**New GUID Partition Table in Linux**

GPT, the GUID partition table is the modern replacement for the antique MS-DOS Master Boot Record (MBR). GPT is part of the UEFI (Unified Extensible Firmware Interface) specification, and because Linux is a real operating system with modern features you can use GPT with both UEFI and legacy BIOS.

**MBR:**

The MBR lives on the first 512 bytes of your storage device, and holds the bootloader and partition table. The bootloader occupies 446 bytes, the partition table uses 64 bytes, and the remaining two bytes is to store boot signature. The MBR is limited to four primary partitions, and a single primary partition can hold an extended partition which can then be divided into logical partitions. Linux supports an unlimited number of logical partitions. In the olden days Linux maxed out at 63 IDE partitions and 15 SCSI partitions because the kernel was limited in the total device numbers it could allocate. Those limitations gone with udev, since it allocates device numbers dynamically,

The MBR is tiny and inflexible, and has lasted this long thanks to clever hacks to get around its limitations. Logical Block Addressing (LBA) gets around the limitations of its original cylinders, heads, and sectors (CHS) addressing. The traditional hard disk block size of 512 bytes limits partitions to 2TB in size, though more clever hackery supports the new 4096-byte sectors for a maximum 16TB partition size

### **GUID Partition Table (GPT)**

Clever hackery takes us far, and modern tools take us even farther. GPT is part of the Unified Extensible Firmware Interface (UEFI) specification, and on Linux you don't need an EFI BIOS to boot from a GPT partition, but can use it with legacy BIOS systems.

GPT does not have primary and logical partitions, but just partitions as GParted

GPT has several advantages over the MBR:

* 64-bit disk pointers allows 264 total sectors, so a hard disk with 512-byte blocks can be as large as 8 zebibytes. With 4096-byte sectors your maximum disk size is really really large
* The default maximum number of partitions is 128, and if your operating system supports it you can have more
* No more CHS cruft or hacky primary-extended-logical partitioning scheme, which falls down if you need Windows because Windows is inflexible and hogs primary partitions
* GPT has fault-tolerance by keeping copies of the partition table in the first and last sector on the disk
* GPT computes a cyclic redundancy check (CRC) checksum to verify its own integrity, and of the partition table
* Unique IDs for disks and partitions.

### **Unique IDs**

The GPT GUIDs (Globally unique identifiers) and our familiar Linux UUIDs (Universally Unique Identifiers) are not the same thing, though they serve the same useful purpose: giving block devices unique names. Linux UUIDs are a function of filesystems, and are created when the filesystem is created. To see Linux UUIDs just fire up the blkid command:

**[root@testvm ~]# blkid**

**/dev/sda1: UUID="c1fcc857-a8ef-4222-98d9-cfa9da4f693d" TYPE="xfs"**

**/dev/sda2: UUID="d37e3a9d-0aca-4e15-b005-ddbe620cd4b5" TYPE="swap"**

**/dev/sda3: UUID="e7b7948b-1210-4cc8-b3b7-b38708a7b549" TYPE="xfs"**

**/dev/sdb: UUID="5MZwkp-T64x-DGbC-oMTd-VoDE-w9AU-rkascT" TYPE="LVM2\_member"**

**/dev/sdc1: UUID="811ea2fa-7ba6-40ba-8efd-fe20cd133194" TYPE="ext4" PARTUUID="d48597fb-e14f-4d31-83d3-fc788eb9def3"**

**To get GUIDs we need the gdisk command:**

**[root@testvm ~]# fdisk -l /dev/sdc**

**WARNING: fdisk GPT support is currently new, and therefore in an experimental phase. Use at your own discretion.**

**Disk /dev/sdc: 53.7 GB, 53687091200 bytes, 104857600 sectors**

**Units = sectors of 1 \* 512 = 512 bytes**

**Sector size (logical/physical): 512 bytes / 512 bytes**

**I/O size (minimum/optimal): 512 bytes / 512 bytes**

**Disk label type: gpt**

**# Start End Size Type Name**

**1 2048 411647 200M Linux filesystem**

**[root@testvm ~]# gdisk /dev/sdc**

**GPT fdisk (gdisk) version 0.8.6**

**Partition table scan:**

**MBR: protective**

**BSD: not present**

**APM: not present**

**GPT: present**

**Found valid GPT with protective MBR; using GPT.**

**Command (? for help): ?**

**b back up GPT data to a file**

**c change a partition's name**

**d delete a partition**

**i show detailed information on a partition**

**l list known partition types**

**n add a new partition**

**o create a new empty GUID partition table (GPT)**

**p print the partition table**

**q quit without saving changes**

**r recovery and transformation options (experts only)**

**s sort partitions**

**t change a partition's type code**

**v verify disk**

**w write table to disk and exit**

**x extra functionality (experts only)**

**? print this menu**

**Command (? for help):**

“This offers immediately useful information: That this disk uses GPT, with an MBR in protective mode. This protected MBR allows booting from a legacy BIOS, and and protects GPT from GPT-unaware utilities (such as fdisk) and operating systems. These will see an MBR disk with no free space. gdisk serves up a wealth of information on partitions. Below we can see all partitions:”

**Command (? for help): p**

**Disk /dev/sdc: 104857600 sectors, 50.0 GiB**

**Logical sector size: 512 bytes**

**Disk identifier (GUID): E044EBAA-B838-4536-8B14-39AE23D653C7**

**Partition table holds up to 128 entries**

**First usable sector is 34, last usable sector is 104857566**

**Partitions will be aligned on 2048-sector boundaries**

**Total free space is 104447933 sectors (49.8 GiB)**

**Number Start (sector) End (sector) Size Code Name**

**1 2048 411647 200.0 MiB 8300**

**Command (? for help):**

To create a partition start GNU parted as follows:

**root@testvm ~]# parted /dev/sdc**

**GNU Parted 3.1**

**Using /dev/sdc**

**Welcome to GNU Parted! Type 'help' to view a list of commands.**

**(parted) mklabel gpt**

**Warning: The existing disk label on /dev/sdc will be destroyed and all data on this disk will be lost. Do you want to continue?**

**Yes/No? yes**

**(parted) print**

**Model: VMware, VMware Virtual S (scsi)**

**Disk /dev/sdc: 53.7GB**

**Sector size (logical/physical): 512B/512B**

**Partition Table: gpt**

**Disk Flags:**

**Number Start End Size File system Name Flags**

**(parted) mkpart**

**Partition name? []? 1**

**File system type? [ext2]? ext4**

**Start? 0GB**

**End? 2GB**

**(parted) print**

**Model: VMware, VMware Virtual S (scsi)**

**Disk /dev/sdc: 53.7GB**

**Sector size (logical/physical): 512B/512B**

**Partition Table: gpt**

**Disk Flags:**

**Number Start End Size File system Name Flags**

**1 1049kB 2000MB 1999MB ext4 1**

**(parted) mkpart**

**Partition name? []? 2**

**File system type? [ext2]? ext4**

**Start? 2GB**

**End? 4GB**

**(parted) print**

**Model: VMware, VMware Virtual S (scsi)**

**Disk /dev/sdc: 53.7GB**

**Sector size (logical/physical): 512B/512B**

**Partition Table: gpt**

**Disk Flags:**

**Number Start End Size File system Name Flags**

**1 1049kB 2000MB 1999MB ext4 1**

**2 2000MB 4000MB 2001MB 2**

**(parted) mkpart**

**Partition name? []? 3**

**File system type? [ext2]? ext4**

**Start? 4GB**

**End? 6GB**

**(parted) print**

**Model: VMware, VMware Virtual S (scsi)**

**Disk /dev/sdc: 53.7GB**

**Sector size (logical/physical): 512B/512B**

**Partition Table: gpt**

**Disk Flags:**

**Number Start End Size File system Name Flags**

**1 1049kB 2000MB 1999MB ext4 1**

**2 2000MB 4000MB 2001MB 2**

**3 4000MB 6000MB 2000MB 3**

**(parted) quit**

**[root@testvm ~]# gdisk /dev/sdc**

**GPT fdisk (gdisk) version 0.8.6**

**Partition table scan:**

**MBR: protective**

**BSD: not present**

**APM: not present**

**GPT: present**

**Found valid GPT with protective MBR; using GPT.**

**Command (? for help): p**

**Disk /dev/sdc: 104857600 sectors, 50.0 GiB**

**Logical sector size: 512 bytes**

**Disk identifier (GUID): BA8BA61E-1521-4BD9-B7F9-D99FF08163EC**

**Partition table holds up to 128 entries**

**First usable sector is 34, last usable sector is 104857566**

**Partitions will be aligned on 2048-sector boundaries**

**Total free space is 93140925 sectors (44.4 GiB)**

**Number Start (sector) End (sector) Size Code Name**

**1 2048 3905535 1.9 GiB 0700 1**

**2 3905536 7813119 1.9 GiB 0700 2**

**3 7813120 11718655 1.9 GiB 0700 3**

**Command (? for help):**

**[root@testvm ~]# lsblk -t /dev/sdc**

**NAME ALIGNMENT MIN-IO OPT-IO PHY-SEC LOG-SEC ROTA SCHED RQ-SIZE RA WSAME**

**sdc 0 512 0 512 512 1 deadline 128 4096 32M**

**├─sdc1 0 512 0 512 512 1 deadline 128 4096 32M**

**├─sdc2 0 512 0 512 512 1 deadline 128 4096 32M**

**└─sdc3 0 512 0 512 512 1 deadline 128 4096 32M**

**[root@testvm ~]#**

**.**

**With gdisk to examine a single partition. Press i, then the number of the partition:**

**[root@testvm ~]# gdisk /dev/sdc**

**GPT fdisk (gdisk) version 0.8.6**

**Partition table scan:**

**MBR: protective**

**BSD: not present**

**APM: not present**

**GPT: present**

**Found valid GPT with protective MBR; using GPT.**

**Command (? for help): i**

**Partition number (1-3): 1**

**Partition GUID code: EBD0A0A2-B9E5-4433-87C0-68B6B72699C7 (Microsoft basic data)**

**Partition unique GUID: BC1FA9FD-016F-4272-9A2D-A4D30084A03F**

**First sector: 2048 (at 1024.0 KiB)**

**Last sector: 3905535 (at 1.9 GiB)**

**Partition size: 3903488 sectors (1.9 GiB)**

**Attribute flags: 0000000000000000**

**Partition name: '1'**

**Command (? for help): i**

**Partition number (1-3): 3**

**Partition GUID code: EBD0A0A2-B9E5-4433-87C0-68B6B72699C7 (Microsoft basic data)**

**Partition unique GUID: 7AFCEBB5-39AB-45FB-B62E-CDF40EE5B2BF**

**First sector: 7813120 (at 3.7 GiB)**

**Last sector: 11718655 (at 5.6 GiB)**

**Partition size: 3905536 sectors (1.9 GiB)**

**Attribute flags: 0000000000000000**

**Partition name: '3'**

**The Partition unique GUID is what you can see and can use in fstab**

**[root@testvm ~]# blkid|grep sdc**

**/dev/sdc1: UUID="24e09aa6-fd42-4719-b72f-ae2ab01114ea" TYPE="ext4" PARTLABEL="1" PARTUUID="bc1fa9fd-016f-4272-9a2d-a4d30084a03f"**

**/dev/sdc2: UUID="3763a010-8174-47a7-b47c-9503c2b3aabe" TYPE="ext4" PARTLABEL="2" PARTUUID="eb95a73f-947e-418b-a407-7511108c4043"**

**/dev/sdc3: UUID="a699f71b-06a6-42b8-8b9b-6df0416be92e" TYPE="ext4" PARTLABEL="3" PARTUUID="7afcebb5-39ab-45fb-b62e-cdf40ee5b2bf"**

### **Linux GPT Support**

As always, Linux has complete support for this newfangled stuff. You'll need GRUB 2

**Devinder Singh**