What's a GPT?

by Rod Smith, rodsmith@rodsbooks.com

Last revision: 6/26/2010, GPT fdisk version 0.7.2

I'm a technical writer and consultant specializing in Linux technologies. This Web page, and the associated software, is provided free of charge and with no annoying outside ads; however, I did take time to prepare it, and Web hosting does cost money. If you find GPT fdisk or this Web page useful, please consider making a small donation to help keep this site up and running. Thanks!

Donate $1.00	Donate $2.50	Donate $5.00	Donate $10.00	Donate $20.00	Donate another value
Donate	Donate	Donate	Donate	Donate	Donate

Note: This page is part of the documentation for my GPT fdisk program.

In order to understand what GPT is, you must first understand the current standard for disk partitioning and its limits. With that knowledge in hand, you can see how GPT can fix MBR's deficiencies.

MBR, Its Annoyances, and Its Limits

Since the earliest hard disks for x86 computers, these disks have been divided into one or more partitions using a partitioning scheme that has, through the ages, gone by several names, such as *MS-DOS disklabels, BIOS partitions,* and *MBR partitions.* By whatever name, the MBR partitioning scheme has several characteristics that have, in one way or another, been limitations or annoyances:

- The MBR system originally provided for only four partitions. As a workaround, one of these original (or *primary*) partitions can be set aside (and is then known as an *extended* partition) to hold an arbitrary number of *logical* partitions. This configuration is awkward and can lead to problems, since some OSes can only boot from primary partitions and since it's not easy to convert a partition from one type to another. (The FixParts program, which is part of the GPT fdisk family, can convert between primary and logical types, within certain limits.)
- MBR partitions include one-byte partition type codes to help OSes identify the partitions' purpose. The single byte has proven somewhat limiting, and there are occasional collisions—a single code with an ambiguous meaning, such as 0x82, which can refer to either Linux swap space or a Solaris disklabel.
- The MBR scheme originally employed cylinder/head/sector (CHS) addresses to identify partitions' locations. This feature can lead to problems because different BIOSes or OSes can interpret the CHS geometry in different ways. The CHS scheme also tops out at a maximum disk size of about 8 GiB. In part to work around this limitation, MBR has been extended to use 32-bit logical block addressing (LBA), but using two systems itself introduces the opportunity for problems, such as mismatched CHS and LBA definitions for a single partition. In practice, modern OSes seem to mostly ignore the CHS values, and they *must* ignore the CHS values on partitions that begin or end beyond the 8 GiB mark.
- In conjunction with the near-universal sector size of 512 bytes, the 32-bit LBA pointers used by MBR partitions impose a 2 TiB limit on disk and partition size. This limit, unlike the earlier 8 GiB limit of CHS, is not easily overcome by a simple extension to MBR; there simply isn't space left

in the size allocated to MBR data structures. With 3 TB drives available now and even larger devices configurable using RAID technologies, this problem is a very serious one and will only become more serious with time.
- MBR partitions are susceptible to damage. The primary partition table is stored entirely within the first sector of the disk, so if it's destroyed or damaged, it will be hard to recover the disk's partitions. Logical partitions are stored in a linked list data structure that's scattered over the extended partition, so if a single link is broken, access to the remaining logical partitions will be lost.

There are techniques you can use to extend the life of MBR; however, these methods are stop-gaps at best. Sooner or later, you'll find MBR to be inadequate as you move to larger and larger disks.

GPT to the Rescue

The heir apparent to MBR is GPT. This new partitioning scheme fixes many of MBR's problems:

- In its most common configuration, GPT supports up to 128 partitions, so there's no need for extended or logical partitions. In case needs change, GPT data structures permit larger partition table sizes, although in practice few tools seem to allow users to adjust this parameter. (GPT fdisk does, though.)
- The GPT partition type code is a 128-bit (16-byte) GUID. In theory, this gives plenty of room for lots of unique partition types. Unfortunately, in practice there are already collisions—most importantly for Linux users, Linux and Windows have used the same type code for their filesystem partitions, although a new Linux-specific code is now available. To supplement the type code, GPT supports a plain-text name for each partition.
- GPT knows nothing about CHS geometries and it uses 64-bit sector pointers, so 2 TiB drives are puny compared to the total supported disk size of GPT.
- GPT adds CRC32 checksums to its data structures and stores those structures twice on the disk—once at the start of the disk and again at the end. These measures help protect the system against accidental damage caused by carelessness or disk errors.

Unfortunately, GPT is not without its problems. These mainly relate to compatibility. Older OSes have no or limited GPT support. For instance, Windows only supports GPT at all on Windows Server 2003, the 64-bit (but not the 32-bit) version of Windows XP, Windows Vista, and later. Through Windows 7, booting from a GPT disk is impossible unless the system uses the Unified Extensible Firmware Interface (UEFI) rather than a legacy BIOS. (Most computers through 2010 still used a legacy BIOS, although by mid-2011 UEFI-based PCs started becoming common in stores.) Linux has long supported GPT, but boot support depends on the boot loader. GRUB through version 0.97 doesn't officially support booting from GPT, but patched versions of GRUB 0.97 with GPT support are common, and GRUB2 officially supports booting from GPT. Intel-based Macs use GPT by default. (The "Booting from GPT" section of this Web page describes GPT boot issues in more detail.)

To protect GPT disks against errant older disk tools, GPT keeps an MBR partition table on the first sector of the disk. This MBR contains a single disk-spanning partition of type 0xEE, which makes older tools think the disk is in use by an unknown OS. Some tools take advantage of this feature to create a hybrid MBR configuration, in which some partitions are accessible via both GPT and MBR definitions. Although this is non-standard, awkward, delicate, and downright dangerous, it can help make the transition from MBR to GPT easier by providing a workaround for OSes that don't understand GPT.

In addition to the protective MBR, GPT features two main types of data structure, each of which is stored on the disk twice:

- The GPT *headers* contain disk meta-data—information on the locations of the partition tables, the disk GUID, and so on. No actual partitions are defined in the headers, though. Each disk has two headers, a main header and a backup header. If one is damaged, you can use the other to recover the damaged one.
- Just as there are two headers, there are two partition tables. These should be byte-for-byte identical to one another, unlike the headers, which vary because they need to point to each other and to their own partition tables.

Because of the hard 2 TiB limit of MBR partitions, chances are you'll be forced to switch to GPT for at least some disks in the not-too-distant future. MBR is likely to remain useful on smaller devices, such as USB flash drives, for years to come. In 2011, most existing x86 and x86-64 PCs still use MBR-partitioned disks, although as noted earlier, UEFI-based PCs are now common in stores. Mac hardware is an exception to this rule; Apple has embraced GPT, as well as the EFI, of which the GPT definition is part.

Go on to "Working Around MBR's Limitations"

Return to "GPT fdisk" main page

If you have problems with or comments about this web page, please e-mail me at rodsmith@rodsbooks.com. Thanks.

Return to my main web page.