Useful Options for mount and umount

In UNIX and Linux operating systems everything resides in a tree like structure rooted at /, the root directory. This includes storage devices with partitions and their own file systems. The *mount* command mounts these somewhere within this tree structure. For instance, a disk partition used for data storage can be mounted on /mnt/data or /media/data or wherever else you find it convenient or prudent.

The mount command accepts options that determine how is the file system mounted. A basic mount command without any options can be as simple as this:

sudo mount /dev/sdb1 /media/usb

Of course the directory you're mounting to (in this example /media/usb) should exist beforehand. To unmount the device from it, thereby removing its file system from the file tree, just use the *umount* command the same way.

To mount a file system with options pass the -o option first followed by a comma separated list of mounting options, which may look something like this:

sudo mount -o noexec,auto /dev/sdb1 /media/usb/

In this case we've used the noexec and auto options, but we might have as well specified any number of other options depending on our needs and desires. Here is a quick overview of those you might find most useful, depending on the situation. They can be used alone or in combination with each other.

defaults

The defaults option is actually a shortcut to a number of

different options that you'll likely want if you have nothing specific in mind, and just want to mount a file system for normal use. They include rw, suid, dev, exec, auto, nouser, and async. In other words running..

sudo mount -o defaults /dev/sdb1 /media/usb

..is the same as running..

sudo mount -o rw,suid,dev,exec,auto,nouser,async /dev/sdb1 /media/usb

So what does each of these options mean?

rw – mount with both read and write access

suid – allows giving users who don't own a specific file in a file system temporary permissions for running the file as if they did own it. In other words the owning user can give other users temporary permissions. This makes commands like passwd and ping usable for normal users even though their operation requires actions which normal users typically don't have access to.

dev – allows creating devices nodes such as /dev/sdc1 within a mounted file system. The opposite option, nodev, would disallow this.

exec – allows executing binaries within the mounted file system. The opposite option, noexec, would make it impossible to execute programs from the mounted device.

auto – specifies that this file system will be automatically mounted on system startup or by running the mount -a command.

nouser – disallows the ordinary non-root user to mount this file system.

async – specifies that all input and output to the file system should be asynchronous as opposed to sync which would make them synchronous. The difference is that the asynchronous mode

allows processing to run even as I/O operations are still ongoing rather than wait for them to finish, which makes sense most of the time, and is therefore a reasonable default option.

If all of these options sound fine to your specific need then just passing the defaults option will be good. However, if you don't want one or more of these default options you may want to specify your own list, or use defaults with overriding subsequent options.

Here are some of the other useful options you might want to consider, some of which override or are the opposite of the above defaults, but could be useful in certain situations.

ro – mounts the file system as read-only, disallowing any write access to the mounted file system. This could be useful if you want to make sure to preserve the data on the file system as is, and especially prevent overwriting any data, when you are mounting the file system only for the purpose of accessing the files on it.

noexec – disallows executing binaries on the mounted file system, which could also be used for file systems used only as data storage where you don't wish programs to run.

users – makes it possible for every user on the system to mount or unmount the file system.

group – allows non-root users to mount the file system if one of their groups is the same as the group to which the device belongs.

remount – useful when you want to mount an already mounted file system, but with different options than those specified previously or within the /etc/fstab configuration file, and without changing the mount point (the path where it is mounted).

noatime – prevents updating access times on inodes of the file system, which could speed up access on a frequently accessed file system for a specialized purpose.

umount options

Finally, you can pass the same options to the umount command using the -O option first, like this:

umount -O noexec /dev/sdb1

Options passed to the umount command only mean that the command will apply to file systems which have the specified option within the /etc/fstab configuration file. In the above example it will not apply if /dev/sdb1 has no *noexec* option specified in /etc/fstab.