FDISK68 / FDISK80
ver. 1.50-16 of 28-May-2015
(1.0-beta)

FDISKxx is a “work-similar” program for the N8VEM systems running the UNA, RomWBW, or Mini-M68k BIOS. It functions to partition hard disk media, such as Compact Flash cards and Secure Digital cards to operate in a manner that provides three-way compatibility among the three CP/M operating systems: RomWBW CP/M 2.2, UNA CP/M 2.2, and Mini-M68K CP/M-68.

Slices: Beginning with RomWBW CP/M 2.2, hard disk media used no partition table, but simply sliced up the lowest sectors on a disk into up to 8 x 8Mb CP/M file systems. Each file system also had a 256 sector boot area at its head. In HEX, each slice occupied 0x4100 sectors, 0x100 sectors at the head as a boot area, and 0x4000 == 16384 sectors of actual CP/M directory and files. 16384 x 512 bytes = 8Mb of disk space. UNA CP/M can use this system of 8Mb slices starting at sector 0, and when doing so, is completely compatible with the RomWBW file system.

Partitions: CP/M-68 for the Motorola 68000 CPU's allows file systems of up to 512Mb. Thus, the Mini-M68k implementation was designed to use the DOS partition table to specify partitions containing CP/M file systems of any size, up to the maximum. When the partitions size is exactly 8Mb, all of the file system parameters (allocation size, number of director entries, &c.) are exactly the same as the Z80 8Mb slices. Hence, if a CP/M-68 partition (of type “CP/M” == 0x52) of exactly 16384 sectors exactly overlays one of the RomWBW/UNA data slices, the three systems can read and write files on that particular slice/partition.

CF and SD cards: These media are widespread nowadays and are normally distributed with the IBM/MS FAT file system, usually FAT16 or FAT32. Luckily, all off-the-shelf media contain a DOS partition table, with the entire device allocated to DOS primary partition 1 (counting 1..4). Commercial operating systems use only partition 1, and do not look at the other partitions, even if the device is re-partitioned. By repartitioning one of these cards, leaving space at the beginning of the “disk” for Slices, and re-allocating partition 1 to be a FAT file system on higher sectors of the disk, such a device is usable on Windows, MAC, and Linux, needing only to be have the proper FAT file system formatted by the commercial system.

FAT12/FAT16: Both the UNA BIOS and the Mini-M68K BIOS can read the two older versions of the FAT file specification. This speeded OS development for UNA and and Mini-M68K operating systems by allowing them to be written to CF/SD media on the cross-development system, and immediately booted on the target SBC. Up to this time, only Linux 'fdisk,' although cumbersome, was able to partition these media. The creation of CP/M “slice overlay” partitions is especially tedious, since proper alignment is absolutely necessary.

FDISKxx: The 'fdisk' tool is distributed as two executable files: FDISK80.COM for UNA or RomWBW CP/M or standalone operation and FDISK68.OUT for standalone operation on the Mini-M68K. Standalone operation means that the file can be executed from a FAT file system on removable media. CP/M operation means that it can be executed from within CP/M and, as such, may be distributed as part of a CP/M volume, specifically, a ROMdisk image.

Program Usage

Commands: All FDISKxx commands are single letters, the first letter of a mnemonic for the operation. The entirety of FDISKxx is case-insensitive. For clarity, this documentation will use both upper and lower case, but use of the shift key within the program is not necessary.

Numbers: Except when a command specifically asks for 'hex' input, numbers may be typed in radix 8, 10, or 16. The usual C-programming language conventions apply. Hexadecimal numbers begin with “0x” (zero, ex); e.g., 0x100, 0xFFFF, 0x157ad. Octal numbers begin with a leading “0” (zero); e.g., 0400, 0177777, 0253655. Decimal numbers begin with “1..9”; e.g., 256, 65535, 87981.

When specifying locations on the disk, and partition sizes, sometimes larger numbers are needed. In these cases, a multiplier suffix may be attached to a number, as defined above; viz., 1M == 1024k. There are three usual multiplier suffixes: “G”, “M”, “K” having the usual meaning: Giga (1024*1024*1024), Mega (1024*1024), and Kilo (1024). There is also the odd suffix, “U”, meaning Unit (1), which has use with hexadecimal values, described below.

1G = 1024M, 1M = 1024K, 1K = 1024U = 1024.

Disk locations, and partition sizes, may also be specified in terms of sectors, tracks/heads, or cylinders, and sizes in terms of bytes. This gives rise to four more numeric suffixes; namely, “Y”, “T”, “S”, “B”, standing for: cYlinder, Track, Sector, and Byte. Further, disk locations bear no prefix, but partition sizes bear the prefix “+”.

100y disk location at cylinder 100
100 disk location at 100; the default unit being cylinders
0x100s disk location at sector 256
1T disk location at track 1
512b == 1s sector 1 location = byte 512 location (oddball usage of B)
+8Mb disk size of 8 megabytes (M is multiplier, B is byte unit)

Hint: S is the most commonly use location suffix, and B is the most commonly use size suffix. Note: 0x11B and 0x11uB are different. The U must be used with a hex number if the Byte size specifier is in use.

Geometry: Traditionally, hard disks use moving platters, moving heads, and were divided into cylinders, that which could be accessed without moving the arm(s) holding the heads, tracks (accessed by multiple heads), that which could be accessed in one disk rotation, and sectors, the addressable unit of conglomerate data, most often 512 bytes each. These three values, often abbreviated “C:H:S” for Cylinder/Head(track)/Sector were fixed by the disk hardware, and were the means of communicating a sector location to the disk hardware. CHS geometry is a legacy of the old disk structure. It has been made obsolete by the advent of intelligent controllers, which simply address sectors with a 28 or 48 bit address. This last is known as Logical Block Addressing, or LBA for short.

CF cards interface through an IDE interface, and a disk geometry, i.e., values for C, H, and S, are still assigned. However, LBA is easier to use, and all CF cards support LBA. SD cards

have abandoned CHS entirely, and are only addressable LBA. However, in aligning partitions, as is needed for CP/M-68 / CP/M 2.2 cross compatibility, CHS has its uses.

Commands: All commands are unique in the first letter and, as stated before, are case insensitive. Hence, “p”, “P”, “Print”, and “punt” are equivalent. If a command begins with an unrecognized letter, the advisory message “? for help” will be displayed.

Here are the commands in alphabetic order:

? -- see Help, below.

Activate – toggle the 'active' flag (“*”) on a partition, which marks the partition as bootable. If more than one partition is marked active at a time, a cautionary message is issued.

Delete – delete the specified partition. The command will prompt for the partition number to be deleted. Delete removes the partition specification from the partition table, and does not affect the partition specification of any of the remaining partitions. Optionally, the partition to be deleted may be specified as a parameter to the command; e.g., “delete 4” or “d 4”.

Geometry – management of disk geometry. The virtual geometry of the disk may be assigned using this command. First the number of sectors per head (track) will be prompted for. <CR> keeps the existing number. Then the number of tracks (heads) per cylinder will be prompted for. <CR> retains the existing number. Finally a summary of the disk geometry to be used is displayed. The CHS values in the partition table are updated if the new geometry is different from the old.

Help – will print a brief command help page. “?” is equivalent. A one-line description of all commands available from the command prompt is printed. These descriptions are just short reminders.

Initialize – wipe the entire partition table, including reserved slices, and start with a fully empty partition table specification. Disk geometry is not affected by this command. Initialize is equivalent to Deleting each partition individually and resizing the reserved slice count to zero.

List – print a brief list of some useful partition types. Since partition types range from 00 to FF (hex), the complete list is very long. About a dozen and a half of those of most interest to the Z80/Z180/MC68000 community are listed. This command is accessible from within the partition Type assignment command, as well as at the command level.

New – create a new partition. The command will first prompt for the partition number to be created. An empty slot must be specified else an error occurs. The second prompt is for the starting location of the partition in default units, usually Cylinders. Any unit of disk position may be specified, other than the default, by using one of the “Y”, “T”, or “S” number suffixes as described above. Cylinders is the most common unit used, and careful selection of disk geometry, described above, can result in particular alignment of the new partition. Third, the ending location is prompted for. Two possible specifications may be used. First the ending location may be specified as a number in the same fashion as the the starting location, and using any of the location overrides above. Or, the ending location may be specified as a partition size by prefixing the number with a “+” sign. The same overrides apply, and in

addition, the “B” (byte) size suffix may be used.

Special cases: cylinder 0, track 0, or sector 0, taken as the partition start, will be rounded to 1T to protect the master boot record (which contains the partition table). An ending location, or size, which extends beyond the end of the media will be truncated to the last sector on the disk.

Print – prints the current partition table and notice of any slices known to be reserved. The printed table lists, in order, the partition number, partition type in text format, if known, the active flag (“*”), partition type in hexadecimal, CHS start of the partition expressed in terms of the current geometry setting, CHS end of the partition in terms of the current geometry, the absolute LBA address of the partition, the LBA sector count, or size of the partition in sectors, and an approximate size in a compact form (in bytes).

Type – change the industry-wide type setting, a byte, of a partition. The type indication is specified in hexadecimal; no “0x” prefix is needed. The actual type of a partition actually depends on the actual contents of the partition. Different operating systems depend more or less on the type field in the partition table. However, it is a good idea to be as accurate as possible in getting this field correct.

Quit – exit FDISKxx without saving changes. Nothing is written back to the disk; all changes being worked on, or experimented with, are lost.

Reserve – reserve space on the lowest sectors of the disk for CP/M slices of 8Mb each, or modify the currently reserved number of slices. As described above, slices are 0x4100 sectors in size = 0x100s boot area + 0x4000s data area. If reserve is used before “New,” then newly created partitions will be unable to share any storage with the CP/M slices. See below for details of creating CP/M-68 partitions that overlay slices.

Unit – prompts for the disk unit to have its partition table edited. This command is automatically invoked when FDISKxx is brought up. Subsequent uses of “Unit” may be to refresh the original partition table, in case one does not like the layout one is currently editing.

Mini-M68K: the unit is specified as the disk letter assigned by the BIOS: “C:”, “D:”, &c.
UNA-BIOS: the unit is specified as the decimal number assigned by the BIOS: “1”, “2”, &c.
RomWBW: TBD (probably a hex number consisting of driver nibble / unit nibble.)

Zebra – is reserved for use in FDISKxx for debugging. It is specified as “Z#”, where # is a digit between 0 and 5. The higher the debug level, the more debugging output is produced by the program. The “Help” command will tell you whether the version of FDISKxx you are running has the capability of producing debugging output.

Examples

Creating slices for CP/M 2.2:

1. Execute FDISKxx, specifying the disk unit to be modified.
2. I – command: wipe the old partition table. Or, D-command: delete partition 1.
3. G – command: set the new geometry to S/H: 16, H/C: 16. This will fudge the cylinder size to 256s (0x100s).
4. R – command: reserve as many slices as you wish, up to 8.
5. N – command: partition 1, start: (use the default), size: specify by location of size, remembering that +xxMb can be larger than the media, and will be chopped back to the last sector on the disk.
6. T – command: set the type of partition 1 to (FAT12/FAT16/FAT32)
7. W – command: update the partition table on the media.

Format partition 1 on your host system. See the final comments below on multi-system usage.

Creating a CP/M 2.2 // CP/M-68 // Windows/MAC multi-platform disk from an new CF or SD card:

1. Execute FDISKxx, specifying the disk unit to be modified.
2. I – command: wipe the old partition table.
3. G – command: set the new geometry to S/H: 16, H/C: 16. This will fudge the cylinder size to 256s (0x100s) facilitating the creation of CP/M-68 slice overlay partitions.
4. N – command: partition: 2, start: 1y, end: +8Mb. This will create a partition with LBA start = 256, and LBA count = 16384.
5. T – command: partition 2, new type: 52. This changes the partition type to 0x52, to be used by CP/M-68 to overlay the first CP/M 2.2 slice, which still has to be reserved. A type 0x52 partition so aligned will not be seen as a conflict by the later R – command.
6. N – command: partition 3, start (default 65): 66y, end: +8Mb
7. T – command: partition 3, new type 52,
8. N – command: partition 3, start (default 130): 131y, end: +8Mb
9. R – command: reserve as many slices as you wish, up to 8.
10. N – command: partition 1, start: (use the default), end: specify the size you want, up to the entire rest of the card.
11. T – command: possibly set a new type (FAT12/FAT16/FAT32) type for partition 1.
12. W – command: if you are satisfied with the result, write the new partition table to the media.

Partition 1 will have to be formatted on your host system to one of the FAT file system types. If FAT12 or FAT16 is used, then UNA-BIOS and the Mini-M68K BIOS will be able to read the directory and execute stand-alone programs. UNA-BIOS can also burn new BIOS images read from the CF or SD card.

The slices will have to be formatted when you bring up CP/M.

(end)