linux when with no extended partition present. Let Linux create the extended partition and extended partition tables. Make sure that the type of the extended partition is not 05 if the extended partition ends cylinder 1024 or later (1025 numbered from 1). The type can be changed to 0F or 85 (Linux extended) using the Linux fdisk t command.
- Note that Linux fdisk numbers cylinders from 1.

## Partition Notes

Svend Olaf Mikkelsen

### Findpart Options

- Findpart for Windows includes the functionality of other programs as mentioned on the utilities pages.
- Findpart for DOS includes the functionality of Editpart, Findfat, Finddir, Getsect, Putsect and Setsize.
- For usage type 'findpart' followed by the utility name. For editing commands the findpart environment variable must be set to 'edit'. Note that for Editpart the word 'editpart' should not be included in the Findpart command line.
- Some options, which are not mentioned in the usage screen:
- In some cases partitions may have been created assuming another disk geometry than reported by the BIOS. This will often be the case if the partitions are made using Linux tools. To search simulating another disk geometry, do as example:

```
findpart 2 heads 16 fp.txt
```

If the Ontrack Disk Manager program has been used on the disk, all partition structures may be 63 sectors longer into the disk than usual. To compensate for this, the keyword 'dm' can be used:

```
findpart 2 dm fp-a.txt
```

To make a listing of IDE (ATA) disks in the system, do in pure DOS:

```
findpart ide fp-b.txt
```

To include disks on other ports than the motherboard main IDE connectors:

```
findpart ide ext fp-c.txt
```

To not use the BIOS for disk access in DOS, the abbreviations pm, ps, sm and ss can be used as disk number for primary master etcetera. This option can be used if the BIOS does not support the disk correctly. Also available are tm, ts, qm, qs, 5m, 5s, 6m and 6s. Example:

```
findpart pm fp-d.txt
```

### FindNTFS

So far just one additional note. The "Trees" in the output summary. If a partition is fully recognized, there will only be 1 tree, the root. If parts of the directory structure is lost, there can be more than one tree in the search result. The explanation can be that the partition is internally damaged, or wrong search parameters were used.