NAME

fixparts – MBR partition table repair utility

SYNOPSIS

fixparts *device*

DESCRIPTION

FixParts (aka **fixparts**) is a text–mode menu–driven program for repairing certain types of problems with Master Boot Record (MBR) partition tables. The program has three design goals, although a few additional features are supported, as well:

* It can remove stray GUID Partition Table (GPT) data, which can be left behind on a disk that was once used as a GPT disk but then incompletely converted to the more common (as of 2011) MBR form.

* It can repair mis-sized extended partitions –– either partitions that extend beyond the physical end of the disk or that overlap with nearby primary partitions. FixParts is designed in such a way that this type of repair occurs automatically, so if it's the only problem with your disk, you can launch the program and then immediately save the partition table, making no manual changes, and the program will fix the problem.

* You can change primary partitions into logical partitions or vice–versa, within constraints imposed by the MBR data structures.

Additional features include the ability to change partition type codes or boot/active flags, to delete partitions, and to recompute CHS values. With the possible exception of recomputing CHS values, these secondary features are better performed with **fdisk**, because **fixparts**' design means that it's likely to alter partition numbering even when such changes are not requested.

The **fixparts** program employs a user interface similar to that of Linux's **fdisk**, but **fixparts** is much more specialized. Most importantly, you can't create new partitions with **fixparts**, although you can change primary/logical assignment.

In the MBR scheme, partitions come in three varieties:

primary
: These partitions are defined in the first sector of the hard disk and are limited in number to four. Some OSes, such as Windows and FreeBSD, must boot from a primary partition.

extended
: Extended partitions are specialized primary partitions. They serve as holding areas for logical partitions.

logical
: A disk can contain an arbitrary number of logical partitions (**fixparts**, however, imposes a limit of 124 logical partitions). All the logical partitions reside inside a single extended partition, and are defined using a linked-list data structure. This fact means that every logical partition must be preceded by at least one sector of unallocated space to hold its defining data structure (an Extended Boot Record, or EBR).

These distinctions mean that primary and logical partitions cannot be arbitrarily interspersed. A disk can contain one to three primary partitions, a block of one or more logical partitions, and one to three more

primary partitions (for a total of three primary partitions, not counting the extended partition). Primary partitions may not be sandwiched between logical partitions, since this would mean placing a primary partition within an extended partition (which is just a specific type of primary partition).

Unlike most disk utilities, **fixparts**' user interface ignores extended partitions. Internally, the program discards the information on the original extended partition and, when you tell it to save its changes, it generates a new extended partition to contain the then-defined logical partitions. This is done because most of the repairs and manipulations the tool performs require generating a fresh extended partition, so keeping the original in the user interface would only be a complication.

Another unusual feature of **fixparts**' user interface is that partition numbers do not necessarily correlate with primary/logical status. In most utilities, partitions 1–4 correspond to primary partitions, whereas partitions 5 and up are logical partitions. In **fixparts**, any partition number may be assigned primary or logical status, so long as the rules for layout described earlier are obeyed. When the partition table is saved, partitions will be assigned appropriately and then tools such as the Linux kernel and **fdisk** will give them conventional numbers.

When it first starts, **fixparts** performs a scan for GPT data. If the disk looks like a conventional GPT disk, **fixparts** refuses to run. If the disk appears to be a conventional MBR disk but GPT signatures are present in the GPT primary or secondary header areas, **fixparts** offers to delete this extraneous data. If you tell it to do so, the program immediately wipes the GPT header or headers. (If only one header was found, only that one header will be erased, to minimize the risk of damaging a boot loader or other data that might have overwritten just one of the GPT headers.)

With the exception of optionally erasing leftover GPT data when it first starts, **fixparts** keeps all changes in memory until the user writes changes with the **w** command. Thus, you can adjust your partitions in the user interface and abort those changes by typing **q** to quit without saving changes.

OPTIONS

The **fixparts** utility supports no command–line options, except for specification of the target device.

Most interactions with **fixparts** occur with its interactive text–mode menu. Specific functions are:

a Toggle the active/boot flag. This flag is required by some boot loaders and OSes.

c Recompute the cylinder/head/sector (CHS) values for all partitions. CHS addressing mode is largely obsolete, but some OSes and utilities complain if they don't like the CHS values. Note that **fixparts**' CHS values are likely to be incorrect on disks smaller than about 8 GiB except on Linux.

l Change a partition's status to logical. This option will only work if the current partition layout supports such a change. Note that if changing a partition's status in this way is not currently possible, making some other change may make it possible. For instance, omitting a partition that precedes the target partition may enable converting a partition to logical form if there had been no free sectors between the two partitions.

o Omit a partition. Once omitted, the partition will still appear in the **fixparts** partition list, but it will be flagged as omitted. You can subsequently convert it to primary or logical form with the **r** or **l** commands, respectively. When you save your changes with **w**, though, the partition will be lost.

p Display basic partition summary data. This includes partition's number, the boot/active flag's status, starting and ending sector numbers, primary/logical/omitted status, whether or not the

partition may be converted to logical form, and the partition's MBR types code.

q Quit from the program *without saving your changes*. Use this option if you just wanted to view information or if you make a mistake and want to back out of all your changes.

r Change a partition's status to primary. This option will only work if the current partition layout supports such a change. Note that every partition can theoretically become a primary partition, although in some configurations, making this change will require omitting some partitions. If **fixparts** refuses to allow changing a partition to primary, you may need to convert other partitions to logical form or omit them entirely.

s Sort partition entries. This option orders partitions in the display to match their on-disk positions, which can make understanding the disk layout easier in some cases. This option has no effect on the ultimate ordering of logical partitions, which are sorted before being saved. The order of primary partitions in the final saved partition table may be affected by this option. In both cases, as already noted, the partition numbers displayed by **fixparts** may not be the same as those used by the kernel or displayed by other partitioning tools.

t Change a partition's type code. You enter the type code using a one−byte hexadecimal number.

w Write data. Use this command to save your changes and exit from the program.

? Print the menu. Type this command (or any other unrecognized command) to see a summary of available options.

BUGS

Known bugs and limitations include:

* The program compiles correctly only on Linux, FreeBSD, Mac OS X, and Windows. Linux versions for x86−64 (64−bit), x86 (32−bit), and PowerPC (32−bit) have been tested, with the x86−64 version having seen the most testing. Under FreeBSD, 32−bit (x86) and 64−bit (x86−64) versions have been tested. Only 32−bit versions for Mac OS X and Windows have been tested.

* The FreeBSD version of the program can't write changes to the partition table to a disk when existing partitions on that disk are mounted. (The same problem exists with many other FreeBSD utilities, such as **gpt**, **fdisk**, and **dd**.) This limitation can be overcome by typing **sysctl kern.geom.debugflags=16** at a shell prompt.

* The program can load only up to 128 partitions (4 primary partitions and 124 logical partitions). This limit can be raised by changing the *#define MAX_MBR_PARTS* line in the *basicmbr.h* source code file and recompiling.

* The program can read partitions only if the disk has correct LBA partition descriptors. These descriptors should be present on any disk over 8 GiB in size or on smaller disks partitioned with any but very ancient software.

* The program makes no effort to preserve partition numbers. This can have consequences for boot loaders and for mounting filesystems via **/etc/fstab**. It may be necessary to edit configuration files or even to re-install your boot loader.

*

The program may change the order of partitions in the partition table.

AUTHORS

Primary author: Roderick W. Smith (rodsmith@rodsbooks.com)

Contributors:

* Yves Blusseau (1otnwmz02@sneakemail.com)

* David Hubbard (david.c.hubbard@gmail.com)

* Justin Maggard (justin.maggard@netgear.com)

* Dwight Schauer (dschauer@gmail.com)

* Florian Zumbiehl (florz@florz.de)

SEE ALSO

cfdisk (8), **cgdisk (8)**, **fdisk (8)**, **mkfs (8)**, **parted (8)**, **sfdisk (8) gdisk (8) sgdisk (8)**

http://en.wikipedia.org/wiki/Master_boot_record

http://www.rodsbooks.com/fixparts/

AVAILABILITY

The **fixparts** command is part of the *GPT fdisk* package and is available from Rod Smith.