



by Guido Socher ([homepage](#))

*About the author:*

Guido likes Linux because of the community. It is a huge project and somehow all the different people around the world are working towards the same goal: Making it better every day.

Linux is unfortunately now so big that it attracts also the bad guys who just try to make profits and don't give anything back. Watch out! Often they appear to be small "pro-linux" businesses from the outside but they are just opportunistic.

## The mystery of mount points



*Abstract:*

This article explains the concept of mount points but I hope the article has also some information which is interesting to readers who are not new to linux.

---

## Introduction

I will first explain the concept of mounting file-systems and then come to more advanced features. If you feel that you are already a Linux expert then you can start to read a bit further down.

## How it works

Under windows you have even until today the problem that adding a new drive will mess up all the naming conventions. What used to be D: drive is suddenly E: and users as well as software gets confused about it.

Under Unix you will never have this problem because all the physical disks are inserted into the directory tree. You can see this if you type "mount" or "df":

```
# df
Filesystem 1K-blocks    Used Available Use% Mounted on
/dev/hda1   9070728 4701180   3908768  55% /
/dev/hda3   24950428 683412   22999584   3% /home
none        257236      0     257236   0% /dev/shm

# mount
/dev/hda1 on / type ext3 (rw)
none on /proc type proc (rw)
none on /dev/shm type tmpfs (rw)
/dev/hda3 on /home type ext3 (rw)
```

Here we have two disk partition one which contains everything except /home. This is the root partition "/" and it is physically connected to hda1. hda1 is the first partition (1) on the first ide harddisk (hda). The third partition (hda3) on the same disk is mounted on /home. Thus if you go to /home then you access files on hda3. You will never have to worry if this is now D: or E: or ??? It is always just a directory tree and it is always the same.

## The file-system busy problem

Linux buffers file system read and write operations. You will notice this when you have usb version 1.1 disk and you access a big file. The first time access is slow but the second time you open the same file it is very very fast the the light on the usb disk does not flash.

Because of this feature you can not just remove a disk that is used. You need to unmount it. For normal internal harddisks this is never an issue but it can be a problem for cdrom drives and USB disks.

If you e.g mount a usb-stick under /mnt/usb and you do "cd /mnt/usb" then bash is using this filesystem. If you then try to umount it from a different shell window you will get "file system busy" and the umount will fail. It is enough to "cd" again out from the /mnt/usb and you can umount it. The problem is sometimes you forgot who is using it and you have so many applications open that it is difficult to find out what is causing the umount to fail.

Ask your computer!

```
# fuser -m -u /mnt/usb
/mnt/usb:  1347c(root)  1348c(guido)  1349c(guido)
```

What you get from the fuser command is a list of processes still using the filesystem in question. Now you can check with "ps auxw" what this is or you can just kill them. After that umount will work.

## But windows can do it!?

Why not just pull out the usb-stick? Well, old dos/win3.1 computers could just be powered off. No shutdown needed. At that time some of my friends complained that you had to shutdown Linux before powering off. Then win95 came and those complains were gone because everybody was used to shut down the computer first.

## Linux can do it too!

There are also solutions for Linux to be able to just pull out the disk (e.g a usb disk). Mandrake Linux has a feature called supermount and it gives exactly the same results (or problems) as windows. But I don't recommend it because it can cause instability and lost files.

It is better to use the mtools (<http://mtools.linux.lu/>). mtools are a set of commands to access floppy drives, usb-sticks ... basically any removable media with a FAT filesystem and you do not need to mount the filesystem.

Here is the trick:

1. Edit `/etc/mtools.conf` and add a line that says

```
drive u: file="/dev/sda1" # or sda4; some usb-stick have partition
                        # 4 created as factory default.
```

2. Now you do NOT mount `/dev/sda1`. Instead you just type  
`mdir u:`  
to see what is on the usb-stick.

```
mcopy * u:
to copy all the files in the current directory to the usb-stick.
```

```
mcopy u:\* .
to copy all files from the usb-stick to the current directory (note the \ in front of the wildcard).
```

Very easy to use.

## Allow anybody to mount a filesystem

Normally you need to be user root to mount filesystems. For devices like cdrom or usb-stick you might want to give any user the right to mount them. All you need to do in this case, is to add a line like this to `/etc/fstab`:

```
# dev      mount point    fs type    flags
/dev/sda1  /mnt/usb       auto       noauto,user 0 0
```

This line means that `sda1` (the first partition on the first scsi disk, usb storage disks are mapped to scsi disks) shall not be mounted automatically at startup (`noauto`) and can be mounted/unmounted by anybody (user). The FS-type field is also set to `auto` so you can mount a usb-stick formatted with `vfat` or `ext2` and the right filesystem will be detected automatically.

## Laptops and nfs

The network file system, NFS, is really good if you want to have centralized backup of home directories for several users. All you need to do is mount the home directories from a central server and all users have their individual home directories available on all the computers in the network. The same setting and preferences and the same data everywhere. It's really good.

But how do we handle mobile users with Laptops? (no backups?)

Once you are off the network your home directory is gone. One solution is to always copy all data (rsync) but this can easily lead to inconsistency if you are not very disciplined and often you do not need your full home directory. I find it a good solution to have a second small home directory with just the settings and data I need when I travel.

When you mount a filesystem (any filesystem/disk including nfs) on a non empty directory then the content of this directory becomes invisible and you see the mounted filesystem.

This is the solution: Create the mount point for the home directories (/home) and automount there the different home directories when the computer is on the net. However before you mount anything create a softlink for the user that owns the Laptop to an off-line home directory:

```
/home/guido -> ../home_nonet/guido
```

When user guido is on the network then he will see his normal home directory because the softlink is shaded out. When he is on the road then he will see /home\_nonet/guido as his home directory (/home/guido).

A very nice solution.

## Conclusion

To build an abstraction layer for the hardware has always been one of the goals of Unix. To to mount/umount filesystem/disks was one of the ways to achieve this. A really advanced and modern idea even tough Unix is much older then other modern PC operating systems.

---

Webpages maintained by the LinuxFocus Editor team © Guido Socher "some rights reserved" see <a href="http://linuxfocus.org/license/">linuxfocus.org/license/</a> <a href="http://www.LinuxFocus.org">http://www.LinuxFocus.org</a>	Translation information: en --> -- : Guido Socher (homepage)
--	---