Singular file system

The simplest, most basic partitioning scheme in any Linux operating system consists of 3 partitions:

Type	File System	Description
EFI System Partition	vfat	Stores boot loaders and bootable OS images in `.efi` format
Root File System	ext4, btrfs, XFS, or other	Stores the Linux OS files (kernel, system libraries, applications, user data)
Swap	Swap partition or file	Stores swapped memory pages from RAM during high memory pressure

This guide assumes the following:

- There is only 1 disk that needs partitioning
- `/dev/nvme0n1` is the primary disk

Preparing the disk

Determine the disks that are installed on your system. This can easily be done with `fdisk`:

```
fdisk -l
```

It outputs a list of disk devices with one or more entries similar to this:

```
Disk /dev/nvme0n1: 232.89 GiB, 250059350016 bytes, 488397168 sectors
Disk model: Samsung SSD 840
Units: sectors of 1 * 512 = 512 bytes
Sector size (logical/physical): 512 bytes / 512 bytes
I/O size (minimum/optimal): 512 bytes / 512 bytes
Disklabel type: gpt
Disk identifier: XXXXXXXX-XXXX-XXXX-XXXX-XXXXXXXXXXXX
```

The line starting the device file with `/dev/` is the relevant one. Start partitioning the disk with `cfdisk`:

```

```

> **WARNING:** Make sure you are modifying the correct device, else you *will* lose data!

```
cfdisk /dev/nvme0n1
```

If the disk has no partition table yet, `cfdisk` will ask you to specify one. The default partition table format for UEFI systems is `gpt`. Create a layout with at least 3 partitions:

Size	FS Type
1G	EFI System
(RAM size)	Linux Swap
(remaining)	Linux root (x86-64)

> **NOTE:** Specifying the correct file system type allows some software to automatically detect and assign appropriate mount points to partitions. See Discoverable Partitions Specification for more details.

You can verfiy that the partitions have been created by running `fdisk -l` again:

```
Disk /dev/nvme0n1: 232.89 GiB, 250059350016 bytes, 488397168 sectors
Disk model: Samsung SSD 840
Units: sectors of 1 * 512 = 512 bytes
Sector size (logical/physical): 512 bytes / 512 bytes
I/O size (minimum/optimal): 512 bytes / 512 bytes
Disklabel type: gpt
Disk identifier: XXXXXXXX-XXXX-XXXX-XXXX-XXXXXXXXXXXX

Device             Start       End   Sectors   Size Type
/dev/nvme0n1p1      2048   2099199   2097152     1G EFI System
/dev/nvme0n1p2   2099200  35653631  33554432    16G Linux swap
/dev/nvme0n1p3  35653632 488396799 452743168 215.9G Linux root (x86-64)
```

This time `fdisk` will also list the partitions present on the disk.

> **NOTE:** You might notice a pattern with how Linux structures its block devices. Partitions also count as "devices" which you can interact with. Each partition has an incrementing counter attached to its name to specify its order in the partition layout.

Formatting partitions

Format the partition with the appropriate `mkfs` subcommand for the file system you want to use, e.g. ext4:

```
mkfs.ext4 /dev/nvme0n1p3        # ext4 root file system
mkfs.fat -F 32 /dev/nvme0n1p1   # EFI System Partition
mkswap /dev/nvme0n1p2           # Swap space
```

Next mount the file systems:

> **ATTENTION:** Depending on which file system you chose earlier for your root file system, additional mount parameters might be beneficial or necessary, e.g. `btrfs` requires specifying the subvolume you want to mount using the option `subvol=NAME`. Refer to the file system's manual to determine relevant mount parameters.

```
mount /dev/nvme0n1p3 -o noatime /mnt
mount /dev/nvme0n1p1 --mkdir /mnt/efi
swapon /dev/nvme0n1p2
```

Revision #5
Created 2023-09-25 16:39:50 UTC by Sebin
Updated 2025-02-03 16:09:24 UTC by Sebin