The GUID Partition Table for BIOS Firmware

Hayden Tremethick

This discussion is limited to disk drives with 512 byte sectors Linux only machines.
Modern drives are increasingly useing 4096 byte sectors.

The original **Master Boot Record (MBR)** based partition scheme uses the first 512 bytes (Sector 0) of a boot disk to hold 446 bytes of code, the Stag 1 *boot loader* (boot.img). Following this is the data for a maximum of four *primary partitions.* or three *primary* and one *extended partition*.
GRUB2 Stage 1 loads stage1.5 (core.img) into RAM, hands over control to GRUB2 and exits.
GRUB2 Stage 1.5 is written tn the sectors between the MBR and the first partition
GRUB2 Stage 1.5 (core.img) is located in the space between the boot sector and the first partition.
Stage 1.5 contains a few common filesystem drivers, such as the standard EXT and other Linux filesystems, FAT, and NTFS.
Stage 1.5 locates the stage 2 files in the /boot filesystem and loads the needed drivers.
Stage 2 files are located in the /boot/grub2 directory and several subdirectories and are loaded as needed.
The function of GRUB2 stage 2 is to locate and load a Linux kernel into RAM and turn control of the computer over to the kernel. The kernel and its associated files are located in the /boot directory. The kernel files are identifiable as they are all named starting with vmlinuz.

GRUB2 supports booting one of a selection of kernels.

By default, GRUB provides a pre-boot menu of the installed kernels, including a rescue option and, if configured, a recovery option. Stage 2 loads the selected kernel into memory and turns control of the computer over to the kernel.
All the kernels are in a self-extracting, compressed format to save space. The kernels are located in the /boot directory, along with an initial RAM disk image, and device maps of the hard drives.

After the selected kernel is loaded into memory and begins executing, it must first extract itself from the compressed version of the file before it can perform any useful work. Once the kernel has extracted itself, it loads *systemd*, and turns control over to it. This is the end of the boot process.

Unified Extensible Firmware Interface (UEFI) replaces the BIOS firmware interface originally present in all IBM PC-compatible personal computers.

GUID Partition Table (GPT) is a subset of the UEFI specification. It uses Globally Unique IDentifiers (128-bit numbers) also called Universally Unique Identifiers (UUID) for identifying UEFI disk drives partitions. GPTs use logical block addressing (LBA) in place of the historical cylinder-head-sector (CHS) addressing.

GPT is a newer standard and is gradually replacing MBR. GPT can be employed on most Linux BIOS systems. however be aware that some buggy BIOSes have problems booting from GPT disks.

For GPT hard drives with 512-byte sectors.
The 32 bit MBR partition table will allow a maximum of 2 TiB ($2^{32} \times 2^{9}$ bytes) or (1024^{2}).
The 64 bit GPT allows a maximum disk size of of 8 ZiB or 9.4 ZB (2^{64} x 2^{9} bytes) and 128 partitions. [ZiB (zebibyte) = 2^{70} bytes = 1024^{7} (IEC August 2005) ZB (zetabyte) = 1000^{7}]

LBA 0 (first sector) **in the GPT is the "protective MBR"** so that booting a BIOS-based computer from a GPT disk is supported; but both the bootloader (GRUB2) and the operating system (recent

Linux) must be GPT-aware. This *protective MBR* prevents legacy MBR-based disk utilities from misrecognizing and possibly overwriting GPT disks. A single partition type of EEh is indicated and identifies it as GPT

LBA 1 (second sector) **is the GPT header,** it has a pointer to the Partition Table or Partition Entry Array, (usually LBA 2).

LBA 1 also defines the usable blocks on the disk and the number and size of the partition entries that make up the partition table.

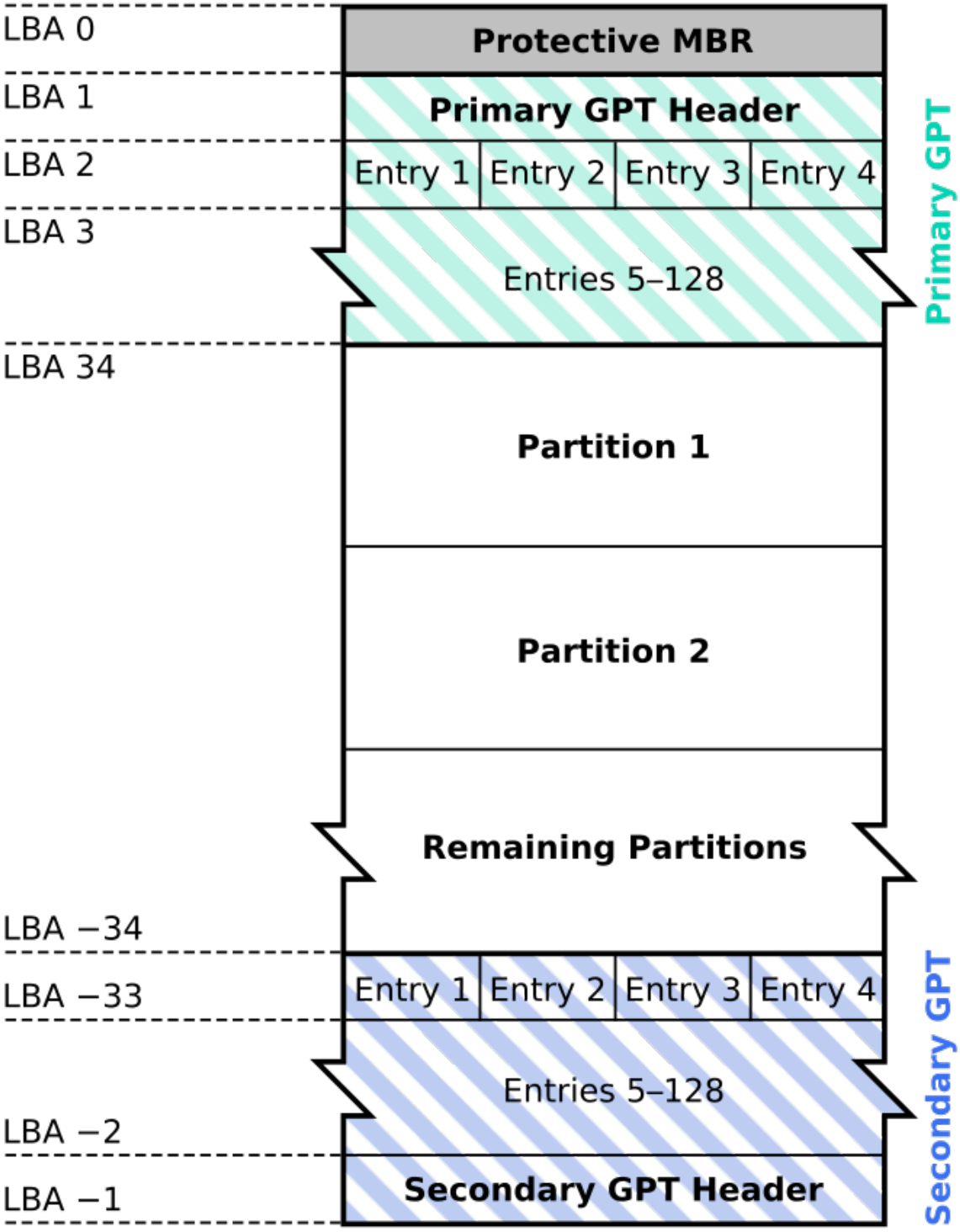
It also contains a CRC32 checksum for itself and for the partition table, which may be verified by the firmware, bootloader, or operating system on boot. Because of this, hex editors should not be used to modify the contents of the GPT. Such modification would render the checksum invalid, making the disk unusable.

LBA 2 - LBA 33. The UEFI specification stipulates that a minimum of 16,384 bytes be allocated for the Partition Entry Array. That gives a disk with 512-byte sectors and a partition entry array size of 16,384 bytes (32sectors) and 128 bytes for each partition entry (4 entries per sector).
LBA 34 is the first usable sector on the disk.
LBA 2048 start of the first Linux partition.
For backward compatibility with most legacy operating systems MBR partitions must always start on track boundaries according to the traditional CHS addressing scheme and end on a cylinder boundary. Extended partitions must start on cylinder boundaries as well.

GPT Utilities

gdisk
gpart
cgpt

GUID Partition Table Scheme



Partition table format

Offset	Length	Contents
0	8 bytes	Signature ("EFI PART", 45 46 49 20 50 41 52 54)
8	4 bytes	Revision (For version 1.0, the value is 00 00 01 00)
12	4 bytes	Header size (in bytes, usually 5C 00 00 00 meaning 92 bytes)
16	4 bytes	CRC32 of header (0 to header size), with this field zeroed during calculation
20	4 bytes	reserved, must be zero
24	8 bytes	Current LBA (location of this header copy)
32	8 bytes	Backup LBA (location of the other header copy)
40	8 bytes	First usable LBA for partitions (primary partition table last LBA + 1)
48	8 bytes	Last usable LBA (secondary partition table first LBA - 1)
56	16 bytes	Disk GUID (also referred as UUID on UNIXes)
72	8 bytes	Partition entries starting LBA (always 2 in primary copy)
80	4 bytes	Number of partition entries
84	4 bytes	Size of a partition entry (usually 128)
88	4 bytes	CRC32 of partition array
92	*	reserved, must be zeroes for the rest of the block (420 bytes for a 512-byte LBA)
LBA Size		**TOTAL**

The following table shows the GPT Header Format:

Offset	Length	Contents
0	8 bytes	Signature ("EFI PART", 45 46 49 20 50 41 52 54)
8	4 bytes	Revision (For GPT version 1.0 (through at least UEFI version 2 value is 00 00 01 00)
12	4 bytes	Header size in little endian (in bytes, usually 5C 00 00 00 mea bytes)
16	4 bytes	CRC32 of header (0 to header size), with this field zeroes duri calculation
20	4 bytes	Reserved; must be zero
24	8 bytes	Current LBA (location of this header copy)
32	8 bytes	Backup LBA (location of the other header copy)
40	8 bytes	First usable LBA for partitions (primary partition table last LB
48	8 bytes	Last usable LBA (secondary partition table first LBA – 1)
56	16 bytes	Disk GUID (also referred to as UUID on UNIXes)
72	8 bytes	Partition entries starting LBA (always 2 in primary copy)
80	4 bytes	Number of partition entries
84	4 bytes	Size of partition entry (usually 128)
88	4 bytes	CRC32 of partition array
92	*	Reserved; must be zeroes for the rest of the block (420 bytes 512-byte LBA)
LBA size	**Total**	