

Control Linux system with fstab

In this article, I will show you how to use fstab to control your Linux system.

In this article, I will show you how to use fstab to control your Linux system.

Learn about fstab

The `/etc/fstab` file allows you to control what filesystems are mounted at startup on your Linux system, including Windows partitions and network shares. You can also use it to control the attachment points of external storage devices like USB and external hard drives. In this article, I will show you how to use it to control your Linux system.

/etc/fstab - Available on all Linux computers, control which Linux filesystem is mounted here.

Its tutorial page, `man fstab`, starts with information:

Fstab is a program read only by other programs, not allowed to write; System administrators should be responsible for creating and maintaining this file correctly.

However, today fstab is usually created by an installer or program. So you should not worry too much about your 'responsibility'.

But if you want to delve into fstab research, you still need to understand how and how to adjust it.

A typical fstab

The fstab file installed by most modern Linux distributions looks a bit scary. This is an fstab file from an Ubuntu system:

```
# / etc / fstab: static filesystem information.
#
# Use 'blkid -o value -s UUID' to print the universally unique identifier
# for a device;  garment này ???c s? d?ng v?i UUID = nh? m?t thêm robust
?? ?? tên
# devices mà ho?t ??ng hi?n th?i n?u ??a này thêm và g? b?
. See fstab (5).
#
#
proc / proc proc defaults 0 0
# / was on / dev / sda6 during installation
UUID = 2ad9188b-9d1c-4102-bf24-4b5ad456a701 / ext3 errors = remount-ro 0 1
```

```

# / boot was on / dev / sda1 during installation
UUID = 3943c247-16e9-405b-9fda-87684b02cc4e / boot ext2 defaults 0 2
# swap was on / dev / sda7 during installation
UUID = 15825096-aef7-41d6-b53a-c86aec2ebde8 none swap sw 0 0
/ dev / sda0 / media / cdrom0 udf, iso9660 user, noauto, exec, utf8 0 0

```

Figure 1 below shows the meaning of the columns.

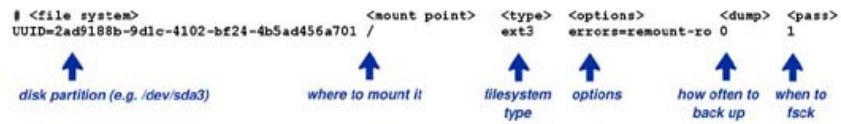


Figure 1

Devices and UUIDs

Let's start with the device: UUID = 2ad9188b-9d1c-4102-bf24-4b5ad456a701. What this means?

Previously, the fstab device field was much simpler, like / dev / hda3 for the third partition on the first IDE disk.

However, systems are becoming more and more complex. USB and SATA disks both use the *sd* disk driver, originally written for SCSI. However, it is impossible to predict their order. If there are several USB and SATA drives, or if you regularly add or remove drives, you may find that your original filesystem appears on *sda2* today but appears on *sdc2* the next day.

To overcome this confusion, fstab can use "Universally Unique Identifier" to identify each filesystem.

However, can I find out which disk partition is mapped with UUID? You can see some comments in fstab:

```

# / was on / dev / sda6 during installation
UUID = 2ad9188b-9d1c-4102-bf24-4b5ad456a701 / ext3 errors = remount-ro 0 1

```

However, you should not trust these annotations completely. It is very likely that your partition is on *sda6*, but it is not sure.

To be safe, we need to check the current value with the *blkid* command:

```

$ blkid
/ dev / sda1: UUID = "702be669-1Aee-4128-8c57-60b58bc91f59" TYPE = "ext2"
/ dev / sda3: UUID = "615aaed5-0dba-4204-9717-c9a00ff411ea" SEC_TYPE = "ext2"
/ dev / sda5: UUID = "0c5121ff-331a-4ae2-b8be-e0b10bcae62f" SEC_TYPE = "ext2"
/ dev / sda6: UUID = "d2a1e4aa-6589-4846-ba58-107d32a25375" SEC_TYPE = "ext2"
/ dev / sda7: UUID = "1533cdc3-635f-4552-818b-1fadce9ea7f8" SEC_TYPE = "ext2"
/ dev / sda8: UUID = "b24fd645-7c28-431b-883d-0a6cf03340ed" SEC_TYPE = "ext2"
/ dev / sda9: TYPE = "swap"

$ blkid -o value -s UUID / dev / sda8
b24fd645-7c28-431b-883d-0a6cf03340ed

```

You can also use `/dev/disk/by-uuid` :

```
$ ls -l / dev / disk / by-uuid
lrwxrwxrwx 1 root root 10 2010-04-18 09:06 0c5121ff-331a-4ae2-b8be-e0b10bcae62
lrwxrwxrwx 1 root root 10 2010-04-18 09:06 615aaed5-0dba-4204-9717-c9a00ff411e
lrwxrwxrwx 1 root root 10 2010-04-18 09:06 1533cdc3-635f-4552-818b-1fadce9ea7f
lrwxrwxrwx 1 root root 10 2010-04-18 09:06 b24fd645-7c28-431b-883d-0a6cf03340e
lrwxrwxrwx 1 root root 10 2010-04-18 09:06 ca8ec122-33c7-4765-bd65-78a15c58def
lrwxrwxrwx 1 root root 10 2010-04-18 09:06 d2a1e4aa-6589-4846-ba58-107d32a2537
lrwxrwxrwx 1 root root 10 2010-04-18 09:06 702be669-1Aee-4128-8c57-60b58bc91f5
```

Some distributions, instead of using UUIDs, can label (*label*) each filesystem:

```
LABEL = / / ext3 defaults 1 1
```

This method allows for much easier reading but is a bit confusing. If you install some Linux distros on different partitions, you may encounter some partitions with the same label. So how can you distinguish which partition is labeled?

Confused by UUID and labels? You do not have to use them. If there is a simple setting with a disk, the simple syntax still works:

```
/ dev / sda6 / ext3 errors = remount-ro 0 1
```

Mounting point and Filesystem type

The next two areas are very simple. *Mount point* is wherever you want to try filesystem: `/`, `/home`, `/boot` or anywhere. It must be an empty directory that exists first. If not, anything inside will be hidden when you mount something on it.

Type (*type*) is filesystem type, like *ext2*, *ext3* , filesystem of Windows *vfat* and *ntfs* , or *iso9660* for CDs. You can also use the *auto* command, and Linux will try to guess the filesystem type. `man filesystems` has a list of supported filesystems.

Options

The *options* section is the most complicated part. This is where you specify 'other issues' - anything that doesn't fit in other parts. If there are no specific options, you only need to use defaults.

You can list many options if you want, distinguish between them with commas. For example, a CDROM can use `ro`, `user`, `noauto`, `exec`, where:

1. **ro** is read-only setup.
2. **The user** means that it does not need root privileges to mount: any user can type `mount / media / cdrom0`, if you do not have any services already attached to it.
3. **noauto** means the system will not try to mount it when booting - a good idea for external devices.
4. **exec** instructs the system to allow you to run programs from that file system. Otherwise it will be disabled on the CDROM and Windows filesystem.

On Windows FAT filesystem, if you use `exec` you may want to `fmask = 111`: Windows filesystem has no permissions, so make sure the executable bit is set if you want to run programs.

Put it all together, you can create a useful entry on systems that don't automatically attach USB devices:

```
/ dev / sdb1 / stuff vfat user, noauto, exec, fmask = 111 0 0
```

If your device is displayed somewhere other than `/ dev / sdb1`, please adjust as needed. Then `sudo mkdir / stuff`, now whenever you plug in the camera, mp3 player or USB, you will be able to mount it by typing `mount / stuff`.

To get the full list of options, see `man mount` and find the **FILESYSTEM INDEPENDENT MOUNT OPTIONS** and **FILESYSTEM SPECIFIC MOUNT OPTIONS** section.

Dump and pass

Dump indicates how often you want the filesystem to be backed up. Most people do not use this field, but if you are running automatic backup software, you may want to use it.

Pass indicates when and when the device should be checked for `fsck` before being mounted. In general, you should use 1 for root filesystem, 2 for all other mounted filesystems and 0 for filesystems that are not mounted by default.

See more:

1. 4 quick access to a Terminal in Linux
2. Guide to network operation for Linux users: 11 commands to know

You finished reading the article "**Control Linux system with fstab**" edited by the [TipsMake](#) team. We hope this article has provided you with many useful tech tips and tricks. You can search for similar articles on tips and guides. Thank you for reading and for following us regularly.