

RPI Adding USB Drives

From eLinux.org

Contents

- 1 USB Flash and Hard Drives
- 2 Powered USB Hubs
- 3 Mounting Drive Partitions
 - 3.1 USB Drive Hierarchy
 - 3.2 Mount Points
 - 3.3 Mounting a Partition
 - 3.4 Mounting a Partition at Boot
 - 3.5 Operations on a Mounted Partition
- 4 Unmounting a Partition Before Drive Disconnection
- 5 Other mount and umount Features
- 6 Robust mounting of multiple USB flash drives

USB Flash and Hard Drives

Sooner or later, you're going to run out of room on the SD card used to boot and run your Pi. One solution is to move to a larger SD card, but, the largest low-cost high-speed cards are 32 GBs for about US\$20 ~ \$45 (May 2012). However, there may be compatibility issues between these higher-capacity SD cards and the Pi, at least, temporarily. USB flash drives (aka "thumb" drives) can provide additional storage for about the same, or less, cost as an SD card of the same capacity. USB flash drives generally also have faster data transfer speeds than SD cards. Another option is to connect a USB hard disk drive, which can provide over a terabyte (1 TB) in capacity and data transfer speeds well beyond 40 MBs/second, much larger and faster than SD cards or USB flash drives. See [Transfer system disk from SD card to hard disk](#).

Powered USB Hubs

In order to expand the storage capacity, you may need to connect a powered USB hub to one of your Pi's USB ports, and then connect the USB drive to the hub. A USB hard drive can either have its own AC/mains power supply or it will draw its power via the USB cable. If it is the latter, then you will most certainly need a powered hub as the Pi may not have enough power for itself and extra devices. A four-port USB 2.0 hub will cost about US\$7 - \$25 though do shop around as quality does vary. Since the current Pi models only has a USB 2.0 interface so you do not need to buy a USB 3.0 device.

If you're running your Pi "headless" (ie. without keyboard, mouse or monitor) and remotely logging in via another computer, both USB ports on the model 'B' will be available for use with USB flash drives, without excessive drain on the Pi's power supply. (**NB:** Only tested on a 'B' model acquired July 2012)

Mounting Drive Partitions

You can plug in a USB flash or hard disk drive while the Pi is running without any problem. The USB device will be sensed by the Pi, however, it probably won't be added to the operating system's file system. That will require some manual intervention to "mount" the device onto the file system. A flash or hard drive

may be configured with one, or more, partitions, which you can create yourself in the unlikely event that the drive doesn't already have any. Creating partitions is covered in a separate tutorial: [TBD]

USB Drive Hierarchy

Linux hardware devices are organized and identified by letters under the file system's device hierarchy, `/dev`. USB drives attached to the Pi are found under the file system as `/dev/sdX`, where "X" starts with the letter "a" representing the first USB drive, "b" for the second drive, "c" for the third, etc. The individual partitions on each drive are represented by incremented numbers, starting with "1". So, the first partition on the first USB drive is located in the file system as `/dev/sda1`, the second partition on that drive is `/dev/sda2`, the third partition on a third drive would be `/dev/sdc3`, etc.

Mount Points

In order to make a partition accessible to the file system, you need to use the Linux "mount" command, referencing the hardware (e.g., `/dev/hda1`) and a "mount point" in the file system, which is any empty directory, usually created for the purpose by the user. There is a canonical (i.e., standard) directory path where mount points are usually created: the `/mnt` directory. You can create a new directory under `/mnt` for each partition to be mounted, and it can be named anything you want as long as it doesn't contain any spaces. You might want to name it to correspond to the drive's physical characteristics, e.g., `/mnt/1GB_USB_flash`, OR `/mnt/120GB_USB_hard_disk`.

Mounting a Partition

After creating the mount point for a partition, the only thing left to do is to actually mount the drive partition, e.g.:

```
sudo mount -o uid=pi,gid=pi /dev/sda1 /mnt/1GB_USB_flash
```

where:

`-o` (lowercase letter "o", not the number zero) specifies that an option string follows

`uid=pi,gid=pi` specifies the user ID is user "pi" and the group ID is "pi" (note there are no spaces allowed between these terms)

`/dev/sda1` is the first partition on the first USB drive

`/mnt/1GB_USB_flash` is the mount point directory

Mounting a Partition at Boot

To automatically mount a partition at boot time, the `/etc/fstab` file must be edited, and a line added for each partition and mountpoint:

```
neil@raspbian:~$ cat /etc/fstab
proc                /proc              proc              defaults          0                0
/dev/mmcblk0p1     /boot              vfat              defaults          0                0
#/dev/mmcblk0p3    none               swap              sw                0                0
/dev/mmcblk0p4     /mnt/sdcard        ext4              rw,defaults      0                0
/dev/sda1          /mnt/usbdisk       ext3              rw,defaults      0                0
/dev/sda3          none               swap              sw                0                0
```

In this example, we are mounting one partition for read-write access at `/mnt/usbdisk`, and also moving the swap file from SD card to USB. The `#` character in the first column means that the rest of that line is treated as a comment. For more information see `man fstab(5)`.

Operations on a Mounted Partition

If all goes well, you will see nothing after pressing the Return key other than another command line prompt, which means there were no errors encountered during the mount process. You can then access the partition by referring it via `/mnt/1GB_USB_flash/...` where the `"..."` would be the file system path to the directories and files on which you want to perform an operation. For example, to list the details about the files at the top of the partition, you would simply type:

```
ls -l /mnt/1GB_USB_flash
```

To copy a file named `"file1.txt"` to another file named `"file2.txt"` in the directory `"my_directory"`, you would type:

```
cp /mnt/1GB_USB_flash/my_directory/file.txt /mnt/1GB_USB_flash/my_directory/file2.txt
```

Of course, you can make this a lot simpler by just changing the current directory to `my_directory` like this:

```
cd /mnt/1GB_USB_flash/my_directory
```

To run an executable file `"my_program.bin"` while in that directory, you would type:

```
./my_program.bin
```

You can copy or move files and directories to or from any directory in the mounted partition by just referring to the appropriate path to that directory in the partition.

Unmounting a Partition Before Drive Disconnection

When you're done with a drive and want to disconnect it from the Pi, you should unmount the drive by using the `"umount"` command (notice there is no letter `"n"` between the `"u"` and the `"m"`) and referring to the mount point:

```
sudo umount /mnt/1GB_USB_flash
```

Other mount and umount Features

You can learn more about `mount` and `umount` features by typing `mount -h` OR `mount --help`, `umount -h` OR `umount --help`, or via their `"man"` (manual) pages by typing `man mount` OR `man umount`.

Robust mounting of multiple USB flash drives

Whilst the approach used in the `/etc/fstab` example above is sufficient for the mounting of a single USB hard disk (as in the example) or USB flash drive (which usually has a `vfat` partition) this does assume that only one disk or drive present. Where multiple drives maybe present mounting by reference to the main `/dev` entries (`sda1`, `sdb1`, `sdc1` ...) is not robust since, if the drives are (re-)plugged into different USB jacks, their `/dev` entry names can change. This issue can be resolved by referring to the UUID of each drive's partition(s), assigned during partition creation. To determine this for new drives first enter this command: `ls -laF /dev/disk/by-uuid`. You should see something like this ...

```
trevor@raspberrypi ~ $ ls -laF /dev/disk/by-uuid/
total 0
drwxr-xr-x 2 root root 120 Aug 25 18:56 ./
drwxr-xr-x 6 root root 120 Jan 1 1970 ../
lrwxrwxrwx 1 root root 15 Jan 1 1970 10b4c001-2137-4418-b29e-57b7d15a6cbc -> ../../mmcblk0p2
lrwxrwxrwx 1 root root 15 Jan 1 1970 A1B1-918F -> ../../mmcblk0p1
```

These are the entries for the Pi's SD card partitions. Now plug a flash drive into a usb port. Wait a few seconds to allow it to be detected etc. then re-enter `ls -laF /dev/disk/by-uuid ...`

```
trevor@raspberrypi ~ $ ls -laF /dev/disk/by-uuid/
total 0
drwxr-xr-x 2 root root 120 Aug 25 18:56 ./
drwxr-xr-x 6 root root 120 Jan 1 1970 ../
lrwxrwxrwx 1 root root 15 Jan 1 1970 10b4c001-2137-4418-b29e-57b7d15a6cbc -> ../../mmcblk0p2
lrwxrwxrwx 1 root root 15 Jan 1 1970 A1B1-918F -> ../../mmcblk0p1
lrwxrwxrwx 1 root root 10 Jan 1 1970 A838-48E2 -> ../../sda1
```

Which shows the new entry for the flash drive **currently** linked to `/dev/sda1` (or equivalent for your Pi setup). Similarly, plugging in a second flash drive should result in something like ...

```
trevor@raspberrypi ~ $ ls -laF /dev/disk/by-uuid/
total 0
drwxr-xr-x 2 root root 120 Aug 25 18:56 ./
drwxr-xr-x 6 root root 120 Jan 1 1970 ../
lrwxrwxrwx 1 root root 15 Jan 1 1970 10b4c001-2137-4418-b29e-57b7d15a6cbc -> ../../mmcblk0p2
lrwxrwxrwx 1 root root 15 Jan 1 1970 A1B1-918F -> ../../mmcblk0p1
lrwxrwxrwx 1 root root 10 Aug 25 18:56 A3BF-8779 -> ../../sdb1
lrwxrwxrwx 1 root root 10 Jan 1 1970 A838-48E2 -> ../../sda1
```

In the examples above the first drive was a "Cruzer2Gb", which had a UUID of "A838-48E2" and was linked to `/dev/sda1`. Subsequently, the second drive, a "DTmicro16Gb", with UUID of "A3BF-8779" was linked to `/dev/sdb1`. (Had the drives been plugged in the other way around their links would have been swapped too). However, this means that, now we know the UUID(s), `/etc/fstab` may be edited (via `sudo nano /etc/fstab`) to include additional mount points such as shown below:

```
trevor@raspberrypi ~ $ cat /etc/fstab
proc /proc proc defaults 0 0
/dev/mmcblk0p1 /boot vfat defaults 0 0
/dev/mmcblk0p2 / ext4 defaults,noatime 0 0
UUID=A838-48E2 /media/Cruzer2Gb vfat auto,users,rw,flush,utf8=1,uid=woody,gid=cruzer,dmask=002,fmask=113 0
0
UUID=A3BF-8779 /media/DTmicro16Gb vfat auto,users,rw,flush,utf8=1,uid=woody,gid=dtmicro,dmask=002,fmask=113 0
0
# a swapfile is not a swap partition, so no using swapon|off from here on, use dphys-swapfile swap[on|off]
for that
```

Retrieved from "http://elinux.org/index.php?title=RPI_Adding_USB_Drives&oldid=318176"

Category: RaspberryPi

- This page was last modified on 2 March 2014, at 21:46.
- This page has been accessed 123,507 times.
- Content is available under a Creative Commons Attribution-ShareAlike 3.0 Unported License unless otherwise noted.