{:title “Partitoning and Mounting (Linux)” :layout :post :date “2016-10-1” :tags [“Computers” “Guide”]}

Partitioning

Think of it as creating a room (partition) in a house (The actual harddrive) for where data can stay.

Creating a Partition on Linux

Typing lsblk will show a quick result of your avaliable devices on your computer. It’ll display two types disks and parts. A quick difference you can see between disks and parts is that disks don’t have a number at the end. Note: There are expections but for the most part USBs don’t have numbers at the end. Parts on the other hand do have numbers that correspond with the disk it is in. Such as sda1 is within the disk sda and thus if there is a sda5 most likely there are four other partitions.

Using a disk management program: Gparted

There are various disk management programs but gparted is most common in linux. There is fdisk and parted but without a gui that can acomplish the same feats as gparted. Im going to be changing the sdb disk so make sure to select that. Note: All paths to devices avaliable on the computer start with /dev/… Now we already have one partition on the sdb disk called sdb1. If we did have more unallocated memory we could create another partition from the unallocated memory. However we don’t so lets delete the partition sdb1 and create 2 partitions splitting the memory. Deleting the partition by right clicking on the partition you want deleted and selecting “Delete”. Now all of the partition memory is turned to unallocated as seen below. Note: No changes actually occur till the green check mark is clicked. Now if you check your devices with lsblk you’ll notice that sdb is all alone without sdb1 or any partitions. This is because all of sdb is unallocated memory at the moment. Now you can select the unallocated memory and create a partition with it. However since you want to make two partitions leave some memory left for the second one depending on how much you want. This is done by right clicking on the unallocated memory and clicking “New”. This should bring you to a menu such as this below … If you want to make a 60 GiB partition make the new size 60*1024 = 61440 MiB. (1024 MiB in 1 GiB) Keep the everything else the same since ext4 is readiable by linux. Then click on the rest of the unallocated memory and click add to create the rest of the memory into a partition. If that went smoothly you should have two partitions now and you can verify this by lsblk. The result is two partitions on the sdb disk detected by your computer. Thats it now you have two partitions.

Mounting the Partitions / Placing files in the Partitions

Note: If you aren’t too familiar with using a terminal please read up on how to use a terminal. mount is a command that will allow you to attach the filesystem (partition) to the computer. In this case we are going to mount the two partitions onto our computer. Begin by opening up your favorite terminal and entering the following command. Ignore the dollar sign it just stands for the prompt. $ mount /dev/sdb1 /mnt - What this does is attach the first partition to the /mnt directory. You can check this by doing lsblk As you can see the mount point of sdb1 is now /mnt You can now copy files into the /mnt directory using cp or make files using echo Whatever is placed into the /mnt directory will now be in the partition. Example: $ echo “hello” >> /mnt/file.txt Now you should see a file called file.txt in the /mnt directory using ls /mnt Using umount you can remove a mount point from the partition unmounting it. Example: $ umount /dev/sdb1 - You can examine through lsblkls /mnt that it is no longer there. You can mount using mount /dev/sdb1 /mnt and again see that the file.txt is now back again which shows that the file indeed is in the partition.